博文视点云原生精品丛书

Broadview
www.broadview.com.cn

深入浅出 Istio

Service Mesh快速入门与实践

崔秀龙 著

中国工信出版集团

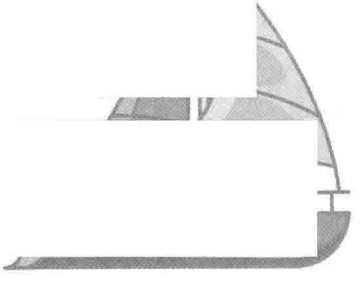
電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

原生精品丛书

深入浅出Istio

Service Mesh快速入门与实践

崔秀龙 著



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

内 容 简 介

Google 联合 IBM、Lyft 推出的 Istio，一经问世就受到了人们的普遍关注，其热度迅速攀升，成为 Service Mesh（服务网格）方案的代表项目。本书整理了 Istio 中的部分概念和案例，以快速入门的形式，对 Istio 的基础用法一一进行讲解，并在书末给出一些试用方面的建议。

在本书中，前 3 章从微服务和服务网格的简短历史开始，讲述了服务网格的诞生过程、基本特性及 Istio 的核心功能，若对这些内容已经有所了解，则可以直接从第 4 章开始阅读；第 4、5 章分别讲解了 Istio 的配置和部署过程；第 6 章至第 9 章，通过多个场景来讲解 Istio 的常用功能；第 10 章结合了笔者的实践经验，为读者提供了 Istio 的一系列试用建议。本书没有采用官方复杂的 Book Info 应用案例，而是采用客户端+简单 HTTP 服务端的案例，读者随时都能在短时间内启动一个小的测试。

本书面向对服务网格技术感兴趣，并希望进一步了解和学习 Istio 的中高级技术人员，假设读者已经了解 Kubernetes 的相关概念并能够在 Kubernetes 上熟练部署和管理微服务。若希望全面、深入地学习 Kubernetes，可参考《Kubernetes 权威指南：从 Docker 到 Kubernetes 实践全接触》和《Kubernetes 权威指南：企业级容器云实战》。

图书在版编目（CIP）数据

深入浅出 Istio：Service Mesh 快速入门与实践 / 崔秀龙著. —北京：电子工业出版社，2019.3
（博文视点云原生精品丛书）
ISBN 978-7-121-35964-4

Ⅰ. ①深… Ⅱ. ①崔… 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（2019）第 012183 号

责任编辑：张国霞
印　　刷：三河市双峰印刷装订有限公司
装　　订：三河市双峰印刷装订有限公司
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：13.5　字数：210 千字
版　　次：2019 年 3 月第 1 版
印　　次：2019 年 3 月第 1 次印刷
印　　数：5000 册　　定价：79.00 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。

推荐序一

Service Mesh 是新兴的微服务架构，被誉为下一代微服务，是云原生技术栈的关键组件之一。从云原生演进的路线来看，Service Mesh 概念是云原生推进过程中的必然产物，基于云原生理念设计实现的微服务应用，需要一个通用的通信层对服务进行统一管控。将该通信层下沉为基础设施的一部分，将极大地减轻云原生应用的负担，并增强云原生应用的弹性和健壮性。

Istio 作为第 2 代 Service Mesh 产品的典型代表，在 Google、IBM 等公司的强力推动下，已经得到社区的广泛认可，成为 Service Mesh 的明星项目，并有可能在未来一两年内成为 Service Mesh 的事实标准，可谓前途远大。

但是，Istio 本身由于具备大量的功能特性和各种外围集成，加上本身在架构上有非常多的模型抽象和解耦设计，导致概念多、术语多、细节多、入门不易。秀龙的这本书，可以帮助读者从基本知识开始，一步一步地掌握 Istio 的知识点，在细致学习理论知识的同时，又有大量的实际操作，非常适合作为 Istio 的入门指引。

本书中的部分内容，得益于作者本人对 Istio 的深入了解和实践积累，秀龙对 Istio 的优缺点有深刻的见解，提供的试用建议非常中肯，对有意在实际项目中尝试使用 Istio 的同学会有非常大的参考价值，值得对 Service Mesh 技术感兴趣，想详细了解 Istio 架构体系，并掌握 Istio 日常使用方法的同学阅读。

蚂蚁金服高级技术专家、Service Mesh 布道师　敖小剑

推荐序二

以 Kubernetes 为代表的云原生应用的生命周期管理的成熟，为使用 Kubernetes 部署和管理微服务打下了坚实的基础。作为云原生基础设施的一部分，Service Mesh 成为云原生演进的下一个重要方向。

秀龙作为畅销书《Kubernetes 权威指南：从 Docker 到 Kubernetes 实践全接触》和《Kubernetes 权威指南：企业级容器云实战》的作者，深刻理解 Kubernetes 在容器化应用编排管理方面的优势，也明白 Kubernetes 在微服务流量控制和管理方面的不足。Istio 作为继 Kubernetes 之后 Google 参与的云原生开源力作，极大地弥补了 Kubernetes 的不足。秀龙写的这本《深入浅出 Istio：Service Mesh 快速入门与实践》可谓适时出版。

在与 ServiceMesher 社区成员交流的过程中，我发现 Istio 中的众多概念及复杂配置令人望而生畏，不利于理解和学习。秀龙经常活跃于社区中，热心解答社区成员的众多疑问。本书是秀龙对 Istio 实战经验的总结，可以帮助读者快速入门和实践。

蚂蚁金服云原生布道师　宋净超

前言

为什么写作本书

Google 联合 IBM、Lyft 推出的 Istio，一经问世就受到了人们的普遍关注，其热度迅速攀升，将 Service Mesh（服务网格）的命名者 Linkerd 远远抛在身后，成为 Service Mesh 方案的代表项目。笔者从 Istio 问世开始，便和 ServiceMesher 社区及众多同样看好 Istio 的朋友一起，持续关注和参与 Istio 项目，并在该过程中对 Service Mesh 的技术生态及 Istio 自身的来龙去脉有了一定的认识。

在和社区互动的过程中，笔者看到有很多用户对这一新生事物一头雾水，因此斗胆写作本书，将 Istio 中的部分概念和案例重新整理，以快速入门的形式，对 Istio 的基础用法一一进行讲解，并在书末给出一些试用方面的建议。

本书读者对象

本书面向对服务网格技术感兴趣，并希望进一步了解和学习 Istio 的中高级技术人员，假设读者已经了解 Kubernetes 的相关概念并能够在 Kubernetes 上熟练部署和管理微服务。若希望全面、深入地学习 Kubernetes，可参考《Kubernetes 权威指南：

从 Docker 到 Kubernetes 实践全接触》和《Kubernetes 权威指南：企业级容器云实战》。

本书概要

本书围绕 Istio 对服务网格的概念、历史和能力，以各种实例为基础，进行了易于上手和理解的讲解。

前 3 章从微服务和服务网格的简短历史开始，讲述了服务网格的诞生过程、基本特性及 Istio 的核心功能，若对这些内容已经有所了解，则可以直接从第 4 章开始阅读。

第 4、5 章分别讲解了 Istio 的配置和部署过程。

第 6 章至第 9 章，通过多个场景来讲解 Istio 的常用功能。本书没有采用官方复杂的复杂 Book Info 应用案例，而是采用客户端+简单 HTTP 服务端的案例，读者随时都能在短时间内启动一个小的测试。

第 10 章结合了笔者的实践经验，为读者提供了 Istio 的一系列试用建议。

希望读者能通过本书快速地对 Istio 的功能特性有一个基本认识，理解其中的优点和不足，并进一步试用和评估。

相关资源

为方便大家学习和实践，本书提供了两个应用项目，其中，sleep 客户端应用项目的地址为 https://github.com/fleeto/sleep，flaskapp 服务端应用项目的地址为 https://github.com/fleeto/flaskapp。另外，笔者深度参与的 Istio 官方文档汉化项目也已上线，地址为 https://istio.io/zh。

致谢

感谢永远不知道笔者在做什么的崔夫人的大力支持；

感谢电子工业出版社工作严谨、高效的张国霞编辑，她在成书过程中对笔者的指导、协助和鞭策，是本书得以完成的重要助力；

另外，笔者在学习、交流 Istio 的过程中，从敖小剑、宋净超两位大咖，以及他们创办的 Service Mesher 社区（http://www.servicemesher.com/）所聚集的大量服务网格技术爱好者身上获得很多启发，在此一并致以诚挚的谢意。

读者服务

轻松注册成为博文视点社区用户（www.broadview.com.cn），扫码直达本书页面。

- ◎ **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）。
- ◎ **交流互动**：在页面下方 读者评论 处留下您的疑问或观点，与我们和其他读者一同学习交流。

页面入口：http://www.broadview.com.cn/35964

目录

1

第 1 章

服务网格的历史

要讨论服务网格（Service Mesh），就必须提到微服务。微服务（Microservices）自 2012 年被提出以来，就继承了传统 SOA 架构的基础，并在理论和工程实践中形成新的标准，热度不断攀升，甚至有成为默认软件架构的趋势。2014 年，马丁·福勒在 *Microservices* 一文中，对微服务做出了纲领性的定义，总结了微服务应该具备的特点，如下所述。

◎ 在结构上，将原有的从技术角度拆分的组件，升级为从业务角度拆分的独立运行的服务，这些服务具备各自的实现平台，并且独占自有数据，在服务之间以智能端点和哑管道的方式通信。

◎ 在工程上，从产品而非项目的角度进行设计，强调迭代、自动化和面向故障的设计方法。

微服务架构在很大程度上提高了应用的伸缩性，方便了部门或业务之间的协作，使技术岗位能够更好地引入新技术并提高自动化程度，最终达到减耗增效的目的。然而和所有新方法一样，微服务架构在解决老问题的同时，也带来了一些新问题，例如：

◎ 实例数量急剧增长，对部署和运维的自动化要求更高；

◎ 用网络调用代替内部 API，对网络这一不可靠的基础设施依赖增强；

◎ 调用链路变长，分布式跟踪成为必选项目；

◎ 日志分散严重，跟踪和分析难度加大；

◎ 服务分散，受攻击面积更大；

◎ 在不同的服务之间存在协作关系，需要有更好的跨服务控制协调能力；

◎ 自动伸缩、路由管理、故障控制、存储共享，等等。

David Wheeler 曾说过："Any problem in computer science can be solved by another layer of indirection." 可将其理解为：计算机科学中的所有问题都可以在新的层次里间接地解决。微服务架构产生的新问题，同样可以在微服务之外的新的层次里间接地解决。

为了解决微服务架构产生的一些问题，以 Kubernetes 为代表的容器云系统出现了。这类容器云系统以容器技术为基础，在进程级别为微服务提供了一致的部署、调度、伸缩、监控、日志等功能。

然而，除了进程本身的问题，微服务之间的通信和联系更加复杂，其中的观测、控制和服务质量保障等都成为微服务方案的短板，因此随着 Kubernetes 成为事实标准，Service Mesh 顺势登场。

自 Service Mesh 技术诞生以来，国内外出现了很多产品，下面选择其中几个重要的产品和事件，大概理理 Service Mesh 相关产品的发展情况。

1.1　Spring Cloud

诞生于 2015 年的 Spring Cloud 应该是 Service Mesh 的老前辈了。事实上，时至今日，Spring Cloud 仍是 Service Mesh 的标杆。

Spring Cloud 最早在功能层面为微服务治理定义了一系列标准特性，例如智能路由、熔断机制、服务注册与发现等，并提供了对应的库和组件来实现这些标准特性。到目前为止，这些库和组件已被广泛采用。

但是，Spring Cloud 也有一些缺点，例如：

◎ 既博采众家之长，也导致了一种散乱的局面，即用户需要学习和熟悉各组件的“方言”并分别加以运维，这在客观上提高了应用门槛；
◎ 需要在代码级别对诸多组件进行控制，包括 Sidecar 在内的组件都依赖 Java 的实现，这和微服务的多语言协作目标是背道而驰的；
◎ 自身并没有对调度、资源、DevOps 等提供相关支持，需要借助其他平台来完成，然而目前的容器编排事实标准是 Kubernetes，二者的部分功能存在重