

O'REILLY®



# 学习 Chef



Learning Chef 云时代的配置管理与自动化运维技术

Mischa Taylor Seth Vargo 著  
闫诺 译

清华大学出版社

---

# 学习 Chef

云时代的配置管理与自动化运维技术

*Mischa Taylor Seth Vargo* 著

闫诺 译

Beijing • Cambridge • Farnham • Köln • Sebastopol • Tokyo

**O'REILLY®**

O'Reilly Media, Inc. 授权清华大学出版社出版

清华大学出版社  
北京

## 内容简介

Chef作为DevOps运动的重要工具之一，用于快速实现云端自动化配置管理。全书共16章，主题涉及Chef配置管理、Ruby及其语法、如何写Chef recipe，如何用Test Kitchen管理沙盒测试环境，如何用Chef客户端管理单个节点，Chef cookbook如何写和如何用，如何用Chef服务器管理多个节点，搜索、数据包、角色、开发环境和测试等都有介绍。

本书是一本介绍Chef基础架构自动化平台的入门书，有比较强的实操性，很适合刚开始接触Chef、配置管理和自动化编程的读者阅读和参考。

Copyright © 2015 Mischa Taylor and Seth Vargo. All rights reserved.

Authorized Simplified Chinese translation edition, by O'Reilly Media, Inc., is published by Tsinghua University Press, 2016. Authorized translation of the original English edition, 2015 O'Reilly Media, Inc., the owner of all rights to publish and sell the same.

All rights reserved including the rights of reproduction in whole or in part in any form.

本书之英文原版由O'Reilly Media, Inc.于2015出版。

本中文简体翻译版由O'Reilly Media, Inc.授权清华大学出版社于2016年出版。此翻译版的出版和销售得到出版权和销售权的所有者——O'Reilly Media, Inc.的许可。

版权所有，未经书面许可，本书的任何部分和全部不得以任何形式复制。

北京市版权局著作权合同登记 图字：01-2015-2863

本书封面贴有清华大学出版社防伪标签，无标签者不得销售。

版权所有，侵权必究。侵权举报电话：010-62782989 13701121933

### 图书在版编目（CIP）数据

学习Chef：云时代的配置管理与自动化运维技术 /（美）泰勒（Taylor, M.），（美）法戈（Vargo, S.）著；闫诺译. -- 北京：清华大学出版社，2016

书名原文：Learning Chef: A Guide To Configuration Management and Automation  
ISBN 978-7-302-43390-3

I. ①学… II. ①泰…②法…③闫… III. ①虚拟处理机 IV. ①TP338

中国版本图书馆CIP数据核字(2016)第074814号

责任编辑：文开琪

封面设计：Ellie Volckhausen，张健

责任校对：周剑云

责任印制：沈露

出版发行：清华大学出版社

网 址：<http://www.tup.com.cn>, <http://www.wqbook.com>

地 址：北京清华大学学研大厦A座 邮政编码：100084

社总机：010-62770175

邮 购：010-62786544

投稿与读者服务：010-62795954，[jsjic@tup.tsinghua.edu.cn](mailto:jsjic@tup.tsinghua.edu.cn)

质量反馈：010-62772015，[zhiliang@tup.tsinghua.edu.cn](mailto:zhiliang@tup.tsinghua.edu.cn)

印刷者：北京鑫丰华彩印有限公司

装订者：三河市漂源装订厂

经 销：全国新华书店

开 本：178mm×233mm 印 张：21.25

字 数：349千字

版 次：2016年7月第1版

印 次：2016年7月第1次印刷

印 数：1~3000

定 价：79.00元

# O'Reilly Media, Inc.介绍

O'Reilly Media通过图书、杂志、在线服务、调查研究和会议等方式传播创新知识。自1978年开始，O'Reilly一直都是前沿发展的见证者和推动者。超级极客们正在开创着未来，而我们关注真正重要的技术趋势——通过放大那些“细微的信号”来刺激社会对新科技的应用。作为技术社区中活跃的参与者，O'Reilly的发展充满了对创新的倡导、创造和发扬光大。

O'Reilly为软件开发人员带来革命性的“动物书”；创建第一个商业网站（GNN）；组织了影响深远的开放源代码峰会，以至于开源软件运动以此命名；创立了Make杂志，从而成为DIY革命的主要先锋；公司一如既往地通过多种形式缔结信息与人的纽带。O'Reilly的会议和峰会集聚了众多超级极客和高瞻远瞩的商业领袖，共同描绘出开创新产业的革命性思想。作为技术人士获取信息的选择，O'Reilly现在还将先锋专家的知识传递给普通的计算机用户。无论是通过书籍出版，在线服务或者面授课程，每一项O'Reilly的产品都反映了公司不可动摇的理念——信息是激发创新的力量。

## 业界评论

“O'Reilly Radar博客有口皆碑。”

——Wired

“O'Reilly凭借一系列（真希望当初我也想到了）非凡想法建立了数百万美元的业务。”

——Business 2.0

“O'Reilly Conference是聚集关键思想领袖的绝对典范。”

——CRN

“一本O'Reilly的书就代表一个有用、有前途、需要学习的主题。”

——Irish Times

“Tim是位特立独行的商人，他不光放眼于最长远、最广阔的视野并且切实地按照Yogi Berra的建议去做了：‘如果你在路上遇到岔路口，走小路（岔路）。’回顾过去Tim似乎每一次都选择了小路，而且有几次都是一闪即逝的机会，尽管大路也不错。”

——Linux Journal

## 作者介绍

---

**Mischa Taylor**是一名Chef咨询师，这家位于西雅图的高成长初创公司，主营业务为打造Chef平台，快速实现自动化配置，并将业务过程移到云端。他的兴趣聚焦于打造高质量的产品和提升组织内部工程实践的生产力。他能说会道且擅长指导，同时在软件开发和神经网络计算方面也颇有专研和造诣。

**Seth Vargo**是HashiCorp的软件工程师和开源倡导者。他先后任职于Chef (Opscode), CustomInk和匹兹堡的多家初创公司。他对技术和组织文化有很强烈的爱好。在写代码和从事开源相关工作之余，他喜欢参与本地用户组和会议。他是匹兹堡DevOps Days的组织者。

## 译者介绍

---

**闫诺**，Chef创始工程师。在Chef软件公司任职期间作为主要工程师研发了首个版本的托管企业Chef服务。2011年离开Chef参与创办消费者推荐搜索引擎公司Livestar，2013年并入以图片为基础的推荐、探索引擎公司Pinterest。在Pinterest任职期间，他参与构建综合性实时认知及响应平台，后者服务于Pinterest的自动化垃圾内容处理、通知推送服务、广告服务及内容推荐服务等。2015年底，他离开Pinterest，目前正在为解决新的工程挑战而忙得乐此不疲。

闫诺毕业于美国华盛顿大学（西雅图），大学主修应用及计算数学。平时对世界各地的文化和各类艺术很感兴趣。工作之余，时常为朋友圈内有需要的创业公司提供帮助，积极参与鼓励大学生学习科学、技术、工程及数学的公益项目。

## 封面介绍

---

《学习Chef》封面上的鸟是沃氏蜜，也称褐背蜜（*Protodiscus regulus*）。

沃氏蜜是一种小型鸟类，生活在南部非洲的金合欢丛生草原以及其他半湿润的区域。这种不显眼的小鸟大多是这里的留鸟，主食介壳虫，以巢寄生的方式繁殖后代：成鸟偷偷将卵产在其他鸟类的巢中，沃氏蜜通常偏爱扇尾莺或其他莺类的球形小窝，它们自己可以省去抚育后代的各种投入。

沃氏蜜得名于瑞典自然科学家约翰·奥古斯都·沃尔贝里。1838年至1856年间，沃尔贝里在南部非洲考察。除了沃氏蜜外，还有一种鹰、一种鸬鹚、一种果蝠、一种蛙和一种树都以他命名。在1856年，他在现今博茨瓦纳境内的塔马拉卡内河畔遭遇一头受伤的大象，不幸被踩踏致死。在他的死讯传到瑞典前，他刚刚被当选为瑞典皇家科学院的成员。这也使他成为瑞典皇家科学院唯一一名死后当选的成员。

O'Reilly公司出版的许多书中，封面上的动物都已濒危。对于我们这个世界，这些动物中的每一种都是独一无二，不可或缺的。如果您愿意伸出援手与我们一起拯救这些美丽的生灵，请访问<http://animals.oreilly.com>。

封面图片来自《卡塞尔自然史》。

---

# 译者序

二十几年前，当互联网刚刚开始逐渐盛行时，人们并没有一个很好的配置管理和基础架构自动化解决方案。通常，公司中的若干系统工程师和管理员掌握一些配置脚本，然后通过运行一系列的脚本来配置各种服务器。公司里只有他们知道什么情况下应该运行哪些脚本。准备服务器以及维护这些脚本的时间成本很高。然而，在当时这并不是什么大问题，因为当时基于网络的应用程序部署并不像现在这么频繁，往往几个星期、几个月甚至半年才会更改一次生产环境。因此花几个星期来准备各种服务器虽然不理想但也无大碍。

随着互联网的迅速发展，不久问题就来了。各大公司开始将更多业务转向互联网，需要更快进行产品迭代。在应用程序开发结束后花费很长时间来准备服务器以及部署逐渐不可以接受。由于当时的应用开发工程师通常并不负责系统运维，在工程师的开发环境中完美运行的应用经常在运维人员负责的生产环境中迟迟无法配置成功。事实上，很多大公司在这上面浪费了很多的时间，从而耽误了产品发布的最佳时期，也无法以最快的速度给用户带来更大价值。

因此，诸如CF Engine及Puppet的配置管理和基础架构自动化工具先后诞生了。这些工具极大提高了系统运维效率，就像应用程序本身一样，工程师们以代码来定义基础架构理想的配置，然后这些工具将基础架构自动配置成理想的配置。这在当时极大地提高了互联网公司的运维效率。然而，两种工具都基于收敛（convergence）的理论，也就是并不需要定义依赖性和运行顺序。比如，在某种服务器上若干服务需要配置，服务A依赖于服务B（意味着如果服务B不存在，安装服务A则会失败），在这些工具中你并不需要指定这样的依赖性，工具会不断尝试执行这些操作（安装服务A、安装服务B），而总有一次它会先安装了服务B再安装服务A导致配置成功。相比于早期的CF Engine，Puppet在这方面更优化一些，支持在代码中定义依赖性，并将所有需要运行的操作顺序生成有向

图（Directed Graph）。然而，根据同样的服务器配置代码生成的有向图不保证唯一性，也就是说很可能十次有九次都可以成功完成运行来配置服务器，而在关键时刻其生成了不同的有向图而改变了运行顺序，而失败。因此，诸如CF Engine及Puppet等工具的最大不足是：①需要多次运行才能保证系统配置完成，因而花费更多的时间成本；②无法轻易地像测试应用程序一样进行自动化集成测试，同样因为一次运行并不保证和最终的状态相同，所以可能发生测试通过，但在生产环境运行时无法成功；或测试失败，但其实可以运行成功等情形。

配置管理和基础架构自动化工具Chef的创始人之一Adam Jacob是资深的系统运维工程师并活跃于开源社区，他在自己使用类似CF Engine以及Puppet等工具十余年的过程中体会到以上不便，并了解到社区中很多工程师有这样的痛点，因此在2008年写了第一个版本的Chef。Chef允许用户直接定义运行顺序，并在每个代码文件（称作配方单）中按顺序执行其内容。这保证了可预测性。因为有明确和固定的运行顺序，使用者可以清楚知道Chef自动化配置基础架构时所执行的步骤。如果哪个步骤有问题，Chef永远会在同一个位置报错，不会发生在不更改代码时有时候运行成功、有时候运行失败的情况。这使工程师提高了效率，并可以为Chef代码写自动化集成测试，保证基础架构配置代码不存在错误，在任何时刻都能够迅速部署基础架构。

Chef的理念随后受到大量工程师的追捧，Adam和来自亚马逊的Jesse Robbins、IBM的Nathan Haneysmith以及微软的Barry Steinglass离开自己的公司并全职投入到将Chef做成世界上最好的配置管理、基础架构自动化工具的使命。2009年初，Chef任命原亚马逊云服务EC2的创始工程师、时任微软工程总监的Christopher Brown作为负责工程的副总裁。2月份，Barry和Christopher找到了我，谈话中我了解到Chef的技术挑战、发展方向以及创始人的愿景。4月份，我作为创始团队第一个全职工程师、8号员工加入Chef软件公司（当时称为Opscode公司），参与所有产品（开源Chef、托管企业Chef、私有企业Chef及Chef超级市场）的研发工作中。

现在，Chef管理着全球各行各业数不胜数的大、中、小企业的基础架构，已成为最受欢迎的基础架构自动化工具之一。随之，掌握Chef技能的工程师也在人才市场炙手可热。然而，对于Chef的中文版教学内容仍不多见。希望本书能够帮助希望了解Chef的初学者入门。

最后，感谢出版社将本书引入中文市场以及付出的努力；感谢好友张翌旻先生无私帮忙翻译后记中对封面濒临灭绝的鸟类的介绍；也感谢家人对我利用工作之余完成本书的翻译工作的支持和理解。

闫诺  
于美国旧金山

# 目录

|                           |    |
|---------------------------|----|
| 前言 .....                  | 1  |
| 第1章 配置管理和Chef.....        | 7  |
| 什么是配置管理 .....             | 7  |
| 为什么需要配置管理工具来实现IT自动化 ..... | 8  |
| Chef是什么 .....             | 9  |
| 为什么企业需要Chef.....          | 9  |
| 接下来要做什么 .....             | 12 |
| 第2章 配置Chef开发环境 .....      | 15 |
| 安装一个适合程序员使用的文本编辑器 .....   | 16 |
| Chef开发工具 .....            | 17 |
| 在Linux下安装Chef开发工具 .....   | 18 |
| 在Windows下安装Chef开发工具 ..... | 32 |
| 小结 .....                  | 41 |
| 第3章 Ruby和Chef语法 .....     | 43 |
| Ruby概述 .....              | 43 |
| Ruby语法和案例 .....           | 44 |
| Chef语法和案例 .....           | 54 |
| 第4章 如何写第一个Chef配方单 .....   | 61 |
| 创建代码的目录结构 .....           | 61 |

|                                        |            |
|----------------------------------------|------------|
| 写下第一个Chef配方单 .....                     | 62         |
| 验证第一个Chef配方单 .....                     | 63         |
| 用配方单指定理想配置 .....                       | 65         |
| 卸载时，须明确指定不要做什么 .....                   | 69         |
| 小结 .....                               | 71         |
| <b>第5章 用Test Kitchen管理沙盒测试环境 .....</b> | <b>73</b>  |
| 安装Vagrant和VirtualBox .....             | 75         |
| 启动自己的第一个虚拟机 .....                      | 79         |
| YAML概览 .....                           | 84         |
| 用.kitchen.yml配置Test Kitchen .....      | 86         |
| 小结 .....                               | 87         |
| <b>第6章 用Chef客户端管理节点 .....</b>          | <b>89</b>  |
| 什么是节点 .....                            | 89         |
| 在一个节点上创建沙盒环境 .....                     | 90         |
| 用Test Kitchen在节点上安装Chef客户端 .....       | 91         |
| 第一次运行Chef客户端 .....                     | 93         |
| Chef客户端的三种模式 .....                     | 95         |
| 命令行工具Ohai .....                        | 96         |
| 访问节点信息 .....                           | 98         |
| 小结 .....                               | 100        |
| <b>第7章 撰写和使用菜谱 .....</b>               | <b>103</b> |
| 你的第一个菜谱：每日消息 .....                     | 103        |
| 你的第一个菜谱：每日消息（使用Chef开发包） .....          | 104        |
| Cookbook_file资源简介 .....                | 106        |
| 你的第一个菜谱：每日消息（Chef客户端） .....            | 107        |
| Cookbook_file资源简介 .....                | 109        |
| 第一次运行Chef .....                        | 110        |
| 验证结果 .....                             | 112        |
| 剖析Chef运行 .....                         | 113        |
| 菜谱结构 .....                             | 116        |

|                                            |            |
|--------------------------------------------|------------|
| 必须了解的四个资源 .....                            | 119        |
| Apache菜谱：手把手教你创建菜谱 .....                   | 119        |
| 小结 .....                                   | 130        |
| <b>第8章 属性 .....</b>                        | <b>133</b> |
| Motd-Attributes菜谱 .....                    | 134        |
| 设定属性 .....                                 | 136        |
| 属性优先级基础 .....                              | 138        |
| Include_Recipe .....                       | 139        |
| 属性优先级 .....                                | 142        |
| 属性排错 .....                                 | 143        |
| 小结 .....                                   | 146        |
| <b>第9章 用Chef服务器同时管理多个节点 .....</b>          | <b>147</b> |
| 手动安装企业Chef .....                           | 149        |
| 使用菜谱来自动化安装企业Chef服务器 .....                  | 151        |
| 幂等性简介 .....                                | 154        |
| 配置企业Chef服务器 .....                          | 159        |
| 测试连接 .....                                 | 165        |
| 准备一个新节点 .....                              | 166        |
| 用Chef Solo配置Chef服务器 .....                  | 173        |
| 小结 .....                                   | 174        |
| <b>第10章 社区以及Chef-Client菜谱 .....</b>        | <b>175</b> |
| 使用社区菜谱 .....                               | 175        |
| Chef-Client菜谱 .....                        | 177        |
| Knife Cookbook Site插件 .....                | 180        |
| 使用Knife Cookbook Site搜索社区菜谱 .....          | 181        |
| 通过Knife Cookbook Site管理Chef服务器中的社区菜谱 ..... | 182        |
| Chef-Client配方单 .....                       | 184        |
| 配置Knife使用生产环境SSL设置 .....                   | 186        |
| 配置Chef-Client使用生产环境的SSL设置 .....            | 188        |
| 小结 .....                                   | 196        |

|                                  |            |
|----------------------------------|------------|
| <b>第11章 Chef Zero</b>            | <b>197</b> |
| Test Kitchen和Chef Zero           | 197        |
| 用Chef-Playground在宿主机上运行Chef-Zero | 200        |
| 小结                               | 205        |
| <b>第12章 搜索</b>                   | <b>207</b> |
| 从命令行搜索                           | 207        |
| 用Knife从命令行搜索                     | 208        |
| 使用Test Kitchen从配方单中搜索            | 211        |
| 小结                               | 215        |
| <b>第13章 数据包</b>                  | <b>217</b> |
| 在配方单中使用数据包项目的数据创建本地用户            | 222        |
| 验证用户                             | 226        |
| 加密数据包                            | 227        |
| 小结                               | 233        |
| <b>第14章 角色</b>                   | <b>235</b> |
| 创建一个网页服务器角色                      | 236        |
| 属性和角色                            | 238        |
| 角色和搜索                            | 239        |
| 角色菜谱                             | 241        |
| 小结                               | 242        |
| <b>第15章 环境</b>                   | <b>243</b> |
| 创建一个开发环境                         | 244        |
| 属性和环境                            | 245        |
| 完整实例                             | 247        |
| 小结                               | 258        |
| <b>第16章 测试</b>                   | <b>259</b> |
| 关于测试                             | 259        |
| 重温Apache菜谱                       | 262        |

|                            |            |
|----------------------------|------------|
| 使用Serverspec进行自动化测试 .....  | 266        |
| 使用Foodcritic进行自动化测试 .....  | 283        |
| 使用ChefSpec进行自动化测试 .....    | 287        |
| 小结 .....                   | 295        |
| <b>第17章 结语 .....</b>       | <b>297</b> |
| 词汇表 .....                  | 298        |
| <b>附录A 开源Chef服务器 .....</b> | <b>301</b> |
| <b>附录B 托管企业Chef.....</b>   | <b>323</b> |

# 前言

## 这是一本什么书

《学习Chef》是一本针对学习Chef基础架构自动化平台的入门书籍。这是一本面向Chef、配置管理和自动化编程初学者的书籍。

通过使用Chef，你将可以对你的基础架构内应用程序的配置、打包和部署以代码的形式表现出来。我们会展示为什么通过代码来为你的基础架构建模会很轻松、自然地带来可在任何时候重建整个基础架构的便利，而手动运行命令则不能如此。一旦你理解了这个用代码来记录和建模整个基础架构的蓝图，就可以在几分钟或小时内建立或重建你的整个应用基础架构，而不是像手动进行那样需要几天、几周或更长时间来完成。本书为入门书籍，你并不需要拥有任何Chef、基础架构自动化或编程的经验。

首先，我们会展示如何在你的个人开发用电脑上配置Chef的开发环境。你将来会使用此开发环境来撰写和测试Chef代码。俗话说“实践出真知”，学写代码也如此。因此我们会在本书中尽早引导你开始实践写代码，并在每个新章节介绍一个新的概念和相应的Chef工具来慢慢引导你前进。我们也会提供动手实例来帮助你加深对概念的理解，实践动手写代码。

在你读完本书之后，你将对Chef和基础架构自动化的基本概念有较好的理解。然后可以继续阅读以下两本对Chef更深入探讨的书籍：Stephen Nelson-Smith写的*Test-Driven Infrastructure with Chef*（第2版）和Jon Cowie写的*Customizing Chef*。

# 谁是此书的读者群体

本书适合没有Chef和基础架构自动化经验的系统管理员和软件工程师。读者需要相对熟悉命令行的使用及能够完成基本的系统管理任务。

本书中所有的动手实例都可以在Linux、Mac OS X或Windows上操作。必要的时候，我们会提供分别针对每个平台的讲解。由于目前并没有针对Chef的易用的集成开发环境（IDE），和Chef交互的主要界面是命令行。幸运的是，除了最开始的安装过程之外，Chef的命令行界面在各个平台上几乎相同，这使我们能够在本书覆盖上述三个操作系统下Chef的使用而不至于过度干扰使用其中任何一个操作系统的读者。

你不需要拥有自动化编程经历来阅读此书，但你至少要写过shell脚本、批处理文件或PowerShell脚本。你应该在学习写Chef代码之前对写任何一种脚本相对熟悉。

## 为什么都是厨房用语

在阅读此书的时候，你也许会注意到Chef的发明者使用厨房用语来描述基础架构自动化工具和概念。Chef软件公司（发明并维护Chef的公司）的工程师们在早期创建Chef的时候发现类似菜谱或按照配方单做菜的说法是对一些诸如一个自动化脚本的集合或运行一份永远能一致重建一个基础架构组件的代码的抽象概念的很好的比喻。如此使用厨房用语来做比喻，能让初学者面对其他专业用语相对更容易接受基础架构自动化的概念。除此以外，从市场和品牌的角度也让人更容易记得Chef。

## 本书采用的约定

本书采用以下排版约定。

### 斜体 (*italic*)

用于表明新的术语、URL、电子邮件地址、文件名和文件扩展名。

### 等宽字体 (`Constant width`)

用于程序清单，在正文段落中出现的程序元素(如变量或函数名)、数据库、数据类型、环境变量、语句和关键字也采用这样的字体。

### 等宽粗体 (**`Constant width bold`**)

用于显示命令或应该由用户键入的其他文本。

### 等宽斜体 (*`Constant width italic`*)

表明这里的文本需要替换为用户提供的值或其他由上下文确定的值。

---

提示：这个图标表示小提示或建议。

---

---

说明：这个图标表示通用的说明。

---

---

警告：这个图标表示警告或需要注意。

---

## 示例代码的使用

补充资料（案例代码、练习等等）可在此链接下载（英文）：<http://learningchef.com>。

本书的目的是帮助读者完成工作。通常情况下，可以在你的程序或文档中使用本书中给出的代码。不必联系我们获得代码使用授权，除非你需要使用大量的代码。例如，在写程序的时候引用几段代码不需要向我们申请许可。但以光盘方式销售或重新发行O'Reilly书中的示例确需要获得许可。引用本书或引用本书中的示例代码来回答问题也不需要申请许可。但是，如果要将本书中的大量范例代码加入你的产品文档，则需要申请许可。

我们欣赏引用时注明出处的做法，但不强求。引用通常包括书名、作者、出版社和ISBN，例如“*Learning Chef* by Mischa Taylor and Seth Vargo (O'Reilly). Copyright 2015 Mischa Taylor and Seth Vargo, 978-1-491-94493-6”。

如果觉得使用示例代码的情况不属于前面列出的合理使用或许可范围，请通过电子邮件联系我们，邮箱地址为[permissions@oreilly.com](mailto:permissions@oreilly.com)。

## Safari® Books Online

---

说明：Safari Books Online是一个按需定制的数字图书馆，以图书和视频的形式提供全球技术领域和经管领域内知名作者的专业作品。

---

专业技术人员、软件开发人员、网页设计人员、商务人员和创意专家将Safari Books Online用作自己开展研究、解决问题、学习和完成资格认证培训的重要来源。

Safari Books Online为组织机构、政府部门、教育和个人提供广泛、灵活的产品组合和定价方式。在这里，订阅者通过一个可以全文检索的数据库中就能够访问数千种图书、培训视频和正式出版之前的书稿，这些内容提供商有O'Reilly Media、Prentice Hall Professional、Addison-Wesley Professional、Microsoft Press、Sams、Que、Peachpit

Press、Focal Press、Cisco Press、John Wiley & Sons、Syngress、Morgan Kaufmann、IBM Redbooks、Packt、Adobe Press、FT Press、Apress、Manning、New Riders、McGraw-Hill、Jones & Bartlett、Course Technology及其他几百家出版社。欢迎访问Safari Books Online，了解更多详情。

## 联系我们

对于本书，如果有任何意见或疑问，请通过以下地址联系出版商：

美国：

O'Reilly Media, Inc.  
1005 Gravenstein Highway North  
Sebastopol, CA 95472

中国：

北京市西城区西直门南大街2号成铭大厦C座807室（100035）  
奥莱利技术咨询（北京）有限公司

本书也有相关的网页，我们在上面列出了勘误表、范例以及其他一些信息。网址如下：  
<http://shop.oreilly.com/product/9781491944936.do>（英文版）。

对本书做出评论或者询问技术问题，请发送E-mail至以下邮箱：[bookquestions@oreilly.com](mailto:bookquestions@oreilly.com)。

如果希望获得关于本书、会议、资源中心和O'Reilly的更多信息，请访问以下网址：  
<http://www.oreilly.com>。

## 致谢

来自Mischa：我要感谢所有对本书创作过程中给予反馈的人。感谢我的姐姐和Jane Maris Sinagub博士给予我的精神支持和鼓励。

我要真心地感谢所有对本书有所贡献的人，包括但不限于：Alex Vinyar、Alyssa Nabors、Anthony Stonebarger、Daniel DeLeo、Deluan Quintao、Eric Helgeson、Gene Harris、Glenna Gorlick、Jason Steele、Jennifer Davis、Jo Rhett、John Keiser、Jennifer Davis、John Fitzpatrick、Jon Cowie、Julian Dunn、Katherine Daniels、Kelly Setzer、Kimberly Lanning、Landon Medlock、Lejo Varughese、Mandi Walls、Michael Goetz、Michael Vitale、Nathen Harvey、Patricia Fernandes、Rhiannon Portwood、Sascha