



高技能人才培训丛书 | 丛书主编 李长虹

云计算系统运维

张燕林 刘立华 编著

- 任务引领训练模式
- 来自企业岗位的真实工作任务
- 目标、任务、准备、行动、评价五步训练法
- 分析问题、解决问题、效果评价完整的工作过程



中国电力出版社
CHINA ELECTRIC POWER PRESS



高技能人才培训丛书 | 丛书主编 李长虹

云计算系统运维

张燕林 刘立华 编著

内 容 提 要

本书是一本以云计算的系统运营、技术维护和实训模拟为重点的教材。本书采用任务引领训练模式编写，以工作过程为导向，以岗位技能要求为依据，以典型工作任务为载体，训练任务来源于企业真实的工作岗位。

全书共由 24 个训练任务构成，从任务 1 云计算与 OpenStack 架构，到任务 24 公有云主机日常运维。24 个训练任务能力目标基于从业人员的职业能力，通过系统学习这 24 个训练任务并达到其能力目标要求，学习者可以完全具备实施云计算系统运维的技术能力。

本书由浅入深、通俗易懂、注重应用，示范操作步骤翔实且图文并茂，既可以作为云计算系统运维技术人员的参考书，也可以作为 IT 部门云计算技术人员或职业院校教师的教材，以及从事高技能职业教育与职业培训课程开发相关人员的参考书。

图书在版编目 (CIP) 数据

云计算系统运维 / 张燕林, 刘立华编著. —北京: 中国电力出版社, 2018. 8

(高技能人才培训丛书)

ISBN 978-7-5198-2138-8

I. ①云… II. ①张…②刘… III. ①云计算—岗位培训—教材 IV. ①TP393.027

中国版本图书馆 CIP 数据核字 (2018) 第 132830 号

出版发行: 中国电力出版社

地 址: 北京市东城区北京站西街 19 号 (邮政编码 100005)

网 址: <http://www.cepp.sgcc.com.cn>

责任编辑: 杨 扬 (010-63412524)

责任校对: 王海南

装帧设计: 赵姗姗

责任印制: 杨晓东

印 刷: 三河市航远印刷有限公司

版 次: 2018 年 8 月第一版

印 次: 2018 年 8 月北京第一次印刷

开 本: 787 毫米 × 1092 毫米 16 开本

印 张: 20.5

字 数: 548 千字

印 数: 0001—2000 册

定 价: 69.00 元

版权专有 侵权必究

本书如有印装质量问题, 我社发行部负责退换

主任 李长虹

副主任 俞益飞 张自忠

顾问 马阳 岳庆来

委员 高勇 刘振 闵光培 李泽明

曾芳丽 刘振鹏 邹少明 邓松

阮友德 张饶丹 王鑫 张迎辉

陈福宝 魏继东 张燕林 刘立华

国务院《中国制造 2025》提出“坚持把人才作为建设制造强国的根本，建立健全科学合理的选人、用人、育人机制，加快培养制造业发展急需的专业技术人才、经营管理人才、技能人才。营造大众创业、万众创新的氛围，建设一支素质优良、结构合理的制造业人才队伍，走人才引领的发展道路”。随着我国新型工业化、信息化同步推进，高技能人才在加快产业优化升级，推动技术创新和科技成果转化发挥了不可替代的重要作用。经济新常态下，高技能人才应掌握现代技术工艺和操作技能，具备创新能力，成为技能智能兼备的复合型人才。

《高技能人才培养丛书》由嵌入式系统设计应用、PLC 控制系统设计应用、智能楼宇技术应用、云计算系统运维、工业机器人设计应用等近 20 个课程组成。丛书课程的开发，借鉴了当今国外发达国家先进的职业培训理念，坚持以工作过程为导向，以岗位技能要求为依据，以典型工作任务为载体，训练任务来源于企业真实的工作岗位。在高技能人才技能培养的课程模式方面，可谓是一种创新、高效、先进的课程，易理解、易学习、易掌握。丛书的作者大多来自企业，具有丰富的一线岗位工作经验和实际操作技能。本套丛书既可供一线从业人员提升技能使用，也可作为企业员工培训或职业院校的教材，还可作为从事职业教育与职业培训课程开发人员的参考书。

当今，职业培训的理念、技术、方法等不断发展，新技术、新技能、新经验不断涌现。这套丛书的成果具有一定的阶段性，不可能一劳永逸，要在今后的实践中不断丰富和完善。互联网技术的不断创新与大数据时代的来临，为高技能人才培养带来了前所未有的发展机遇，希望有更多的课程专家、职业院校老师和企业一线的技术人员，参与研究基于“互联网+”的高技能人才培养模式和课程体系，提高职业技能培训的针对性和有效性，更好地为高技能人才培养提供专业化的服务。



全国政协委员
深圳市设计与艺术联盟主席
深圳市设计联合会会长

《高技能人才培养丛书》由近 20 个课程组成，涵盖了云计算系统运维、嵌入式系统设计应用、PLC 控制系统设计应用、智能楼宇技术应用、工业控制网络设计应用、三维电气工程设计应用、产品造型设计应用、产品结构设计应用、工业机器人设计应用等职业技术领域和岗位。

《高技能人才培养丛书》采用典型的任务引领训练课程，是一种科学、先进的职业培训课程模式，具有一定的创新性，主要特点如下：

先进性。任务引领训练课程是借鉴国内外职业培训的先进理念，基于“任务引领一体化训练模式”开发编写的。从职业岗位的工作任务入手，设计训练任务（课程），采用专业理论和专业技能一体化训练考核，体现训练过程与生产过程零距离，技能等级与职业能力零距离。

有效性。训练任务来源于企业岗位的真实工作任务，大大提高了操作技能训练的有效性与针对性。同时，每个训练任务具有相对独立性的特征，可满足学员个性能力需求和提升的实际需要，降低了培训成本，提高了培训效益；每个训练任务具有明确的判断结果，可通过任务完成结果进行能力的客观评价。

科学性。训练实施采用目标、任务、准备、行动、评价五步训练法，涵盖从任务（问题）来源到分析问题、解决问题、效果评价的完整学习活动，尤其是多元评价主体可实现对学习效果的立体、综合、客观评价。

本课程的另外一个特色是训练任务（课程）具有二次开发性，且开发成本低，只需要根据企业岗位工作任务的变化补充新的训练任务，从而“高技能人才任务引领训练课程”确保训练任务与企业岗位要求一致。

“高技能人才任务引领训练课程”已在深圳高技能人才公共训练基地、深圳市的职业院校及多家企业使用了五年之久，取得了良好的效果，得到了使用部门的肯定。

“高技能人才任务引领训练课程”是由企业、行业、职业院校的专家、教师和工程技术人员共同开发编写的。可作为高等院校、行业企业和社会培训机构高技能人才培养的教材或参考用书。但由于现代科学技术高速发展，编写时间仓促等原因，难免有错漏之处，恳请广大读者及专业人士指正。

编委会主任 李长虹

前 言

中国互联网用户众多，信息终端普及率很高，企业及消费者对于 IT 技术的了解和接受速度也非常快，再加上政府大力支持云计算产业，这就使得中国在云计算部署规模、技术以及商业模式创新方面迅猛发展。

工信部在 2017 年 4 月发布《云计算发展三年行动计划（2017—2019）》，提出到 2019 年，中国云计算产业规模达到 4300 亿元，突破一批核心关键技术，涌现两三家在全球云计算市场中具有较大份额的领军企业。

云计算技术的兴起，提供了一种适应于企业信息化发展需要的解决方法。云计算的出现为信息技术领域和企业信息化建设带来了新的挑战和机遇。然而，真正系统、深入、全面地阐述云计算系统运维技术及应用的书籍寥寥无几。本书作为一本全面、系统、深入论述云计算系统运维、技术和架构的云计算专著，正好弥补了这一空白。

本书重点介绍了云计算的系统运营与技术维护，各章节结构做了精心的设计和安排，有较强的逻辑性、系统性、全面性、专业性和场景实用性，并尽可能照顾到不同层次、不同专业层面的技术人员的学习水平。从内容上看，本书有以下特点：

(1) 全新的教材编写框架。本书完全打破传统教材的章节框架结构，基于“任务引领型一体化训练及评价模式”，训练任务全部来源于企业真实的工作任务，经过提炼，转化为训练任务。

(2) 以企业岗位要求为能力目标。训练任务的能力目标，以云计算系统运维职业岗位从业人员的职业能力为基准。

(3) 训练与评价可实现一体化。训练任务具有独立性、完整性，目标明确且可实现、可考评，能够满足个性化的能力提升要求。

此外，在训练任务实施部分中，示范操作步骤详实且图文并茂，每一步操作都有操作结果的效果状态图，力求做到学习者在没有老师指导的情况下，也能够完成示范操作的内容，因此本书也非常适合自学。

本书既可以作为开发人员的参考书，也可以作为企业员工培训或职业院校学生的教材，以及从事职业教育与职业培训课程开发相关人员的参考书。

本书在云联盟、中国智慧城市研究院与深圳中科慕课指导下，由张燕林、刘立华主持编写，全书由李长虹审定统稿。

云计算系统运维技术是一个比较新的领域，并且由于时间仓促，书中难免有疏漏和不当之处，敬请读者批评指正。同时也希望本书的读者在了解到云计算技术的同时，都能够积极的投身到云计算产业实践中来。只有更多的人认识到云计算的价值，通过系统运维才能挖掘出更多云计算的价值，云计算产业才会有源源不断的动力来蓬勃发展，相信读者中的很多人都将成为云计算产业的中坚力量。

目 录

序	
丛书序	
前言	
任务 1	云计算与 OpenStack 架构 1
任务 2	Linux 操作系统安装及基础优化 12
任务 3	OpenStack 组件和认证服务部署 29
任务 4	OpenStack 镜像和计算服务部署 42
任务 5	OpenStack 网络和 Dashboard 服务部署 56
任务 6	OpenStack 块存储和对象存储服务安装 69
任务 7	OpenStack RDO 自动化部署 86
任务 8	OpenStack 认证服务详解 92
任务 9	虚拟机模板制作 103
任务 10	OpenStack 镜像、卷、实例类型创建 115
任务 11	云计算网络基础 125
任务 12	OpenStack 网络服务 Neutron 实现 146
任务 13	基于 OpenStack 的 FLAT、VLAN 网络配置 156
任务 14	OpenStack 存储服务 Cinder 应用 169
任务 15	openfiler 外置存储部署 178
任务 16	配置 NFS 为 Cinder 后端存储 187

任务 17	云计算虚拟化之 KVM	198
任务 18	KVM 管理工具之 libvirt	207
任务 19	KVM 管理工具之 virsh	218
任务 20	公有云服务选型与规划	227
任务 21	公有云服务安全选型与规划	236
任务 22	公有云实例管理	248
任务 23	公有云弹性基础设施规划与配置	269
任务 24	公有云云主机日常运维	291
附录	训练任务评分标准表	310



任务 1

云计算与OpenStack架构

该训练任务建议用 6 个学时完成学习。

1.1 任务来源

云计算技术已经成为当今 IT 界非常成熟的技术，面对市场上种类繁多的云计算产品，作为初学者如何选择一款主流的云计算产品显得非常重要。OpenStack 就是众多云计算产品中的佼佼者，选择它的理由是产品成熟稳定、开放源代码和众多大公司参与代码贡献。

1.2 任务描述

通过对典型云计算产品 OpenStack 的架构进行剖析，宏观掌握 IT 系统架构、云计算相关知识及其应用。

1.3 能力目标

1.3.1 技能目标

完成本训练任务后，读者应当能（够）掌握以下技能。

1. 关键技能

- (1) 掌握 IT 系统底层的基本架构原理。
- (2) 掌握 OpenStack 云产品的架构并进行剖析。
- (3) 掌握 OpenStack 云产品在 IT 系统中所扮演的角色。

2. 基本技能

- (1) 能够对不同云的特征属性进行比较。
- (2) 会 IaaS、PaaS、SaaS 云平台的基本搭建步骤。
- (3) 能够撰写云计算产品架构的简要剖析报告。

1.3.2 知识目标

完成本训练任务后，读者应当能（够）学会以下知识。

- (1) 理解云计算技术能够解决的基本问题。
- (2) 熟悉 OpenStack 云产品的架构原理。

(3) 掌握云计算的基本概念。

1.3.3 职业素质目标

完成本训练任务后，读者应当能（够）具备以下素质。

- (1) 具有守时、诚信、敬业精神。
- (2) 具有安全意识、质量意识、保密意识。
- (3) 遵守系统调试标准规范，养成严谨科学的学习态度。
- (4) 养成总结训练过程和结果的习惯，为再次实训总结经验。
- (5) 培养喜爱云计算运维管理工作的心态。

1.4 任务实施

1.4.1 活动一 知识准备

- (1) 云计算技术的发展趋势。
- (2) OpenStack 的发展趋势。

1.4.2 活动二 示范操作

1. 活动内容

- (1) 云计算概览。
- (2) OpenStack 产品架构剖析。

2. 操作步骤

(1) 步骤一：IT 系统架构发展的三个阶段。从计算机发明到今天，信息技术伴随时代的发展也变得越来越强大，从信息技术的诞生到现在，信息技术系统架构经历了如下三个阶段。

1) 纯物理设备的发展阶段。这一阶段，应用部署和运行在各自独立的物理机上。比如一套企业的 ERP 系统，传统的部署方式是找 3 台物理机，分别部署 Web 服务器、应用服务器和数据库服务器。如果规模大一点，各种服务器采用集群架构，且每个集群成员分别部署在各自的物理机上，然后通过负载均衡技术对各自服务进行调度访问。

2) 面向资源的虚拟化计算发展阶段。随着物理服务器计算能力的增强，虚拟化技术的发展大大提高了物理服务器的资源使用率。这个阶段，一台物理机上运行若干虚拟机，应用系统直接部署到虚拟机上。这样部署的好处主要体现在减少了需要管理的物理机数量，同时节省了维护成本。

3) 面向服务的云计算发展阶段。随着虚拟化技术的应用，IT 环境中虚拟机也变得越来越，这时新的需求产生了，如何对 IT 环境中的虚拟机进行统一和高效的管理。由此，云计算技术应运而生。计算（CPU/内存）、存储和网络三类 IT 资源通过云计算平台实现统一管理。当用户需要虚拟机的时候，只需要向平台提供虚拟机的规格即可，平台会快速从三个资源池分配相应的资源，部署一个满足规格的虚拟机来承接这个服务。这时，虚拟机的管理者不再需要关心虚拟机运行在哪台物理机之上，存储空间从哪里来，IP 是如何分配的。

(2) 步骤二：云计算概述。

1) 云计算的定义。对云计算（cloud computing）一般的理解是基于互联网的相关服务的增加、使用和交付模式，通常涉及通过互联网来提供动态易扩展且经常是虚拟化的资源。公认相对比较权威的是美国国家标准与技术研究院（NIST）的定义：云计算是一种按使用量付费的模式，

这种模式提供可用的、便捷的、按需的网络访问，进入可配置的计算资源共享池（资源包括网络、服务器、存储、应用软件、服务），这些资源能够被快速提供，只需投入很少的管理工作，或服务供应商进行很少的交互。

2) 云计算的特点。云计算是通过使计算分布在大量的分布式计算机上，而非本地计算机或远程服务器中，企业数据中心的运行将与互联网更相似。这使得企业能够将资源切换到需要的应用上，根据需求访问计算机和存储系统。

被普遍接受的云计算特点如下。

a) 超大规模。“云”具有相当的规模，公有云的提供商每个云计算中心的装机容量多达几万台。截至 2016 年的有关报告显示，Google 云计算已经拥有 100 多万台服务器，而对亚马逊 AWS 拥有服务器的预测值在 200 万台左右，像内蒙古的大数据云计算中心服务器装机能力达 70 万台，投入使用 30 万台。企业私有云一般也拥有数百甚至上千台服务器。

b) 虚拟化。云计算支持用户在任意位置、使用各种终端获取应用服务。所请求的资源来自“云”，而不是固定的有形的实体。应用在“云”中某处运行，但实际上用户无需了解、也不用担心应用运行的具体位置。只需要一台笔记本或者一部手机，就可以通过网络服务来实现需要的一切，甚至包括超级计算这样的任务。

c) 高可靠性。“云”使用了数据多副本容错、计算节点同构可互换等措施来保障服务的高可靠性，使用云计算比使用本地计算机更可靠。

d) 通用性。云计算不针对特定的应用，在“云”的支撑下可以构造出千变万化的应用，同一个“云”可以同时支撑不同的应用运行。

e) 高可扩展性。“云”的规模可以动态伸缩，满足应用和用户规模增长的需要。

f) 按需服务。“云”是一个庞大的资源池，用户可以按需购买，同时可以像自来水、电、煤气那样尽量标准化地计费。

g) 费用低廉。由于“云”的特殊容错措施可以采用极其廉价的节点来构成云，“云”的自动化集中式管理使大量企业无需负担日益高昂的数据中心管理成本，“云”的通用性使资源的利用率较之传统系统大幅提升，因此用户可以充分享受“云”的低成本优势，经常只要花费几百美元、几天时间就能完成以前需要数万美元、数月时间才能完成的任务。

h) 潜在的危险性。云计算服务除了提供计算服务外，还必然提供了存储服务。但是云计算服务当前大多由私人机构（企业）经营，对于政府机构、特定商业机构（如像银行这样持有敏感数据的商业机构），应谨慎选择云计算服务。虽然云计算中的数据对于数据所有者以外的其他云计算用户是保密的，但是对于提供云计算的商业机构而言有时是毫无秘密可言的。

3) 云计算的三种模式。

a) 基础设施即服务（Infrastructure as a Service, IaaS）：这个是指系统、网络、存储等这些基础设施，通过用户的选择来购买不同相应的服务，就是基础设施服务。

b) 平台即服务（Platform as a Service, PaaS）：这个是指云服务供应商提供一个开发平台给用户，包括开发工具、系统、Web 服务等给用户去开发。

c) 软件即服务（Software as a Service, SaaS）：这个是指供应商会提供各种应用程序和软件给用户，这些软件由供应商去负责安装、管理和运营，用户可以直接登陆云系统去使用它们。

4) 云计算的四种部署模型。

a) 公有云。平时我们在使用的百度搜索引擎、还有像 51CTO 博客，大家注册后都可以去写博客，用里面的资源；还有百度网盘，开放给公众使用，上传各种资料，这些就是公有云。说到百度网盘，如果真是每个用户都分配给 2T 空间，那是不现实的，那云供应商就会因成本太高而

垮掉，其实它们仅仅是做了链接，当用户上传某些相同文件的时候，如果它探测到该文件在库里已经存在，就不会再上传这个文件，而是建立一个软链接。所以有的时候，用户上传某些文件到百度网盘，只需要几秒钟就上传上去了，实际上这个文件，它们的库里面早有人存放了。

b) 私有云。在企业中，假设一个公司有三十位开发人员，每位开发人员分配给他们一台测试机，那是不可能的，而且他们也不会完全用足一台机器的资源，这样会造成资源浪费；可以找三台性能配置比较高的计算机，用虚拟化技术给每台电脑装十台虚拟机，然后给他们每人用一台，这样就可以节省资源，但是只能供给企业内部用，不对外开放，这就是私有云。又或者有一个很大的数据资源，是公司的共享资源，但每个部门需要共享的内容不一样，可以通过给他们不同的权限到每一个不同的云角落找相应的数据，这也是私有云。

c) 混合云。就是既有公有云，又有私有云，公有云和私有云的混合体，这就是混合云。有的大公司私有云建设好却又有资源闲置情况，就可对外提供云服务。

d) 社区云。指由几家企业或几个团队组成的一个社区云，只能由他们这些圈子内的人去登录使用，而对其他人不开放，但是利用的资源也是由他们这些企业或者团队去分摊消费。

(3) 步骤三：OpenStack 产品架构剖析。OpenStack 已经走过了 8 个年头。每半年会发布一个版本，版本以字母顺序命名。现在已经到第 15 个版本 Ocata (字母 O)。OpenStack 最初只有两个模块，现在已经有 20 余个，每个模块作为独立的子项目开发。OpenStack 含七个核心项目：Identity Management (Keystone)、User Interface Dashboard (Horizon)、Compute (Nova)、Networking (Neutron)、Image Service (Glance)、Block Storage (Cinder)、Object Storage (Swift)。各个服务之间的关系如图 1-1 所示。

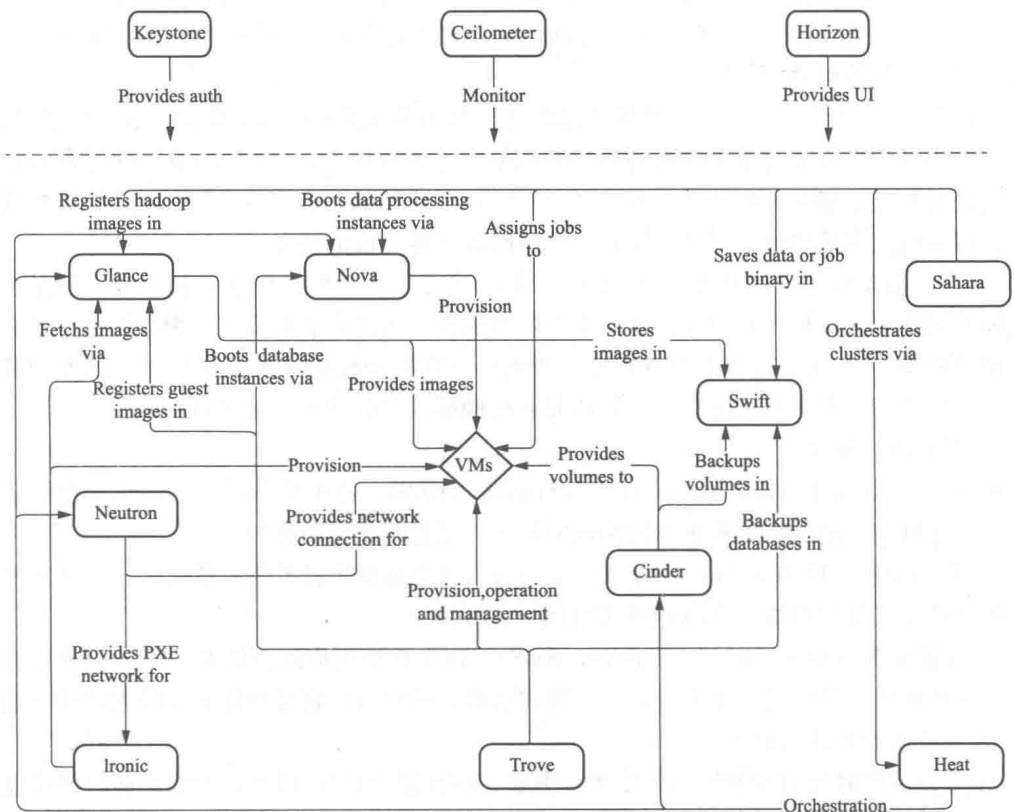


图 1-1 OpenStack 各服务之间的关系

1) 认证服务 Keystone。Keystone (OpenStack Identity Service) 行使 OpenStack 框架中负责身份验证、服务规则和服务令牌的功能, 实现了 OpenStack 的 Identity API。Keystone 类似一个服务总线, 也可以说是整个 OpenStack 框架的注册表, 其他服务通过 Keystone 来注册其服务的 Endpoint (服务访问的 URL), 任何服务之间相互的调用, 需要经过 Keystone 的身份验证来获得目标服务的 Endpoint 来找到目标服务。

Keystone 基本概念介绍如下:

a) User。User 即用户, 他们代表可以通过 Keystone 进行访问的人或程序。Users 通过认证信息 (credentials, 如密码、API Keys 等) 进行验证。

b) Tenant。Tenant 即租户, 它是各个服务中的一些可以访问的资源集合。例如, 在 Nova 中一个 Tenant 可以是一些机器, 在 Swift 和 Glance 中一个 Tenant 可以是一些镜像存储, 在 Neutron 中一个 Tenant 可以是一些网络资源。

c) Role。Role 即角色, Roles 代表一组用户可以访问的资源权限, 例如 Nova 中的虚拟机、Glance 中的镜像, Users 可以被添加到任意一个全局的或租户内的角色中。在全局的 Role 中, 用户的 Role 权限作用于所有的租户, 即可以对所有的租户执行 Role 规定的权限; 在租户内的 Role 中, 用户仅能在当前租户内执行 Role 规定的权限。

d) Service。Service 即服务, 如 Nova、Glance、Swift。根据前三个概念 (User、Tenant 和 Role) 一个服务可以确认当前用户是否具有访问其资源的权限。但是当用户尝试着访问其租户内的 service 时, 他必须知道这个 service 是否存在以及如何访问这个 service, 这里通常使用一些不同的名称表示不同的服务。在上文中谈到的 Role, 实际上也是可以绑定到某个 service 的。例如当 Swift 需要一个管理员权限的访问进行对象创建时, 对于相同的 Role 并不一定也需要对 Nova 进行管理员权限的访问。为了实现这个目标, 应该创建两个独立的管理员 Role, 一个绑定到 Swift, 另一个绑定到 Nova, 从而实现对 Swift 进行管理员权限访问不会影响到 Nova 或其他服务。

e) Endpoint。Endpoint, 翻译为“端点”, 可以理解它是一个服务暴露出来的访问点, 如果需要访问一个服务, 则必须知道它的 Endpoint。因此, 在 Keystone 中包含一个 Endpoint 模板 (Endpoint template, 在安装 Keystone 的时候可以在 conf 文件夹下看到这个文件), 这个模板提供了所有存在的服务 Endpoints 信息。一个 Endpoint Template 包含一个 URLs 列表, 列表中的每个 URL 都对应一个服务实例的访问地址, 并且具有 public、private 和 admin 这三种权限。public url 可以被全局访问 (如 <http://compute.example.com>), private url 只能被局域网访问 (如 <http://compute.example.local>), admin url 被从常规的访问中分离。

2) 用户接口仪表盘 (Horizon)。在整个 OpenStack 应用体系中, Horizon 就是整个应用的入口。提供了一个模块化的、基于 Web 的图形化界面服务门户。用户可以通过浏览器使用这个 WEB 图形化界面来访问、控制他们的计算、存储和网络资源。

3) 计算服务 (Nova)。Nova 提供虚拟机的全生命周期管理, 属于 OpenStack 核心组件之一。主要包括: API 服务器 (nova-api)、计算服务器 (nova-compute)、网络控制器 (nova-network)、调度器 (nova-schedule)、卷控制器 (nova-volume)、消息队列 (queue)、DashBoard。

API 服务器作为云控制器扮演着 Web 服务前端的角色。这个云框架的核心是 API 服务器。API 服务器命令和控制 Hypervisor, 存储还有网络, 让用户可以实现云计算。API 端点是一个基础的 HTTP 网页服务, 通过使用多种 API 接口来提供认证、授权、基础命令和控制功能, 增强了 API 和多种其他供应商已经存在的资源池的兼容性。

计算控制器 (Compute Controller) 提供了计算服务器资源。Compute 控制器控制运行在

宿主机上的计算实例，可以通过使用 API 的方式把命令分发到 Compute 控制器，进行以下的操作：运行实例、结束实例、暂停实例运行、重启实例、迁移实例、映射卷、分离卷、获得控制台输出、对象存储（Object Store）、授权管理器（Auth manager）、卷控制器（Volume controller）。

4) 网络服务（Neutron）。OpenStack 网络服务（Neutron）管理 OpenStack 环境中所有虚拟网络基础设施（VNI）以及物理网络基础设施（PNI）的接入层等服务，同时也提供更高等级的网络服务，例如防火墙功能、负责均衡服务等。网络服务提供网络、子网以及路由这些对象的抽象概念。每个抽象概念都有自己的功能，可以模拟对应的物理设备：网络包括子网，路由在不同的子网和网络间进行路由转发。OpenStack Networking（neutron）允许创建、插入接口设备，这些设备由其他的 OpenStack 服务管理。插件式的实现可以容纳不同的网络设备和软件，为 OpenStack 架构与部署提供了灵活性。

它包含下列组件：

- a) Neutron-Server：接收和路由 API 请求到合适的 OpenStack 网络插件，以达到预想的目的。
- b) OpenStack 网络插件和代理：插拔端口，创建网络和子网，以及提供 IP 地址，这些插件和代理依赖于供应商和技术而不同，OpenStack 网络基于插件和代理为 Cisco 虚拟和物理交换机、NEC OpenFlow 产品、Open vSwitch、Linux Bridging 以及 VMware NSX 产品穿线搭桥。
- c) 消息队列：用于在 Neutron-Server 和各种各样的代理进程间路由信息，也为某些特定的插件扮演数据库的角色，以存储网络状态。

OpenStack 网络主要和 OpenStack 计算交互，以提供网络连接到它的实例。

5) 镜像服务（Glance）。Glance 服务提供虚拟机镜像的发现、注册和获取等服务。Glance 提供 Restful API 可以查询虚拟机镜像的 metadata，并且可以获得镜像。通过 Glance，虚拟机镜像可以被存储到多种存储上，比如简单的文件存储或者对象存储。

OpenStack 镜像服务包括以下组件。

- a) glance-api：接收镜像 API 的调用，诸如镜像发现、恢复、存储。
- b) glance-registry：存储、处理和恢复镜像的元数据，元数据包括项诸如大小和类型。
- c) 数据库：存放镜像元数据。
- 6) 块存储服务（Cinder）。块存储服务（Cinder）为实例提供块存储，存储的分配和消耗是由块存储驱动器后端配置的驱动器决定的。

块存储服务通常包含下列组件：

- a) cinder-api：接受 API 请求，并将其路由到“cinder-volume”执行。
- b) cinder-volume：与块存储服务和调度例进程进行直接交互，它也可以与这些进程通过一个消息队列进行交互，该服务响应送到块存储服务的读写请求来维持状态，它也可以和多种存储提供者在驱动架构下进行交互。
- c) cinder-scheduler 守护进程：选择最优存储提供节点来创建卷。
- d) cinder-backup 守护进程：该服务提供任何种类备份卷到一个备份存储。
- e) 消息队列：在块存储的进程之间路由信息。

7) Object Storage（Swift）。Swift 使用普通的服务器来构建冗余的、可扩展的分布式对象存储集群，存储容量可达 PB 级。Swift 的是用 Python 开发，前身是 Rackspace Cloud Files 项目，随着 Rackspace 加入到 OpenStack 社区，Rackspace 也将 Cloud Files 的代码贡献给了社区，并逐渐形成现在 Swift。Swift 的基本架构如图 1-2 所示。

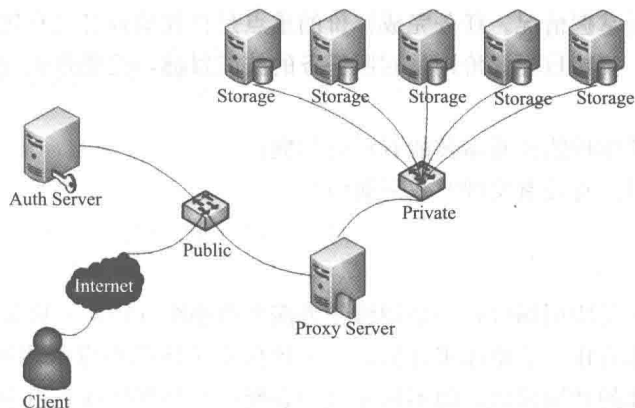


图 1-2 Swift 的基本架构

1.4.3 活动三 能力提升

查看 OpenStack 官网，了解 OpenStack 各组件之间的联系和 OpenStack 架构设计的原理。具体要求如下。

- (1) 掌握 OpenStack 架构设计原理。
- (2) 理解并掌握 OpenStack 七大核心组件各自的功能和原理。
- (3) 理解 OpenStack 七大核心组件之间是如何协调进行工作。

1.5 效果评价

在技能训练中，效果评价可分为学习者自我评价、小组评价和教师评价三种方式，三种评价的方式相互结合，共同构成一个完整的评价系统。训练任务不同，三种评价方式的使用也会有差异，但必须要突出以学习者自我评价为主体。

训练任务既可以作为培训使用，也可以用于考核评价使用，不论在哪种使用场合，对训练的效果进行评价都是非常必要的，但两种使用场合评价的目的略有不同。如果作为培训使用，则效果评价应以关键技能的掌握情况评价为主，任务完成情况的评价为辅，即重点对学习者的训练过程进行评价，详细考评技能点、操作过程的步骤等；如果作为考核使用，则效果评价应重点对任务的完成情况进行考核，如果任务未完成，则考核结果为不及格，在这种情形下需要考评者明确指出任务未完成的原因，如果需要给出一个对应的分数，则可以根据每一项关键技能目标的掌握情况，给予合理的配分。

1.5.1 成果点评

由教师组织，可以小组形式也可以全体学员一起，对学生学习的成果进行展示、点评。

- (1) 学生展示（演示）学习成果。
- (2) 教师对优秀成果进行点评。
- (3) 教师对共同存在的问题进行总结。

1.5.2 结果评价

1. 自我评价

自我评价是指由学习者对训练任务目标的掌握情况进行评价，评价的主要内容是训练任务的

完成情况和技能目标的掌握情况。任务完成评价的重点是自我检查有没有按照质量要求在规定的时间内完成训练任务，技能目标评价则是对照任务的技能目标，主要是关键技能目标，逐条检查实际掌握情况。

(1) 训练任务的关键技能及基本技能有没有掌握？

(2) 训练任务的目标有没有实现？效果如何？

评价情况：

2. 小组评价

小组评价有两种主要应用场合：一是训练任务需要由小组（团队）成员合作完成，此时需要将小组所有成员的工作看作一个整体来评价，个人评价所关注的重点可能不是小组的工作重点，小组评价更加注重整体的共同成就，而不是个人的表现；二是训练任务由学习者独立完成的，没有小组（团队）成员合作，在这种情况下，小组评价中参与评价的成员承担着第三方的角色，通过参与评价，也是一个学习和提高的过程。

在小组评价过程中，被评级人员通过分析、讲解、演示等活动，不仅可以展示学习效果，还可以全面提供综合能力。当然，从评价的具体结果来看，小组评价是对任务完成情况进行评价，主要包括任务完成质量、效率、工艺水平以及被评价者的方案设计、表达能力等，小组评价可以见表 1-1 的评价标准进行。

(1) 训练任务的关键技能及基本技能有没有掌握？

评价情况：

(2) 训练任务的目标有没有实现？效果如何？

评价情况：

参评人员：

3. 教师评价

教师评价是对学习者的训练过程、训练结果进行整体评估，并且在必要的时候考评学习者的设计方案、流程分析等内容，评价标准可以参考表 1-1。

(1) 训练任务的完成情况及完成质量。

(2) 训练过程中有没有违反安全操作规程，有没有造成设备损坏及人身伤害？

(3) 职业核心能力及职业规范。

评价情况：

参评教师：

表 1-1

结果评价

评价项目	评价内容	配分	完成情况	得分	合计	评价标准
安全操作	未按安全规范操作，出现设备及人身安全事故，则评价结果为 0 分					
能力目标	1. 符合质量要求的任务完成情况	50	是 <input type="checkbox"/> 否 <input type="checkbox"/>			若完成情况为“是”，则该项得满分，否则得 0 分
	2. 完成知识准备	5	是 <input type="checkbox"/> 否 <input type="checkbox"/>			
	3. 云计算能够解决什么问题	20	是 <input type="checkbox"/> 否 <input type="checkbox"/>			
	4. 讲述 OpenStack 架构原理	25	是 <input type="checkbox"/> 否 <input type="checkbox"/>			