

为初学者量身打造的

# DevOps

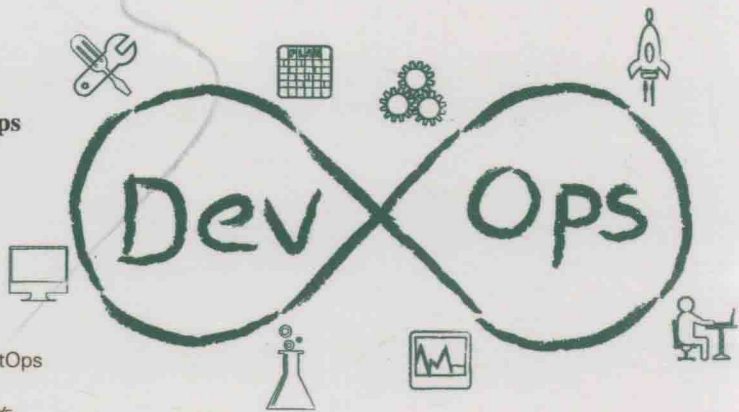
## 入门与实践

[日] DevOps引入指南研究会 / 著  
[河村圣悟 北野太郎 中山贵寻 日下部贵章]  
刘斌 / 译

### 手把手教你

在开发现场引入DevOps  
的具体流程

Vagrant • Ansible  
Serverspec • Git • GitHub  
Docker • Jenkins  
微服务架构 • 蓝绿部署  
日志收集 • 敏捷开发  
ticket驱动开发 • SRE • ChatOps  
持续集成 • 持续交付  
ELK技术栈 • 不可变基础设施



中国工信出版集团

人民邮电出版社  
POSTS & TELECOM PRESS

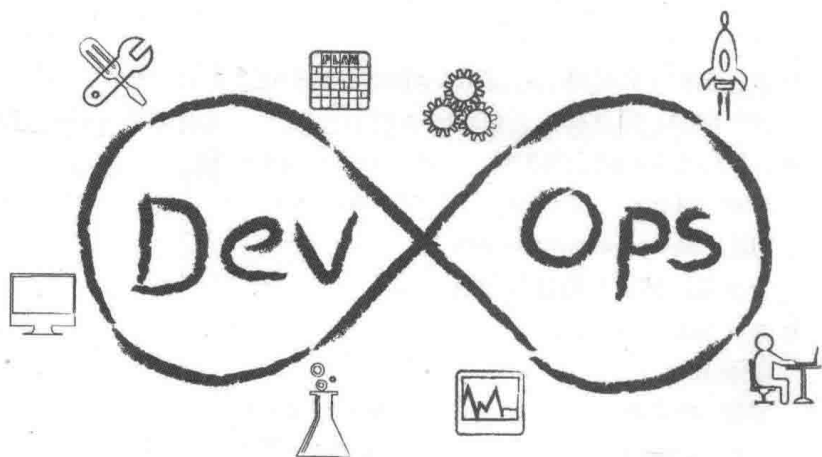
TURING 图灵程序设计丛书

# DevOps

## 入门与实践

[日] DevOps引入指南研究会 / 著  
[河村圣悟 北野太郎 中山贵寻 日下部贵章]

刘斌 / 译



人民邮电出版社  
北京

## 图书在版编目(CIP)数据

DevOps入门与实践 / 日本DevOps引入指南研究会著;  
刘斌译. -- 北京: 人民邮电出版社, 2019.7

(图灵程序设计丛书)

ISBN 978-7-115-51256-7

I. ①D… II. ①刘… ②刘… III. ①软件工程 IV.

①TP311.5

中国版本图书馆CIP数据核字(2019)第094213号

## 内 容 提 要

本书结合大量实例,详细介绍了在开发现场引入DevOps的具体流程。在对DevOps出现的背景和相关概念进行说明之后,首先介绍了如何在个人环境中引入DevOps,接着介绍了在团队中开展DevOps的方法,最后介绍了引入DevOps的最佳实践。内容全面涵盖了DevOps相关的工具、技术和开发思想。

本书条理清晰,逐步深入,特别适合对DevOps感兴趣的初学者以及希望在团队中引入DevOps的基础设施工程师和开发人员阅读。

---

◆ 著 [日] DevOps引入指南研究会

译 刘 斌

责任编辑 杜晓静

责任印制 周昇亮

◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路11号

邮编 100164 电子邮件 315@ptpress.com.cn

网址 <http://www.ptpress.com.cn>

天津翔远印刷有限公司印刷

◆ 开本: 880×1230 1/32

印张: 12.25

字数: 377千字

2019年7月第1版

印数: 1-3 000册

2019年7月天津第1次印刷

著作权合同登记号 图字: 01-2017-5048号

---

定价: 69.00元

读者服务热线: (010) 51095183转600 印装质量热线: (010) 81055316

反盗版热线: (010) 81055315

广告经营许可证: 京东工商广登字20170147号



## 作者简介

“DevOps引入指南研究会”由以下四位成员组成。

### 河村圣悟

曾在日本索尼公司任架构师，2014年进入Recruit公司，负责全面推进基础设施即代码的部署方式。著有《为工程师打造的Git教科书》。

### 北野太郎

2013年进入Recruit公司，现主要负责Recruit基础设施的构建和自动化运维工作。著有《Apache Solr入门》。

### 中山贵寻

大学毕业后进入NS Solutions公司基础设施事业部，负责为构建和运维自动化项目提供支持。对Hadoop和OpenStack等抱有兴趣。

### 日下部贵章

2014年进入Recruit公司，参与大规模本地部署环境的相关工作，喜欢的技术是KVM和Ceph。

## 译者简介

### 刘斌

软件工程师，专注于后台开发、运维技术和云计算技术，崇尚不可变基础设施和基础设施即代码。



微信连接



回复“运维”查看相关书单



微博连接

关注@图灵教育 每日分享IT好书



QQ连接

图灵读者官方群I: 218139230

图灵读者官方群II: 164939616

**图灵社区**

**iTuring.cn**

在线出版, 电子书, 《码农》杂志, 图灵访谈

站在巨人的肩上  
**Standing on Shoulders of Giants**



iTuring.cn

## 版 权 声 明

DevOps 導入指南

(DevOps Dounyu Shinan: 4760-4)

Copyright © 2016 by Seigo Kawamura / Taro Kitano / Takahiro Nakayama / Takaaki Kusakabe  
Recruit Technologies Co.,Ltd.

Original Japanese edition published by SHOEISHA Co.,Ltd.

Simplified Chinese Character translation rights arranged with SHOEISHA Co.,Ltd.  
through CREEK & RIVER Co.,Ltd. and CREEK & RIVER SHANGHAI Co., Ltd.

Simplified Chinese Character translation copyright © 2019 by Posts & Telecom Press.

本书中文简体字版由 SHOEISHA Co.,Ltd. 授权人民邮电出版社独家出版。  
未经出版者书面许可，不得以任何方式复制或抄袭本书内容。

版权所有，侵权必究。

---

※ 本书中出现的网址有可能会发生变更。

※ 本书在出版时尽可能地确保了内容的正确性，但对运用本书内容或示例程序的一切结果，本书作译者和出版社概不负责。

※ 本书中出现的示例程序、脚本以及运行结果画面等都是在特定环境下再现的一例。

※ 本书中出现的公司名、商品名分别是各公司的商标及注册商标。

※ 本书内容基于 2016 年 8 月执笔时的情况。



# 译者序

DevOps 这个术语进入国内已经很多年了。对初学者来说，最常见的问题就是什么是 DevOps，怎么做才算实践了 DevOps。DevOps 是一个职位？一个部门？还是一套工具？问 10 个人，可能会得到 10 种不同的答案，每个人的理解都不尽相同。

对初学者来说，在学习什么是 DevOps 之前，亲自在实践中体验一下 DevOps 会比较好。相比抽象的概念，通过实际操作得到的结果可能会更直接、更形象。在了解了 DevOps 的各种实践之后，回过头来再去理解什么是 DevOps 就易如反掌了。到那时，你可能都不关心 DevOps 的定义到底是什么了。

本书就是一本适合 DevOps 初学者的入门书。

首先，第 1 章介绍了 DevOps 出现的背景。要了解 DevOps 的本质，就需要了解它的历史，否则就会被它的表象所迷惑，从而迷失目标。在介绍 DevOps 的背景之后，第 1 章还介绍了 DevOps 的几个重要特征，包括抽象化、自动化、持续集成和监控等。本书的后面几章都是基于第 1 章的概述展开讨论的，读完整本书之后再回过头来阅读第 1 章，相信你会对 DevOps 有更深一层的理解。

本书的结构非常有特色，可以说是为初学者和初学团队量身打造的。书中先从个人场景开始，介绍了如何提高个人开发效率和自动化程度。在个人掌握了相关技能之后，又将 DevOps 上升到团队的高度，介绍了在团队内部开展 DevOps 时需要引入什么样的工具，选择什么样的架构，采用什么样的组织形式等。最后还介绍了很多将 DevOps 引入组织内部的方法和最佳实践。对想要在自己的团队内开展 DevOps 的人来说，这部分内容会起到一定的作用。

本书的另一个特色是结合实例进行讲解。相比枯燥的理论，本书更注

重手把手教读者进行实践。即使你对某项技术不熟悉也没有关系，甚至不必去查阅复杂的官方文档，也不必去网上搜寻各种教程，本书中的实例就会让你轻松上手，快速理解一门新的技术。可能只要花上十几分钟，你就能学会如何使用 Logstash 采集 Web 服务的日志并存储到 Elasticsearch 中，然后通过 Kibana 创建图表和仪表盘进行可视化。在对某项技术有一个直观、生动的认识之后，你就可以发挥自己的主观能动性，深入挖掘这一技术的潜能了。

从以上两点来说，本书非常适合 DevOps 的初学者使用。

另外，即使你不关心 DevOps，本书也会对你有一定的帮助。DevOps 涵盖的范围非常广，我们在不知不觉中可能就已经在实践 DevOps 了。本书也涵盖了云计算时代开发云原生应用程序所需要的技术和开发思想，比如蓝绿部署、不可变基础设施、持续集成和持续部署等，这些内容即使单独拿出来学习也非常有价值。从这一方面来说，本书也适合那些熟悉 DevOps 的开发人员，以及只对某些技术感兴趣的开发人员。翻阅一下本书的目录，相信一定会有某些章节吸引到你。

浅显易懂、生动形象是日本技术类图书的特点，这应该和日本传统技艺的修行文化有关。相信很多人都听说过“断舍离”一词，这一词汇是在 2000 年以后才产生的。与此类似，日本还有一个叫作“守破离”的词汇，而这一词汇出现的时间比“断舍离”早了几百年。古时候讲究师承，无论是学习文化还是技能，都需要先拜师，再学艺。“守破离”就描述了从掌握一门技能到成为一派宗师的 3 个阶段。

- 守，指的是遵循老师的教诲，学习定式，掌握基本技能
- 破，指的是在掌握基本技能之后，能进行自我反省和改善，找出做得不好的地方，同时拓宽自己的视野，吸收其他流派的优点，打破定式
- 离，指的是脱离原定式，创造新定式

也就是说，要想学好一门技艺，就得先找到一个好老师，再寻求突破，而本书就是一本带你认识、理解并在实践中实现 DevOps 的书：

- 守，基于本书介绍的示例和方法开展 DevOps
- 破，找到实践中不好的地方，借鉴他人的经验，改进自己的 DevOps
- 离，创建符合自己团队的效率最高的 DevOps 模式

说完“守破离”，我们再来看看“术与道”。

“道为术之灵，术为道之体”是说道是根本，术是表现；道是世界观，术是方法论；道是目的，术是手段。如果说提高商业价值是道，而本书则是术的集合。

“以道统术，以术得道”。我们在学习某项技能时，通常都是从术开始的，循序渐进而最终悟道。如果只知术，而不知术与术之间的关系、规律以及变化，就只能生搬硬套，始终困于术的牢笼中而不能自我突破。道不变，术却可以不同。

总之，道要悟，术要通。

# 前言

听到 DevOps 这个词，你会想到什么呢？在网上搜索一下就可以知道，DevOps 所涉及的领域非常广，而且其含义也因人而异。因此，即使想去学习 DevOps，也会产生“到底怎么做才算实现了 DevOps”的疑惑。

DevOps 是指通过 Dev（开发）和 Ops（运维）的紧密合作来提高商业价值的工作方式和文化。DevOps 思想涉及的范围很广，不仅包括新技术和新工具的使用，还包括与这些技术、工具相关的组织和文化，以及能实现持续改善的运维架构，所以并没有办法明确指出只要进行了某种特定的工作就算进行了 DevOps 实践，这也使得学习和实践 DevOps 变得非常困难。

即使无法明确指出什么是 DevOps 实践，我们也可以学习 DevOps 思想产生的原因、DevOps 的目的、支持 DevOps 的方法和工具等，而本书的目的就是帮助读者学习这些内容。

日文书名副标题中的基础设施即代码是指将服务器、网络设备等基础设施的设置和架构代码化，把软件开发的开发模式应用到基础设施运维中的方法。基础设施即代码是在 DevOps 实践中支持开发和运维紧密合作的一个非常有效的方法。

本书不仅介绍什么是基础设施即代码，还将深入探讨如何进行实践。此外，书中还会介绍基础设施即代码是如何支撑 DevOps 思想的，以及能取得什么样的效果等内容。为了让刚接触 IT 工作的人以及虽已成为团队的中坚力量但尚不熟悉 DevOps 的人都能阅读本书，本书会全面讲解 DevOps 的技术和工具，从入门到应用。

此外，已经具备 DevOps 的相关知识但苦于无法在团队中实施的人也可以阅读本书。本书将按照从个人到组织的顺序进行讲解，首先介绍如何在个人环境中以基础设施即代码为中心阶段性地引入 DevOps 相关的技术，然

后介绍如何在团队中实施 DevOps，并将其运用到服务的开发和运维上，最后从团队成员的角度出发，介绍如何将 DevOps 方法引入组织内部。

无论是面对外界的快速变化而被迫应对的学生，还是每日每夜都在思考如何提高商业价值的 IT 公司员工，对他们来说，DevOps 都是一个强大的工具。本书可以帮助读者全面掌握 DevOps 的基础知识乃至应用，希望各位读者能够阅读完本书。

# ○ 本书的阅读方法

## ■ 章节结构

从第 1 章开始按顺序阅读，读者可以阶段性地学习如何引入 DevOps。第 1 章介绍 DevOps 的概要；第 2 章介绍如何从个人层面开始实施 DevOps；第 3 章介绍如何在团队中实施 DevOps；第 4 章和第 5 章介绍具体的实践和应用。越往后阅读，内容的难度会越大，需要读者在前一章知识的基础上掌握更高难度的知识。第 6 章将介绍如何在组织中应用前面 5 章学到的知识。

## ■ 关于命令的印刷格式

如下所示，以符号 \$ 开始的行表示输入的命令以及该命令的执行结果，因此读者不需要输入 \$ 符号。

```
$ echo hello  
hello
```

此外，在命令行或配置文件中，# 之后的部分都表示注释。注释用于解释该行的内容，所以读者不需要输入 # 之后的内容。

```
$ echo hello # 输出"hello"字符串
```

# ○ 目录

<b>第1章 认识DevOps</b>	<b>1</b>
<b>1-1 DevOps 出现的背景</b> .....	<b>2</b>
1-1-1 DevOps 诞生的背景.....	3
1-1-2 以敏捷开发为代表的持续开发方式的出现.....	4
1-1-3 持续开发带来的运维问题.....	5
1-1-4 DevOps 的诞生和历史.....	11
1-1-5 小结.....	14
<b>1-2 认识 DevOps</b> .....	<b>15</b>
1-2-1 以迅速满足商业需求为目标.....	15
1-2-2 PDCA 循环和 DevOps.....	16
1-2-3 抽象化.....	17
1-2-4 自动化.....	21
1-2-5 统一管理.....	21
1-2-6 持续集成.....	24
1-2-7 监控.....	25
1-2-8 目的意识、同理心和自主思考.....	26
1-2-9 小结.....	27
<b>1-3 组织和 DevOps</b> .....	<b>28</b>
1-3-1 DevOps 能解决组织、团队中的什么问题.....	28
1-3-2 康威定律.....	31
1-3-3 小结.....	31
<b>第2章 从个人开始实践DevOps</b>	<b>33</b>
<b>2-1 从小的地方开始实践 DevOps</b> .....	<b>34</b>
<b>2-2 个人也能够实现 DevOps</b> .....	<b>36</b>
2-2-1 从哪里开始入手.....	36
2-2-2 构建本地开发环境.....	37
<b>2-3 从个人环境到团队环境的准备</b> .....	<b>45</b>
2-3-1 使用 Vagrant 实现本地开发环境的代码化.....	46

2-3-2	使用 Ansible 将构建工作通用化,并向其他环境展开	56
2-3-3	使用 Serverspec 实现基础设施测试代码化	76
2-3-4	使用 Git 在团队内共享配置信息	90
2-3-5	基础设施即代码和 DevOps 的目标	103

## 第3章 在团队中普及DevOps 105

<b>3-1</b>	在团队内实施 DevOps 的意义	106
<b>3-2</b>	实现团队工作的效率化	108
3-2-1	使用 GitHub 进行团队开发	108
3-2-2	使用 Docker 进一步提高开发效率	129
3-2-3	使用 Jenkins 管理工作	163
3-2-4	使用持续集成和持续交付优化发布	197
<b>3-3</b>	在团队中实施 DevOps 的效果	211

## 第4章 面向DevOps的架构变革 213

<b>4-1</b>	以 DevOps 为中心对架构进行变革	214
<b>4-2</b>	改变应用程序架构	215
4-2-1	The Twelve-Factor App	215
4-2-2	微服务架构	218
<b>4-3</b>	改变基础设施架构	222
4-3-1	使用不可变基础设施进行高效管理	222
4-3-2	使用蓝绿部署切换服务	227
4-3-3	本地部署和公有云	233
4-3-4	SaaS	236
4-3-5	日志收集和分析	240
<b>4-4</b>	改变团队	245
4-4-1	DevOps 和敏捷开发	245
4-4-2	ticket 驱动开发	255
4-4-3	网站可靠性工程	257
4-4-4	ChatOps	263
<b>4-5</b>	DevOps 团队的作用	271
4-5-1	故障处理	271
4-5-2	实现持续集成和持续交付	273
4-5-3	性能优化	273
4-5-4	建立开发和运维之间的合作体制	274



<b>第5章 实践基础设施即代码</b>	<b>277</b>
<b>5-1 实践持续集成和持续交付</b>	<b>278</b>
5-1-1 持续集成和持续交付的组成要素和集成	278
5-1-2 集成 GitHub 和 Slack: 将 GitHub 的事件通知给 Slack	281
5-1-3 集成 GitHub 和 Jenkins: git push 之后的自动化处理	286
5-1-4 集成 Jenkins 和 Slack: 将任务事件通知到 Slack	295
5-1-5 集成 Jenkins 和 Ansible: 通过任务触发基础设施构建	299
5-1-6 集成 Jenkins 和 Serverspec: 通过构建任务触发基础设施测试	305
5-1-7 从 GitHub 触发 Jenkins 的 Provisioning	311
5-1-8 使用持续集成和持续交付, 将开发、构建和测试组合到一起	312
5-1-9 如何实现更实用的架构	314
<b>5-2 实践 ELK 技术栈</b>	<b>317</b>
5-2-1 ELK 技术栈的构成要素和集成	317
5-2-2 构建 ELK 技术栈	318
5-2-3 访问日志的可视化	330
5-2-4 可视化让我们距离 DevOps 更近一步	340
<b>5-3 实践不可变基础设施</b>	<b>342</b>
5-3-1 实现不可变基础设施所需要的要素以及发布流程	342
5-3-2 使用 CloudFormation 构建基础环境	345
5-3-3 基于蓝绿部署进行发布工作	353
5-3-4 发生故障时切换基础设施	355
5-3-5 更具实用性的架构	356
5-3-6 不可变基础设施会从根本上改变基础设施的使用方式	358
<b>第6章 跨越组织和团队间壁垒的DevOps</b>	<b>359</b>
<b>6-1 普及 DevOps 的困难之处</b>	<b>360</b>
<b>6-2 在组织中实施 DevOps</b>	<b>361</b>
6-2-1 在新的组织中实施 DevOps	361
6-2-2 在既有组织中实施 DevOps	361
6-2-3 实施 DevOps 的反模式	370
6-2-4 在组织形式方面是否有实施 DevOps 的最佳实践	373
<b>6-3 团队整体的 DevOps</b>	<b>377</b>