



微服务 容器化开发实战

尹为强◎著

基于SpringCloud、Docker、Rancher、Kubernetes
从设计、开发、部署到运维的整体解决方案

微服务容器化开发实战

尹为强 著

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

微服务和 Docker 容器技术是目前分布式架构中常用的主流技术,结合 Kubernetes 强大的容器服务编排与调度能力,中小企业能够简单快速地构建自己的分布式互联网架构系统。本书结合实际生产项目的架构重构过程和经验,详细介绍从单体架构过渡到微服务容器化的分布式架构的各个方面,主要包括微服务开发和运维技术栈的选型、Spring Cloud 微服务开发、Rancher 和 Kubernetes 生产离线环境下的安装与部署、微服务的容器化与服务编排、日志调用链的采集和展示、基于 Prometheus 的系统监控,以及微服务的持续集成等要点。本书基于 Spring Boot、Spring Cloud、Docker、Rancher、Kubernetes 等技术,主要提供从设计、开发、部署到运维的微服务容器化的生产环境整体解决方案。

本书内容通俗易懂,每章都结合实际生产项目进行讲解,特别适合将 Spring Cloud、Docker 和 Kubernetes 等技术应用到生产中的技术管理人员、架构师与开发人员阅读,也适合微服务、容器化的入门读者阅读。本书还可作为微服务、容器化的运维人员,以及对相关技术感兴趣的人员的参考用书。

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

微服务容器化开发实战 / 尹为强著. —北京: 电子工业出版社, 2020.1

ISBN 978-7-121-38292-5

I. ①微... II. ①尹... 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(2020)第 014138 号

责任编辑: 宋亚东 特约编辑: 田学清

印 刷: 三河市华成印务有限公司

装 订: 三河市华成印务有限公司

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: 14.75 字数: 296 千字

版 次: 2020 年 1 月第 1 版

印 次: 2020 年 1 月第 1 次印刷

定 价: 89.00 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 一

在计算机技术日新月异的今天，层出不穷的新技术推动了生产力的发展，也推动了整个社会的进步。软件架构从单体架构到分布式架构，从 SOA 架构到微服务架构，再到服务网格。数据库从 Oracle 包打天下到 MySQL 大行其道，再到内存数据库、NoSQL 数据库遍地开花。容器编排调度技术从 Mesos、Swarm、Kubernetes 三足鼎立，到最终 Kubernetes 一统天下。“工欲善其事，必先利其器。”在互联网转型的大潮中，需要在思想、方法、技术等方面全面转型。互联网架构的高可靠、高并发、高扩展与低耦合，共同创造了互联网应用的无限可能。

微服务的核心思想是将整个系统拆分为多个相互独立的业务，各模块可以独立地开发、测试、部署、运行和运维。这种低耦合、高内聚的灵活特性正是所有架构设计人员所期盼的。因此，微服务架构在短时间内便得到大型互联网企业的青睐，成为打造复杂互联网应用的一把利器。Kubernetes 作为目前主流的容器编排调度平台，从新生事物逐渐发展成为影响全球 IT 技术的基础设施平台，更推动了微服务架构、云原生应用、服务网格等热门技术的普及和落地。

本书内容循序渐进，概念清晰明了，由浅入深，从易到难，为读者呈现了一幅包括微服务、容器化及技术实践在内的全景图。本书融合了作者在企业推广实施微服务的诸多经验，不但理论架构完整，而且包含典型的实践案例。

目前，大数据、云计算、物联网、人工智能、区块链等技术遍地开花，借助 5G 技术的翅膀，我们定能展翅翱翔！

黄文良

中国联通集团首席业务创新官 电子商务中心总经理

序 二

不断增长的用户访问量，快速多变的业务需求，随之带来的资源高效利用和敏捷开发迭代需求，对于开发人员和运维人员来说都是新的挑战。为了解决应用快速部署、资源弹性使用问题，容器化和微服务等新技术正在不断深入人心。Docker 容器技术、Google 的 Kubernetes 和 Mesosphere 的 Mesos 等分布式互联网架构技术应运而生并日益成熟。DevOps 理念正在不断得到业界认同和落地实践。

本书作者尹为强是我多年的老同事，是一位多年来潜心研究互联网架构新技术的资深研发工程师，在云化架构、容器化微服务等方面有独特的见解和心得体会，并在实际生产系统中得到了成功的应用。现在他利用业余时间将工作中形成的宝贵经验汇集成册同大家分享，可以使有志学习容器化和微服务方面的技术爱好者少走弯路，这是一件值得庆贺的好事。

本书有两大特点：一是理论知识论述深入浅出、通俗易懂；二是内容基于作者在设计互联网亿级用户的高并发、分布式架构平台过程中的实际经验和深刻总结。每个关键知识点都并非纸上谈兵，而是有详细的实操案例，具有很强的实战指导意义。在实际开发工作中，人们经常遇到诸多挑战，如开发环境和生产环境保持一致、大规模应用的快速部署、服务平滑升级、应用的微服务化拆分、微服务编排和自动扩缩容、资源的弹性使用、容器化集群的高效管理等。通过阅读本书，人们可以很好地应对这些挑战。本书可以作为初级和中级开发人员、架构师、技术经理等人员的学习参考书。

赵锡成

中国联通集团电子商务中心软件开发部总监

前言

微服务的概念最早是在 2011 年 5 月的软件架构会议上提出的, Martin Fowler 于 2014 年全面阐释了微服务架构。Docker 作为一种容器化技术诞生于 2013 年的开源项目, 如今, Docker 已经成为容器技术领域的领头羊。Kubernetes 这个名字起源于古希腊, 是舵手、罗盘的意思。Google 于 2014 年 6 月推出 Kubernetes (Borg 的开源版本), 并于 2015 年 7 月推出 Kubernetes v1.0 版本。微服务、Docker 和 Kubernetes 涵盖了系统开发、运行环境与编排调度技术, 是目前分布式系统架构中主流的技术栈。

为什么写本书

如今, 无论是开发人员、运维人员还是技术管理者, 言系统架构必谈微服务和容器化, 必谈分布式和互联网架构。但是对于大多数 IT 从业者来说, 微服务、容器化、Kubernetes 等涉及很多新技术, 如果完全摸索着去学习, 无形中会增加学习时间, 并延长系统的新架构改造周期, 而且其他人遇到的问题还要重新解决。

笔者也曾遇到相同的问题, 微服务、容器化和 Kubernetes 相关技术栈过于庞大, 在自行摸索过程中, 由于主机环境、软件版本、参数配置、网络模式等存在一定的差异, 所以其他人分享的经验无法复用。笔者在进行订单系统进行微服务、容器化的新架构开发和升级过程中, 总结和提炼了一些经验与注意事项, 将这些内容加以整理并固化形成本书, 希望对读者的实践有所帮助。

对于微服务开发工程师、容器化和 Kubernetes 运维工程师、系统架构师、技术管理者等来说, 本书不但能够引导他们少走弯路, 而且能够给予他们系统架构、技术栈选型等方面的参考。阅读本书, 读者能够使单体架构的系统简单快捷地进行微服务容器化的架构升级, 从而使新项目能够基于微服务容器化的新架构迅速地开发、部署和运维。

本书主要内容

本书共包括 10 章, 以微服务、容器化、Kubernetes、Rancher 等的概念、解决方案及生产实践操作为基础和主线, 进行了易于理解和上手操作的讲解。

第 1 章至第 3 章主要围绕微服务开发展开: 从单体架构和微服务架构的特点、两者之间的差异开始, 介绍了如何拆分微服务; 然后介绍微服务开发的技术框架选型, 以及微服务开发中

的代码编写；最后介绍了微服务运行的 3 种运行环境，以及每种运行环境如何配置参数等。

第 4 章主要围绕微服务的容器化部署和生产展开：详细介绍在离线的生产环境内网中，如何快速、便捷地部署和管理 Kubernetes 集群，如何创建安全的数字证书，以及如何创建私有镜像仓库等。

第 5 章至第 7 章主要围绕微服务间的调用展开：首先介绍了基于 ELK 组件的日志采集模型及其在生产中的应用；然后介绍了微服务间调用链的概念、调用链模型及调用链的动态跟踪；最后介绍了服务间调用的熔断器，为服务调用提供一种稳定、可靠的容错和自保护机制。

第 8 章主要围绕微服务到容器化的生产构建展开：从精简、安全、标准化的基础镜像构建到业务镜像构建，再到生产环境中服务编排及特定服务涉及的目录挂载、环境变量设置等。

第 9 章和第 10 章主要围绕系统监控、持续集成部署展开：介绍了系统监控的类型、范围，以及 Exporter + Prometheus + Grafana 的监控模型体系。系统持续集成部署部分涉及基于 Jenkins 的自动编译、打包、部署的概念和生产实践。

致谢

首先要感谢我的家人给予我的理解和坚定支持。特别感谢我的妻子，她的理解、包容和支持使我能够全身心投入书稿的编写工作。

同时，衷心感谢电子工业出版社博文视点的宋亚东为我耐心讲解书籍出版要求以及书稿编写的注意事项，并且基于作者的角度提供了诸多帮助，尽量减轻作者工作量。他工作严谨，技术理解深刻，对书稿质量把控严格，还多次邀请专家对本书提出有益意见，对本书的修改和完善起到了重要作用。

由于作者水平有限，书中难免存在不足之处，敬请广大读者批评指正。

尹为强



读者服务

微信扫码回复：38292

- 获取博文视点学院 20 元付费内容抵扣券；
- 获取免费增值资源；
- 加入读者交流群，与更多读者互动；
- 获取精选书单推荐。

轻松注册成为博文视点社区（www.broadview.com.cn）用户，您对书中内容的修改意见可在本书页面的“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。

目 录

| | |
|-------------------------------|----|
| 第 1 章 单体架构与微服务架构 | 1 |
| 1.1 背景介绍 | 2 |
| 1.2 单体架构 | 3 |
| 1.2.1 单体架构的特点 | 3 |
| 1.2.2 单体架构的优点 | 4 |
| 1.2.3 单体架构的缺点 | 4 |
| 1.3 微服务架构 | 4 |
| 1.3.1 微服务架构的特点 | 6 |
| 1.3.2 微服务架构的优点 | 6 |
| 1.3.3 微服务架构的缺点 | 6 |
| 1.4 微服务拆分 | 7 |
| 1.4.1 微服务设计原则 | 7 |
| 1.4.2 微服务拆分原则 | 8 |
| 1.5 微服务开发原则 | 9 |
| 1.5.1 微服务命名 | 9 |
| 1.5.2 项目依赖配置 | 9 |
| 1.5.3 配置文件 | 10 |
| 1.5.4 代码命名 | 10 |
| 1.6 本章小结 | 11 |
| 第 2 章 Spring Boot 微服务开发 | 12 |
| 2.1 微服务框架 | 13 |
| 2.1.1 Dubbo 服务框架 | 13 |

| | | |
|--------------|---|-----------|
| 2.1.2 | Spring Cloud 微服务框架 | 15 |
| 2.1.3 | Dubbo 服务框架和 Spring Cloud 微服务框架的对比 | 17 |
| 2.2 | 技术栈选型 | 17 |
| 2.3 | 搭建开发环境 | 19 |
| 2.3.1 | 安装开发软件 | 19 |
| 2.3.2 | Maven 配置 | 20 |
| 2.3.3 | IDE 配置 | 21 |
| 2.4 | 编写框架代码 | 22 |
| 2.4.1 | 创建 Maven 工程 | 22 |
| 2.4.2 | 创建微服务模块 | 24 |
| 2.4.3 | 微服务代码结构 | 26 |
| 2.5 | 编写业务代码 | 29 |
| 2.5.1 | Controller 文件 | 30 |
| 2.5.2 | Mapper 文件 | 31 |
| 2.5.3 | Service 文件和 ServiceImpl 文件 | 31 |
| 2.6 | 打包与测试 | 33 |
| 2.6.1 | 编译打包 | 33 |
| 2.6.2 | 运行测试 | 35 |
| 2.7 | 本章小结 | 36 |
| 第 3 章 | 搭建微服务运行环境 | 37 |
| 3.1 | 运行环境概述 | 38 |
| 3.2 | 搭建开发环境 | 38 |
| 3.2.1 | IDE 环境 | 40 |
| 3.2.2 | 非 IDE 环境 | 41 |
| 3.3 | 搭建测试环境 | 41 |
| 3.3.1 | 非容器化环境 | 41 |
| 3.3.2 | 容器化环境 | 42 |
| 3.4 | 搭建生产环境 | 43 |

| | | |
|--------------|--|-----------|
| 3.5 | 切换运行环境 | 45 |
| 3.6 | 本章小结 | 45 |
| 第 4 章 | Rancher 和 Kubernetes 离线部署 | 46 |
| 4.1 | 下载离线文件 | 47 |
| 4.1.1 | 下载 Docker 安装文件 | 47 |
| 4.1.2 | 下载 docker-compose 文件 | 48 |
| 4.1.3 | 下载 Harbor 安装文件 | 48 |
| 4.1.4 | 下载 Rancher 离线文件 | 49 |
| 4.2 | 创建 CA 证书 | 51 |
| 4.3 | Docker 离线安装 | 52 |
| 4.3.1 | 安装 Docker | 52 |
| 4.3.2 | 自定义 Docker 目录 | 53 |
| 4.3.3 | 创建用户 eos | 54 |
| 4.4 | Harbor 离线安装 | 54 |
| 4.4.1 | 配置 docker-compose | 54 |
| 4.4.2 | 载入 Harbor 镜像 | 55 |
| 4.4.3 | 配置 Harbor | 56 |
| 4.4.4 | 启动 Harbor | 57 |
| 4.4.5 | 创建用户和项目 | 58 |
| 4.5 | Docker 登录设置 | 61 |
| 4.5.1 | 配置 Harbor 证书 | 61 |
| 4.5.2 | 登录 Docker | 62 |
| 4.6 | Rancher 离线安装 | 62 |
| 4.6.1 | 载入 Rancher 镜像 | 63 |
| 4.6.2 | 安装 Rancher | 65 |
| 4.7 | Kubernetes 安装 | 66 |
| 4.7.1 | 配置私有仓库 | 66 |
| 4.7.2 | 添加 Control 节点 | 67 |

| | |
|-------------------------------|------------|
| 4.7.3 添加 Worker 节点..... | 71 |
| 4.7.4 配置 kubectl 客户端..... | 74 |
| 4.8 本章小结..... | 76 |
| 第 5 章 日志采集 | 78 |
| 5.1 日志采集模型..... | 79 |
| 5.2 配置微服务日志..... | 80 |
| 5.3 容器日志采集..... | 81 |
| 5.4 日志汇集..... | 82 |
| 5.4.1 日志格式..... | 83 |
| 5.4.2 配置 Filebeat..... | 84 |
| 5.4.3 配置 Logstash..... | 86 |
| 5.4.4 配置 Elasticsearch..... | 91 |
| 5.5 日志展示..... | 95 |
| 5.5.1 配置 Kibana..... | 95 |
| 5.5.2 启动和关闭 Kibana..... | 96 |
| 5.5.3 Kibana 索引设置..... | 96 |
| 5.5.4 展示 Kibana 图形化..... | 98 |
| 5.6 本章小结..... | 108 |
| 第 6 章 调用链跟踪 | 109 |
| 6.1 调用链概述..... | 110 |
| 6.1.1 系统完整调用链..... | 111 |
| 6.1.2 调用过程追踪..... | 111 |
| 6.2 Spring Cloud 调用链模型..... | 112 |
| 6.2.1 Sleuth 组件..... | 112 |
| 6.2.2 Zipkin 组件..... | 114 |
| 6.2.3 Spring Cloud 调用链跟踪..... | 115 |
| 6.3 调用链跟踪配置..... | 116 |
| 6.3.1 微服务配置..... | 116 |

| | | |
|--------------|-------------------------------|------------|
| 6.3.2 | Zipkin 服务端配置 | 120 |
| 6.4 | 调用链跟踪展示 | 124 |
| 6.4.1 | Zipkin 图形化展示 | 125 |
| 6.4.2 | Kibana 图形化展示 | 133 |
| 6.5 | 本章小结 | 136 |
| 第 7 章 | 熔断器 | 137 |
| 7.1 | 熔断器概述 | 138 |
| 7.1.1 | 熔断器介绍 | 138 |
| 7.1.2 | Hystrix 介绍 | 138 |
| 7.2 | 熔断器配置 | 139 |
| 7.3 | 熔断器使用 | 140 |
| 7.3.1 | EnableCircuitBreaker 注解 | 140 |
| 7.3.2 | HystrixCommand 注解 | 141 |
| 7.3.3 | Fallback 函数 | 142 |
| 7.4 | 熔断器监控 | 144 |
| 7.5 | 本章小结 | 145 |
| 第 8 章 | 容器化与服务编排 | 146 |
| 8.1 | 容器化与服务编排概述 | 147 |
| 8.1.1 | Docker 介绍 | 147 |
| 8.1.2 | Docker 与虚拟机 | 148 |
| 8.1.3 | 服务编排与调度 | 149 |
| 8.2 | 构建基础镜像 | 149 |
| 8.2.1 | 母镜像的选择 | 150 |
| 8.2.2 | 基础镜像的特点 | 151 |
| 8.2.3 | 生成基础镜像 | 151 |
| 8.3 | 构建业务镜像 | 154 |
| 8.3.1 | 微服务打包 | 154 |
| 8.3.2 | 镜像构建的 Dockerfile 文件 | 156 |

| | |
|-----------------------------|------------|
| 8.3.3 镜像集成构建..... | 157 |
| 8.4 服务编排..... | 161 |
| 8.4.1 服务创建..... | 161 |
| 8.4.2 服务版本升级..... | 165 |
| 8.4.3 参数配置..... | 166 |
| 8.4.4 服务版本降级..... | 167 |
| 8.5 目录挂载..... | 168 |
| 8.5.1 创建持久化卷..... | 169 |
| 8.5.2 绑定持久化卷..... | 170 |
| 8.5.3 挂载持久化卷..... | 170 |
| 8.6 环境变量..... | 171 |
| 8.7 管理 Config Map..... | 172 |
| 8.7.1 创建 Config Map..... | 172 |
| 8.7.2 使用 Config Map..... | 173 |
| 8.8 本章小结..... | 175 |
| 第 9 章 系统监控 | 176 |
| 9.1 系统监控概述..... | 177 |
| 9.1.1 主机监控..... | 177 |
| 9.1.2 安全监控..... | 179 |
| 9.1.3 中间件监控..... | 181 |
| 9.1.4 业务监控..... | 183 |
| 9.1.5 容器监控..... | 184 |
| 9.1.6 监控工具..... | 184 |
| 9.2 Kubernetes 监控..... | 185 |
| 9.3 Node Exporter 组件..... | 187 |
| 9.4 Prometheus 组件..... | 191 |
| 9.4.1 Prometheus 组件的特征..... | 191 |
| 9.4.2 Prometheus 组件的架构..... | 193 |

| | |
|------------------------------|------------|
| 9.4.3 Prometheus 组件的部署 | 194 |
| 9.5 Grafana 组件 | 206 |
| 9.5.1 Grafana 组件介绍 | 206 |
| 9.5.2 Grafana 组件的部署 | 207 |
| 9.5.3 Grafana 组件的展示 | 209 |
| 9.6 本章小结 | 211 |
| 第 10 章 持续集成部署 | 212 |
| 10.1 持续集成部署的概念 | 213 |
| 10.1.1 持续集成 | 213 |
| 10.1.2 持续部署 | 213 |
| 10.2 持续集成部署的模型 | 214 |
| 10.3 基于 Jenkins 的 CICD | 215 |
| 10.3.1 GitLab 代码分支管理 | 215 |
| 10.3.2 CICD 模型 | 215 |
| 10.3.3 CICD 流程 | 216 |
| 10.3.4 CICD 实践 | 218 |
| 10.4 本章小结 | 220 |

单体架构与微服务架构

由量变到质变。从单体架构向微服务架构演进，不只是软件架构的改变，更是方法论和生产力的大幅度提升。本章主要介绍单体架构和微服务架构的优缺点，以及微服务的设计原则和拆分原则。

1.1 背景介绍

本书以笔者所参与的 IT 系统的微服务化、容器化为例，介绍单体架构如何演进成微服务架构，并融合容器、Kubernetes、日志与调用链、监控与运维等生产实践。该 IT 系统是国内某家电信运营商所有线上订单的生产处理系统，即现在比较火热的“中台”系统。该系统是电子商务平台的后台支撑系统，支撑自有网上商城、淘宝店铺、天猫旗舰店、京东旗舰店等网上销售渠道的订单的生产交付。

电信运营商的线上订单一般包括手机终端订单、手机配件订单、手机卡订单、合约机订单、宽带订单等，由于这些订单的类型不同，所以后续生产交付流程的差异也很大。订单的生产交付流程如图 1-1 所示。

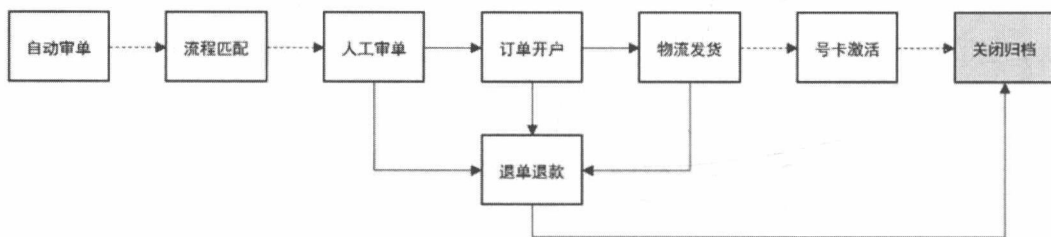


图 1-1 订单的生产交付流程

- (1) 自动审单：审核黑名单用户、一证五户、人证相符等。
- (2) 流程匹配：根据订单来源、订单类型、订单信息智能分配后续生产交付流程。
- (3) 人工审单：审核订单信息的有效性、完整性、实名信息等。
- (4) 订单开户：手机卡订单、合约机订单等在 BSS（业务支撑系统）系统中创建用户、客户、账户信息等。
- (5) 物流发货：调用第三方社会物流企业接口，申请并打印发货电子面单。
- (6) 号卡激活：用户拿到手机卡后，视频激活手机卡。
- (7) 退单退款：处理订单退单和订单支付款的退款。
- (8) 关闭归档：订单状态置为完成状态，标记订单的生产交付完成。