



OpenStack 实战

OpenStack
IN ACTION

〔美〕V. K. Cody Bumgardner 著
颜海峰 译

OpenStack 实战

OpenStack
IN ACTION

〔美〕V. K. Cody Bumgardner 著
颜海峰 译

人民邮电出版社
北京

异步社区电子书

感谢您购买异步社区电子书！异步社区已上架电子书 500 余种，社区还会经常发布福利信息，对社区有贡献的读者赠送免费样书券、优惠码、积分等等，希望您在阅读过程中，把您的阅读体验传递给我们，让我们了解读者心声，有问题我们会及时修正。

社区网址：<http://www.epubit.com.cn/>

反馈邮箱：contact@epubit.com.cn

异步社区里有什么？

图书、电子书（[半价电子书](#)）、优秀作译者、访谈、技术会议播报、赠书活动、下载资源。

异步社区特色：

纸书、电子书同步上架、纸电捆绑超值优惠购买。

最新精品技术图书全网首发预售。

晒单有意外惊喜！

异步社区里可以做什么？

博客式写作发表文章，提交勘误赚取积分，积分兑换样书，写书评赢样书券等。

联系我们：

微博：

@ 人邮异步社区

@ 人民邮电出版社 - 信息技术分社

微信公众号：

人邮 IT 书坊

异步社区

QQ 群：368449889

图书在版编目（C I P）数据

OpenStack实战 / (美) V. K. 科迪·布姆加德纳著 ;
颜海峰译. -- 北京 : 人民邮电出版社, 2017. 5
书名原文: OpenStack in Action
ISBN 978-7-115-45013-5

I. ①O… II. ①V… ②颜… III. ①计算机网络
IV. ①TP393

中国版本图书馆CIP数据核字(2017)第043853号

版 权 声 明

Original English language edition, entitled *OpenStack in Action* by V.K. Cody Bumgardner published by Manning Publications Co., 209 Bruce Park Avenue, Greenwich, CT 06830. Copyright ©2016 by Manning Publications Co. Simplified Chinese-language edition copyright ©2017 by Posts & Telecom Press. All rights reserved.
本书中文简体字版由 Manning Publications Co.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。

-
- ◆ 著 [美] V. K. Cody Bumgardner
 - 译 颜海峰
 - 责任编辑 杨海玲
 - 责任印制 焦志炜

 - ◆ 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>
 - 北京市昌平百善印刷厂印刷

 - ◆ 开本: 800×1000 1/16
 - 印张: 20.5
 - 字数: 440 千字 2017 年 5 月第 1 版
 - 印数: 1 - 3 000 册 2017 年 5 月北京第 1 次印刷

 - 著作权合同登记号 图字: 01-2016-3956 号
-

定价: 79.00 元

读者服务热线: (010)81055410 印装质量热线: (010)81055316

反盗版热线: (010)81055315

广告经营许可证: 京东工商广字第 8052 号

内容提要

本书的主题是通过 OpenStack 来部署企业私有云。本书不只是像技术手册一样介绍如何部署 OpenStack，还会解释各个步骤涉及的原理以及这项技术对业界的影响。

本书分为 3 个部分，第一部分（第 1 章～第 4 章）是入门指南，先介绍 OpenStack 云操作系统，然后让读者直接通过一个快速部署工具和最小化的基础设施来快速体验 OpenStack，再介绍 OpenStack 命令行工具（CLI），并通过使用 OpenStack 来理解组件的功能和整个 OpenStack 框架里各个组件之间的交互；第二部分（第 5 章～第 8 章）关注整个生态系统，深入介绍 OpenStack 的一个核心项目，并带领读者进行多节点环境下的 OpenStack 手动部署；第三部分（第 9 章～第 12 章）阐述在生产环境中如何使用 OpenStack，重点介绍与生产环境中 OpenStack 部署相关的架构、组织和策略决策，Ceph 存储的基本部署和操作，使用 Fuel 来进行 OpenStack 自动化 HA（高可用）部署，以及通过 OpenStack Heat 和 Ubuntu Juju 进行云编排。此外，还有一个附录介绍从裸设备安装 Linux 操作系统的详细步骤。

本书适合对使用 OpenStack 来构建私有云环境有兴趣的基础设施专家、工程师、架构师和技术支持人员阅读。因为阅读本书不需要有很好的技术基础，只要了解 Linux 的基本操作，所以不同背景和技术能力的人群都可以阅读本书。

对本书的赞誉

开源项目 OpenStack 从 2010 年 7 月诞生以来得到了迅猛发展，一路竞争，击败了其他各路云计算的开源项目，得到了全球 400 多家企业和近 5 万名个人会员的支持，成为云计算提供“基础架构即服务”的事实标准。CERN、沃尔玛、Comcast、NTT Group、Paypal、中国移动和国家电网等一批知名企业和机构都相继选择使用 OpenStack 作为自己的业务支撑平台，国内各企业、高校和开源社区也掀起了学习 OpenStack 的热潮，各类 Meetup 分享会、开源论坛和相关报道层出不穷，数不胜数。但 OpenStack 进入中国这几年，OpenStack 的运维和开发方面的书籍并不多，系统地介绍 OpenStack 各主要模块部署的书更是鲜见，这与国内学习需求和热情产生了供需矛盾。究其原因，一是开源项目代码更新快，二是一般情况下国外运用实践早于国内且经验丰富，外加语言的因素。

《OpenStack 实战》这本书系统地阐述了 OpenStack 计算、网络和块存储等节点的部署，内容丰富，得到了 Jay Pipes 等人的高度赞扬。及时地将这本书引进国内并翻译出版，让国内读者能轻松地理解私有云构建架构和过程，从部署实战的角度出发帮助广大国内 OpenStack 运维人员和开发者深入了解和掌握 OpenStack，意义重大。相信通过阅读这本书，广大读者可以为规划和运维基于 OpenStack 的企业私有云打下坚实的基础。

——王庆

OpenStack 基金会个人独立董事，英特尔开源技术中心云计算和网络部研发经理
一个开源软件的成熟的特征就是有相应的中文书籍，OpenStack 也不例外。把国外的优质 OpenStack 书籍翻译成中文，可以培养更多的 OpenStack 专业人才，大大加快国内 OpenStack 在企业的普及速度。

——陈沙克

OpenStack 技术博主 (<http://www.chenshake.com/>)，九州云信息科技有限公司副总裁
《OpenStack 实战》不仅提供了真实案例，供读者学习如何开发自己的云平台，而且介绍了为什么要构建私有云、私有云技术栈选择、物理硬件集群规划等内容。阅读本书，读者将会系统地

了解如何构建面向用户的基础架构服务以及部署和运行 OpenStack 云的技术细节。

——叶璐

OpenStack 中国区大使，Qunar DevOPS 工程师

动手实践一直是技术人员对复杂技术加深理解的有效途径，这样一本理论和实践结合的 OpenStack 指导手册不仅为入门使用者提供快速参考，更为深入研究 OpenStack 者提供辅助。

——郭长波

OpenStack 基金会个人独立董事，Oslo 项目 PTL

异步社区
www.epubit.com.cn

译者序

OpenStack 项目的特点通过它的名字就很好地展示了——Open（开放）。我开始接触 OpenStack 时，CloudStack 被认为是最稳定的，而且那时全球范围内有几家电信企业采用了 CloudStack。电信企业的采用是很重要的信号，因为电信行业对可用性要求比较高。但后来居上的 OpenStack 从一开始就推崇“开放”，号称 4 个开放：开源（Open Source）、开放式设计（Open Design）、开放式开发（Open Development）和开放式社区（Open Community）。2016 年下半年参加的 OpenStack 巴塞罗那峰会，在主题演讲会场上，OpenStack 基金会首席运营官 Mark Collier 再次重申了上面这 4 个开放。

- 开源：有很多开源软件的所有者同时发布社区版与企业版（专业版），但 OpenStack 从一开始就只有社区版，没有任何企业版。
- 开放式设计：OpenStack 有很多项目，每个项目都有一个项目技术领导（Project Technical Leader, PTL），还有若干个核心成员（Core Member）。这些职位都不是某个公司指派的，而是在很透明的机制下提名产生的。每年两次的 OpenStack 峰会就包括设计峰会，所有需求讨论都是公开的，并使用公开的文档记录工具记录，如社区使用比较多的 Etherpad（<http://etherpad.openstack.org/>）。除此之外，还使用 IRC 和邮件列表，IRC 更适用于即时通信，即所有参与讨论者都是在线的情况，邮件列表则更多地考虑可能有不在线的参与者有必要参与讨论的情况。
- 开放式开发：除了在设计峰会上讨论需求外，平时的新功能（Blueprints, BP）实现需求都通过 Launchpad（<https://launchpad.net/>）进行管理，当然 bug 也通过 Launchpad 管理。所有人都可以注册为开发者，在 Launchpad 提 bug 或者 BP，代码审查是通过社区基础设施团队架设的 Gerrit 服务器（<https://review.openstack.org>）进行的，所有代码都是先提交到 Gerrit 上，所有开发者都可以评论和打分。因此，整个开发过程都是透明的。
- 开放式社区：基金会的 13 名技术委员（Technical Committee, TC）全部由所有活跃开发者（Active Technical Contributors, ATC）选举产生，基金会有 3 类董事成员，每类

8 名, 共 24 名, 其中有 8 名是个人董事, 通过所有社区参与者选举产生。另外, 在 2016 年的峰会上还讨论了成立用户委员 (User Committee, UC), 由 CERN 的 Tim Bell 来帮忙建立。这种大的决策不是某人关起门做出的, 而是摆上桌面来讨论的。

也许正是 OpenStack 如此开放, 才吸引了众多公司与全球 170 多个国家超过 5 万名开发者参与进来。

感谢带领我进入实验室接触到 OpenStack 的邝颖杰老师, 还有当时的队友区家华和林泽强两位同学。记得当时是从 F 版开始的, 坑还是比较多的。那时很多人的博客文章都给了我很大的帮助, 感谢他们无私奉献, 分享自己的经验。感谢带领我进入唯品会构建云平台的陈展奇先生, 让我真正接触到企业级需求。从接触 OpenStack 到现在 4 年左右的时间, 我看到了 OpenStack 越来越稳定, 也学到了很多東西, 也有很多人都给予了我帮助, 感谢各位。

颜海峰

OpenStack 巴塞罗那峰会有感

异步社区
www.epubit.com.cn

译者简介

颜海峰，目前就职于 HPE，在云计算研发部门担任软件工程师。华南农业大学学生 IT 研发中心首届成员，也正是在此实验室开始了云计算的奇幻之旅。曾任唯品会企业私有云计算平台、网宿弹性计算服务的首批研发工程师。

异步社区
www.epubit.com

序

很难相信从我开始看到 Nova 的最初源代码到现在已经 5 年了。当时这些代码刚发布，它们是由 Anso Labs 团队为 NASA 创建的。当时我在 Rackspace 公司工作，公司正在寻找一套新的代码库作为下一代 Rackspace 云。几个月后，Rackspace 开源了它的 Rackspace 云文件平台作为 Swift 项目。Nova 和 Swift 成为新生的 OpenStack 项目最开始的两个顶梁柱。

从那时开始，这两个项目都经历了显著的变化。Swift 项目的核心团队和代码库还相对稳定些，尽管项目增加了不少新功能，性能和扩展性也有所提升。此外，与开始时的简陋相比，现在 Nova 项目的源代码基本上算是“脱胎换骨”了。新的代码库（如 Glance、Cinder、Keystone 和 Neutron）的创建都是为了提供原本 Nova 提供的功能。

这些新的源代码被创建出来处理大规模计算基础设施的功能性管理，同时一个新型的开源社区也开始形成。在操作系统分发和打包、配置管理、数据库设计、自动化、网络和存储系统有经验的开源开发者和支持者聚拢起来为 OpenStack 贡献自己的力量。

我们的社区以极快的速度成长（并继续成长），快速成为全球最大、最活跃和最有影响力的开源社区之一。为了社区更好地发展，更好地面对管理上的挑战，OpenStack 基金会应运而生。每年在世界各地举办的设计峰会和会议都会吸引全球超过 3500 名贡献者参与。社区创建了一个世界级的持续集成和构建系统来支持源代码和贡献者的快速增长。这些自动化构建系统的规模和范围可以比肩甚至超过了一些比较老的开源社区，如 Apache 和 Eclipse 基金会。

OpenStack 的生态系统成为了一些新成立的公司（如 SwiftStack 和 Piston Cloud）的沃土。同样，对原有的公司（如惠普、Mirantis 和 Red Hat）也是相当有益，它们通过 OpenStack 社区旗下多个项目持续推动创新。

OpenStack 社区的“膨胀”也让如何部署这些分布式软件组件以及如何运维它们变得更加复杂。如果想要“从零开始”部署 OpenStack，就必须掌握从网络和存储到虚拟化和配置管理这些广泛的知识。获取这些必备的知识 and 技能已经成为使用 OpenStack 构建云平台要面临的关键挑战之一。这本书就提供了部署和运营 OpenStack 必备的知识。

在本书中，作者为读者剖析了 OpenStack 部署的复杂过程，介绍了 3 种部署方式：通过一

个叫作 DevStack 的脚本工具,通过手动安装操作系统软件包,以及通过 Fuel OpenStack 安装器。在每一节中,网络和存储的设置都有详细的解释,让读者逐步接触云计算,并在读完本书后能够很轻松地深入云计算的海洋。

作者除了介绍了 OpenStack 技术,还解释了如何评估和怎样让你的组织从云计算中受益。云并不能神奇到把很多组织里基于人力的人工和耗时的过程问题解决掉。但通过灵活合理的应用过程,云可以让 IT 组织发生变革,并显著提高它们提供服务的质量。在第 9 章中, Bumgardner 提供了任何正在考虑把现有的虚拟化 IT 基础设施替换成 OpenStack 或者正在为内部用户构建一个新的私有云平台的 IT 总监都必读的内容。

总的来说,本书就像是复杂的云计算世界和 OpenStack 软件生态系统里专业的启蒙导师。阅读它,吸收它所讲的知识,就可以彻底变成一个“Stacker”!

Jay Pipes

OpenStack 技术委员会成员

Mirantis 公司技术总监

前言

我首次接触 OpenStack 是在 2011 年夏天,那时我就职于肯塔基大学(University of Kentucky)。我和我的同事兼好友 Brent Salisbury 被邀请参与一家财富 50 强技术公司关于产品研发项目的讨论。在会议期间,项目的执行发起人给了我们两个选择,使用现有的商业工具或者调研使用一个名为 OpenStack 的社区项目。自然结果是我们选择了调研我们一无所知的框架,就这样开始了我们的 OpenStack 之旅。这个开发项目没有任何产品产出,但与 OpenStack 的“不期而遇”却成为了我学术和职业生涯的转折点。Brent 离开了肯塔基大学,与别人合办了一家创业公司,后来该公司被 Docker 收购了,于是他现在正就职于 Docker 公司。而我,走了与他不同的道路,从硕士变成博士并写了这本书。

在 2013 年年初, OpenStack 的 Grizzly 版本跟当前的版本类似,但由于当时功能快速增加导致不稳定,让我们觉得在企业环境里, OpenStack 还不能用于生产环境。虽然我没有冒险在企业里使用 OpenStack,但研究计算是另一回事。作为研究生独立研究课程,我在研究计算时记录了使用 OpenStack 的案例、架构和策略。不止这样,我还描述了这个过程,最后在我们企业内部采用这个平台作为私有云。

我在原创的学术报告里使用图 1 来表示 OpenStack 在组件层面的分布式体现。我推测要烹饪一头大象,应该跟吃它一样,一次只能一块。在技术领域里,我们经常能接受技术分离作为一个组织的稳健做法——“我是存储人”或者“我是网络人”——但对很多人来说,首要任务就是只吃大象的一部分。在本书中,我会尽量将一些易懂的知识和新概念结合起来,让读者更容易理解。虽然可能你不想品尝大象的脚,但如果想成功地玩转云计算,最好在原理上知道它们是怎样工作的。

现在我写这个前言,刚好是 Manning 出版社的编辑第一次找我的两年后。当我开始这个项目时,还只有不到 500 名 OpenStack 贡献者,但现在已经有好几千名了。OpenStack 不但成为了成长最快的开源社区之一,同时也被全球规模靠前的很多组织采用。更重要的是,至少对你来说, OpenStack 现在已经足够成熟了,可以作为你的组织的私有云的基础。

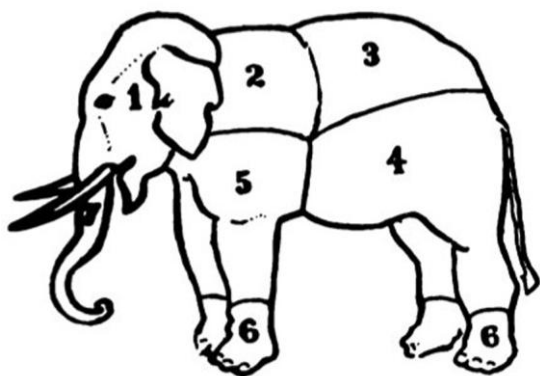


图 1 这幅图来自于 16 世纪 Maestro Martino 写的 *Libro de Arte Coquinaria* (关于烹饪艺术的书)

致谢

没有我的博士导师兼好友 Victor W. Marek 教授的鼓励，就没有本书的出版。我有责任把这种鼓励和信心传递下去。

如果不是自己经历过，我无法想象出版一本书需要付出如此多的努力。这份努力是否能产生预期的结果将取决于读者，但毫无疑问，审稿人、编辑和其他贡献者都花费了大量时间来致力于打造一本高质量的书。通过我参与贡献和审阅其他作者的书，以及我现在写的这本书，我可以很肯定地说，Manning 出版社为出版更好的书尽了最大的努力。我特别要感谢本书的策划编辑 Susan Conant 的孜孜不倦的工作，持续推进内容的改进。同时也要感谢出版人 Marjan Bace 以及编辑与生产团队的每一位成员，包括 Mary Piergies、Cynthia Kane、Andy Carroll、Katie Tennant 和幕后的所有工作人员。非常感谢 Bill Bruns、Andy Hill、Michael Kidd、Jeff Lim 和 Fabrizio Soppelsa 在本书编写期间帮忙做技术审阅。最后，还要感谢 Andy Kirsch、Chris Snow、Fernando Rodrigues、Hafizur Rahman、Kosmas Chatzimichalis、Matt Harting、Mayur Patil、Michael Hamrah、Peeyush Maharshi 和 Toby Lazar，他们阅读本书初稿并提供了很多建议。

特别感谢我的妻子 Sarah，感谢她照顾我们的两个孩子，同时理解和支持我出差、做研究生工作、编写这本书和其他各个方面，这些都是无私的。虽然论文、讲稿和书上都只有我的名字，但同样会带上我们共同拥有的姓。Sarah、Sydney 和 Jack，很抱歉没有花足够多的时间和精力陪伴你们。我希望，就像我以你们为傲一样，你们也以我为傲。我爱你们。

关于本书

本书的主题是通过 OpenStack 来部署企业私有云。在本书中，我把私有云看成是企业内部的基础设施资源池，即基础设施即服务（Infrastructure as a Service, IaaS）。相反，公有云 IaaS 资源是由第三方服务提供商拥有和运营的。

从财务角度看，可以把私有云看成主要资产成本，而公有云通常是运营成本。很容易区别两者，在私有云部署中，不管实际使用量的大小，企业通常都要购买或者把在其他地方正常工作的基础设施挪过来使用。在公有云部署中，成本通常是跟直接的占用小时（开机或关机）数和通信开销有关。

组织采用私有云还是公有云通常与组织的 IT 职责的大小和规模有关。企业的 IT 部门负责为组织其他部门集中提供技术架构和资源，是使用私有云的“既得利益者”。一个多租户、充分编排的私有云为企业 IT 提供了非常高效的资源管理。在这方面，企业 IT 部门变成一个“云代理商”。相反，部门的 IT 单元通常缺乏数据中心设施和部署性价比高的私有云的人员。通常他们只有少量的资源需求，因此部门可以充分利用公有云资源。如果可以，部门也可以充分利用由他们的企业 IT 单元提供的私有云资源。基于工作负载同时使用私有云和公有云就是混合云架构了。

尽管云和想充分利用云的企业类型不同，但不同的云可以使用相同的技术来构建。虽然构建云资源的要素可以是相同的，但是使用形式和方式可以千变万化。

OpenStack 是一个可以用来构建私有云和公有云的强有力的框架。从本质上讲，OpenStack 为构建云的硬件和软件而抽象和提供了一组通用的 API 接口。这个框架提供了两个非常重要的东西：

- 硬件和软件资源的抽象，这避免了特定组件的厂商锁定；
- 一组通用的资源管理 API 接口，这可以实现连接的组件的完全编排。

第一点从财务角度来看是非常好的，第二点是现代 IT 变革的关键。对于企业 IT 部门来说，OpenStack 为云部署带来了相同水平的高效变革。

为什么选这本书

本书希望通过一步步、自底向上的方式，为构建计算资源云提供指导。本书的目标读者包括打算部署 OpenStack 环境的研究人员、系统管理员和学生。阅读本书不需要有很好的技术基础，只要了解 Linux 的基本操作，本书的内容适合不同背景和技术能力的人群。同样，OpenStack 适用于多种用例。

尽管不同的用例都使用相同的 OpenStack 框架，但不同服务提供商的私有云的要求和设计会有很大区别。企业都希望为企业内部提供私有资源云。这种类型的私有云不只是代表了一种额外的服务，更是代表了组织提供计算资源方式的一个转变。

本书由以下几部分组成：

- 介绍自动化部署 OpenStack 单节点开发环境；
- 通过一步步手动部署多节点环境来深入理解 OpenStack；
- 从 IT 运维角度来介绍私有云技术（OpenStack、Ceph、Juju 等）带来的影响；
- 使用厂商提供的自动化部署和管理工具来部署生产级别的 OpenStack 环境。

本书介绍的架构适合于从小规模（5 个节点）到大规模（100 个节点）企业的私有云部署。另外，第 12 章还会介绍如何在新构建的私有云上使用应用编排工具，如 OpenStack Heat 和 Ubuntu Juju。

本书会让读者理解私有云技术、这些技术的部署和运维，以及云编排对传统 IT 角色的长期影响。本书会帮助读者更好地说服企业在其内部部署 OpenStack 私有云，同时帮助读者提高私有云方面的技能。本书的配置和操作脚本可以到 GitHub 下载：<https://github.com/codybum/OpenStackInAction>。

对读者而言，需要理解的最重要的一点是，OpenStack 私有云不是另一种简单的虚拟化工具。OpenStack 是充分利用现有的虚拟化工具来构建和管理云的框架。读者将会学到如何构建、部署和管理云。从技术层面来看，读者将会理解 OpenStack 各个组件和支持技术，特别是 OpenStack 计算、网络、块存储、Dashboard 和 API 组件。

内容路线图

本书划分为 3 个部分，第一部分（第 1 章～第 4 章）是入门指南，第二部分（第 5 章～第 8 章）深入介绍整个生态系统，第三部分（第 9 章～第 12 章）介绍在生产环境中如何使用 OpenStack。

第 1 章介绍 OpenStack 云操作系统、开发这个框架的动机和 OpenStack 能为你的组织做些什么。

第 2 章将直接通过一个快速部署工具和最小化的基础设施来快速体验 OpenStack。这个体验

不只是演示 OpenStack Dashboard 的使用，还提供一个学习 OpenStack 框架时可以运行的模型。在第 2 章结尾，读者可以通过自己的 OpenStack 环境来创建虚拟机。

第 3 章会使用第 2 章构建的环境，介绍 OpenStack 命令行工具 (CLI)。这一章介绍 OpenStack 的基本操作，如创建新租户（项目）、用户、角色和内部网络等。

第 4 章中，通过使用 OpenStack 来理解组件的功能和整个 OpenStack 框架里各个组件之间的交互。读者可以学到多种云设计方法，让自己可以准备自己的多节点部署。这一章还会介绍 OpenStack 组件是如何协同工作的，以及它们与厂商资源的关系。

第 5 章～第 8 章，每章分别深入介绍 OpenStack 的一个核心项目。这几章会带领读者进行多节点环境下的 OpenStack 手动部署。通过这几章，读者将会更好地理解在 OpenStack 生态里面它们是如何工作的。另外，这些手动部署工作将会带给读者宝贵的问题排查经验。

第 9 章介绍与生产环境中 OpenStack 部署相关的架构、组织和策略决策。第 10 章介绍 Ceph 存储的基本部署和操作。第 11 章会使用 Fuel 来进行 OpenStack 自动化高可用 (HA) 部署。最后，第 12 章会介绍通过 OpenStack Heat 和 Ubuntu Juju 进行云编排。

谁应该读这本书

本书适合对使用 OpenStack 来构建私有云环境有兴趣的基础设施专家、工程师、架构师和技术支持人员阅读。本书对身为领导者和战略角色的人有一定的战略价值，其内容同时也适合技术型读者。阅读本书不需要有很好的技术基础，只要了解 Linux 的基本操作就可以。

代码约定和下载

本书所有的代码都会用等宽字体与其他文本内容加以区分。代码注释伴随在很多代码清单中，突出重要的概念。在一些例子中，数字编号链接到代码清单后面的解释。

读者可以在 Manning 出版社的网站 www.manning.com/books/openstack-in-action 和 <https://github.com/codybum/OpenStackInAction> 下载本书的示例代码。

作者在线

本书包含一个可以免费访问的由 Manning 出版社运营的私有网络论坛，读者可以在这里发表对本书的评论，询问技术类问题，可以得到作者或者其他读者的帮助。可以通过 www.manning.com/books/openstack-in-action 访问这个论坛。这个网页提供了注册后如何访问这个论坛，可以获得哪些帮助，以及在论坛发表文字的规则。

Manning 出版社为读者与读者之间、读者和作者之间的沟通提供了场所。但由于作者参与论坛讨论和解答问题是完全自愿的（无偿的），因此作者对在论坛解答读者问题不作任何承诺。我

们建议读者向作者提出有挑战性的问题来激发作者的兴趣！当本书正式出版时，作者在线论坛和前面提到的文档都可以通过出版社网站访问。

关于作者

Cody Bumgardner (<http://codybum.com>) 有超过 20 年的 IT 行业的从业经验，在 IT 架构、软件开发、网络、研究、系统和安全领域扮演过技术、管理和销售角色。最近几年，作者主要专注于研究、实现和介绍云计算和计算经济学。他现在还是肯塔基大学计算机科学的博士生，专注于计算经济学和分布式资源管理。Cody 现在还是一家公立大学的首席技术架构师（CTA）。作者作为首席技术架构师为就职的学校制定了一个 5 年的云计算战略和路线图。这个路线图概述了颠覆性的云技术，并给出了全体 IT 人员的相关转变。这个计划以企业 OpenStack 私有云部署为核心，支持超过 4 万个包括学术、研究和医疗卫生（学术）方面的用户。Cody 主要负责 OpenStack 私有云、计算研究和云计算其他前沿技术的架构、账务建模、部署和长期战略规划。

关于封面

本书封面上的图像的说明文字是“一位来自库唐斯的挤奶女工”。这张插图取自经 Louis Curmer 编辑并于 1841 年在巴黎出版的很多艺术家的作品汇集而成的作品集。这个作品集的名字是 *Les Français peints par eux-mêmes*，翻译出来意思是“法国人眼中的自己”。每一幅图集都是精工细作并手工着色的，这些丰富多样的图集形象生动地提醒着，200 多年前的文化是如何分隔在世界不同地区、城镇、乡村和居民区的。人与人之间相互独立，说着不一样的方言和语言。站在大街或者乡村小道上，从衣着打扮就很容易分清他们住在哪里、卖些什么、处于什么样的地位。

那时如此多元化的区域差异逐渐消失了，而从那时起衣服着装也开始变化。现在不同州的居民都很难辨认出来，更别说不同城镇和地区的了。也许我们以文化多元化换得了一个更加丰富多彩的个人生活，当然也是一个更加多变、快节奏的科技生活。

在一个科技类图书众多的时代，Manning 以印有两个世纪以前丰富多样的地区生活的图书封面赞颂了计算机事业的创新性和自主精神，这些插图又将创新性和自主精神重新带回到我们的生活。

目录

第一部分 入门指南

1	第 1 章 介绍 OpenStack	2
	1.1 OpenStack 是什么	4
	1.2 理解云计算和 OpenStack	7
	1.3 关联 OpenStack 及其控制的计算资源	8
	1.3.1 OpenStack 和 hypervisor	9
	1.3.2 OpenStack 和网络服务	11
	1.3.3 OpenStack 和存储	12
	1.3.4 OpenStack 和云专业术语	13
	1.4 OpenStack 组件介绍	14
	1.5 OpenStack 发展历史	14
	1.6 小结	15

2	第 2 章 体验使用 OpenStack	16
---	----------------------	----

2.1 DevStack 是什么	17
2.2 部署 DevStack	18
2.2.1 搭建服务器	20
2.2.2 准备服务器环境	21
2.2.3 准备 DevStack	22
2.2.4 执行 DevStack	24
2.3 使用 OpenStack Dashboard	30
2.3.1 Overview 界面	32

2.3.2 Access & Security 界面	32
2.3.3 Images & Snapshots 界面	35
2.3.4 Volumes 界面	38
2.3.5 Instances 界面	39
2.4 访问第一个私有云服务器	43
2.4.1 为实例分配浮动 IP	44
2.4.2 允许到达浮动 IP 的网络访问	45
2.5 小结	45

3	第 3 章 OpenStack 基本操作	46
---	----------------------	----

3.1 使用 OpenStack CLI	47
3.2 使用 OpenStack API	49
3.3 租户模型操作	50
3.3.1 租户模型	50
3.3.2 创建租户、用户和角色	52
3.3.3 租户网络	55
3.4 配额	67
3.4.1 租户配额	67
3.4.2 租户用户配额	69
3.4.3 额外配额	70
3.5 小结	71

4	第 4 章 理解私有云构建模块	72
---	-----------------	----

4.1 OpenStack 组件间如何关联	72
-----------------------	----

- 4.1.1 理解组件间交互 73
- 4.1.2 分布式计算模型 78
- 4.2 OpenStack 与厂商技术的
关系 82
 - 4.2.1 OpenStack 使用厂商存储
系统 82
 - 4.2.2 OpenStack 里使用厂商提
供的网络系统 86
- 4.3 为什么要手动部署 93
- 4.4 小结 94

第二部分 手动部署

第 5 章 控制器部署 96

- 5.1 部署控制器必备软件
97
 - 5.1.1 准备环境 99
 - 5.1.2 配置网络接口 99
 - 5.1.3 更新安装包 102
 - 5.1.4 安装软件依赖 103
- 5.2 部署共享服务 106
 - 5.2.1 部署身份认证服务
(Keystone) 107
 - 5.2.2 部署镜像服务 (Glance)
116
- 5.3 部署块存储 (Cinder) 服
务 123
 - 5.3.1 创建 Cinder 数据存
储 123
 - 5.3.2 配置 Cinder 的 Keystone 用
户 125
 - 5.3.3 创建 Cinder 服务和端点
126
 - 5.3.4 安装 Cinder 127
- 5.4 部署网络 (Neutron) 服务
128
 - 5.4.1 创建 Neutron 数据存
储 128
 - 5.4.2 配置 Neutron 的 Keystone
用户 129
 - 5.4.3 安装 Neutron 131
- 5.5 部署计算 (Nova) 服务
133
 - 5.5.1 创建 Nova 数据存
储 133
 - 5.5.2 配置 Nova 的 Keystone 用户

- 134
- 5.5.3 分配角色给 nova 用户
135
- 5.5.4 创建 Nova 服务和端点
135
- 5.5.5 安装 Nova 控制器 136
- 5.6 部署 Dashboard (Horizon)
服务 138
 - 5.6.1 安装 Horizon 139
 - 5.6.2 访问 Horizon 139
 - 5.6.3 调试 Horizon 139
- 5.7 小结 140

第 6 章 网络节点部署 141

- 6.1 准备网络节点部署环境
142
 - 6.1.1 准备环境 143
 - 6.1.2 配置网络接口 143
 - 6.1.3 更新安装包 147
 - 6.1.4 软件和配置依赖 147
 - 6.1.5 安装 Open vSwitch 149
 - 6.1.6 配置 Open vSwitch 152
- 6.2 安装 Neutron 154
 - 6.2.1 安装 Neutron 组件 155
 - 6.2.2 配置 Neutron 155
 - 6.2.3 配置 Neutron ML2 插件
156
 - 6.2.4 配置 Neutron L3 代理 157
 - 6.2.5 配置 Neutron DHCP 代理
158
 - 6.2.6 配置 Neutron 元数据代理
158
 - 6.2.7 重启和验证 Neutron 代理
159
 - 6.2.8 创建 Neutron 网络 159
 - 6.2.9 关联 Linux、OVS 和
Neutron 168
 - 6.2.10 检查 Horizon 170
- 6.3 小结 171

第 7 章 块存储节点部署 172

- 7.1 准备块存储节点部署环境
173
 - 7.1.1 准备环境 174
 - 7.1.2 配置网络接口 175
 - 7.1.3 更新安装包 178
 - 7.1.4 安装和配置逻辑卷管理器
178

- 7.2 部署 Cinder 182
 - 7.2.1 安装 Cinder 183
 - 7.2.2 配置 Cinder 184
 - 7.2.3 重启和验证 Cinder 代理 185
- 7.3 测试 Cinder 186
 - 7.3.1 创建 Cinder 卷: 命令行 186
 - 7.3.2 创建 Cinder 卷: Dashboard 188
- 7.4 小结 190

8 第 8 章 计算节点部署 191

- 8.1 准备计算节点的部署环境 192
 - 8.1.1 准备环境 193
 - 8.1.2 配置网络接口 193
 - 8.1.3 更新安装包 196
 - 8.1.4 软件和配置依赖 196
 - 8.1.5 安装 Open vSwitch 197
 - 8.1.6 配置 Open vSwitch 198
- 8.2 安装 hypervisor 200
 - 8.2.1 验证作为 hypervisor 平台的主机 200
 - 8.2.2 使用 KVM 201
- 8.3 在计算节点安装 Neutron 203
 - 8.3.1 安装 Neutron 软件 203
 - 8.3.2 配置 Neutron 204
 - 8.3.3 配置 Neutron ML2 插件 204
- 8.4 在计算节点安装 Nova 205
 - 8.4.1 安装 Nova 软件 205
 - 8.4.2 配置核心 Nova 组件 206
 - 8.4.3 检查 Horizon 207
- 8.5 测试 Nova 208
- 8.6 小结 211

第三部分 构建生产环境

9 第 9 章 设计自己的 OpenStack 架构 214

- 9.1 替换现有的虚拟服务器平

台 214

- 9.1.1 部署选择 216
- 9.1.2 使用何种类型的网络 217
- 9.1.3 使用何种类型的存储 218
- 9.1.4 使用何种服务器 220
- 9.2 为什么要构建私有云 221
 - 9.2.1 公有云规模经济的观念 222
 - 9.2.2 全球规模或严格控制 222
 - 9.2.3 不公开的数据引力 222
 - 9.2.4 混合云 223
- 9.3 构建私有云 223
 - 9.3.1 OpenStack 部署工具 224
 - 9.3.2 私有云的网络 225
 - 9.3.3 私有云存储 226
- 9.4 小结 227

第 10 章 部署 Ceph 228

- 10.1 准备 Ceph 节点 228
 - 10.1.1 节点认证与授权 229
 - 10.1.2 部署 Ceph 软件 232
- 10.2 创建一个 Ceph 集群 233
 - 10.2.1 创建初始配置 233
 - 10.2.2 部署 Ceph 软件 234
 - 10.2.3 部署初始配置 235
- 10.3 添加 OSD 资源 236
 - 10.3.1 准备 OSD 设备 237
 - 10.3.2 创建 OSD 239
- 10.4 Ceph 基本操作 240
 - 10.4.1 Ceph 存储池 240
 - 10.4.2 对 Ceph 集群进行基准测试 241
- 10.5 小结 244

11 第 11 章 使用 Fuel 进行自动化的高可用 OpenStack 部署 245

- 11.1 准备你的环境 246
 - 11.1.1 网络硬件 246
 - 11.1.2 服务器硬件 249
- 11.2 部署 Fuel 255

11.3 基于 Web 的基本 Fuel OpenStack 部署	257
11.3.1 服务器发现	258
11.3.2 创建 Fuel 部署环境	259
11.3.3 为环境配置网络	260
11.3.4 分配主机到你的环境	261
11.3.5 完成设置和验证	263
11.3.6 部署变更	264
11.4 小结	264

第 12 章 利用 OpenStack 进行云 编排	265
-------------------------------	-----

12.1 OpenStack Heat	265
12.1.1 Heat 模板	266
12.1.2 Heat 演示	268
12.2 Ubuntu Juju	272
12.2.1 为 Juju 准备 OpenStack	273
12.2.2 安装 Juju	274
12.2.3 部署 charms CLI	277
12.2.4 部署 Juju GUI	279
12.3 小结	283

附录 安装 Linux	284
-------------	-----

第一部分

入门指南

本书的第一部分是对 OpenStack 框架的介绍：为什么要使用它和如何使用它。剖析 OpenStack 各个组件，解释它们与底层资源（计算、存储、网络等）的关系。这一部分将会带领你在单个节点上通过 DevStack 部署工具来部署 OpenStack。同时这一部分内容还会帮你思考如何将 OpenStack 用在你的环境中，并激发你对这个框架的兴趣，继续探索本书后面的部分，更深入地了解它是如何运作的。

第 1 章 介绍 OpenStack

本章主要内容

- OpenStack 和云生态系统
- 选择 OpenStack 的理由
- OpenStack 可以为你和你的组织做些什么
- OpenStack 的核心组件

一二十年前，很多大型的计算机硬件公司都通过自己生产制造专门的处理器来保持竞争优势。但随着成本的上升，能制造出足够数量的芯片来保持盈利的公司越来越少。于是，专门生产芯片的厂商出现了，它们可以大规模生产通用处理器，并且大大降低了成本。从一开始的只有少数计算机芯片厂商“鼓吹”的基于英特尔 x86 指令集的标准化的台式机和服务器平台，到最后形成了采用通用硬件的客户-服务器的市场格局。

在 21 世纪初的互联网风潮下，互联网快速发展，从而出现了大量大规模使用通用硬件的数据中心。虽然通用硬件设备强大且便宜，但它的架构就跟我们看到的台式机一样，不是按中心化管理的思想来设计的。没有现成的工具可以用来像管理资源池一样管理这些通用硬件设备。更糟糕的是，在那时，这些服务器缺少硬件管理的能力（辅助管理卡），看起来跟台式机一样。不像大型机和大型对称多处理结构（symmetric multiprocessing, SMP）的机器，这些通用服务器跟台式机一样，需要通过软件管理层来协调其他独立的资源。

在这个阶段，公共或者私有的组织在自己内部开发出很多管理框架来管理公共资源。图 1-1 展示了跨越多个数据中心的相互连接的资源池。通过管理框架，这些公共资源可以基于其可用性或者用户需求来灵活使用。不知道谁创造了这个术语，这种通过管理框架来灵活使用通用硬件设备的计算方式，可以说是拥有了资源“云”。

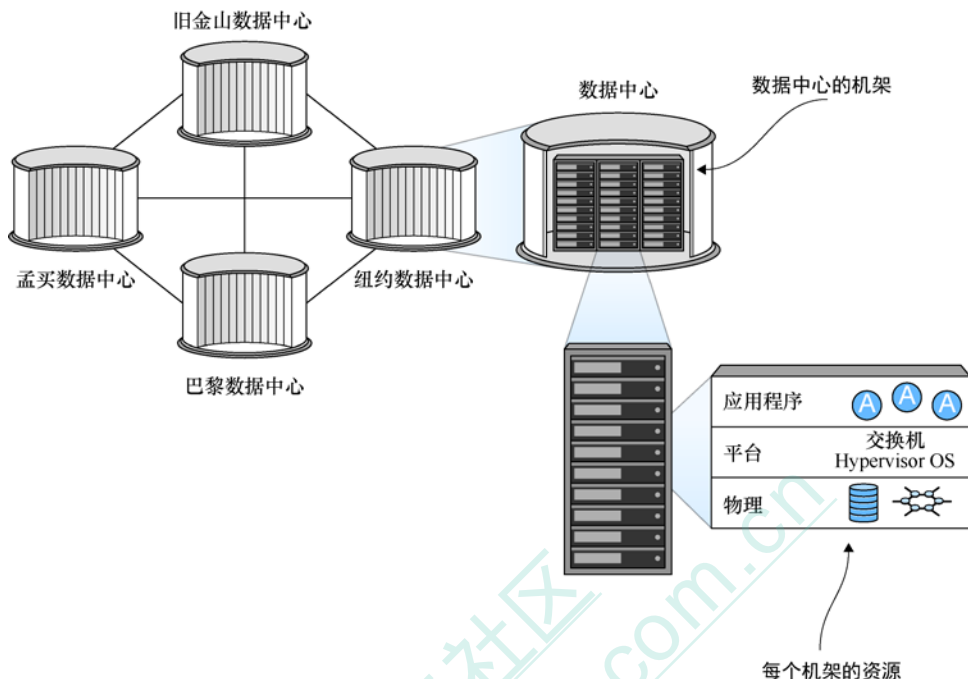


图 1-1 彼此互联的通用资源的云

在这个阶段，在很多商用或者开源的云管理软件之中，OpenStack 是最为流行的一个。OpenStack 提供了一个通用的平台来控制云计算里面的服务器（计算）、存储、网络，甚至应用资源。OpenStack 可以通过基于 Web 的界面、命令行工具（CLI）和应用程序接口（API）来进行管理。这个管理平台不仅能管理这些资源，更妙的是它不需要你去选择特定硬件或者软件厂商。厂商特定组件可以轻松地被替换成通用组件，OpenStack 为 IT 业界各类从业人员创造了价值。

一种更好理解 OpenStack 的方式是了解在亚马逊网站上购物的过程。用户登录亚马逊网站，然后购物，商品将会通过快递派送。在这种场景之下，一个高度优化的编排步骤是尽可能快并且以尽可能低的价格把商品买回家里。亚马逊成立 12 年后推出 AWS (Amazon Web Services)。AWS 把用户在亚马逊网站购买商品这种做法应用到了计算资源的交付上。一个服务器请求可能要花费本地 IT 部门几周的时间去准备，但在 AWS 上只需要准备好信用卡，然后点击几下鼠标即可完成。OpenStack 的目标就是提供像 AWS 或者其他服务提供商一样水准的高效资源编排服务。

OpenStack 是什么？

- 对于云计算平台/系统/存储/网络管理员来说，OpenStack 可以控制多种类型的商业或者开源的软硬件，提供了位于各种厂商特定资源之上的云计算资源管理层。磁盘和网络配置这些重复性手动操作任务现在可以通过 OpenStack 框架来进行自动化管理。事实上，提供虚拟机甚至上层应用的整个流程都可以通过使用 OpenStack 框架进行自动化管理。
- 对于开发者来说，OpenStack 是一个在开发环境中可以像 AWS 一样获得资源（虚拟机、

存储等)的平台,还是一个可以基于应用模板来部署可扩展应用的云编排平台。可以想象一下,通过 OpenStack 框架,可以为应用提供基础设施(X 虚拟服务器有 Y 容量内存)和相应的软件依赖(MySQL、Apache2 等)资源。

- 对于最终用户来说,OpenStack 是一个提供自助服务的基础设施和应用管理系统。用户可以做各种事情,从简单的像 AWS 一样提供虚拟机(VM),到构建高级虚拟网络和应用,这些都可以在一个独立的租户(项目)内完成。租户,也就是我们所说的项目,是 OpenStack 用来对资源分配进行隔离的方式。租户隔离了存储、网络和虚拟机这些资源,因此,最终用户可以拥有比传统虚拟服务环境更大的自由度。可以想象一下,最终用户被分配了一定额度的资源,他们可以随时获得他们想要的资源。

虚拟机和租户

在本书中,虚拟机指的是模拟物理服务器的一个实例。与物理机一样,虚拟机执行相同的功能,从操作系统的角度来看,无法区分是运行在虚拟机还是物理机上。导致虚拟机被使用的原因多种多样,但大多数的虚拟化推动力可以归结为:以牺牲性能来获得通过软件对资源的灵活控制。从一个更高的角度来说,你可以认为 OpenStack 之于数据中心,就像操作系统之于服务器,都带来了相同水平的运行效率。

读者将在本书多处看到租户这个词,在 OpenStack 里面这个词有特定含义。我们可以认为租户就是资源的配额限制集合,被虚拟机用来在逻辑上与不同租户互相隔离。例如,一个用户在租户 A 配错了网络,但租户 B 并不会受到影响。

OpenStack 基金会拥有数以百计的官方企业赞助商,以及数以万计的覆盖 130 多个国家或地区的开发者组成的社区。像 Linux 一样,很多人将会被 OpenStack 吸引,作为其他商业产品的一个开源的替代品。但他们将会逐渐认识到,对于云框架来说,没有哪个云框架拥有 OpenStack 这样的服务深度和广度。也许更为重要的是,没有其他产品,包括商业或者非商业的,能被大多数的系统管理员、开发者或者架构师使用并为他们组织创造这么大的价值。

1.1 OpenStack 是什么

让我们来详述 OpenStack 作为管理、规定和利用云资源的框架的定义。OpenStack 官方网站(www.openstack.org)这样描述这个框架:“创建私有云和公有云的开源软件。”接着是:“OpenStack 软件是一个大规模云操作系统。”如果读者有服务器虚拟化的经验,也许读者会很快地得出这样不正确的结论:OpenStack 只是提供虚拟机的另外一种方式。虽然虚拟机是 OpenStack 框架可以提供的一种服务,但这并不意味着虚拟机是 OpenStack 的全部。

图 1-2 展示了 OpenStack 通过其几个资源组件协调来提供公有云服务和私有云服务。如图所示,OpenStack 没有取代资源提供者,它只是通过框架内部的控制点来简单地管理这些资源。

一个有经验的系统管理员也许会非常怀疑 OpenStack 是一个“云操作系统”的描述。OpenStack

不像管理员通过启动盘引导启动几百台传统操作系统服务器那样，直接在裸设备上引导启动。相反，它通过对资源的管理，在云计算环境里共享操作系统的特性。

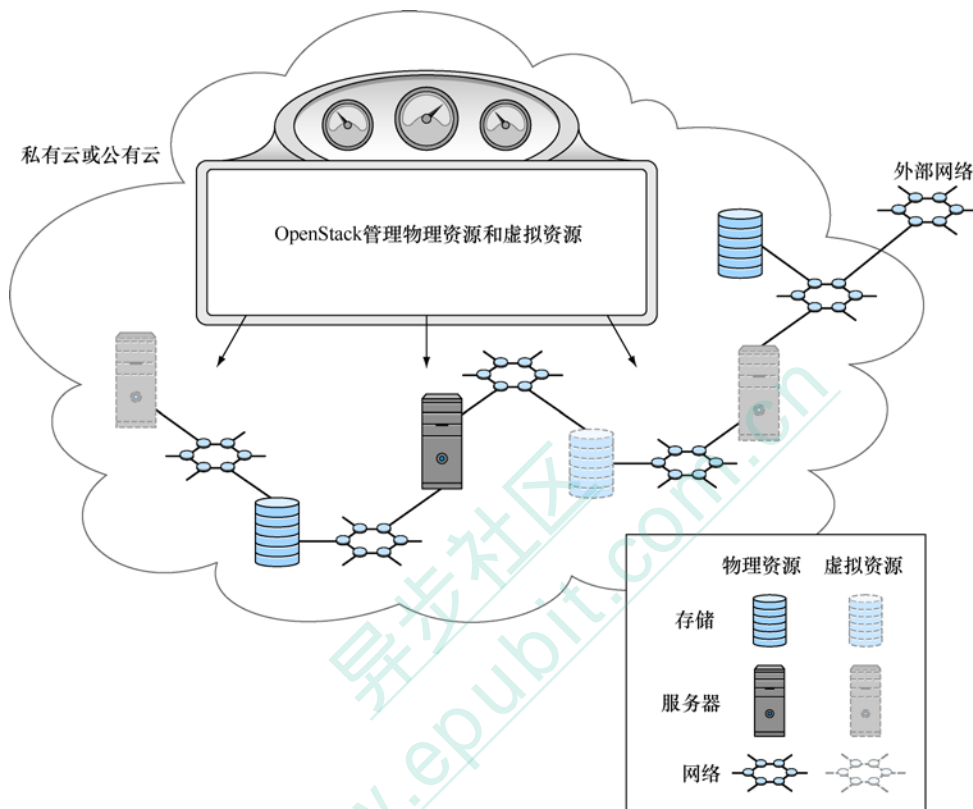


图 1-2 OpenStack 是一个云操作系统

在 OpenStack 云平台上用户可以：

- 充分利用物理服务器、虚拟服务器、网络 and 存储系统资源；
- 通过租户、配额和用户角色高效管理云资源；
- 提供一个对底层实现透明的通用的资源控制接口。

乍看之下，OpenStack 确实不像是个传统操作系统，但“云”同样不像传统计算机。我们必须回过头来重新考虑一个操作系统的根本作用。

最初，操作系统乃至硬件层面抽象语言（汇编语言）、程序都是用二进制机器码来编写的。然后传统操作系统出现了，允许用户不仅可以编写应用程序代码，还可以管理硬件功能。现在管理员可以使用通用的接口管理硬件实例，开发者可以为通用操作系统写代码，用户只需要学习一个用户交互接口即可。这样可有效地对底层硬件透明化，只需要操作系统是一样的。在计算机进化演变过程中，操作系统的发展和新操作系统的出现，给系统工程和管理领域带来了风险。

图 1-3 展示了现代计算系统的各个抽象的层次。

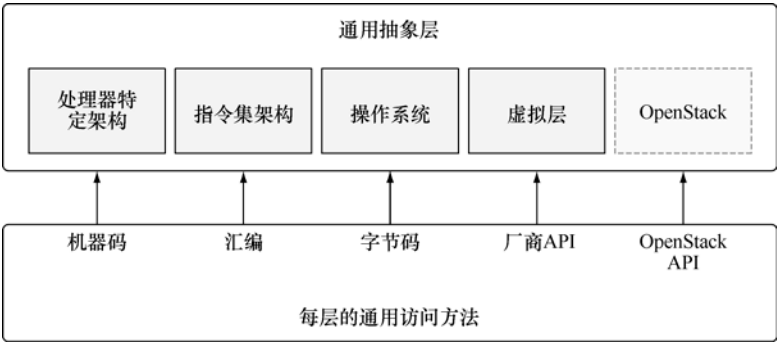


图 1-3 计算抽象的层次

毫无疑问，过去的一些开发者不想因为使用操作系统而失去了对硬件的直接控制，正如有些管理员不想因为服务器虚拟化而失去对底层硬件和操作系统的控制。在每次转变过程中，从机器码到汇编，再到虚拟层，我们一直没有失去对底层的控制；每次都是通过抽象手段简单标准化而已。我们仍然拥有高度优化的硬件，我们仍然拥有操作系统，只不过更常见的是我们拥有这些层面之间的硬件虚拟化层。

新的抽象层被广泛接受，通常是因为对标准实现优化的好处大于在这些层面上做（虚拟化）转换。也就是说，当整体计算资源的使用率能通过牺牲原生性能来得到很好的提升，那这一个层面的抽象就会被接受。这个现象可以通过中央处理器（CPU）的例子来清晰展现，这几十年，中央处理器都遵守相同的指令集，但它们内部的架构却发生了翻天覆地的变化。

大多数人想到中央处理器时，都没想到硬件层面的虚拟化和执行形式的变化，但事实就是这样。很多在 x86 处理器上执行的指令可以被处理器内部虚拟化，一些复杂的指令可以通过一系列更简单、更快速的指令来执行。指令层面优化的复杂度不在本书的讨论范围内，但很有必要去了解，即使是使用裸设备，即使是在处理器层面，也是应用到了某种形式的虚拟化。现在，与其关注失去了控制，不如想象一下，通过使用一个共同的框架来管理、监控和部署基础设施和应用的私有和公有云。只有向前迈出转变的步伐，才会真正领会 OpenStack。

数十年间 CPU 的抽象和虚拟化

英特尔的 x86 指令集首次出现在 1978 年推出的英特尔 8086 处理器上，作为英特尔 8080 处理器的向后兼容替代品。x86 指令集定义了一系列对处理器变化透明的汇编指令。从那以后，新的“处理器扩展特性”不断被添加进来，处理器时钟周期也不断提升，但已存在的指令依旧保留下来。

随着更快的处理器需求的增长，因此产生了软件能在不同代处理器之间互操作的需求。CPU 设计者需要对低级别抽象进行弹性优化，同时还要保留指令级别的兼容性（标准化）。设计者不用担心关于如何保持底层硬件一致的问题，这样他们可以在不同代处理器间极大地提升处理器的时钟速度。1995

年，英特尔的奔腾 Pro（Pentium Pro）处理器引入了微操作解码（micro-op decoding）这个概念。之前一个特定指令就是一个指令时钟周期，通过翻译这个指令为多个简单微指令，每个微指令就是一个指令时钟周期。

除了微操作，奔腾 Pro 处理器还引入了指令的无序执行和内存的虚拟化（通过 32 位总线对 36 位内存进行寻址）来对处理器进行优化。但这些对开发者来说是完全抽象化的，允许相同的应用运行在不同厂商出品的不同代的处理器上。这种保持指令级别兼容性的方式依然使用在当前的 x86_64 处理器中。

1.2 理解云计算和 OpenStack

本书主要关注如何通过使用 OpenStack 来部署企业私有云。同时，我会把私有云描绘成其所在的组织内部拥有和管理的基础设施（虚拟机、存储等）资源池，也就是我们所说的基础设施即服务（Infrastructure as a Service, IaaS）。相反，公有云 IaaS 资源由第三方服务提供商拥有和运营，如亚马逊的 AWS、微软的 Azure 等。本书的目标是帮你把公有云服务的简单和高效带到你的企业。

经济考量：私有云与公有云

从财务角度看，我们可以把私有云看成主要资本支出，公有云则是运营支出。这个区别很容易理解，在私有云部署时，你的组织要购买基础设施或者把在其他地方正常工作的基础设施挪过来使用。而在公有云中，成本直接与资源使用挂钩（使用才付费，不使用不付费），当然还有网络通信开销。

组织选择私有云还是公有云通常取决于使用资源的规模和组织内部 IT 部门的职责范围。企业 IT 部门的职责如果是集中为其他部门提供技术架构和资源供给，部署私有云更有利。一个多租户（数据、配置和用户管理在逻辑上是租户间隔离的）、完全实现编排的私有云可以让企业 IT 部门成为私有云代理。

多租户和完全编排

多租户指的是在云平台中以部门级别管理计算资源的能力。例如，一个市场营销部门可以分配一定比例的共享资源（X 台虚拟机、Y 容量的存储等），这个部门可以随意使用这些资源而不影响中心组织（回顾一下亚马逊购买模型）。同样，完全编排描述了应用程序依赖资源的分配能力。例如，一个会计应用程序和它的 Web 服务、数据库服务等依赖可以被程序化部署在一个环境里。因此，市场营销部门不只能在一个特定租户内管理它的资源，还能通过平台编排服务来部署基础设施（虚拟机）和应用程序（MySQL、Apache2、定制的应用程序等）。

相比之下，如果是部门中的 IT 单位，通常缺乏数据中心基础设施和部署高性价比的私有云

的人员。由于他们相对小的资源需求，部门中的 IT 单位通常可以利用公有云资源，或者利用他们企业的 IT 部门提供的私有云资源。

如果基于工作负载同时使用私有云和公有云，那么这种结合使用称为混合云。公有云和私有云都是使用相同的技术来构建，不过虽然构建组件可能一样，但使用私有云和公有云的动机可能完全不同。例如，用户经常因为安全合规性原因而使用私有云。通常来说，下列这些工作负载会使用公有云：本质上是周期性的工作负载，或者需要一个对企业来说非常昂贵的全局规模的工作负载。

虽然本书主要介绍如何使用 OpenStack 构建私有云，但也有很多内容是介绍如何基于 OpenStack 的 API 来直接转换成公有云提供商的服务。

抽象和 OpenStack API

从根本上讲，OpenStack 抽象和提供了一个通用的 API 接口来控制不同厂商提供的硬件和软件资源。这个框架提供了两个很重要的内容。

- 硬件和软件的抽象，这样避免了所有特定组件的厂商锁定问题。这是通过使用 OpenStack 管理资源来实现的，而不是直接通过厂商组件。这样做的缺点是除了通用的必要功能外，并不是所有的厂商功能都被 OpenStack 支持。
- 一个通用的 API 管理所有资源，允许连接各个组件进行完全编排服务。

第一点从财务角度来看是非常好的，第二点则是现代 IT 变革的关键。

这有什么内幕呢

OpenStack 提供了可伸缩和被抽象的底层硬件的各种功能的支持。OpenStack（或者其他云框架）不能做到的是主动顺应你当前的技术实践。为了充分利用云计算的能力，你必须对当前的业务和架构实践进行相应的转变。

如果你的架构标准是基于使用厂商提供的适当功能来对数据中心所有服务器实现某些功能，这样会与对厂商抽象的云部署冲突。如果你的业务实践只是按用户需求创建虚拟机，那你就没有抓住云自助服务的本质。如果最终用户的请求可以被高效自动化执行，或者用户可以自我供给资源，那你就是充分利用了云计算的能力。

1.3 节将把 OpenStack 与其他相应技术（也许是你熟悉的）关联起来。

1.3 关联 OpenStack 及其控制的计算资源

前面介绍了 OpenStack 能带来的好处，但它是如何工作的呢？也许，理解 OpenStack 是怎样工作的最简单的方式是把这个框架与企业环境内的常见技术关联起来。

在本节中，读者将会了解 OpenStack 是如何与它控制的基础资源（计算、存储、网络等）关联起来的。如你所见，OpenStack 通常不提供实际意义上的资源，它只是简单控制这些低层次的

资源。图 1-4 展示了 OpenStack 是如何管理资源的提供者的，它们轮流被虚拟机使用。

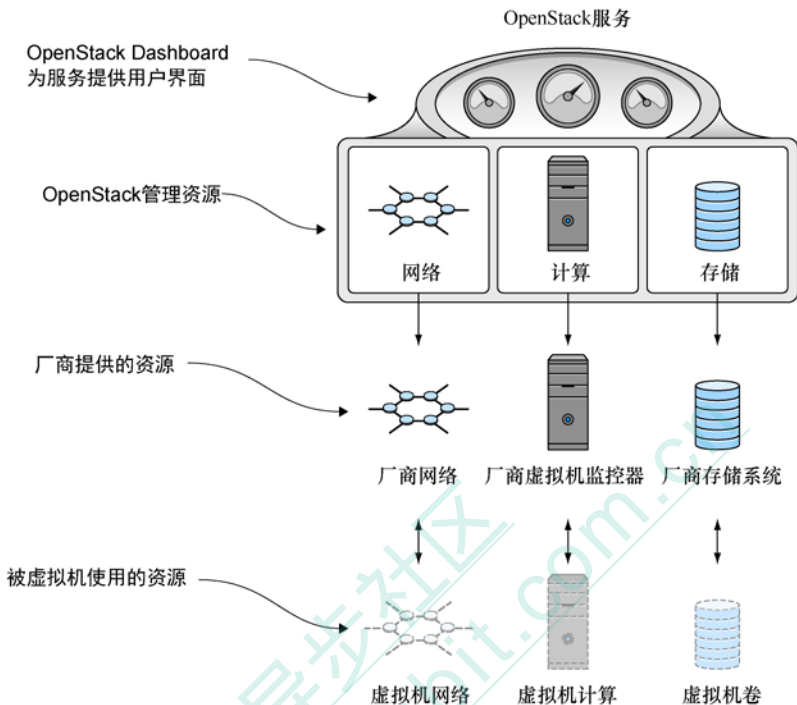


图 1-4 OpenStack 资源管理模型

在接下来的小节里，读者将会看到关联特定资源组件的详情：服务器虚拟化，通过对 hypervisor（虚拟机管理器）的控制；网络，通过对厂商提供的硬件和 OpenStack 服务的控制；块和对象存储，通过对厂商和 OpenStack 服务的控制。最后，我们会看到 Openstack 各个服务和常见的云术语的关联。如你所见，OpenStack 是一个协调资源和服务的框架，而不关心有哪家底层技术厂商。

1.3.1 OpenStack 和 hypervisor

hypervisor 或者虚拟机监控器（Virtual Machine Monitor, VMM）是一种为虚拟机进行物理硬件仿真的管理软件。OpenStack 不是一个 hypervisor，但它确实控制着 hypervisor 的操作。OpenStack 框架支持多种 hypervisor，包括 XenServer/XCP、KVM、QEMU、LXC、ESXi、Hyper-V、BareMetal 和其他（可通过下列网址查看 hypervisor 的支持列表：<https://wiki.openstack.org/wiki/HypervisorSupportMatrix>）。读者可能对 VMware ESX、VMware ESXi 和 Microsoft Hyper-V 比较熟悉，因为这些是当前企业虚拟化市场主流的 hypervisor。因为许可限制、成本和其他因素，OpenStack 社区对这些商业 hypervisor 的支持要少于开源的 hypervisor。

图 1-5 展示了 OpenStack 如何管理物理硬件上被 hypervisor 虚拟化的资源。在一个 OpenStack 集群内，OpenStack 协调多个 hypervisor 资源和虚拟机的管理。

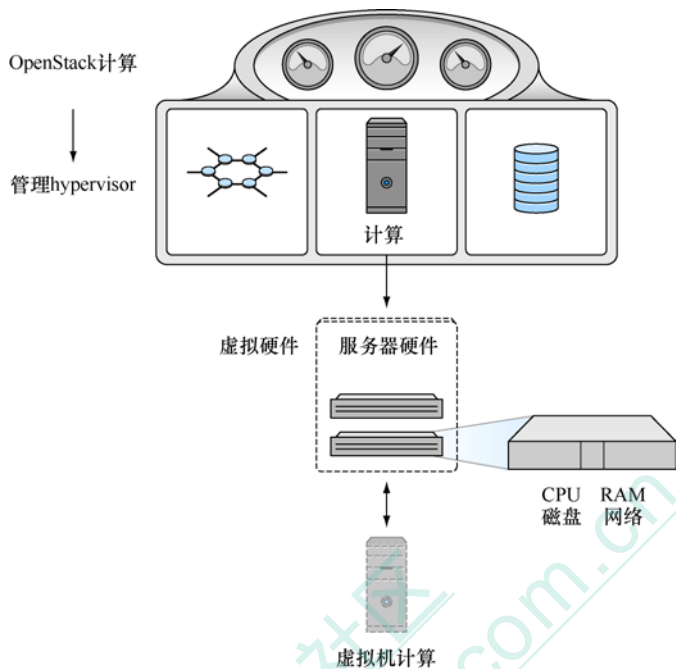


图 1-5 OpenStack 管理着 hypervisor

无论部署规模多大，大多数的个人和组织采用的 hypervisor 是 XenServer 或者 KVM，它们也是支持最多功能的 hypervisor。XenServer 是思杰（Citrix）公司的产品，从严格意义上来说，它是开源的 hypervisor，但商业支持通过思杰公司提供。KVM 已经是 Linux 内核的一部分，因此，很多 Linux 发行版的维护者提供 KVM 的商业支持，包括红帽（Red Hat）、Ubuntu、SUSE 等。

你通过认证了吗 随着大量提供商开始设计基于 OpenStack 框架的公有 IaaS 服务，他们很快意识到自己的客户可能需要微软对运行在 Windows 主机上的 hypervisor 进行认证。当时，思杰公司的 XenServer 已经满足了认证条件，并通过了微软的认证过程。但是，尽管思杰公司有一个以 CloudStack 形式竞争的平台，很多组织还是使用了 XenServer 作为他们的 OpenStack hypervisor。自从很多 Linux 发行厂商通过了微软的认证以后，现在可以完全支持 Windows 运行在 KVM hypervisor 上，包括那些被 OpenStack 控制的 hypervisor 上。

本书将采用基于内核的虚拟机（Kernel-based Virtual Machine，KVM）作为 hypervisor。自 2007 年发布的 Linux 2.6.20 开始，KVM 被并进 Linux 内核，完全被 OpenStack 支持。KVM 还提供了半虚拟化，但需要操作系统原生支持，或者通过在虚拟操作系统镜像添加 hypervisor 特定驱动来进行支持。使用开源的 hypervisor 的传统问题是部署和维护它的学习曲线陡峭，经常需要拥有系统、网络和应用管理经验。幸运的是，在组织内部提供集中化支持的虚拟化资源，资源申请必须通过组织的网络、系统、安全和财务供给流程。通常用户有以下 3 种选择。

- 使用社区代码，自给自足——社区维护的软件使用社区的支持，自己负责部署的设计、开发和运维。
- 使用社区代码，商业支持——社区维护软件使用厂商支持，你和厂商或者只是厂商负责部署。
- 使用社区项目的厂商分支，商业支持——使用厂商提供的软件和支持，你通常只需要负责部署关联的运维和厂商管理。

虽然很多厂商提供 OpenStack 和 KVM 的商业支持，但很多为工作负载构建的内部云不需要商业支持或者认证，因此，用 OpenStack 支持没有购买商业支持的 KVM 也是很流行的做法。无论你怎么部署和采用哪种支持方式，本书提供的材料都一样有用。

Linux 容器 最近，一些人对操作系统级别的虚拟化应用产生了浓厚的兴趣，而不是 OpenStack 实例提供的基础设施级别的虚拟化。操作系统级别的虚拟化可以在单一服务器上运行多个相互隔离的操作系统实例（容器）。但它不是 hypervisor 技术——它运行在系统级别，所有容器共享相同的内核。你可以把容器想象成在需要的地方提供虚拟的隔离，而没有全虚拟化的模拟开销。

目前最流行的两个操作系统级别的虚拟化项目是 Docker (<https://www.docker.com/>) 和 Rocket (<https://github.com/coreos/rkt>)。虽然容器是否比基础设施级别实例更适用于应用程序运行时传递还存在争议，但毫无疑问的是，基于容器的技术将会在构建云时广泛采用。

1.3.2 OpenStack 和网络服务

OpenStack 不是一个虚拟交换机，但它确实管理多个物理、虚拟的网络设备和虚拟覆盖网络（overlay network）。不像 OpenStack 控制虚拟机控制器那样受限于 hypervisor 提供的服务，OpenStack 直接提供网络服务，如 DHCP、路由等。但与 hypervisor 管理类似，OpenStack 对底层厂商技术透明，可以是商业或者开源的技术。

更重要的是，后端技术的改变，如从一种网络/厂商切换到另一种网络/厂商，并不需要客户端配置进行改动。对于涉及网络的大量专有的硬件、软件 and 用户接口，经常从一个厂商或者技术转换到另一个并非易事。通过 OpenStack，这些接口都被 OpenStack API 抽象化了，如图 1-6 所示。

OpenStack 可以管理多种类型的网络技术（实现机制），包括由 Arista Networks、Cisco Nexus、Linux bridging 和 Open vSwitch（OVS）等提供的技术。在

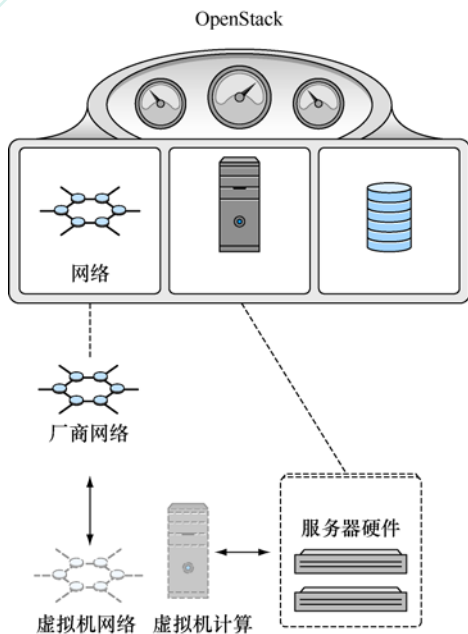


图 1-6 OpenStack 管理网络

本书中，我们将使用 OpenStack 和 OVS 提供的网络服务。OVS 是 OpenStack 部署中常被选择的一种，用户可以简单地在自己的环境里获得和复制，不需要特定硬件环境。除了网络实现机制，还有很多被 OpenStack 支持的网络类型（VLAN 和各种隧道技术等），这些内容将会在第 6 章中详细介绍。

1.3.3 OpenStack 和存储

OpenStack 不是一个存储阵列，至少应该不是你通常认为的存储那种形式。OpenStack 没有从物理上提供被虚拟机使用的存储。

如果你曾经使用过文件共享（NFS 和 CIFS 等），就会用过“基于文件”的存储。这种存储的类型很容易被人使用和被计算机访问，但它通常是另外一种存储类型的抽象：块存储。你可以认为操作系统或者文件系统是块存储的主要用户。

还有另外一种系统管理员可能不熟悉的存储类型：基于对象的存储。这种类型的存储通常是通过软件 API（如 GET /obj=xxx）接口进行访问。基于对象的存储是文件或块存储的更高层面的抽象，但没有后两者的限制。基于对象的存储可以很容易地在多个参与节点之间进行分布和复制。不像块存储那样需要被虚拟机快速访问，分布式的对象存储允许更大的延迟，将不能用作虚拟机的卷（volume，挂载到一个实例上的块设备）。通常做法是在创建时就指明使用对象存储来存放卷和镜像（包含操作系统）的备份。

下面首先介绍 OpenStack 是如何管理块存储的，然后介绍对象存储的相关内容。

1. 块存储

OpenStack 现在没有为最终用户管理基于文件的存储。由图 1-7 可以看出，OpenStack 管理块（虚拟机）存储与管理 hypervisor 和网络类似。

图 1-7 从基础虚拟机资源管理展望的角度展示了其全貌。OpenStack 可以管理很多厂商提供的存储解决方案，包括来自 Ceph、戴尔（Dell）、EMC、惠普（HP）、IBM 和 NetApp 等厂商的方案。与 hypervisor 和网络组件一样，OpenStack 提供灵活切换存储厂商和技术的能力，并且不需要改变客户端的配置。

2. 对象存储

虽然 OpenStack 不是一个用于块存储（用来启动虚拟机）的存储阵列，但它天生拥有提供对象存储的能力。与在物理硬件上运行 Linux 的支持版本不同，OpenStack 提供分布式对象存储集群时并不需要其他软件。这种存储类型可以用来存放卷备份，也通常用来存放大量可以被分割成二进制对象的数据。图 1-8 展示了一个基本的对象服务器部署，当然这些都包含在 OpenStack 环境中。

对象存储不是必须在同一地点。事实上，节点（代理节点和存储节点）可以在多个不同的地点，互为冗余。

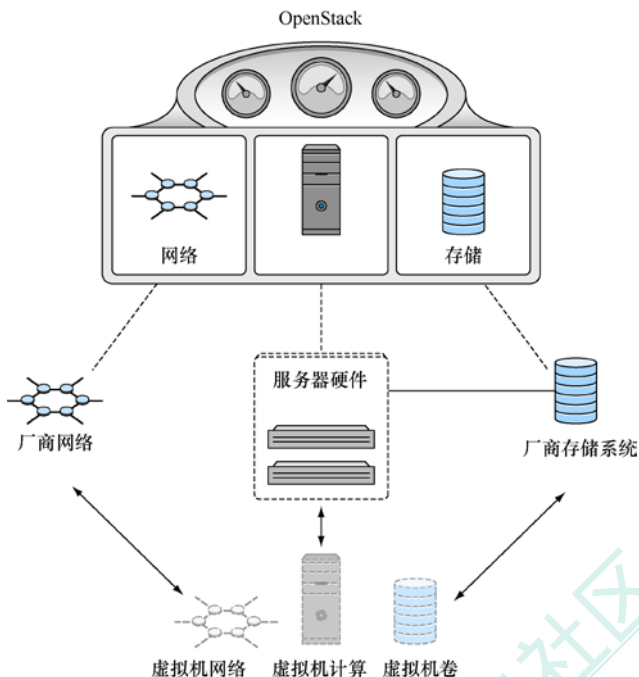


图 1-7 OpenStack 管理块（虚拟机）存储

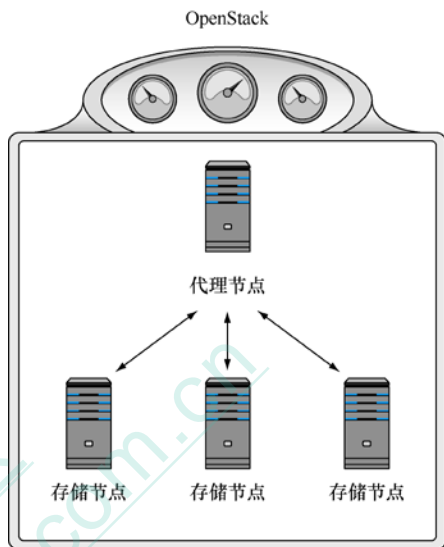


图 1-8 OpenStack 提供基于对象的（API）存储

对象存储传统的用法是存储那些被应用访问的数据，如被用户的应用程序使用的一个文档或文件。在 OpenStack 环境中，对象存储有几种用法。例如，使用对象存储作为虚拟机镜像的仓库。这样并不是说虚拟机直接使用了这些存储，它们只是通过这个存储系统维护的数据被提供出来。这样做是合理的，因为这个提供过程不需要对随机数据的低延时访问。对象存储还会用来备份一个现有的虚拟机的快照，用于长期保存备份。

1.3.4 OpenStack 和云专业术语

OpenStack 是一个用来构建云的框架，可以构建公有云和私有云。除了公有云和私有云的定义，还有“即服务”的云定义。OpenStack 是什么即服务呢？OpenStack 是多个即服务云的基础。

假如你对为自己的企业提供一个类似于 AWS 供应虚拟机和存储资源感兴趣，那么 OpenStack 可以认为是基础设施即服务（Infrastructure as a Service, IaaS）。在这种场景下，用户具有提供个人的直接访问的虚拟机，并由用户直接管理。虽然构成云的物理组件对用户是隐藏的，但是可以直接访问它们。OpenStack 的职责是控制为最终用户提供基础设施的资源。

现在假设你的云用户不能对基础设施直接访问，用户只能访问由 OpenStack 提供和支持的应用编排功能。在这种场景下，OpenStack 可以认为是平台即服务（Platform as a Service, PaaS）的后端提供者。底层的物理和虚拟基础设施组件对用户是隐藏的。想象一下这样的场景，一个开发

团队需要一个独立的应用环境（应用层部署在 IaaS 上）来进行软件测试。通过云编排，OpenStack 可以用来作为部署测试平台的后端提供者。

现在假设你的公司通过使用 OpenStack 提供的基础设施即服务（IaaS）或平台即服务（PaaS）为客户提供一种服务。在这种场景下，OpenStack 服务作为软件即服务（Software as a Service, SaaS）的后端组件。你可以看到，OpenStack 可以用作云计算多个层面的基础组件。

现在你对 OpenStack 可以做什么和如何做有了更深的理解，是时候介绍 OpenStack 各个组件是如何工作的了。1.4 节将会介绍 OpenStack 各个独立组件和它们在整个框架中的作用。

1.4 OpenStack 组件介绍

1.1 节介绍了 OpenStack 基本的功能，本节我们将会分析组成 OpenStack 框架的基本组件。

表 1-1 列举了多个 OpenStack 组件或核心项目。虽然还有更多现处在不同开发阶段的项目，但表 1-1 中所列的是 OpenStack 的基本组件。最新的 OpenStack 服务路线图可以在 OpenStack 路线图网页（www.openstack.org/software/roadmap/）中找到。

表 1-1 核心项目

项 目	代码名称	描 述
计算（Compute）	Nova	管理虚拟机资源，包括 CPU、内存、磁盘和网络接口
网络（Networking）	Neutron	提供虚拟机网络接口资源，包括 IP 寻址、路由和软件定义网络（SDN）
对象存储（Object Storage）	Swift	提供可通过 RESTful API 访问的对象存储
块存储（Block Storage）	Cinder	为虚拟机提供块（传统磁盘）存储
身份认证服务（Identity）	Keystone	为 OpenStack 组件提供基于角色的访问控制（RBAC），提供授权服务
镜像服务（Image Service）	Glance	管理虚拟机磁盘镜像，为虚拟机和快照（备份）服务提供镜像
仪表盘（Dashboard）	Horizon	为 OpenStack 提供基于 Web 的图形界面
计量服务（Telemetry）	Ceilometer	集中为 OpenStack 各个组件收集计量和监控数据
编排服务（Orchestration）	Heat	为 OpenStack 环境提供基于模板的云应用编排服务

现在你应该了解了 OpenStack 及其作用，让我们快速回顾一下它的发展历史。

1.5 OpenStack 发展历史

2009 年，美国总统奥巴马在上任的第一天就签署了针对所有联邦机构的备忘录，引导他们打破横亘在联邦政府和联邦政府服务的人民之间的有关透明度、参与度、合作方面的屏障。这份备忘录就是开放政府令。

该法令签署 120 天后，美国宇航局（NASA）宣布它的开放政府框架，其中包括 Nebula 工具

的共享。开发 Nebula 是为了加快向美国宇航局科学家和研究者提供 IaaS 资源的速度。与此同时，云计算提供商 Rackspace 宣布开源它的对象存储平台——Swift。

2010 年 7 月，Rackspace 和美国宇航局携手其他 25 家公司启动了 OpenStack 项目。在过去 5 年中，已经产生了 12 个发行版本。OpenStack 发行版本见表 1-2。

表 1-2 OpenStack 发行版本

名称	日期	核 心 组 件
Austin	2010 年 10 月	Nova、Swift
Bexar	2011 年 2 月	Nova、Glance、Swift
Cactus	2011 年 4 月	Nova、Glance、Swift
Diablo	2011 年 9 月	Nova、Glance、Swift
Essex	2012 年 4 月	Nova、Glance、Swift、Horizon、Keystone
Folsom	2012 年 9 月	Nova、Glance、Swift、Horizon、Keystone、Quantum、Cinder
Grizzly	2013 年 4 月	Nova、Glance、Swift、Horizon、Keystone、Quantum、Cinder
Havana	2013 年 10 月	Nova、Glance、Swift、Horizon、Keystone、Neutron、Cinder、Ceilometer、Heat
Icehouse	2014 年 4 月	Nova、Glance、Swift、Horizon、Keystone、Neutron、Cinder、Ceilometer、Heat、Trove
Juno	2014 年 10 月	Nova、Glance、Swift、Horizon、Keystone、Neutron、Cinder、Ceilometer、Heat、Trove、Sahara
Kilo	2015 年 4 月	Nova、Glance、Swift、Horizon、Keystone、Neutron、Cinder、Ceilometer、Heat、Trove、Sahara、Ironic
Liberty	2015 年 10 月	Nova、Glance、Swift、Horizon、Keystone、Neutron、Cinder、Ceilometer、Heat、Marconi、Trove、Sahara、Ironic、Searchlight、Designate、Zaqar、DBaaS、Barbican、Manila

OpenStack 现在保持 6 个月发行一个新版本的周期，与 OpenStack 峰会举办周期一致。该项目的参与公司已经从过去的 25 家发展为现在的超过 200 家，超过 130 个国家或地区的数千名用户参与其中。

1.6 小结

- 基础设施即服务（IaaS）云是通用资源的集合，可以通过管理框架协调。
- OpenStack 是一个管理框架，为最终用户的基础设施服务（IaaS）和应用编排（PaaS/SaaS）提供自助服务协调。
- OpenStack 控制现有的商业和社区技术，如 Hypervisor、存储系统、网络硬件和软件。
- OpenStack 是由多个具有特定目的的项目组成的。
- OpenStack 每个项目都有一个相关联的代码名称。