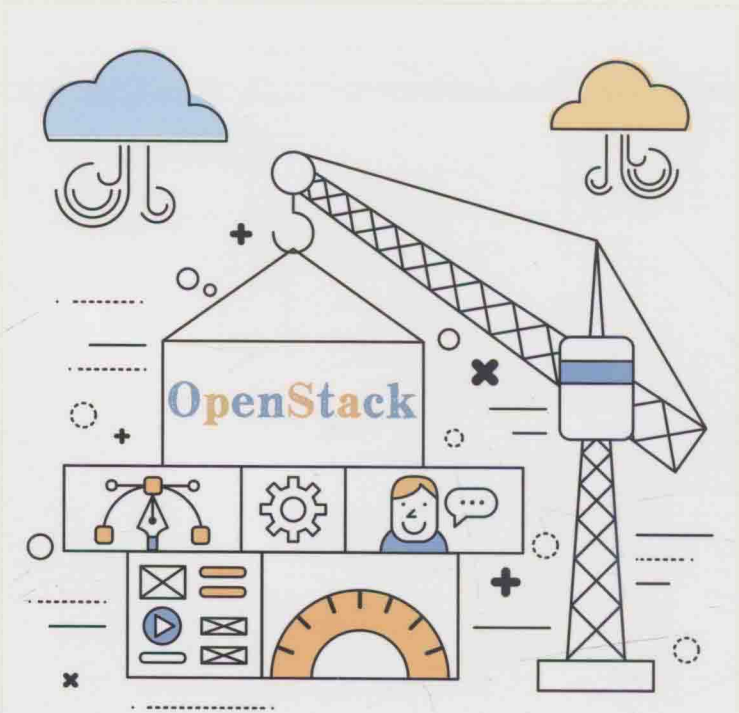
云技术 CSDN 联合力荐

Broadview
www.broadview.com.cn

深度分享OpenStack实践经验
促进OpenStack的持续敏捷开发、部署和测试



OpenStack 最佳实践

测试与CI/CD

徐超 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

OpenStack 最佳实践

测试与CI/CD

徐超 著

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

内 容 简 介

目前，以云计算等为代表的新兴技术得到了大量的运用与普及。同时，凭借着众多极其独特的优势，OpenStack 业已成为开源云计算技术领域的既定事实标准。

这是一本介绍 OpenStack 测试和 CI/CD 实践的书，基于此，书中内容以实践操作为主，从理论到实践，循序渐进，依次讲解了 DevOps 和 CI/CD 的理论概念；软件测试基础和有效设计 OpenStack 测试用例的方法；如何参与 OpenStack 社区贡献及其沟通交流，以及 OpenStack 社区 CI/CD 系统和企业互操作性测试认证；OpenStack 不同维度测试的实现和方法；基于 OpenStack 构建和运行服务于企业研发测试的 CI/CD 应用。

本书适合于云计算相关专业的高校师生和具有一定软件测试或云计算技术基础的读者使用，对于在云计算企业中从事技术工作的管理人员、QA 测试人员和研发人员，本书也非常适用。

图书在版编目（CIP）数据

OpenStack 最佳实践：测试与 CI/CD / 徐超著. —北京：电子工业出版社，2017.4
ISBN 978-7-121-31034-8

Ⅰ. ①O… Ⅱ. ①徐… Ⅲ. ①计算机网络－研究 Ⅳ. ①TP393

中国版本图书馆 CIP 数据核字（2017）第 043424 号

策划编辑：杨中兴
责任编辑：葛 娜
印 刷：三河市华成印务有限公司
装 订：三河市华成印务有限公司
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：19.75 字数：400 千字
版 次：2017 年 4 月第 1 版
印 次：2017 年 4 月第 1 次印刷
定 价：69.00 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

OpenStack 生产环境的使用和部署，目前最大的挑战是如何保持稳定性！有一种说法，OpenStack 将复制 Linux 在开源世界中的成功，成为有史以来第二大开源软件。Linux 解决系统层的问题，OpenStack 则将所有的资源整合起来，实现资源的统一分配和使用。

Linux 系统也是通过十几年的时间才逐步完善的，同样作为开源软件的 OpenStack，虽然成熟度越来越高，但是不可否认，在 OpenStack 使用过程中，还是经常会踩到各种各样的坑，甚至造成生产环境的损失。如何尽量避免踩到这样的坑，答案就是予以充分测试，测试是非常有效的提前发现问题、避免踩坑的手段。

但是如何做好 OpenStack 测试、OpenStack 和其他软件项目测试有什么异同点，以及 OpenStack 测试应该遵循的流程和方法是什么，目前鲜有资料能看到，《OpenStack 最佳实践——测试与 CI/CD》一书的出现，正好填补了这方面的空白。本书不但解答了 OpenStack 生产环境上线部署前，如何通过系统化的测试流程和方法规避风险，提升整体云环境的健壮性，而且更令人眼前一亮的是，书中还介绍了当前日益普及的 DevOps、CI/CD 应用，以及在 OpenStack+Docker 背景结合下的开发、测试、运维的深度实践，在保持业务稳定的前提下，持续完成敏捷开发和测试。

通过测试促进软件的质量和稳定，通过 CI/CD 促进软件项目的敏捷开发。试想，假如有两个通过率分别为 50%和 99%的软件系统，相信后者更受青睐吧。本书正是通过对多个方面内容的介绍，致力于后者，并通过大量的实践回答了如下一些重要问题。

一是阐述了如何运用 DevOps 和 CI/CD；二是如何从小的 OpenStack 测试用例设计，再到针对 OpenStack 进行不同维度、层次的大的系统化测试；三是如何参与社区，从社区中获取帮助，并贡献其中；四是如何基于 OpenStack+Docker 设计与实现 IT 企业中用于研发

测试的 CI/CD 服务。

基于此，本书的魅力在于，不仅深度分享了 OpenStack 测试的经验，还介绍了与测试相关的一套体系，通过这套体系能有效促进 OpenStack 的持续敏捷开发、部署和测试，实现软件系统稳定应用的最终目的。

肖　力

云技术社区创始人

推荐序二

忆往昔，2010 年夏美国著名云厂商 Rackspace 和美国国家航空航天局（NASA）合作，贡献出 Rackspace 云文件平台代码和 NASA Nebula 平台代码，并以 Apache 许可证方式开源发布了 OpenStack。从那时起至今，OpenStack 已经走过了 6 个多年头，以其开源原则和包容精神，一步步吸收新的项目和创新想法，进而从最初仅含两个项目的 Austin 版本，发展到现在具有 50 多个项目的 Newton 合集，OpenStack 俨然成为全球仅次于 Linux 的第二大开源社区。

OpenStack 开源、开放、包容的基因是优秀的，但是由于项目数量发展太快，以及代码更新太快的原因，OpenStack 必然会被各种 Bug 和 Issue 所困扰，这就需要严格且频繁地对 OpenStack 进行各种测试。幸运的是，OpenStack 测试自始就基于 Jenkins 采用持续集成持续交付（CI/CD）的方式，在最大程度上保证了 OpenStack 上游发行版的软件质量。

关于 OpenStack 测试，市面上鲜有书籍详细介绍，大部分书籍或是介绍 OpenStack 的运维，或是介绍 OpenStack 的开发，或是单独全面地介绍 OpenStack 某一模块，比如软件定义存储或软件定义网络。徐超的《OpenStack 最佳实践——测试与 CI/CD》一书正好填补了市场上的这片空白。本书首先从软件测试理论讲起，介绍了什么是 CI/CD，以及 OpenStack 的 CI/CD 内容，阐述了互操作性 InterOp 测试认证操作；然后对 OpenStack 的不同维度和不同底层硬件模块测试进行了深入分析；最后介绍了基于 OpenStack+Docker 的 CI/CD 部署，以及研发测试实践。

本书的重点是 OpenStack 不同维度的测试实践，以及基于 CI/CD 服务的 OpenStack 开发和测试实践，其次是结合 OpenStack 和 Docker 实现的 CI/CD 应用，这些理论和实践对读者了解 OpenStack 测试原理、开发基于 OpenStack 的发行产品、管理和保证软件质量等都是十分有帮助的。掌握了基本软件测试理论和 CI/CD 测试方法，即使是对于非 OpenStack

的其他软件开发和测试，也是十分有借鉴意义的。

除了我们所熟知的那些国际企业之外，在中国诸如中国移动、中国电信、国家电网、中国银联、东风汽车和百联集团等许多企业和电信运营商都选择了 OpenStack 作为公有云或 IT 支撑平台，还有越来越多的企业已确定或正在考虑将 OpenStack 作为企业虚拟化和私有云平台，越来越多的政府机构也将其作为智慧城市应用或电子政务的支撑平台。在 OpenStack 大规模部署和企业应用过程中，首当其冲，稳定性绝对是用户首先考虑的因素，软件测试和质量必然是用户最关注的方面，而在这种背景下，本书是读者的不二选择。

王庆（Shane Wang）

OpenStack 基金会独立个人董事

前　言

一年前，在我即将离职之际，领导偶然对我说道：“那么喜欢写资料分享，考虑写本书吗？”我默笑了下，后来这种感觉愈加强烈，驱使着我真应该做点什么。

由于常写博客的习惯，加之为了更方便地让读者阅读和丰富 OpenStack 的整个测试体系，最终决定利用空闲时间写成一本书予以分享。能坚持下来，也算是最大的慰藉了。

现如今，各种容器技术及云计算、大数据、人工智能等技术应用层出不穷，同时又不断催生出一些诸如 DevOps、CI/CD（持续集成/持续交付）、极限编程和敏捷开发等软件开发模式。

在我初涉 OpenStack 工作时，亦曾在其相关的诸多岗位间徘徊，但随着时间和工作事务的变化，愈加吸引了我对 QA 测试的兴趣和探索。回头看，无疑，测试为我开启了一扇认识 OpenStack、QA 测试和 CI/CD 的大门。

鉴于软件测试体系博大、内容众多，为了更好地把握方向和主题，本书的内容首先将重点放在了针对 OpenStack 不同维度的测试实践上；其次是基于 CI/CD 服务的 OpenStack 研发和测试实践；最后是基于 OpenStack+Docker 技术设计与实现 CI/CD 应用，以及相关的软件测试理论和方法等方面。

本书的目的旨在推动 OpenStack 测试的专业化、系统化。以解决实际问题为出发点，用大量的实际操作来阐述测试的思想与实践。并不是要告诉读者如何使用一个测试工具，这并非我的初衷。我希望读者在学习本书的内容后能够提高综合或专业的素质，摆脱简单的手工或单一测试，以及对测试理解的片面化，从而向更长远的目标迈进。诚然，本书也并非一本万能书，并不是有了它，测试便可以永无 Bug，解决一切困难。

OpenStack 云计算由计算、存储和网络三大基础构成，相对于其他方面，OpenStack 测试是一个相对狭窄的领域，但基于这样的一个事实标准是，以 OpenStack 为代表的云计算已经成为一个既定事实。随着行业的不断渗透、生态环境的不断拓展等，对云计算的测试需求，特别是对高质量的复合型测试人才的需求将更加旺盛。在这里，希望本书能为有需要的读者起到帮助。

我想，本书能够出版需要感谢创造了这世界上仅次于 Linux 的第二大开源项目 OpenStack 的众多社区开发者，以及对我有养育之恩的父母，是你们为这本书的出版创造了可能。

我深知，限于自身个人水平，加之时间有限，本书可能存在某些错误，如你发现，恳请指出，不胜感激，联系邮件：faq@phei.com.cn。

徐　超

目　录

第1章
DevOps的起源和应用

相信读者应该或多或少地听闻或学习过 DevOps 吧。同时，这种模式在 IT 企业中得到了大量普及。这里，本书基于先理论后实践的方式对 DevOps 的起源、发展和应用等方面进行阐述。

1.1 DevOps 的起源和发展

知过去，方能更好行未来。使用 DevOps，自然不能不探究其起源和发展。目前，DevOps 这个名词闪耀于互联网等诸多媒体上，吸引了无数眼球，引起了众多 IT 企业和行业人员的兴趣。不谈 DevOps，仿佛已然落后世界一大截，那什么是 DevOps 呢？

1.1.1 什么是 DevOps

通俗而言，DevOps 不是一个具体的软件工具，拿来即用，而是一套理念、方法和思想，通过这些理念的指导进行活动的最佳实践。顾名思义，DevOps（开发+运维）是软件开发、运维和质量保证等部门之间为了沟通、协作和集成所采用的流程、方法和体系的一个集合。它是人们为了及时生产软件产品或服务，以满足某个业务目标，对开发与运维之间相互依存关系的一种新的理解。这恰好体现了 DevOps 用于弥补开发与运维之间存在的“鸿沟”，实现合作，达到快速交付软件服务的目的。

如何弥补呢？关键有两点：一是思维，要从软件交付的全局出发，加强各角色之间的合作；二是工具，应该选择那些能够实现自己目的的工具，如 SaltStack、Puppet、Chef 和 Ansible 等这样的自动化管理工具。

举一个简单的例子，开发人员把一个软件产品交付给运维/QA 测试人员，后者拿到该产品后开始准备部署和测试。此时，有两种情况：一是部署、测试验证成功；二是部署失败，并测试发现新的缺陷。

这里，假定发生了第二种情况，开发人员会被要求帮助排除故障和修复 Bug，运维/QA 测试人员会说开发团队给的软件存在问题，而开发人员则会回应称该产品在他们的环境下运行良好，因此一定是在部署的过程中做错了什么。由于文件配置、运行环境和操作流程的不同，开发人员诊断问题也并非一件易事。

DevOps 适用于敏捷开发中的开发阶段、部署阶段和测试阶段等。需要强调的是，DevOps 是一个非常广泛的概念，诸如 CI/CD、敏捷开发、开发测试等均属于其范畴。

1.1.2 DevOps 的起源

DevOps 起源于 AWS 和 Google 等大型互联网公司，由于地理位置分散、员工人数众多、产品研发和部门组织关系复杂等诸多原因，这些公司无疑都需要员工紧密协作、提高效率，同时又不希望出现部门分裂。

可以说，DevOps 的诞生、发展是互联网时代大规模兴起和云计算、容器技术大规模普及的综合性产物，是一个借鉴了开源世界许多协作理念的文化运动，本质上是团队的协作文化。

具体到一个业务流程操作的自动化，DevOps 的理念是让代码编译—打包—测试—部署—交付由过去的手工操作完成，变成自动化完成，从而大大节省成本。DevOps 的发展和应用，从来都不是某个人的事情，而是包括了 CI/CD、敏捷开发和团队等要素于一体的协同工作。

DevOps 的出现是由于软件行业日益清晰地认识到：为了按时交付软件产品和服务，开发和运维工作必须紧密合作。也只有结合实际的业务场景需求，才能做到 DevOps 的最佳实践，不要为了敏捷而敏捷，为了 DevOps 而 DevOps。须知，人员的密切配合、流程的自动化和团队文化才是推动 DevOps 最佳实践、提高效率、实现快速交付服务的三驾马车。

1.1.3　DevOps 的发展

一个既定事实是，把开发和运维作为整体来看待的 DevOps 思想已经逐步深入人心，那么我们除了探讨 DevOps 的起源之外，还应该关注它的发展。

从横向比较来看，DevOps 的每一次发展历程都随着互联网开源技术的变化而变化，与其息息相关。随之也逐步有了对 DevOps 系统性需求，用户希望能有个平台或流程来统一支持开发和运维的交付工作及之后的管理工作，即需要一系列的持续集成、持续交付、自动化部署、自动化测试监控、自动化扩展伸缩等，以提升开发—测试—运维过程中的效率，简化这些过程的管理，降低交付风险和沟通成本及运营成本。

从广义上来讲，不管是云管理平台（如 OpenStack），还是各种 PaaS 平台（如 CloudFoundry、Docker），抑或是自动化部署工具（如 Chef、Puppet 和 Ansible）等都是 DevOps 系统的一部分，都是为了解决开发过程中的交付环节问题和交付后的运维管理问题。

一个共用的场景是，在开发和测试过程中，这些平台帮助开发、测试人员搭建和管理环境，以便在代码变更后自动部署以做测试。

随着云计算和容器技术的发展，以及产品业务对 IT 能力的需求推动，DevOps 也愈加清晰和被广泛使用。回顾其发展路径和变迁过程，基本可以将其分为三代：基于物理机的部署时代、基于 IaaS 云可弹性的部署时代和基于 Docker 容器的部署时代。随着这三代的改进，DevOps 系统的应用能力也越来越强。

比如，要把一套运行在 AWS 云虚拟机上的服务迁移到 OpenStack 云虚拟机上，同时这些服务是运行在 Docker 容器中的，当迁移时，只需要在 OpenStack 端的虚拟机上 pull（拉取）镜像，并进行少量配置，即可重新运行，真正实现了便捷的跨云迁移和弹性动态的伸缩服务。

1.2　DevOps 在研发测试中的应用

在传统的软件研发活动中，经常会遇到诸如部门或团队之间在开发、测试或运维中的鸿沟，彼此之间的分离，直接或间接影响到软件产品的产出效率。DevOps 模式具有众多优势，在软件活动中，该如何使用呢。