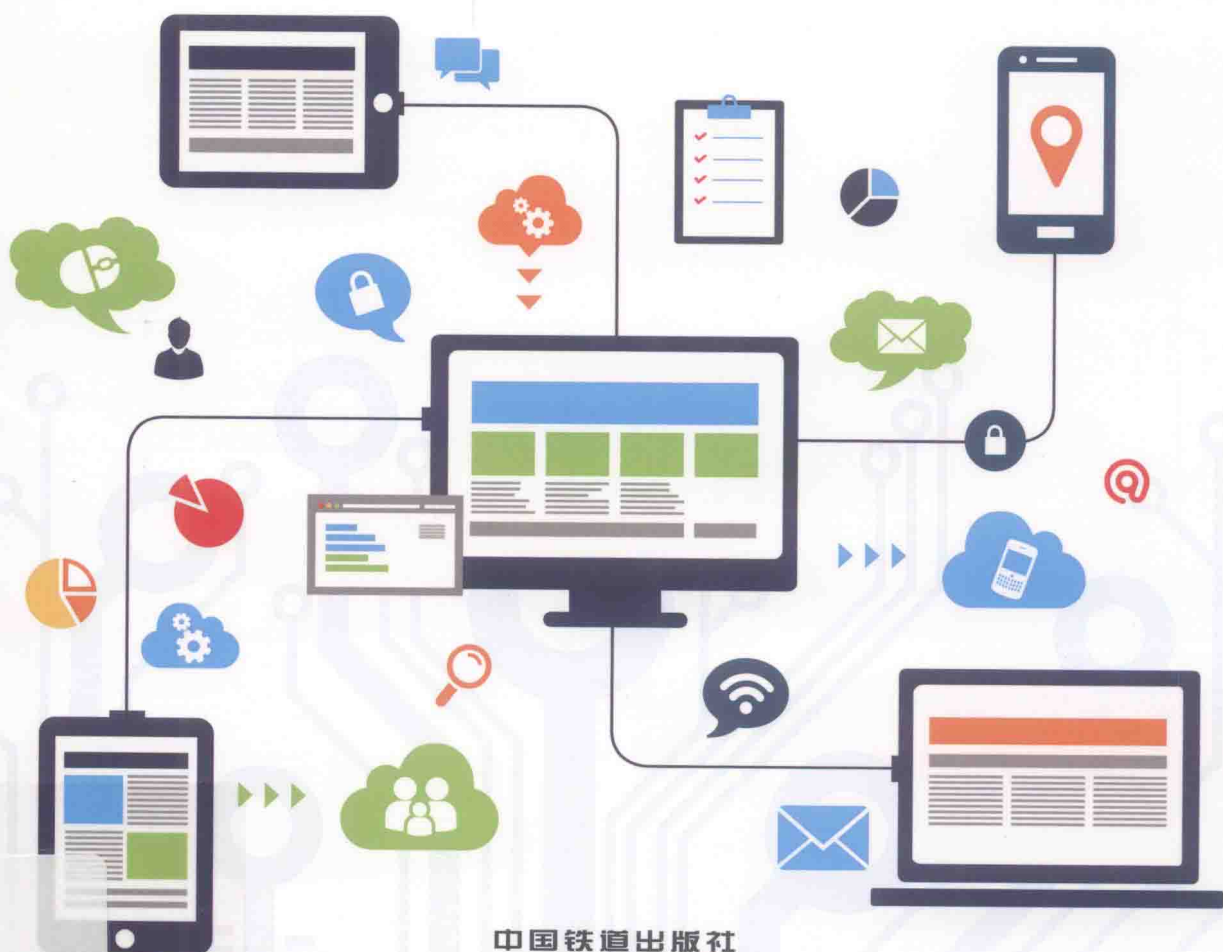


# OpenStack

## 架构分析与实践

管增辉 曾凡浪◎编著



中国铁道出版社  
CHINA RAILWAY PUBLISHING HOUSE

# OpenStack

## 架构分析与实践

管增辉 曾凡浪◎编著

## 内 容 简 介

本书以实战开发为原则,以关键模块架构分析及项目开发为主线,通过 OpenStack 开发中常用的 8 个典型组件和若干典型项目案例,详细介绍了云平台中的计算、网络、存储、服务编排、智能运维等模块,并针对 OpenStack 中通用的关键技术进行了详细介绍。对于每一部分内容的讲解,章节的最后都会配备相应的实战案例供大家参考。

本书内容全面丰富,案例典型、常见,实用性强,适合各个层次想要学习 OpenStack 开发技术的人员阅读,尤其适合有一定 OpenStack 基础而要进行 OpenStack 应用开发的人员阅读。本书内容通俗易懂,章节安排由浅入深,因此,也适合作为高校学生云计算的入门书籍。

### 图书在版编目 (CIP) 数据

OpenStack 架构分析与实践/管增辉,曾凡浪编著. —北京:  
中国铁道出版社, 2018. 12  
ISBN 978-7-113-24963-2

I. ①O… II. ①管… ②曾… III. ①计算机网络IV. ①TP393

中国版本图书馆 CIP 数据核字 (2018) 第 215395 号

书 名: OpenStack 架构分析与实践  
作 者: 管增辉 曾凡浪 编著

责任编辑: 荆 波  
责任印制: 赵星辰

读者服务热线: 010-63560056  
封面设计: **MX** DESIGN  
STUDIO

出版发行: 中国铁道出版社 (100054, 北京市西城区右安门西街 8 号)  
印 刷: 三河市兴博印务有限公司  
版 次: 2018 年 12 月第 1 版 2018 年 12 月第 1 次印刷  
开 本: 787mm×1092mm 1/16 印张: 24.5 字数: 575 千  
书 号: ISBN 978-7-113-24963-2  
定 价: 69.00 元



版权所有 侵权必究

凡购买铁道版图书,如有印制质量问题,请与本社读者服务部联系调换。电话:(010) 51873174

打击盗版举报电话:(010) 51873659

## 本书缘起

仿佛就在一夜之间，云计算火了，一跃成为业内很受关注的热点话题之一。如果读者关注互联网圈子的话，应该听说过早在 2011 年时，国内大大小小的公司就瞄准了云计算的百万级市场，也就是从那时起，云计算成了互联网上毫无疑问的又一大风口。

2018 年新年伊始，小米就与微软达成了云计算合作意向，进一步深化战略合作伙伴关系，以微软在云计算、人工智能等领域的先进技术实力，与小米在多样化的移动、智能设备及服务领域的市场优势相结合，共同致力于打造更加优质的产品和服务，并全力助推小米产品进军国际市场。

通信巨头华为更是把云计算作为 2018 年的重点投入领域，并着力发展其公有云和私有云业务。

AWS 推出的无服务器计算功能，从另一个方面助力于云计算的发展。无服务器计算，意味着开发人员不再需要担心设置或管理服务器。相反，他们可以将代码上传到云上运行。

万物互联的物联网（IOT）时代，更是要求云计算为基础设施层提供更强大的能力和更加丰富的功能。从移动互联网向 IoT 的迅速转变会对基础设施带来新一轮的挑战。这个挑战表现在几个层面：第一个是敏捷性，第二个是成本，第三个是复杂性。以上三个层面的问题可以通过云平台轻松得以解决。

诸如此类，我们这里就不一一列举了，但云计算的火爆程度，已然可见一斑。

OpenStack 作为云计算领域最大的开源项目，结构复杂、内容繁多，官网上给出的参考资料不足以让初学者达到快速入门的目的；目前图书市场上关于 OpenStack 的图书不少，但写作水平参差不齐，大部分书籍都是基于较低版本的 OpenStack 进行写作的，内容上略显陈旧，相较于 OpenStack 每年发布两个新版本的速度而言，不足以让读者全面细致地了解 OpenStack 的最新功能和它所采用的前沿技术。

除此之外，OpenStack 对其中的一些组件做了较大的变动，如果不是专业的开发及架构人员，很难全面把握应用与实践的方向。

考虑到以上因素，本书基于官方资料及个人相关从业经验，立足架构、着眼开发、兼顾实践，包含系统的 OpenStack 架构原理与云平台实践方法，全面提高读者的 OpenStack 实战能力。

## 本书特色

尽管云计算引入我国不过七八年时间，但在这短短的时间里，其相关的教学和科研成果层出不穷，与云计算相关的书籍可谓是“百花齐放，百家争鸣”，因此，读者在学习云计算

时，可供选择的参考资料更多了、可以研习的材料也更多了，在众多的选择中，却让人感到有些无所适从。

观察目前市面上的云计算书籍，大部分教材注重原理介绍和理论阐述，在很大程度上忽略了相关知识的应用与实现，作为典型的理工类书籍，脱离实战讲理论，对于初学者而言，难免枯燥与晦涩，对于具备一定经验的工程师而言，这样的书籍又显得毫无营养可言。

本书力求避其短而扬其长，以典型模块、典型案例为研究与讲解对象，将理论及原理介绍与实战相结合，巧妙地将大量实战经验与原理介绍融入到本书的每一个章节，最大限度地提高本书的含金量，以给读者奉上一顿云计算的营养大餐。

笔者一直本着重点突出、结合实战、求同存异的思想努力将本书写好。总体上而言，本书的主要特色有以下三点：

### 1. 深入浅出、通俗易懂

注意从实际生产中择取典型问题，针对关键模块本着“模块分析，架构先行”的理念进行重点分析，配备典型实战案例，引导读者一步一步学习，以期达到“学中做，做中学”的学习效果。

### 2. 问题典型、案例经典

始终本着“关键问题，重点分析”的宗旨，本书介绍的基本理论知识，都是使用非常贴近生活场景的实例来引导的，这样就避免了知识讲述过于抽象，非常易于理解。通过对典型问题的分析和经典案例的讲解，增加了本书的趣味性和实效性，以期达到“理论联系实际”的最佳效果。

### 3. 道法自然、内外兼修

笔者在过去的工作当中，累积了大量生产实践经验，可以帮助学有所长的读者快速进入到一个实际操作的场景中进一步提高自己的实操能力，从而使得学习之路变得不那么坎坷。

### 4. 基于 OpenStack 较新版本进行写作，技术实效性强

本书基于 Ocata 版本（个别章节会涉及 P 版本的内容），从技术新颖度与成熟度而言，都极具参考性。P 版本的 OpenStack 中新功能的添加并不是很多，更多的是针对 OpenStack 易用性的改进；而新版本 Q 版本中尚有许多不足之处，并且生产实践中部署比较少。

### 5. 描述清晰，讲解透彻

书中尽量避免使用大量的文字进行描述，能用图表来说明的地方尽量使用简单明了的图表表示，坚持“一图胜千言”的原则。

## 本书愿景

在信息化飞速发展的今天，由于大量时间的碎片化，学完的东西总感觉不是那么的系统，知识是学到了，但是结构比较零散，遇到实际问题的时候还是会有一种“知之但难用之”的感觉。

作为技术类的书籍，本书希望从浅显易懂的知识切入，贯穿实操，做到既能满足大家系

统掌握基础知识的欲望，也要能让读者在学习的过程中品尝到新鲜有营养的高规格“饕餮盛宴”。

“授之以鱼，不如授之以渔”，看别人写的书只是我们学习的一种途径，我们更希望读者在看完本书的内容后，在实际工作中可以“自行其书”、“不囿于书”，将本书中学到的知识应用到实际问题的解决中去。读其书会其书，是为法；学习其书用其书，是为道。这是我们对每一位读者的期望，更是本书的真诚愿景。

### 赠君云梯

万丈高楼平地起，层层都会设楼梯。本书考虑到不同层次的读者水平，在内容组织方面也是有所考虑，目的是让每一位“小白”都能一步一个台阶的登堂入室，踏实地迈向 OpenStack 高手的云中楼阁。

云中楼阁第一梯：成为圈中人。技术是开放的，开源的技术更是如此，谈开源但又不迈入开源，那么只能是门外汉；开篇向大家介绍了如何更好更快地加入到 OpenStack 的开源社区中。

云中楼阁第二梯：云中拾零。武侠中我们会经常看到这样一幕：一位武林高手使出“大招”后，可以瞬间定胜负。这个“大招”其实并非简单的一招一式，而是经过马步、梅花桩等小技巧的点滴修炼共集而成。OpenStack 就像是这个“大招”，它是一个庞大的系统，但是无论系统多么的庞大，它也总是由一个个小的“部分”组成，在这里我们称之为服务或组件。这里的“云中拾零”，拾的就是这些零零散散的组件，各个组件各个击破。

云中楼阁第三梯：云中舞剑。本书的学习过程中，我们会嵌入有许许多多的实践示例，意在让大家能够理论与实践相结合，在实践中领略个中乐趣的同时，还能将知识学扎实了。

云中楼阁最高梯：云中望月。我们期望读者在学完本书后，能在脑海中形成一幅相关 OpenStack 架构和实践的云天大图，在实践中游刃有余。

### 学习建议

书中个别章节相对独立，目的在于顾及不同读者需求的同时，但又不失系统化。第一章为入门章节，主要介绍如何“入门”社区。后续章节是分别对 OpenStack 中不同的模块进行讲解，理论讲解的同时还有实操。

- 如果您具有一定的开发经验又对开源有过切身经历，那么第一章可以略过，直接开始正文的学习。
- 如果你对 OpenStack 已经轻车熟路，只是想学习了解其中的某一个模块，那么您大可以单刀直入，直接进入到您感兴趣的章节。
- 如果您是一名初学者，建议您从第一章开始就仔细研读所有的知识点，这对后续的学习非常重要。

## 二维码下载包

- 书中源代码
- OpenStack Super User

## 本书读者

- IT 部门首席信息官 (CIO)
- 企业首席技术官 (CTO)
- 云计算基础设施建设者
- IT 主管
- IT 技术工程师
- 互联网公司员工
- 网络运维人员

除以上关键人员，教育机构的师生通过阅读本书，可以很好地建构自己的云计算知识体系。总之，每位读者都能从中获益，至少对云计算不再“云里雾里”了。

编者

2018 年 5 月



## 第 1 章 走进 OpenStack

1.1	OpenStack 是什么.....	1
1.1.1	OpenStack 的作用.....	1
1.1.2	OpenStack 的应用场景.....	2
1.1.3	什么类型的工作要学 OpenStack.....	3
1.2	为什么要学习 OpenStack.....	3
1.2.1	OpenStack 在云计算中的地位.....	3
1.2.2	云计算新时代：容器 vs 虚拟化.....	4
1.3	如何学习 OpenStack.....	4
1.3.1	对学习者的技术要求.....	4
1.3.2	OpenStack 的学习路线.....	4
1.4	OpenStack 的基本架构.....	5
1.5	OpenStack 的核心组件.....	7
1.5.1	计算资源管理：Nova 组件.....	7
1.5.2	存储资源管理：Cinder/Swift 组件.....	8
1.5.3	网络资源管理：Neutron 组件.....	9

## 第 2 章 OpenStack 部署与社区贡献流程

2.1	OpenStack 部署方式.....	12
2.1.1	DevStack 方式部署.....	13
2.1.2	手动部署分布式 OpenStack 环境.....	18
2.1.3	RDO 方式部署 OpenStack.....	22
2.2	为 OpenStack 社区作贡献.....	25
2.2.1	提交前的环境准备.....	26
2.2.2	代码贡献流程.....	28
	【示例 2-1】代码贡献流程之 bug Fix.....	28
2.2.3	文档贡献流程.....	30
	【示例 2-2】以 heat 为例来演示 HTML 的生成过程.....	31
2.2.4	其他内容的贡献流程.....	32
2.3	开发工具之 Pycharm.....	33



2.3.1 Pycharm 的安装与配置 .....	33
2.3.2 使用 Pycharm 对代码进行远程调试 .....	34
【示例 2-3】通过 Pycharm 调试 OpenStack 中 nova list 的代码 .....	34
2.3.3 Pycharm 与 PDB 的选用比较 .....	35
【示例 2-4】开发工具之 PDB 断点调试 .....	35

### 第 3 章 虚拟化

3.1 虚拟化技术的现状 .....	37
3.2 KVM 的管理工具 Libvirt .....	38
3.2.1 Libvirt 简介 .....	38
【示例 3-1】通过 Libvirt 提供的 API virsh 对虚拟机生命周期实现管理 .....	39
3.2.2 Libvirt 的体系结构 .....	40
3.3 OpenStack 与虚拟化的结合 .....	42
3.4 虚拟机配置 libvirt.xml 详解 .....	45

### 第 4 章 OpenStack 通用技术

4.1 RPC 服务实现分析 .....	49
【示例 4-1】在 OpenStack RPC 中创建 Server 并实现 Client 向 Server 发送请求 (以 rpc、calll 为例) .....	53
4.2 消息队列服务分析 .....	54
4.2.1 透彻理解中间件 RabbitMQ .....	54
【示例 4-2】通过“Hello World”演示如何 RabbitMQ 的消息收发过程 .....	55
4.2.2 RabbitMQ 实现 RPC 通信 .....	58
【示例 4-3】RabbitMQ 之 RPC 通信案例 .....	58
4.3 RESTful API 开发框架 .....	64
4.3.1 灵活但不易用：基于 Pastedeploy 和 Routes 的 API 框架 .....	65
【示例 4-4】通过 nova list 获取虚拟机的命令，根据 Nova 的 api-paste.ini 来说明 是如何路由的 .....	68
4.3.2 基于 Pecan 的 API 框架 .....	69
4.4 TaskFlow 的实现 .....	72
4.4.1 TaskFlow 常见使用场景 .....	72
4.4.2 TaskFlow 中必须理解的重要概念 .....	73
4.4.3 TaskFlow 具体实现 .....	74
【示例 4-5】TaskFlow 仔细看，重实践得体验 .....	74
【示例 4-6】TaskFlow 功能多，长流程特别火 .....	76
4.5 基于 Eventlet 的多线程技术 .....	78

4.5.1 进程、线程与协程 .....	78
4.5.2 Eventlet 依赖的两个库: greenlet 和 select.epoll .....	79
【示例 4-7】greenlet 库应用之协程切换 .....	79
4.5.3 创建协程的常用 API .....	80
4.5.4 定时和监听: Hub .....	81
4.5.5 Eventlet 中的并发机制 .....	83

## 第 5 章 Nova——计算组件

5.1 Nova 架构 .....	84
5.1.1 Nova 基本架构及服务组成 .....	85
5.1.2 Nova 内部服务间的通信机制 .....	86
5.1.3 Nova 内部服务间协同工作 .....	88
5.2 nova-api 服务 .....	89
5.2.1 nova-api 服务的作用 .....	89
5.2.2 nova-api 服务的启动流程 .....	91
5.3 nova-scheduler 服务 .....	95
5.3.1 基本原理及代码结构 .....	96
5.3.2 调度过程 .....	97
5.3.3 配置分析 .....	100
5.4 nova-compute 服务 .....	101
5.4.1 nova-compute 服务的作用 .....	101
5.4.2 nova-compute 服务的启动流程 .....	103
5.4.3 nova-compute 服务的日志分析 .....	105
5.5 周期性任务的实现 .....	106
5.5.1 什么是周期性任务 .....	107
5.5.2 周期性任务的代码 .....	108
5.6 资源及服务刷新机制 .....	111
5.6.1 服务上报机制 .....	111
5.6.2 主机资源刷新机制 .....	112
5.7 典型流程分析 .....	117
5.7.1 nova-scheduler 服务的启动流程 .....	117
5.7.2 虚拟机创建的流程 .....	120
5.8 案例实战——Nova 以 Ceph 作为后端存储 .....	122

## 第 6 章 Neutron——网络组件

6.1 Neutron 的发展历程 .....	126
-------------------------	-----

6.2	网络基础.....	127
6.2.1	网络的基本概念.....	127
6.2.2	常用的网络设备.....	131
6.2.3	虚拟网络技术.....	131
6.2.4	Neutron 网络的基本概念.....	133
6.3	Neutron 核心架构.....	135
6.3.1	Neutron 部署结构.....	135
6.3.2	Neutron 组成部件.....	136
6.3.3	ML2 Core Plugin.....	138
6.3.4	DHCP 服务.....	141
6.3.5	路由服务.....	142
6.3.6	元数据服务.....	144
6.3.8	Neutron 使用示例.....	147
6.4	高级服务 (Advanced Services).....	149
6.4.1	Load Balancer as a Service (LBaaS).....	149
6.4.2	Firewall as a Service (FWaaS).....	153
6.4.3	VPN as a Service (VPNaaS).....	155
6.5	典型网络模型分析.....	156
6.5.1	Linux Bridge + Flat/VLAN 网络模型.....	156
6.5.2	Open vSwitch + VxLAN 网络模型.....	161
6.5.3	小结.....	171

## 第 7 章 Heat——服务编排组件

7.1	Heat 架构分析.....	172
7.1.1	Heat 组件的基本架构.....	173
7.1.2	Heat 对资源的管理.....	175
7.1.3	认识 HOT 模板.....	177
7.1.4	小实例: 通过 HOT 模板创建虚拟机.....	180
7.2	Heat 中的锁机制.....	182
7.3	Heat 中的 Hook 机制.....	184
	【示例 7-1】在通过 Heat 进行资源定义时, 应该如何使用 Hook (钩子).....	185
	【示例 7-2】通过 Heat 创建一个 Stack, 在创建 Stack 时, 需要通过 Environment 来定义 Hook (钩子).....	186
7.4	案例实战——Heat 典型案例.....	189
7.4.1	通过 Heat 模板创建 Stack.....	189
7.4.2	Heat Stack 创建流程.....	195

## 第 8 章 Keystone——认证组件

8.1	Keystone 的架构 .....	198
8.1.1	Keystone 的作用 .....	199
8.1.2	Keystone 与其他组件间的关系 .....	201
8.1.3	基本架构解析 .....	203
8.1.4	自定义 Keystone Plugin .....	205
8.1.5	支持使用 External Plugin .....	206
8.2	Keystone 中的基本概念 .....	207
8.2.1	API V2 和 API V3 .....	207
8.2.2	其他常见概念 .....	208
8.2.3	多区域 multi-region .....	209
8.3	Keystone 的安装部署与基本操作 .....	211
8.3.1	Keystone 的安装部署 .....	211
8.3.2	Keystone 基本操作 .....	212
	【示例 8-1】使用 OpenStack user create 创建一个名为 test 的用户 .....	212
8.4	Keystone 的认证流程 .....	215
8.4.1	认证方式 .....	215
	【示例 8-2】以查看虚拟机列表为例，使用 X-Auth-Token 构造一个合法的 HTTP 请求 .....	215
8.4.2	令牌生成方式 .....	216
8.4.3	Keystone 工作流程 .....	220

## 第 9 章 Cinder——块存储组件

9.1	Cinder 架构分析 .....	222
9.2	Cinder 的安装 .....	225
9.2.1	安装与配置存储节点 .....	225
9.2.2	安装与配置控制节点 .....	227
9.2.3	安装与配置 Backup 服务 .....	231
9.2.4	安装正确性验证及 Cinder 基本操作 .....	232
9.2.5	Cinder 配置存储后端 .....	234
	【示例 9-1】LVM 作为 Cinder 的后端存储 .....	234
9.3	案例实战——通过 Heat 模板创建 Cinder Volume .....	235
9.4	Cinder API 服务启动过程分析 .....	238
9.4.1	cinder-api 代码目录结构 .....	239
9.4.2	cinder-api 服务启动流程 .....	240

9.4.3	REST 请求的路由 .....	242
9.5	案例实战——关键代码分析 .....	245
9.5.1	Volume 创建示例 .....	245
9.5.2	代码分析之 cinder-api 接收请求 .....	247
9.5.3	代码分析之 cinder-scheduler 进行资源调度 .....	249
9.5.4	代码分析之 cinder-volume 调用 Driver 创建 Volume .....	251

## 第 10 章 Ceilometer——数据采集组件

10.1	Ceilometer 架构分析 .....	254
10.1.1	Ceilometer 中的基本概念 .....	255
10.1.2	旧版 Ceilometer 架构 .....	256
10.1.3	新版 Ceilometer 架构 .....	258
10.2	数据处理 .....	260
10.2.1	Notification Agents 数据收集 .....	261
10.2.2	Polling Agents 数据收集 .....	262
10.2.3	数据转换与发布 .....	263
10.3	Pipelines .....	265
10.4	计量项 .....	267
10.5	Agent 和 Plugin .....	269
10.5.1	Polling Agents .....	270
10.5.2	Plugins .....	272
10.6	案例实战——Heat 与 Ceilometer 结合，搭建一个弹性伸缩系统 .....	274
10.6.1	系统介绍 .....	274
10.6.2	准备模板 .....	275
10.6.3	创建系统 .....	277

## 第 11 章 Glance——镜像组件

11.1	Glance 架构分析 .....	279
11.2	状态分析 .....	280
11.3	代码结构与概念分析 .....	281
11.3.1	Metadata 定义 .....	283
11.3.2	Domain 模型 .....	285
【示例 11-1】	自定义 Gateway 方法 .....	285
11.3.3	Task 定义 .....	287
11.4	Glance 的安装与配置 .....	287
11.4.1	Glance 安装部署 .....	288

11.4.2	Glance 基本配置 .....	292
	【示例 11-2】修改 Glance 后端存储为 RBD .....	292
11.5	镜像缓存 .....	293
11.6	案例实战——Glance 常见场景之镜像创建 .....	294

## 第 12 章 智能运维 Vitrage——RCA 组件

12.1	Vitrage 架构 .....	297
12.1.1	High Level 架构设计 .....	298
12.1.2	Low Level 架构设计 .....	300
12.2	Vitrage 安装部署 .....	301
12.2.1	手动方式安装部署 Vitrage .....	301
12.2.2	通过 DevStack 安装 Vitrage .....	303
12.3	Vitrage 模板 .....	304
12.3.1	Templates (模板) 的结构 .....	304
	【示例 12-1】Host 处于 ERROR 状态时, 触发告警的模板 .....	305
12.3.2	模板的加载过程 .....	306
12.3.3	添加自定义模板 .....	307
12.4	Vitrage Evaluator .....	310
12.5	自定义 Datasources .....	312
12.6	案例实战——Vitrage 中的告警解决方案 .....	314

## 第 13 章 OpenStack 其他组件及智能运维方案

13.1	Mistral—— workflow 组件 .....	317
13.1.1	Mistral 应用场景 .....	318
13.1.2	Mistral 中的重要概念 .....	318
13.1.3	Mistral 功能介绍 .....	320
13.1.4	Mistral 架构分析 .....	322
13.1.5	Mistral 实战应用 .....	322
	【示例 13-1】为 Mistral 添加用户自定义 Action .....	322
	【示例 13-2】通过 Mistral 获取虚拟机数据 .....	323
13.2	OpenStack 智能运维解决方案 .....	326
13.2.1	可视化的 Dynatrace .....	327
13.2.2	VirtTool Networks .....	327
13.2.3	智能运维 Vitrage .....	329

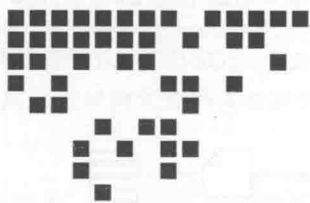
**第 14 章 OpenStack 应用实战：自动编排和配置高可用 Redis 系统**

14.1 利用 cloud-init 配置虚拟机 .....	332
14.1.1 cloud-init 的安装与配置 .....	333
14.1.2 cloud-init 对 VM 进行配置 .....	338
【示例 14-1】通过 cloud-init 配置虚拟机 .....	341
14.1.3 cloud-init 调试过程与问题分析 .....	343
14.2 Redis 数据库的 HA 实现及 Redis 集群的创建 .....	347
14.2.1 Redis HA 方案实现 .....	347
14.2.2 Redis Cluster 集群实现 .....	358

**第 15 章 OpenStack 架构与代码实践**

15.1 OpenStack 架构设计思路 .....	366
15.1.1 业务架构设计思路 .....	366
15.1.2 部署架构设计思路 .....	368
15.1.3 平台用户角色设计 .....	369
15.2 案例实战——向 Heat 中添加自定义资源 .....	370
15.2.1 实现原理及思路分析 .....	370
15.2.2 向 Heat 中添加 Zabbix 资源 .....	371
15.2.3 定义 Zabbix Action .....	373
15.2.4 实现 AutoScaling 模板 .....	375
15.2.5 资源查看 .....	377





## Chapter 1 第 1 章

# 走进 OpenStack

本书是以 OpenStack 为例，讲解云计算架构与实践的书。云计算是 21 世纪初才兴起的热门名词，当 Google 率先在 2006 年的搜索引擎大会上提出后，各大公司纷纷投向云计算开发的大潮。

云计算是一个比较大的概念，在发展之初，市面上就存在多种云计算产品，随着技术的发展，OpenStack 开始一枝独秀，占据了不可小觑的份额。本章从简单介绍 OpenStack 开始，循序渐进地带领读者通过基础架构、学习路线等方面全面了解 OpenStack。

通过对本章的学习，希望读者有以下几点收获：

- 掌握 OpenStack 的基础架构及关键组件；
- 理解 OpenStack 与云计算的关系；
- 理解 OpenStack 与容器化的关系；
- 了解如何学习 OpenStack。

## 1.1 OpenStack 是什么

本书主要讲解 OpenStack 的相关内容，可能有些读者完全没有接触过 OpenStack，只闻其名，不知其实。本节就对 OpenStack 做个简单的介绍，让读者对它有一个大概的认识。

### 1.1.1 OpenStack 的作用

OpenStack 是一个云计算平台，它能管理数据中心的大量计算、存储、网络资源，并且向用户提供一个管理资源的 Web 界面，同时提供功能相同的命令行和 RESTful API 接口。图 1.1 展示了 OpenStack 所提供的功能。

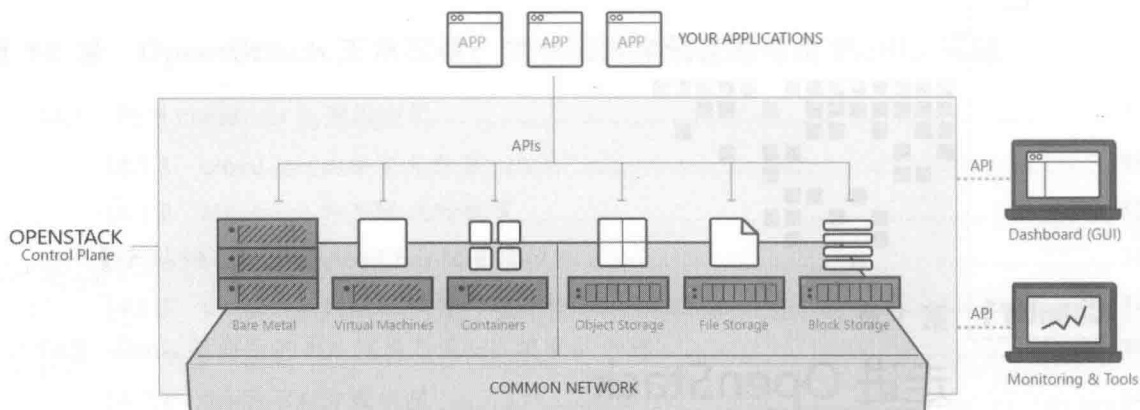


图 1.1 OpenStack 平台功能

OpenStack 位于基础设施之上，管理集群中的物理机、虚拟机、存储和网络等资源，应用程序运行在 OpenStack 之上，从 OpenStack 获取它所需要的资源，和传统运行在物理服务器上的方式一样。OpenStack 暴露一组管理资源的 API，它的 Web 界面 Dashboard 向用户提供图形操作界面，后端则是调用这些 API。同时也提供采集和分析集群中资源使用情况数据的工具。

OpenStack 是一个自由和开源的软件项目，它由美国的计算公司 Rackspace 和美国国家航空航天局 NASA 在 2010 年联合发布。在 2016 年，OpenStack 由 OpenStack 基金会接管。OpenStack 基金会是在 2012 年 9 月成立的一个非盈利性公司，旨在推动 OpenStack 软件和社区发展。如今全球有 600 多家企业加入了 OpenStack 基金会，AT&T、Ericsson、华为、Intel、Rackspace、Red Hat, Inc. 和 SUSE 是其中的白金会员。

### 1.1.2 OpenStack 的应用场景

OpenStack 由于兼有高成熟度和开源的特点，因此适用于多种环境。对于技术和资源尚不充裕的中小公司，可以很容易地使用原生的 OpenStack 部署出一套私有云环境，为线上应用提供基础设施平台。而对于技术资金雄厚的大型企业，还可以基于 OpenStack 进行二次开发，实现满足自己需求的功能，或者对外提供公有云服务和私有云解决方案。基于 OpenStack 衍生出的企业服务大致分为以下几类。

- 培训业务：针对 OpenStack 的培训课程，比如 Linux Foundation、Oracle University 等。
- 发行版和应用：继承 OpenStack 的发行版，方便企业快速部署私有云平台，比如 Mirantis 的 MCP、VMware 的 VIO、Cisco 的 Metacloud、EasyStack 的 ESCloud、联想的 ThinkCloud、华为的 FusionSphere 等。
- 公有云服务：Rackspace、Cloud&Heat，国内的 UnitedStack、网易云、金山云等。
- 私有云托管服务：Ubuntu 的 BootStack、IBM 的 Bluemix Private Cloud Local 等。
- 咨询和技术支持服务：Red Hat、Aptira、NEC、99Cloud 等。
- 厂商驱动：设备厂商为 OpenStack 提供自己的硬件支持，使用户能够将 OpenStack 运行在自己的设备上。这类涉及的主要是存储和网络设备厂商，如 Dell EMC、IBM、HPE、Hitachi、Scality、Cisco、Big Switch、Juniper、华为、H3C。

注意：OpenStack 的应用场景并非限于上述简单的几种，随着业务的不断发展，相信它可以为更多场景提供 IaaS、PaaS 和 SaaS 服务。