

Serverless 架构

无服务器应用与AWS Lambda

Serverless Architectures on AWS :
With examples using AWS Lambda

[英] Peter Sbarski 著
[澳] Sam Kroonenburg

覃宇 李源 罗丽 译



Serverless 架构

无服务器应用与AWS Lambda

Serverless Architectures on AWS :
With examples using AWS Lambda

[英] Peter Sbarski 著
[澳] Sam Kroonenburg

覃宇 李源 罗丽 译

电子工业出版社
Publishing House of Electronics Industry
北京•BEIJING

内 容 简 介

无服务器是软件架构世界中的热门新话题，它充分利用大量的云平台服务，让开发者只需关注核心业务逻辑的实现。同时，它按需分配和使用资源的运行方式，在降低基础设施成本的同时还能获得强大的弹性。

本书的作者 Peter Sbarski 是最早完全拥抱无服务器架构的开发者之一，他将自己在应用无服务器架构中获得的宝贵经验总结成本书，呈现给广大的读者。更难能可贵的是，全书贯穿了 AWS Lambda 实例和大量相关的练习。读者一边阅读，一边动手完成练习，就可以循序渐进地掌握无服务器架构的核心技术。无论你是软件开发新手，还是架构师老兵，本书都是不可多得的了解无服务器架构的绝佳读物。

Original English Language edition published by Manning Publications, USA. Copyright © 2017 by Manning Publications. Simplified Chinese-language edition copyright © 2018 by Publishing House of Electronics Industry. All rights reserved.

本书简体中文版专有出版权由 Manning Publications 授予电子工业出版社。未经许可，不得以任何方式复制或抄袭本书的任何部分。专有出版权受法律保护。

版权贸易合同登记号 图字：01-2017-6970

图书在版编目（CIP）数据

Serverless 架构：无服务器应用与 AWS Lambda /（英）彼得·萨巴斯基（Peter Sbarski），（澳）萨姆·克鲁内伯格（Sam Kroonenburg）著；覃宇，李源，罗丽译．—北京：电子工业出版社，2018.8
书名原文：Serverless Architectures on AWS: With examples using AWS Lambda

ISBN 978-7-121-34268-4

I. ①S… II. ①彼… ②萨… ③覃… ④李… ⑤罗… III. ①移动终端—应用程序—程序设计 IV.①TN929.53

中国版本图书馆CIP数据核字(2018)第109343号

责任编辑：张春雨

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：24 字数：568.32 千字

版 次：2018 年 8 月第 1 版

印 次：2018 年 8 月第 1 次印刷

定 价：108.00 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。

译者序

过去二十年，硅谷的创业神话激励着 IT 从业者们前赴后继地投入创业大潮。而在这汹涌的大潮中，快速发展的技术和不断涌现的平台和服务也让创业的技术门槛越来越低。三五个好友一起想出一个点子，再有一个程序员，租一些云服务器，就可以大干一场了。然而，事情就真的是这么简单吗？有的团队开发的爆款应用几天之内流量即可迅速飙升，日活用户可以达到数百万；却由于最初架构设计时对扩展性的忽视，服务器不堪重负，页面加载开始变慢，甚至无法访问。开发团队加班加点想尽各种办法纠正架构的设计问题并扩展资源才让应用恢复正常，却为时已晚。用户早已失去了耐心，也让应用错失了最佳的扩张机会。有的团队却正好相反，它们会先花上好几个月时间精心地分析、设计、实现、测试可扩展的应用架构，甚至还会开发全套的自动化部署运维工具。然而，等到精心打磨的产品推出的那一天，市场上早已被同类竞品完全瓜分。

这样的案例在现实中屡见不鲜。市场的机会转瞬即逝，爆款的压力突如其来，而创业团队却被架构的扩展性和基础设施的规划运维束缚了手脚，疲于应付。除了提升团队自身的架构和运维能力之外，还要想办法降低基础设施运维的门槛。无服务器技术也许就是这些创业团队的希望。采用了无服务器技术架构之后，开发者使用现成的第三方服务并专注于应用的业务功能代码开发，不用再关心以服务器为代表的基础设置的规划、配置和维护。而应用运行时使用的服务完全是按需分配的，一旦出现吞吐量的激增，这些服务会自动快速扩展以达到可用性的要求。而应用使用的服务都是按照实际的使用量来计费的，这将大大降低基础设施投入的成本。在基础设施即服务、容器即服务和平台即服务的基础上，无服务器架构对云计算进行了更高层次的抽象，大大降低了运维的门槛和成本。

然而，无服务器架构并不是软件架构的银弹。在拥抱大量第三方服务的同时，也让系统架构和提供服务的特定供应商锁定在了一起，日后如果迁移不会那么轻松。第三方服务天然的分布式特质要求架构设计事先就要考虑网络故障和延迟，以及一致性问题。无服务器架构运行资源的限制，也会要求运算要简单，并能在短时间内完成。事件驱动和推送要求计算被拆成许多小巧的无状态函数，其集成和管理势必变得更加复杂。那么，无服务器架构有哪些典型的使用场景？它又可以运用哪些常见的设计模式？使用无服务器架构时有哪些需要特别留意的问题？它使用的工具和实践与传统的开发方式有什么区别呢？

本书的作者 Peter Sbarski 是最早实践无服务器架构的开发之一，他担任工程副总裁的 A Cloud Guru 这个在线视频教育服务就是完全基于无服务器架构创建的。贯穿本书的视频分享网站实例和练习几乎就是 A Cloud Guru 的迷你版，作者将他实施无服务器架构的经验毫无保留地呈现在我们面前。作者在本书的前两章总结了无服务器架构常见的使用场景和架构模式。从第 3 章开始直到第 9 章，分别介绍了无服务器技术中的身份认证和授权、函数计算、网关、对象存储及数据库等重要内容，所有内容的学习都是围绕着视频分享网站的练习展开的，十分生动。作者在第 10 章对全书进行了总结，并介绍了一些值得关注的工具、实践和趋势。本书末尾的附录则介绍了许多服务和工具的使用。

无服务器架构的话题始终无法回避 AWS 这个现象级产品。AWS 于 2014 年和 2015 年先后推出的 Lambda 和 API Gateway 两款产品将 AWS 的各种服务集成在了一起，让无服务器架构变成了触手可及的现实。本书的练习就需要使用 AWS 的重要服务 S3、Lambda 和 API Gateway。而一些更专业的领域也涌现了一批明星服务和产品，例如本书中介绍的身分认证和授权服务 Auth0 和实时数据产品 Firebase(已经被 Google 收购)。无服务器技术发展日新月异，本书英文版出版已经快两年，尽管译者在翻译时已经就其中一些过时内容进行了标注。但其中涉及 AWS 和其他第三方服务的讲解难免落后，读者可以结合这些产品的官方文档来进行阅读。AWS 的中文官方文档质量颇高，值得阅读。

本书的翻译初稿于 2018 年春节后结束，在此要感谢所有译者期间的辛勤付出以及家人的理解和支持。同时也要感谢编辑们的耐心和严谨，谢谢你们！

序一

Patrick Debois

DevOps Jedi、DevOpsDays 创始人

Small Town Heroes 首席技术官

编写专注于一个目标的程序，并尽可能把它做好，编写能够互相协同工作的程序——这是 Unix 哲学的核心思想，由 Unix 的设计者 Ken Thompson 首先提出。最近几年以 Google、Netflix、Uber 和 Airbnb 为代表的公司已经证明，在现代的分布式系统中，可以简单地将这些核心思想中的程序这个词替换成服务。这种思想的最新变化：无服务器计算，宣示了托管服务和自我管理基础设施的智能组合是如何显著改善开发时间和运营成本的。

本书很好地通过一系列实用的落地案例研究搭配了新兴的无服务器设计模式的内容，初学者和实干家都是本书理想的目标读者。无服务器是一门崭新的学科，作者成功地兼顾了广泛的主题，同时还不失深度和重点。本书饱含着作者真挚的热情、细致的洞察力，以及值得分享的知识宝藏。

无服务器计算需要你改变构建软件架构的方式，随着许多范式的改变，有一些习惯你必须忘记。在热情洋溢地拥抱新技术的同时，作者也用了不小的篇幅来罗列这些新架构类型的优势和局限。作为惊喜，它还总结了自己运营真实无服务器架构旅程中的洞见。这种“实践出真知”的态度展示了无服务器所带来的丰厚回报，这将帮助你的企业专注并取得成功。

序二

Donald F. Ferguson 博士

Seeka TV 联合创始人、首席技术官

哥伦比亚大学计算机科学系客座教授

很多技术都在颠覆应用程序的开发、测试和交付。云计算和各种形式的“某某即服务”（as-a-service）就是这些重新定义应用程序开发和交付的例子。许多团队和项目在尝试引入新技术的时候都特别挣扎，有时甚至还会失败。而失败的罪魁祸首就是把这些截然不同的技术强加在现有的应用程序架构和编程模型上。精心设计、实现和交付的跨云应用程序和传统应用程序有着天壤之别。本书在介绍新的应用程序架构方面做得十分出色，还提供了如何成功应用无服务器技术的详尽而实用的指导。

基础设施即服务（Infrastructure as a Service, IaaS）、软件即服务（Software as a Service, SaaS）以及平台即服务（Platform as a Service, PaaS）是内部部署应用程序和基础设施架构的云平台版本。这些模型可以提供价值但却无法彻底释放云平台的潜力。SaaS 提供了关于业务问题的不完全标准的解决方案，但无法快速地开发和交付更多有针对性的应用程序。IaaS 和 PaaS 提供了资源使用效率，却无法消除配置和管理软件服务器基础设施的成本。这些模型都无法借用 Web 可调用 API 的大爆发，基于它们的 API 经济已经渐成气候。无服务器架构是唯一一种架构，这种架构可以消除软件服务器成本、提供快速开发和交付灵活性，并管理有针对性的专注的云平台应用程序。

本书的第一部分——踏上征途为创建无服务器架构打下基础。这部分说明了这种新架构的基本特性和优势，包括该技术优缺点的清晰解释和选择指导。同样重要的是，这部分还介绍了架构设计模式。成功采纳转型中的计算技术的唯一重要因素就是通过应用这些设计模式实现来了解其最佳实践。这部分是在一个真实解决方案的上下文中解释了这些模式，这个解决方案就是作者使用无服务器架构实现的：“一

言不合就秀代码”。作者的实践经验和成功经历也是我推荐本书的主要原因。

人们往往错误地将无服务器等同于某一项具体的技术。比如，AWS Lambda 函数。但无服务器架构范围更广，包括 UI 设计、发布 / 订阅基础设施、工作流 / 编排、活动数据库、API 网关和管理，以及数据服务。总的来说，这些技术势不可挡。本书介绍了做出贡献的技术所扮演的角色以及发挥的作用，详细地介绍了如何使用 Amazon Web Services 中的这些技术集体实现来构建一个可以工作的应用程序。入门手册和教程是能够重复并可靠地使用该技术的核心。

数据层和安全性是任何应用程序都不可回避的最困难的两个架构领域。本书对两个话题都有详尽的阐述。这部分内容解释了概念（例如身份认证和授权），将概念落实到了应用程序场景（例如 Web 应用程序）中，并提供了如何设计实现安全性和数据层的具体而详尽的例子。这些细节还包括使用 Auth0 和 Google Firebase 这些非 AWS 技术的示例。

我的公司也正在使用 AWS 和无服务器架构来构建解决方案，我发现这本书和作者的其他材料对我们的进步至关重要。我在哥伦比亚大学教授计算机科学的高级课程，该课程主要关注互联网应用程序和跨云应用程序。本书的内容是我所教授的大部分内容的基础。我的经验表明，本书是一本释放云计算潜力不可取代的关键书籍。真实应用程序上下文中关于 AWS 的详细信息是无价的，这些概念和模式适用于使用任何技术的任何无服务器解决方案。

前言

我是从 Sam Kroonenburg 那儿第一次听说 AWS Lambda 的。那时 Lambda 刚刚发布，但 Sam 对它的前景已经十分看好了。他和我谈起了在云端执行函数、AWS 中自动化的巨大潜力，以及基于事件的工作流的开发。一切都如此令人着迷，充满了无限可能。不用配置和管理基础设施就能执行我的代码，这个想法看起来很酷而且马上就能梦想成真。作为软件工程师，我一直都想更专注于架构和代码，而不是基础设施、运维和系统管理。我用 Amazon Web Services 来实现这一切的机会来了。

几个月之后，API Gateway 横空出世，并解决了当时 Lambda 最大的问题。使用标准的 HTTP 请求调用 Lambda 函数成为可能。无须服务器就创建快速且可扩展的应用程序后端，这个梦想在我们眼前变成了现实。我的第一个无服务器项目是由 Sam Kroonenburg 发起的 A Cloud Guru，现在已经发展成了一个大型的学习管理平台。这个平台是完全无服务器的，运行成本非常低并可以快速地迭代。我们只需要专注于添加业务价值和新特性，而不需要担心基础设施管理和复杂的运维，这使我们的工作更有意思，而且这个平台还可以不可思议地扩展。

在构建 A Cloud Guru 的过程中，我们也意识到无服务器并不是在 Lambda 中运行代码这么简单，它还要用到第三方服务和产品。我们使用了托管身份认证服务和托管数据库，节省了数周甚至数月的开发时间。我们识别出了这个系统中的一些重要但不必自己构造的部分，比如支付处理和客户消息。这些部分十分重要但我们不用重新造轮子。我们找到了完美契合我们无服务器后端的不错的第三方服务，并将它们和系统其他部分集成到了一起。

第三个关键的话题当然就是选择正确的设计模式和架构了。我们发现，事件驱动的架构天然地适用于无服务器应用程序，而我们也致力于让我们的系统完全由事

件驱动。我们还考虑到了安全性、可靠性以及可扩展性，还有如何组合函数和后端服务来达到更好的效果。

自从帮助构建了第一个大规模无服务器应用程序并回顾了其他无服务器系统之后，我越来越清晰地意识到：可扩展的云函数、可靠的第三方服务，以及无服务器架构和设计模式的组合就是云计算演进的未来。未来几年，我们将看到初创公司和老牌企业都将采用无服务器架构，这将帮助它们不断创新并赶超竞争对手。本书是对未来趋势的展望，也是如何起步的指导手册。我希望你喜欢本书并和我一起进入探索无服务器的旅程。

致谢

没有来自我的同事、同行、家人和朋友的鼓励、反馈和支持，本书是不可能完成的。我很幸运地遇到了身边这些才华横溢的人，他们倾听并提出了宝贵的意见和观点。

我要感谢太多帮助过我的人，但有一些必须提及的名字。首先，我要感谢我的编辑 Toni Arritola，她为本书的写作带来了绝佳体验。Toni 对本书的结构、语言和叙述提出了深刻的反馈，这非常有帮助。她对细节的关注、快速的回应以及热情都是无与伦比的。

Serverless Framework 的创造者 Austen Collins 主要贡献了本书中 Serverless Framework 的内容。没有谁比它的创造者更适合来撰写关于这个框架的内容，非常感谢 Austen 志愿奉献了他的时间和精力。我希望每个阅读了本书——特别是阅读了 Austen 精彩论述——的读者都能花时间学习、理解并采用 Serverless Framework。

我还要感谢 Sam Kroonenburg，他引领我走上了无服务器的道路，并在本书的写作过程中给予我独到的反馈和评论。Sam 对 AWS Lambda 的热情，以及对架构和设计的想法激励着我第一时间把它们记录下来。另一个特别要感谢的人是 Ryan Brown，他阅读了本书并给予了面面俱到的评论和建议。没有 Ryan 阅读本书并提出深刻的反馈，这本书将黯然失色。

我还要感谢 Donald Ferguson 和 Patrick Debois 为本书撰写了特别的序言。Donald 和 Patrick 为软件工程和无服务器社区做了很多工作。我非常钦佩他们的成就，也非常感谢他们抽出时间参与本书。

我还要感谢其他给过我反馈和鼓励的人。他们是 Ryan Kroonenburg、Mike Chambers、John McKim、Adrian Cantrill、Daniel Parker、Allan Brown、Nick

Triantafillou、Drew Firment、Neil Walker、Alex Mackey 和 Ilia Mogilevsky。我要感谢帮助完成本书的来自 Manning 的 Mike Stephens、Kostas Passadis 和 David Fombella Pombal。此外，这份致谢中不能少了在本书出版过程中慷慨阅读文字并提出建议的校审团队，包括 Alain Couniot、Andy Wiesendanger、Colin Joyce、Craig Smith、Daniel Vásquez、Diego Santiviago、John Huffman、Josiah Dykstra、Kent R. Spillner、Markus Breuer、Saioa Picado Fernández、Sau Fai Fong、Sean Hull 和 Vijaykumar Borkar。

最后，我要表达对家人的感谢，包括我的父亲和兄长，以及其他所有亲人，感谢你们在每次聚会时寻找内在力量倾听我谈论本书。我要感谢在整个写作过程中给予我极大帮助的 Durdana Masud，从积极的鼓励和启发开始，到看着不断变化的调色板努力帮助我为书稿中使用的原始图片选择颜色。谢谢你们。

关于本书

无论你是初学者还是专家，无论是刚开始 IT 职业生涯还是有数年的工作经验，这本书都将带你进入无服务器架构的旅程。你会学习关键的设计模式，了解使用无服务器方法的优劣，并使用 AWS Lambda、API Gateway、Elastic Transcoder、S3、Auth0 和 Firebase 构建你自己的视频分享网站。你还会学习许多 AWS 知识，以及那些值得推荐的用来管理和发布你的无服务器应用程序的框架。

本书分为三部分。第一部分介绍了基本的无服务器原理，探讨了关键的架构和设计模式。你会开始使用 AWS Lambda 构建第一个事件驱动的管道，并学习关键的 AWS 服务。例如，无所不在的万能的 Identity and Access Management 服务。

第二部分重点关注了身份认证和授权、AWS Lambda 以及 API Gateway。这部分的所有章节对理解和构建无服务器应用程序都很重要。读完这部分并完成示例之后，你将彻底掌握无服务器应用程序所需的关键技术。

第三部分讨论了构建真实应用程序所需的其他服务和架构。这部分的关键重点是文件和数据存储，分别使用 S3 和 Google 的 Firebase 实现。最后一章补充了更多技术和服务的信息，它们可以用来扩展你的无服务器应用程序。

在这本书的末尾，你会发现七个附录，它们为你提供了关于各种主题的额外信息。例如，最后一个附录涵盖了 Serverless 框架以及 Serverless Application Model（无服务器应用程序模型，SAM）。你应该仔细阅读并尝试这个附录中的步骤。

AWS 和其他服务（如 Auth0 和 Firebase）的发展非常迅速，所以当你阅读本书时，如果某些截图或操作有所不同，请不要惊讶。无服务器事件驱动架构的基本原理会保持不变，但是一些微末的细节，比如按钮的位置或是按钮的标签，可能会随着时间不断变化。本书适用于那些 AWS 和云计算的新手开发者和解决方案架构师，

也适合云计算开发老兵。我希望你会发现一种新方式，来构建成本更低、扩展性更好和趣味更多的应用程序！

代码约定

本书提供了很多代码示例。有些散落在正文中，有些以独立的代码清单形式提供。代码都使用了等宽字体，一眼就可以看出来。

获取源代码

本书中所有源代码都可以在 Manning 的网站 <https://manning.com/books/serverless-architectures-on-aws> 和我的 GitHub 仓库 <https://github.com/sbarski/serverless-architectures-aws> 中找到。我喜欢 GitHub，如果你愿意贡献源代码，请提出 Pull Request。如果你发现了问题，也请提交 Issue。

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/>

关于作者

Peter Sbarski 是 A Cloud Guru 的工程副总裁，也是 Serverlessconf 的组织者，Serverlessconf 是全球首个专门针对无服务器架构和技术的会议。他喜欢举办面对面的工作坊，时常撰写一些关于无服务器架构的博客。Peter 拥有丰富的 IT 工作经验，领导过专注于 Web 和 AWS 云技术的大型企业解决方案团队。他的专长包括后端架构、微服务和系统编排。

Peter 拥有澳大利亚莫纳什大学计算机科学博士学位，读者可以在 Twitter（@sbarski）和 GitHub（<https://github.com/sbarski>）上关注他。

关于封面

本书的封面画像题为《Man from Stupno/Sisak, Croatia》。图片来自于 19 世纪中期由尼古拉·阿森诺维奇 (Nikola Arsenović) 绘制的克罗地亚传统服饰画册的复制品，由克罗地亚斯普利特民族博物馆于 2003 年出版。这些绘画由斯普利特民族博物馆一位热心馆员提供，博物馆位于中世纪罗马帝国时期城镇中心的核心地带：戴克里先宫废墟。公元前 304 年左右戴克里先皇帝 (Emperor Diocletian) 退位后居住于此。这本画册包括了来自克罗地亚不同地区人物的精美彩色画像，伴随着其服饰和日常生活的描述。

过去两百年间，人们的着装风格和生活方式不断发生着变化，如此丰富的地区多样性也逐渐消失。现在很难区分不同大陆的居民，更不用说只有数英里之隔的村庄或城镇了。也许我们在追求更加精彩的个人生活——毫无疑问是更多样化和快节奏的科技生活——的同时已经忽略了文化多样性。Manning 以呈现两个世纪前丰富多彩的地区生活的书籍封面来颂扬计算机产业的独创性和首创性，并让像这本画册这样的古籍和藏品中的绘画作品重现生机。

目录

序一	XIII
序二	XV
前言	XVII
致谢	XIX
关于本书	XXI
关于作者	XXIII
关于封面	XXIV
第 1 部分 踏上征途	1
1 无服务器初探	3
1.1 来龙去脉	4
1.1.1 面向服务的架构与微服务	5
1.1.2 软件设计	6
1.2 无服务器架构原则	9
1.2.1 根据需要使用计算服务执行代码	9
1.2.2 编写单一用途的无状态函数	10
1.2.3 设计基于推送的、事件驱动的管道	10
1.2.4 创建更厚、更强大的前端	11