Broadview®
www.broadview.com.cn

Java微服务实战

本书由浅入深地讲解了微服务的相关技术，包括基础框架、服务框架和监控部署三大部分，以实战为主、理论为辅，内容丰富，实用性强。

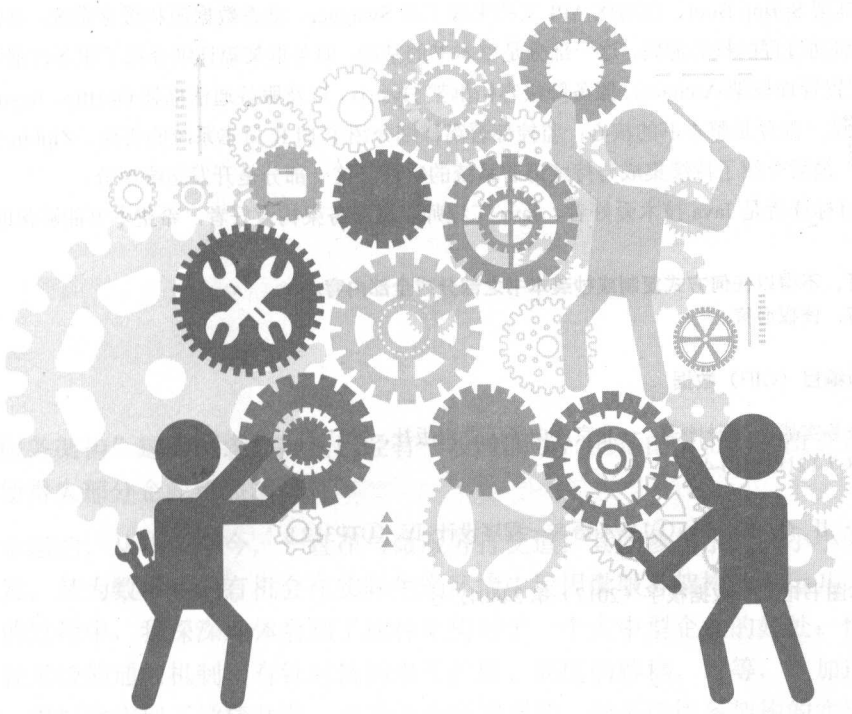
赵计刚 / 著

中国工信出版集团
電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

关于作者

赵计刚

现任网易高级Java开发工程师。毕业于哈尔滨工业大学软件学院。2016年3月加入51信用卡，开始接触微服务架构，之后一直从事微服务的开发与研究，学习与总结了不少微服务架构相关的理论与实践经验。个人是开源技术的拥趸，对新技术充满浓厚的兴趣，尤其是微服务架构相关技术。



Java微服务实战

赵计刚 / 著

電子工業出版社
Publishing House of Electronics Industry
北京•BEIJING

内 容 简 介

本书分为三部分：基础框架篇（第 1~6 章）、服务框架篇（第 7~10 章）、监控部署篇（第 11~13 章），由浅入深地讲解了微服务的相关技术。基础框架篇从微服务架构的基本概念与技术选型出发，详细介绍了微服务基础框架 Spring Boot、自动化 API 文档生成工具 Swagger、动态数据源和缓存系统，并深入分析了 Spring Boot 启动过程的核心源码，这一部分是整本书的基础；服务框架篇详细介绍了服务注册与发现框架 Consul、热配置管理框架 Archaius、服务降级容错框架 Hystrix，以及服务通信框架 OkHttp、AsyncHttpClient 和 Retrofit，这一部分是整本书的核心；监控部署篇详细介绍了 ELK 日志系统的实现、Zipkin 全链路追踪系统的实现，最后介绍了持续集成与持续部署系统的实现，这一部分是开发运维部分。

本书的目标读者是 Java 技术爱好者、Java 工程师、微服务架构爱好者，希望本书能够帮助到你们。

图书在版编目（CIP）数据

Java 微服务实战 / 赵计刚著. —北京：电子工业出版社，2017.11
ISBN 978-7-121-32840-4

Ⅰ. ①J… Ⅱ. ①赵… Ⅲ. ①JAVA 语言－程序设计 Ⅳ. ①TP312.8

中国版本图书馆 CIP 数据核字（2017）第 244012 号

策划编辑：付　睿
责任编辑：牛　勇
印　　刷：三河市良远印务有限公司
装　　订：三河市良远印务有限公司
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：18.75　　字数：386 千字
版　　次：2017 年 11 月第 1 版
印　　次：2017 年 11 月第 1 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 zlts@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。

前言

“微服务架构”这个概念的提出已经有一段时间了，但是由于资料的匮乏以及实现的复杂性，使得大部分企业望而却步。

我是幸运的，从毕业至今，一直在与微服务打交道，其间参与了大大小小多个微服务项目的开发，是为数不多的有机会在实际生产环境中运用微服务架构的幸运儿。在使用微服务架构的过程中，我深深地体会到了这种架构对于一个大中型企业的好处：快速的开发与部署、轻量级的通信机制、有针对性的水平扩展、高度的解耦，等等，这加速了一个项目的迭代，很好地实现了敏捷开发，正是企业所需要的。但是微服务架构的实现也是有一定的复杂性的：服务拆分的边界怎么来定义；原本的单机事务在服务拆分之后变成了分布式事务，这怎么处理；由于服务拆分了，服务之间的通信需要走网络，怎样尽可能地减少网络通信的消耗；怎样防止服务雪崩；怎么梳理链路调用关系，怎么快速定位导致调用链发生错误的服务；怎样监控服务的健康状态，等等，这都是使用了微服务架构后需要解决的问题。本书结合我在实际使用微服务架构中积累的经验给出了其中大多数问题的解决方案，可以为读者朋友提供一个参考。

不可否认的是，正如文章开头所讲的，国内关于微服务架构的开发学习资料与课程都非常有限，这使得微服务架构在国内的推广并没有想象中那样火热。而且，国内的相关资料大多数以理论为主或者没有实战基础。所以，当电子工业出版社博文视点的付睿老师提议写一本以实战为主的微服务书籍之后，我毫不犹豫地抓住了这个机会。本书以实战为主，以理论为辅，真正给出了能在实际生产中使用的技术方案。由于篇幅限制以及以实战为主的特点，本书不会介绍太多的理论（哪怕这个理论很重要），比如在介绍 Consul 的时候，本书不会详细介绍 Raft 一致性协议，但是会介绍与其相关的一些在使用中需要注意的问题，

如果读者对相关问题有兴趣，可以查看相关的论文资料。

本书的组织结构

本书从组织结构上来讲，分为三部分：基础框架篇（第 1~6 章）、服务框架篇（第 7~10 章）、监控部署篇（第 11~13 章）。

第 1 章　微服务概述

本章首先介绍了微服务架构的概念与优缺点，之后简略介绍了微服务中需要的各种组件与常见的技术选型。

第 2 章　微服务基础框架

本章首先介绍了 Spring Boot 在微服务方面的优势，之后通过从零开始开发一个 Spring Boot 项目来介绍 Spring Boot 的基本使用方法，使没有使用过 Spring Boot 的同学可以快速入门。最后在“再学一招”部分，介绍了一个非常好用的 Maven 命令：Maven 依赖树，该命令是查看 SpringBoot-Starter 的依赖以及处理依赖冲突的利器。

第 3 章　微服务文档输出

本章首先介绍了自动化文档输出工具 Swagger 的概念，之后介绍了 Swagger 与 Spring Boot 的集成以及 Swagger 的常用注解。最后在“再学一招”部分，介绍了一个很好用的消除模板代码的框架 Lombok 的安装与使用方法。

第 4 章　微服务数据库

本章以 MySQL 为例，首先介绍了在单数据源的情况下，Spring Boot 与 MyBatis 的集成。之后使用 AbstractRoutingDataSource 实现了对多数据源情况的处理，并简要介绍了实现多数据源的原理。最后在“再学一招”部分，介绍了 MyBatis-Generator 的基本用法。

第 5 章　微服务缓存系统

本章首先介绍了常用的缓存技术的优缺点与选型方案，之后介绍了当使用 Redis 2.x 版本时，使用 Spring Boot 集成 ShardJedis 实现客户端分片的方法。然后介绍了 Redis 3.x 集群的搭建与使用 Spring Boot 集成 JedisCluster 实现服务端集群的方法。最后简要分析了 JedisCluster 的源码。在本章的“再学一招”部分，介绍了使用 GuavaCache 实现本地缓存的方法。

第 6 章　Spring Boot 启动源码解析

本章详细分析了 Spring Boot 启动过程的源码，掌握这一章对于后续章节的学习至关重要。在本章的“再学一招”部分，简要介绍了在开发过程中获取配置信息的 4 种方法。

第 7 章　微服务注册与发现

本章首先介绍了 Consul 的基本概念和功能，之后搭建了服务提供者和服务调用者两个项目来实现使用 Consul 进行服务注册和服务发现的功能，最后介绍了使用 Consul 与 SpringBoot-Actuator 实现服务健康检查的功能。在本章的“再学一招”部分，简要介绍了 Consul 自身提供的几种健康检查的方式及原理。

第 8 章　微服务配置管理

本章首先介绍了为什么要使用 Archaius 以及 Archaius 实现服务热配置的原理，之后展示了使用 Consul-KV 实现配置中心的方式以及结合 Archaius 实现配置动态获取的方式，最后提供了一种将 Archaius 配置信息与 Spring 的 PropertySource 结合的方案。在本章的“再学一招”部分，笔者详细分析了使用 Archaius 构造动态属性源以及动态获取属性的源码。

第 9 章　微服务进程间通信

本章首先介绍了三种服务通信框架：OkHttp、AsyncHttpClient 和 Retrofit，之后分别展示了使用三种框架进行服务通信的代码编写方法。最后在本章的“再学一招”部分，详细分析了使用 Retrofit 进行服务通信的核心源码。

第 10 章　微服务降级容错

本章首先详细介绍了为什么使用 Hystrix、Hystrix 的工作原理以及执行流程，之后展示了在实际项目中如何使用 Hystrix 实现服务降级容错，最后展示了怎样结合 Turbine 来搭建一个完整的 Hystrix 监控系统。在本章的“再学一招”部分，介绍了设置 Hystrix 配置参数的两种方法以及最常使用的 11 个配置项。

第 11 章　微服务日志系统

本章首先详细介绍了为什么使用 ELK 以及 ELK 最常用的两种架构，之后搭建了 ELK 缓冲系统，然后展示了怎样将项目中的日志发送到日志系统中，最后简单介绍了 Kibana 的常见用法。在本章的“再学一招”部分，介绍了怎样使用 Elasticsearch-Curator 进行日志的定时删除。

第 12 章　微服务全链路追踪系统

本章首先详细介绍了为什么使用 Zipkin、Zipkin 的工作流程、数据模型以及工作原理，之后搭建了 Zipkin 全链路追踪系统，然后分别展示了使用 AsyncHttpClient 和 OkHttp 实现服务通信时进行链路追踪的方式，并且介绍了将追踪信息进行持久化的方法。在本章的“再学一招”部分，详细分析了 Brave（Zipkin 的官方 Java 客户端）的核心源码。

第 13 章　微服务持续集成与持续部署系统

本章首先详细介绍了为什么需要搭建持续集成与持续部署系统，之后介绍了构建这套系统的技术选型：GitLab、Jenkins、Docker-Registry 与总体架构，然后分别介绍了使用 jar 包部署服务和使用 Docker 镜像部署服务时持续集成与持续部署系统的工作原理。之后，搭建了这套系统，最后分别展示了在使用 jar 包部署服务和使用 Docker 镜像部署服务时，持续集成与持续部署系统的实现方式。在本章的“再学一招”部分，介绍了最常用的 10 条 Docker 命令。

目标读者

本书面向的读者群：

- Java 技术爱好者
- Java 工程师
- 微服务架构爱好者

本书特点

- 以实战为主，理论为辅，代码编写占了绝大部分的篇幅。
- 代码由浅到深，会介绍表层代码下的核心源码实现。
- 除了第 1 章外，每章的结尾都会提供一个“再学一招”部分，介绍好用的技术或者解析源码。

勘误与支持

由于作者经验水平有限，书中难免有错漏之处。在本书出版后的任何时间，若您对本书有任何问题，可以发邮件到 1197596604@qq.com，我会对所有问题给予回复。也可以加

入 QQ 群 341027254 进行技术交流！

致谢

感谢 51 信用卡，感谢你为我提供这样一个平台，让我能够学习到很多感兴趣的技术，并能将这些技术应用到实际的项目中。也感谢在 51 信用卡中并肩作战的伙伴们，能在这里与一群牛人一起工作让我感到非常自豪。

感谢 51 信用卡首席架构师孔晨，你敢为人先的精神和雄厚的技术沉淀使得在国内并不盛行的微服务架构在 51 信用卡内部运行得炉火纯青。也感谢你不辞辛劳的指导，打通了我在技术道路上的诸多症结。

感谢电子工业出版社博文视点的付睿老师，如果没有你的提议和引导，就不会有这本书，你严谨认真的工作态度让我非常敬佩。

感谢我亲爱的老婆，是你的支持与谅解，才会让我能够有足够多的时间来完成这本书！

谨以此书献给我最敬佩的技术人员、最亲爱的家人，以及众多热爱微服务的朋友们！

目录

第 1 篇　基础框架篇