HZ BOOKS
华章科技

高级DevOps、系统架构师十余年工作总结，基于一线运维工作提炼，从生产环境下的Python和Shell脚本、Python自动化运维、Docker、Jenkins这些流行的DevOps技术方向多角度讲解，以实践案例指导读者掌握DevOps与自动化运维的技巧和难点。

姊妹篇《构建高可用Linux服务器》被《程序员》杂志和51CTO权威媒体评为“十大最具技术影响力的图书”和“最受读者喜爱的原创IT图书”。

Linux/Unix
技术丛书

DevOps
和自动化运维实践

余洪春 著

The Practice of DevOps and Automation Operations

机械工业出版社
China Machine Press

Linux/Unix
技术丛书

DevOps
和自动化运维实践

余洪春 著

机械工业出版社
China Machine Press

图书在版编目（CIP）数据

DevOps 和自动化运维实践 / 余洪春著 . —北京：机械工业出版社，2018.10
（Linux/Unix 技术丛书）

ISBN 978-7-111-61002-1

I. D… II. 余… III. Linux 操作系统 IV. TP316.85

中国版本图书馆 CIP 数据核字（2018）第 221119 号

DevOps 和自动化运维实践

出版发行：机械工业出版社（北京市西城区百万庄大街 22 号 邮政编码：100037）
责任编辑：赵亮宇 责任校对：殷 虹
印 刷：北京市兆成印刷有限责任公司 版 次：2018 年 10 月第 1 版第 1 次印刷
开 本：186mm × 240mm 1/16 印 张：25.75
书 号：ISBN 978-7-111-61002-1 定 价：89.00 元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换
客服热线：（010）88379426 88361066 投稿热线：（010）88379604
购书热线：（010）68326294 88379649 68995259 读者信箱：hzit@hzbook.com

华章科技
HZBOOKS | Science & Technology

Foreword 1 推荐序一

随着互联网业务的高速发展，工作内容更细分化、专业化，所以工作职责也逐渐分出开发（Dev）和运维（Ops）两个完全独立的角色。DevOps 就是为了解决开发团队与运维团队之间存在已久的冲突及矛盾：开发团队责怪运维团队的机器出了问题，运维团队则把问题归咎于开发团队的代码上。

运维人员看重的是保障系统的稳定性、可靠性和安全性，而开发人员则想着如何尽快发布新的版本，增加新的功能，这两者本身就是一种矛盾和冲突，尽管他们的共同目标都是为用户提供软件产品或服务。

那么，如何才能更好地实现 DevOps 工具和文化就变得愈发重要。

洪春恰巧是一位一直在一线奋斗的技术人员，他将结合自己多年的理论与实战经验，教大家如何快速上手实现 DevOps 。

本书以实现 DevOps 为主线，涵盖了实现过程中的核心要素：Shell 应用、Python 应用、基础环境搭建、自动化运维工具、自动化部署管理等。相信无论你是 DevOps 新兵还是老将，都能从书中汲取不少精华，受益良多。

沪江资深架构师　曹林华

2018 年 9 月

推荐序二 *Forword 2*

本书作者余洪春先生和我相识于 ChinaUnix 举办的一次技术交流活动——“千万级 PV 高性能高并发网站架构与设计交流”，当时他已经在宣传自己的第一本著作——《构建高可用 Linux 服务器》，该书凝聚并整合了他多年来在一线工作的经验，时至今日，该书仍是一本在国内非常经典的运维原创著作，现在已经更新到第四版。这种对技术不断进行完善的坚持及工匠精神让我深深折服。这次能受邀为本书写推荐序，让我倍感荣幸。

本书覆盖了 DevOps 中的许多方面，介绍了基于 Python 语言构建的主流自动化运维工具，包括 Ansible、Saltstack、Docker 和 Jenkins 等，这些都是 DevOps 工具元素周期表中最闪亮的内容，也是运维人员必备的技能。本书中分享的案例是余洪春多年实战经验的精华，具有非常高的参考价值及借鉴意义。

书中内容从互联网业务平台构建及自动运维的场景出发，以常见的业务服务为基础，给出了大量的实战案例，相信会给读者带来不少启发及思考。

更难能可贵的是，作者能从易于理解的角度出发，由浅入深地剖析自动化运维管理之道。这对于不同技术水平的读者来说，有助于其有效地阅读和吸收这些知识，也能根据实际需要各取所需。

最后，感谢余洪春先生给中国互联网从业者带来这么好的图书，我相信阅读本书的每一位读者都能从中获取提升的能量，为企业及行业做出自己的贡献。

腾讯高级工程师　刘天斯

2018 年 9 月

Foreword 3 推荐序三

在全球“互联网 +”的大背景下，互联网创业企业如雨后春笋般大量出现并得到了快速发展！很大程度上，对“互联网 +”提供有力的支撑就是 Linux 运维架构师、云计算和大数据工程师，以及自动化开发工程师等。

但是，随着计算机技术的发展，企业对 Linux 运维人员的能力要求越来越高，这就使得很多想入门运维的新手不知所措，望而却步，甚至努力了很久却仍然徘徊在运维岗位的边缘；而有些已经从事运维工作的人也往往疲于奔命，没有时间和精力去学习企业所需的新知识和新技能，从而使得个人的职业发展前景大大受限。

本书就是在这样的背景下诞生并致力于为上述问题提供解决方案的，本书是余洪春先生 10 多年来一线工作经验的再结晶，此前余洪春先生已经出版过 Linux 集群方向的图书（《构建高可用 Linux 服务器》），本次出版的书是作者对运维行业的再回馈。

书中不仅涵盖企业运维人员需要的大规模集群场景下必备的运维自动化 Shell 和 Python 企业开发应用实践案例，还包括热门的自动化运维工具在企业中的应用，以及 Docker 和 Jenkins 实践等。

本书能够帮助运维人员掌握业内运维实战专家的网站集群的企业级应用经验的精髓，从而以较高的标准胜任各类企业运维的工作岗位，并提升自己的运维职业发展竞争力，值得一读！

老男孩 Linux 实战运维培训中心总裁

“跟老男孩学 Linux 运维”系列图书作者　老男孩

2018 年 9 月

前　言 Preface

我的系统架构师之路

从 2006 年接触 Linux 系统并从事 Linux 系统管理员的工作以来，我担任过 Linux 系统工程师、项目实施工程师 / 高级 Linux 系统工程师、运维架构师，到如今的高级系统开发工程师、系统架构师，这一路走来，我深感开源技术和 Linux 系统的强大及魅力。

现阶段我的职务是高级运维开发工程师（DevOps）、系统架构师，主要工作是负责公司的 CDN 业务系统的运维自动化及公司 APP 产品的 CI/CD 工作及自动化部署工作。CDN 系统相对于其他领域而言，海量机器的自动化运维工作是一件比较复杂的事情，关于这项工作，我们可以通过 Python 自动化配置管理管理工具，例如 Ansible 和 SaltStack 来进行二次开发，结合公司的 CMDB 系统，提供稳定的后端 API，方便前端人员或资产人员进行调用，这样大家都可以利用界面来完成自动化运维工作。至今为止，令我印象最为深刻的还是公司的 APP 项目，该项目现在全部部署在云平台（国内云平台）并且 Docker 容器化了，从前端到后端包括大数据接口，全部采用容器化的项目方式部署上线，整个自动化流程跟传统的自动化方式大相径庭。尤其是现在公司正在使用的 Kubernetes，整个架构设计非常复杂，学习成本也是非常高的，但带来的容器的自动化管理也是非常便利的。目前，无论是国外的 AWS、Google 还是国内的阿里云和腾讯云等主流公有云均提供 Kubernetes 的容器服务，可以说 Kubernetes 在当前容器行业是热门的，而 Docker 技术正是 Kubernetes 的基石，建议大家尽快熟练 Docker 的使用方法。

撰写本书的目的

云计算和容器技术是当前的流行技术和发展趋势，云计算和容器技术的流行对于传统的运维知识体系其实也是一种冲击，传统运维工程师的工作性质也在不断地发生变化，要掌握很多新的技能和知识。大家经常会在工作中看到 DevOps 这个词。DevOps 为什么会这么火？

这跟最近几年的云计算和容器技术的快速普及有很大关系：云计算平台上（包括 Kubernets）的各种资源，从服务器到网络，再到负载均衡都是由 API 创建和操作的，这就意味着所有的资源都可以由“软件定义”，这给各种自动化运维工具提供了一个非常好的基础环境。而在传统的互联网行业，例如笔者目前正在从事的 CDN 领域，由于机器数量众多、网络环境错综复杂，也需要由 DevOps 人员来设计工具，提供后端的自动化运维 API，结合公司的 CMDB 资产管理系统，提供自动化运维功能，简化运维的操作流程及步骤，提高工作效率。

工作之余，许多读者朋友们也在向我咨询工作中的困惑，比如从事系统运维工作 3 ～ 5 年以后就不知道如何继续学习和规划自己的职业生涯了。我想通过此书，跟大家分享一下这么多年的工作经验和心得（尤其是近几年流行的 DevOps 技术），解决大家工作中的困惑。通过此书的项目实践和线上环境案例，让大家能迅速了解 Linux 运维人员的工作职责和方向，迅速进入工作状态，快速成长，希望大家通过阅读本书，能够掌握 Linux 系统集群和自动化运维及网站架构设计的精髓，轻松而愉快地工作，提升自己的职业技能，这是我非常高兴看到的，也是我编写本书的初衷。

读者对象

本书的读者对象如下所示：

- 系统管理员或系统工程师
- 中高级运维工程师
- 运维开发工程师
- 开发工程师

如何阅读本书

本书的内容是对实际工作经验的总结，涉及大量的 DevOps 及自动化运维知识点和专业术语，建议这方面经验还不是很丰富的读者先了解第 1 章的内容，这章比较基础，如果大家在学习过程中根据这章的讲解进行操作，定会达到事半功倍的效果。

系统管理员和系统工程师们可以通篇阅读本书，并重点关注第 1 章、第 2 章和第 4 章，其他章节的内容可以选择性地阅读，借此来拓宽知识面，确定学习方向。

对于运维工程师而言，除了第 3 章的内容不要求掌握以外，其他章节的内容均可以做深层次的阅读、实践和思考，书中提到的很多自动化案例，读者可以尝试结合自己公司的实际情况来进行应用。

对于运维开发工程师来说，上述章节描述的内容都与运维开发工作息息相关，建议大家多花些精力和时间，抱着一切从线上环境去考虑的态度去学习和思考，实践后多思考一下原

理性的内容。

对于开发工程师来说，由于其只需对运维系统知识体系有一个大概的了解，重点可以放在本书的第 1 ～ 3 章。如果想了解自动化运维相关知识体系，建议熟悉本书的第 6 ～ 8 章。

大家可以根据自己的职业发展和工作需要选择不同的阅读顺序和侧重点，同时也可以对其他相关的知识点有一定的了解。

致谢

感谢我的家人，她们在生活上对我无微不至的照顾，让我更有精力和动力去工作和创作。

感觉好友刘天斯、老男孩的支持和鼓励，闲暇之余和你们一起交流开源技术和发展趋势，也是一种享受。

感谢朋友曹林华，与我一起花了大量时间调研并且实践电子商务系统中关于秒杀系统的架构及设计。

感谢机械工业出版社华章公司的编辑杨福川和杨绣国，在你们的信任、支持和帮助下，这本书才能如此顺利地出版。

感谢朋友冯松林，感谢他这么多年来对我的信任和支持，在我苦闷的时候陪我聊天，自始至终对我予以支持和信任。

感谢生活中的朋友们——曹江华、何小玲、郑桦、徐江春、张薇（排名不分顺序），工作之余能一起闲聊和打牌，也是非常开心和快乐的事情。

感谢在工作和生活中给予我帮助的所有人，感谢你们，正是因为有了你们，才有了本书的问世。

关于勘误

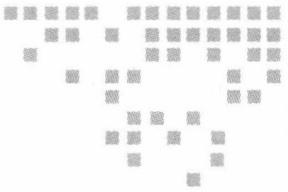
尽管我花了大量时间和精力去核对文件和语法，但书中难免还会存在一些错误和纰漏，如果大家发现问题，希望可以反馈给我，相关信息可发到我的邮箱 yuhongchun027@gmail.com。尽管我无法保证每一个问题都会有正确的答案，但我肯定会努力回答并且指出一个正确的方向。

如果大家对本书有任何疑问或想进行 Linux 的技术交流，可以访问我的个人博客与我交流，博客地址为 http://yuhongchun.blog.51cto.com。另外，我在 51CTO 和 CU 社区的用户名均为抚琴煮酒，大家也可以直接通过此用户名在社区与我交流。

余洪春（抚琴煮酒）

2018 年 2 月于武汉

Contents 目　录



第 1 章 Chapter 1

DevOps 与自动化运维的意义

随着近几年云计算的兴起，相信大家对 DevOps 也越来越熟悉了。DevOps（英文 Development 和 Operations 的组合）是一组过程、方法与系统的统称，用于促进开发（应用程序 / 软件工程）、技术运营和质量保障部门之间的沟通、协作与整合。DevOps 其实是一个体系，而不仅仅是某个岗位，其目的是从总体提高企业 IT 部门的运作效率。关于如何提高运作效率这个问题比较复杂也难以抽象，因此很多人就将 DevOps 具象成了建立一套有效率的开发运维工具，通过这个工具提升个体与团队协作的效率。为了建立和使用这些工具，运维人员必须具备一系列的技能，比如会使用 Python、Go 语言进行开发，会使用 Puppet、Ansible、Saltstack 等一系列工具，并能对这些工具进行二次开发。

1.1 DevOps 在企业中存在的意义

DevOps 如今的兴起与最近两年云计算的快速普及有很大的关系：在云计算平台上，各种资源，从服务器到网络、负载均衡都是有 API 可以创建和操作的，这就意味着所有的资源都是可以用“软件定义”的，这就为各种自动化运维工具提供了一个非常好的基础环境。

事实上，需要频繁交付的公司或企业更应该具有 DevOps 能力，大量的互联网在线应用需要根据用户的反馈随时进行迭代开发，持续改进用户体验，这就需要内部支撑部门（一般是运维开发部门）提供每天几十次乃至上百次的从测试环境发布到线上环境的持续发布能力，这种能力称为持续集成（CI）。如下几个因素更加促进了 DevOps 的发展。

- 虚拟化和云计算基础设施的日益普及。
- 业务负责人要求加快产品交付的速度。

❑ 数据中心自动化配置管理工具的普及。

DevOps 是一个完整的、面向 IT 运维的工作流，其以 IT 自动化以及持续集成（CI）、持续部署（CD）为基础，用于优化程序开发、测试、系统运维等所有环节。DevOps 一词来自于 Development 和 Operations 的组合，尤其重视软件开发人员和运维人员的沟通合作，通过自动化流程来使得软件构建、测试、发布更加快捷、频繁和可靠。DevOps 可用于填补开发端和运维端之间的信息鸿沟，改善团队之间的协作关系。不过需要澄清的一点是，从开发到运维，中间还有测试环节。DevOps 其实包含了三个部分：开发、测试和运维，如图 1-1 所示。

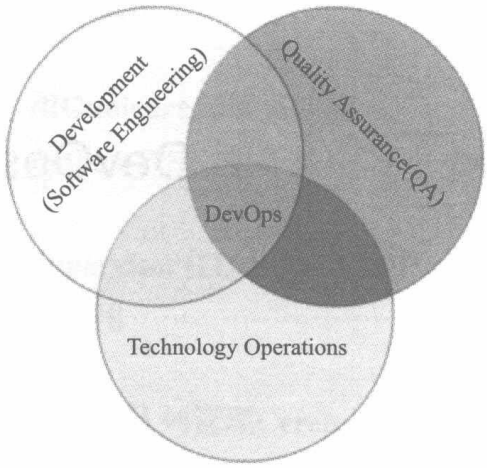


图 1-1 DevOps 工作范畴涵盖了研发、测试及运维三部分图示

我们可以将传统运维的一种极端情况描述为“黑盒运维”。在这种文化中，运维与开发是分开的，相互之间一般不进行合作，就算要合作，也是极不情愿的。“黑盒运维”的特点是开发和运维的目标是相反的。开发团队的任务是为产品增加新功能、不断升级产品，并以此制定绩效；运维团队的目标则是稳定第一。如果没有进行足够的沟通和交流，那么两个团队必然会产生矛盾，当开发人员兴致勃勃地快速开发新功能的时候，运维人员可不情愿部署新功能。对稳定系统实施任何类型的变更，都会导致系统产生隐患，因此运维人员会尽可能地避免变更。这里举个例子说明一下，应用开发人员提交的代码中有一个 Bug，在特定的边界条件下该 Bug 会导致无限循环，而 QA 和测试人员均没有发现这个问题。如果运维人员部署了这个变更，则会导致一些服务器 CPU 使用率飙升至 100%，造成服务不稳定。如果运维人员不去实施变更，那么就不会发生问题，至少是没有新的问题。这就是最左边传统运维的理念。如果我们在这个工作流中引入 DevOps，那么在这里开发和运维是同一个角色。这时，开发就是运维，运维就是开发，团队的共同目标是既要增加新特性，又要确保一定程度的可靠性。所以我们实施 DevOps 就会更加容易，利益也更加明确，结果也是可以预期的。

换句话说，DevOps 希望做到的是打通软件产品交付过程中的 IT 工具链，它的核心理念在于生产团队（研发、运维和 QA）之间的高效沟通和协作，使得各个团队减少时间损耗，从而更加高效地协同工作。

DevOps 早在九年前就已有人提出来，但是，为什么近两年才开始受到越来越多企业的重视和实践呢？因为 DevOps 的发展是独木不成林的，现在具备了越来越多的技术支撑。微服务架构理念、容器技术使得 DevOps 的实施变得更加容易，计算能力的提升和云环境的发展使得快速开发的产品可以立刻获得更广泛的使用。

那么 DevOps 的好处是什么呢？

DevOps的一个巨大的好处就是可以高效地交付，这也正好是它的初衷。Puppet和DevOps Research and Assessment（DORA）主办了2016年DevOps调查报告，根据全球4600位来自各IT公司的技术工作者的提交数据统计，可以得知，高效公司平均每年可以完成1460次部署。

与低效组织相比，高效组织的部署频繁200倍，产品投入使用速度快2555倍，服务恢复速度快24倍。在工作内容的时间分配上，低效者要多花22%的时间用在规划或者重复工作上，而高效者却可以多花29%的时间用在新的工作上。所以这里的高效不仅仅是指公司产出的效率得到了提高，还指员工的工作质量得到了提升。

DevOps的另外一个好处就是其还能改善公司组织文化、提高员工的参与感。员工们变得更高效，也更有满足和成就感；调查显示高效员工的雇员净推荐值（eNPS:employee Net Promoter Score）更高，即对公司更加认同。

快速部署同时还能提高IT稳定性。可能有读者会问，这难道不是矛盾的吗？

快速部署其实可以帮助团队更快地发现问题，产品被更快地交付到用户手中，团队就可以更快地得到用户的反馈，从而更快地响应。而且，DevOps小步快跑的形式所带来的变化也是比较小的，出现问题的偏差每次都不会太大，修复起来相对也会容易一些。因此，认为速度就意味着危险是一种偏见。此外，滞后软件服务的发布也并不一定能够完全地避免问题，在竞争日益激烈的IT行业，这反而可能会错失了软件的发布时机。

DevOps会持续流行下去的原因主要有两点，具体如下。

1）条件成熟：技术配套发展。

技术的发展使得DevOps有了更多的配合和技术支撑。早期的时候，大家虽然也意识到了这个问题，但是苦于当时没有完善、丰富的技术工具支持，处于一种“理想很丰满，但是现实很骨感”的情况。DevOps的实现可以基于新兴的容器技术；也可以进行自动化运维工具Puppet、SaltStack、Ansible之后的延伸；还可以构建在传统的Cloud Foundry、OpenShift等PaaS厂商之上。

2）来自团队的内在动力：工程师也需要。

对于工程师而言，他们也是DevOps的受益者。微软资深工程师Scott Hanselman说过“对于开发者而言，最有力的工具就是自动化工具”（The most powerful tool we have as developers is automation）。工具链的打通使得开发者们在交付软件时可以完成生产环境的构建、测试和运行；正如Amazon的VP兼CTO Werner Vogels那句让人印象深刻的话：“谁开发谁运行”（You build it，you run it）。

另外，很多时候，我们都必须得关注CI/CD（持续集成/持续部署）。

持续集成（Continuous Integration，CI）是一种软件开发实践，即团队开发成员经常集成它们的工作，通常每个成员每天至少集成一次，这也意味着每天可能会发生多次集成。每次集成都通过自动化的构建（包括编译、发布、自动化测试）来验证，从而尽早地发现集成错误。