

云生态

Cloud Ecosystem

专刊

第1期 创刊号

Vol.01

观点&趋势

生态圈新闻

对话大咖

Docker专栏

技术热点

打造中国最优质的云生态媒体



InfoQ

云生态专刊

Cloud developer ecosystem

本刊由InfoQ中国制作出品

出品人：霍泰稳

总策划：崔康

本期主编：郭蕾

流程编辑：丁晓昀

读者反馈/投稿
editors@cn.infoq.com

商务合作
sales@cn.infoq.com

目录

创刊序.....	2
嘉宾寄语.....	3
热点回顾.....	4
对话大咖.....	7
探究AWS开发者生态最佳实践.....	8
技术热点.....	12
从一个OpenStack的失败案例看Ironic 和Neutron组件的现状.....	13
成功部署OpenStack的十个小技巧.....	15
Docker专栏.....	17
etcd：从应用场景到实现原理的全方位解读.....	18
观点&趋势.....	26
BaaS服务的定义、发展以及未来.....	27
谷歌关于容器的观点分享.....	30
生态圈新闻.....	33



创刊序

2015年是中国云计算产业的生态之年，“单打独斗”的模式已经没有出路，越来越多的云服务厂商认识到构建生态系统的重要性，从提供创业孵化环境到通过共享、共赢的理念打造一个自上而下的健康生态链，参与者都有“肉”吃，客户也更加满意，这样的场景令人向往。这也是国内IT产业成熟的一个标志。

中国的云计算市场开始进入加速发展的阶段，在未来5年，市场增长率预计都在30%以上，云厂商及相关合作伙伴将超过千家，云领域相关的IT从业人员将达数万人。公有云目前处于低总量、高增长的时期，IaaS规模在迅速扩大，竞争也最激烈；PaaS依然在培育期，发展前景看好；SaaS最成熟，盈利状况最好；私有云、混合云则处于闷声发财的阶段。

InfoQ有幸参与并见证了中国IT产业的高速发展。技术变革是推动产业发展的主要力量，我们尝试站在技术浪潮的前沿，让国内的架构师、开发者了解和借鉴整个技术社区的成果。《云生态专刊》是InfoQ为大家推出的一个新产品，目标是“打造中国最优质的云生态媒体”，包含的内容包括：

- 引入国外云计算领域的先进思想和技术；
- 报道国内外云厂商和生态圈的发展动态；
- 分享深度和前沿的热点技术；
- 关注产业发展的难点、痛点，分享解决方案；
- 挖掘产业发展的亮点，分享宝贵经验。

我们期望通过这种形式推动国内云计算产业的发展，让生态圈良性循环，也希望大家对《云生态专刊》的发展多提建议和批评，让我们走的更踏实一些。

感谢时代给我们机遇，我们只能奋勇向前，与大家携手进步。

——InfoQ中国总编辑 崔康

嘉宾寄语

作为一家杰出的技术媒体，这几年里InfoQ的成就有目共睹。2015年对于云产业是大好的一年。像七牛这样的国内云公司是否能重建传统IT时代由国外公司把控关键技术产品的IT格局，云生态的构建和成熟极为关键。产业共赢的到来只是一个早晚的问题，贵在我们所有云公司的努力和坚持，而InfoQ打造的《云生态专刊》将居功至伟。

——七牛创始人 许式伟

在移动网络、物联网、智能硬件应用、云服务、大数据应用快速普及的时代，云计算产业链也相应的走上了快车道，技术越来越丰富，分工越来越细，同时产业链上下游的用户、企业、服务者也遇到了谜题，如何高效利用纷繁复杂技术推动自己的业务发展或让自己享受更好的服务，《云生态专刊》正应运而生，希望专刊能够分享更多先进经验、优秀案例，介绍更多前沿和落地的技术与产品，助力生态良性共赢。

——环信创始人CEO 刘俊彦

云计算正在重新统一和颠覆传统IT基础设施和应用，UnitedStack的梦想是利用OpenStack为中国云计算市场带来新一代安全、可靠、高性能的基础设施云环境。InfoQ作为中国IT领域不可缺少且颇有影响的媒体，对推动云计算技术进步影响重大，祝愿《云生态专刊》杂志成为促进中国云生态系统构建路上的重要纽带，愿我们携手共进，用互联网、云计算、开源等技术和方法论去革新中国企业IT和数据中心。

——UnitedStack创始人 程辉

衷心祝贺InfoQ《云生态专刊》开刊，IT时代的未来是云、大数据、移动化和安全并驾齐驱的新型态，希望《云生态专刊》能够分享全球最领先的开源云计算理念、技术和商业模式，推动其在中国市场的落地，打造中国的云计算生态圈，进一步助力中国云计算产业发展。

——惠普云计算集团中国区总经理 张永健

热点回顾



云生态专刊

VMware CTO：数据存储功能或将向闪存转移

近日，VMware公司首席技术官发表了一篇博客文章，文章中透露他相信存储阵列和VSAN层的存储数据服务功能将向闪存介质转移，并认为这对于动态而不是静态的物理块地址是符合逻辑的。

存储数据服务向闪存介质的转移，其好处据称是加速服务器应用，能够将更多的应用加载到服务器和存储基础设施的简化中。例如，如果VMware VSAN可以卸载这种数据服务到闪存介质中，那么这将把更多服务器CPU资源留给运行虚拟机。

OpenShift 3功能开发完成进入测试阶段

红帽在2014年8月就宣布，第三代OpenShift平台将以Docker和Google Kubernetes为基础来进行开发。红帽OpenShift产品管理总监Joe Fernandes近期接受国外科技媒体的采访时透露，OpenShift 3已完成功能开发，目前已进入测试阶段，预计2015年2月将会推出第三代OpenShift正式测试版。

OpenShift目前提供3种产品，包含OpenShift Online、OpenShift Enterprise和OpenShift Origin，其中OpenShift Online是红帽的公共云PaaS平台，提供主机代管服务，开发者可用来开发、建立、部署应用程序。OpenShift Enterprise则是红帽提供企业可就地部署的私有云PaaS平台，而OpenShift Origin则是一个开源解决方案，它是OpenShift Online和OpenShift Enterprise的基础。

微软发布第二季度财报云服务与移动设备业务强劲增长

1月27日，微软发布了2015财年第二季度的财报，财报中显示，截止至2014年12月31日，微软毛利润已经达到163亿美元，运营收入为78亿美元。其中，企业级云服务收入增长114%，实现55亿美元的年收入，手机硬件业务收入达到23亿美元。

云端服务和硬件项目仍然是微软整体营收的发力点，不过微软现在面临的问题是增长预期有限，周期性的产品销量下降很有可能会在下季度出现，还有“一次性购买”的Office产品收入降低，微软真的需要施展浑身解数才能在接下来的几个季度中达到华尔街的期望。

谷歌为其云平台用户开放Docker私有仓库服务

日前，谷歌发布了针对其云平台的Google Container Registry，该服务支持开发者在企业云平台上存储、分享、管理他们的私有Docker仓库，它的功能与Quay.io类似，但Google的Container Registry主要是面向谷歌云平台的企业用户。

谷歌一直很亲睐Docker，开源的容器集群管理项目Kubernetes就是其多年大规模容器管理技术的开源版本。同样，云计算巨头AWS也很重视Docker，去年11月AWS就发布了高性能容器管理服务EC2 Container，开发者可以在其上使用任何的第三方Docker registry，但AWS目前没有提供registry服务。

Red Hat的OpenStack 野心超越Linux OS

Red Hat公司的重点已经开始从其流行的Linux操作系统向OpenStack转移，去年该公司的系列举措奠定了它在云计算生态中的位置，股票因此也上涨了25%。Red Hat的愿景是提供开放式混合云服务，基于开源软件构建其新业务。Red Hat的高级副总裁Tim Yeaton将OpenStack和Unix、Linux的发展做了类比，Unix只开源了其核心部分，而Linux的全部代码都是开源的，正是得益于开源Linux才借助社区的力量得到了更好的发展。

同样OpenStack也面临同样的问题，很多厂商都把OpenStack当成一个“开源的核心”，并在其基础上构建属于自己的中间件。Red Hat认为这样做会重蹈Unix覆辙，正确的思路应该是基于社区构建开放的云平台。

Linux基金会发布 《开放云技术概览》报告

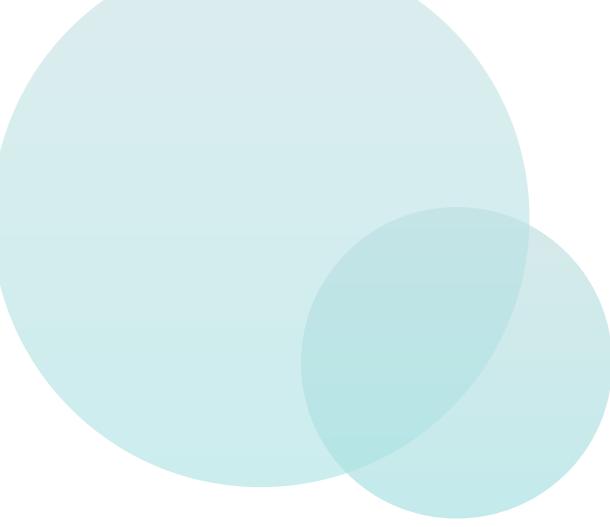
Linux基金会发布《开放云技术概览》报告，报告中汇总了目前所有与云计算相关的开源项目，包括Hypervisor与容器、云操作系统、IaaS、PaaS、服务提供和管理工具、存储、SDN等方面。报告整理了相关技术涉及的开源项目并详细列举了项目的背景、历史、主要开发者、开源协议、商业支持、开发语言、主要使用者信息。值得一读。不过报告也有小小的纰漏，比如容器部分的Docker的贡献者中应该没有Citrix，而报告中却有列举。

BaaS市场增长迅猛 2017年将达到77亿美元

BaaS (Backend as a Service) 是一种新型的云服务，旨在为移动和Web应用提供后端云服务，包括云端数据、文件存储、账户管理、消息推送、社交媒体整合等。BaaS是垂直领域的云服务，随着移动互联网的持续火热，BaaS也受到越来越多的开发者的青睐。它作为应用开发的新模型，可以降低开发者成本，让开发者只需专注于具体的开发工作。

分析机构Marketsand Markets称BaaS市场到2017年将会达到77亿美元，而2012年仅为2.165亿美元，年增长率达到了104%。而根据资料显示，这几年IaaS的增长率为50%左右。由于移动互联网的持续火热，所以BaaS受宠也是情理之中。如果用同样的思路来推测的话，几年之后是不是基于物联网的云服务也会很火热？

对话大咖



探究AWS开发者生态最佳实践

作者：包研

根据Gartner的[云基础设施服务魔力象限](#)显示，AWS在行业中遥遥领先，这与其成功的开发者生态建设不无关系。可能没有第二家云基础设施服务商像AWS这样重视开发者生态，这不仅因为AWS起步较早，更重要的是他们找到了一条与开发者互动的最佳实践，即由开发者驱动AWS业务发展的法则：开发者决定上线哪些服务。同时，由于AWS内部的工程师不断与开发者进行互动，有些创新是由AWS内部的工程师发起的。

开发者就像建筑师，在云上设计形形色色的服务，对于云基础设施服务商而言，开发者生态是否健康是其业务能否长远发展的关键。2012年开始，AWS在国内陆续举行免费的线下培训，尽管AWS截止到2014年12月仍处在有限预览阶段，开发者培训一直在继续。2014年12月12日，AWS Summit首次来到中国在北京举行，35个课程、3场动手实验课程吸引了数千名开发者，而这一切都是免费。如此大规模投入开发者的服务和教育背后的动机是什么？AWS有哪些与开发者互动最佳实践值得其他公司和团队借鉴？在AWS Summit大会当天，InfoQ带着这些问题专访了AWS全球开发者营销主管Adam FitzGerald，以下为与Adam对话内容。

问：在众多的开发者需求当中你们是如何优先选择服务的，你们如何判断哪些用户的需求要先满足的？

答：事实上，我觉得开发者可能会有自己很多的需求和不同的兴趣点，AWS能够在许多领域给开发者提供巨大的价值。而我们在选择的时候，在哪个领域可能会影响最大就选择哪个领域。对大部分的开发者而言，他们都愿意做一些比较新的尝试。比如说我们是不是能够提供一些工具，把一些重复性的准备工作，或者是在技术流程上重叠、重合以及繁琐的工作，通过自己的努力把它自动化。很显然这些领域就是我们的优先领域。有的时候AWS和亚马逊会先在内部寻求一些灵感，比如在AWS re: Invent宣布的编码部署服务，实际上最早就出自于内部的一个产品的应用。最后我们发现这个编码部署在整个的亚马逊的基础设施上取得了非常好的应用效果。像今天早晨在主题演讲中所说的，当我们发现一个产品在内部的应用是如此成功的话，我们就很容易作出推广的决策，因为我们发现客户面临的是同样的挑战，所以他们所面临的是同样的对产品的需求。

问：**AWS**新服务出现的后，怎么决定应该在哪些区域进行推广呢？

答：事实上这是一个非常复杂的决定。这取决于这个产品本身的类型是什么，目标客户是什么，以及这个产品本身的技术属性是什么，所以很大程度上我们产品的推广路线图基本上是由客户所驱动的。因为事实上我们所推出的这些产品有90%的功能是由客户的一些需求决定，所以客户一旦有需求对我们来讲就是非常重要的信息，我们会非常认真地对待这些信息，然后来决定究竟如何做。所以主要衡量产品服务上线的标准是两条，第一是客户的需求，第二就是本身所蕴含的科技。

一个产品推出的时候并不是每次都只在一个地区，有的时候可能是涵盖我们所服务的地区的一半的区域，甚至是有的产品是同时全球上市的，这并不是不可能。之所以有的产品只是在有些地区提供有限预览，这主要是看一下用户群的反应，如果这个地区的用户对这个产品和科技还处在早期酝酿的阶段，或者因为产品本身的特性的问题的话，我们就要斟酌看下一步如何做了。

我们在全球190个国家和地区都有自己的用户，因为全球范围内服务器的不同和客户的不同，所以这对我们做决策来说是一个巨大的挑战。而在全球的范围内，中国的经验就可以给我们提供很多的帮助，告诉我们应该如何来操作。

问：为什么中国北京区的预览服务中，在海外提供的移动服务并没有完全在中国有限预览的版本里提供？

答：事实上，我们所有的服务，在进行发布的时候，在全球的不同的市场都是分步实施的。因为AWS所提供的服务是非常多样的，而且我们在不断地进行开放的创新，所以让每一项服务在世界上所有的地区同步开展，应该是不太可能的，我们通常所做的是和本地区的客户进行直接的交流，并且知道他们的需求是什么。而我们在选择某一个地区的时候也会看一下这个地区原本的客户的积攒厚度是多少，也会看看我们和他们进行交流之后得到的反馈是什么，另外我们在这个地区已经取得的经验是否让我们有足够的基础来推出这项服务。所以我们通常都是在某个地区进行整个的调研之后，再来决定是不是要延续到其他的地区去。对中国而言很显然一项核心的业务是云计算，当然对其他方面的服务，我们需要时间来看一下，我们会以尽快的速度，在倾听客户的声音之后，逐步地把它带到中国来。因为只有和客户交流之后我们才能知道他们对我们的需求是什么，有这方面的需求我们才能以更快的方式把这个带过来，所以不可能实现全球的同步。事实上我们正在进行的有限预览，就是跟客户进行对话的一部分，以此来了解客户的声音，看一下究竟他们需要什么。

问：无论在今天的峰会上还是**re: Invent**上都有大量的培训的课程，亚马逊这么重视基础的培训的初衷是什么？

答：这种培训对我们的开发者掌握相关的技能是极为重要的，因为他们需要掌握所有的新技术。而我们现在非常大的重点是能够帮助中国的开发者去尽快地掌握他们所需要的一些基本的技能，尤其是如何更好地使用云。第二个部分是关于认证，因为必须要让开发者逐步地了解到，在他自己的知识在逐步进阶到一定程度的时候，AWS的认证就能认定他在这个领域已经是一个专家了。所以我们有几种典型的不同方面的培训，比如说关于系统架构的，关于运营方面的，以及关于DevOps。这是我们在re: Invent上刚刚推出的一个新领域培训。所以我再次强调，我们的培训目的应该是使我们的开发者能够更好地掌握技能并且进行学习的一个手段。

问：您如何总结过去的一年中，AWS在开发者生态做了哪些工作？

答：事实上我们工作主要是集中于以下的两个领域，第一，做科技领域的传播者，我们把它叫做Evangelist（布道者），他们主要负责和开发者探讨AWS的平台，我们在全世界都有这些科技的传播者。第二，主要是集中于开发者的社区的建设，以促进开发者彼此之间进行交流，让开发者能够彼此分享他们在AWS上面的一些经验，这样我们就能够建立一个非常活跃的开发者的社群，他们可以彼此互动。因此我们做的工作主要有三大块，第一，组织一些用户的团体；第二是Hackathon；第三“社区英雄”（“AWS Community Heroes”），我们会在所有开发者的成员中选择一些在AWS上做得非常好的成功经验，与所有其他社区的开发者进行分享。

问：开发者在地域性上有什么特点？

答：首先，我们在每个地区都要关注该地区重要的任务是什么。对中国而言，现在我们更需要关注的是移动平台，因为我们已经看到中国在这个领域的开发者群体有非常强烈的意愿提供移动应用和移动服务，这也许是中国地区的开发者社区和其他国际的开发者社区有所不同的地方，所以我们特别鼓励中国社区的开发者能够利用好AWS的平台，以便他能够在这一平台上提供移动服务以及相关的应用。

第二，我们必须要尊重在一个地区的地方特点，比如说就文化方面而言，我们要知道这个地区人们是如何来进行学习的。他是通过一些理论的教科书的方式还是通过培训的方式。再比如说人们是如何进行交流和组织事情的。所以我们必须要对这个地区的特点有所了解，这样才能够在各个地方对自己的开发者项目有不同的运作。比如说巴西可能会和中国有所不同，而中国可能又和美国有所不同。

但我想说的是，目前的国际经验告诉我们，开发者之间的共同点，要比他们之间的不同点更多。开发者的共同点是关注科技的问题，并且提出解决方案。而AWS恰好提供了这样一个绝佳的平台，让他们进行交流。

问：有一种声音认为**AWS**在国内做了大量的基础培训和教育市场的工作，但因此获益的不仅仅是**AWS**，而是本土大量云计算的厂商。**AWS**做了许多公益的工作，您对此怎么评价？

答：事实上，我们在尽力帮助我们的客户取得成功，既然我们有好的科技和解决方案，**AWS**又能够提供最好的工具和平台，我们为什么不这么做呢？我觉得我们愿意这么做。

技术热点



云生态专刊

从一个OpenStack的失败案例看Ironic和Neutron组件的现状

作者：杨赛

2015年1月12日，[Packet公司](#)平台部门VP David Laube在公司官方博客上发布了一篇名为《谈谈我们把四个月的工作量扔进垃圾堆的经验，或者应该说这是OpenStack的失败案例》的文章，介绍他们在尝试OpenStack中遇到的一些挫折。

Packet公司提供裸金属基础架构服务（也就是以前叫做物理机托管的服务），跟Softlayer和RackSpace做的生意差不多，但是做的时间短，是一家创业公司，大约在2014年上半年正式启动，本文作者David就是在去年夏天应邀加入的这家公司。Packet的创始人Zachary Smith在创立Packet之前做过Voxel这家公司，也是做服务器托管服务，后来在2011年卖给了Internap，成为Internap的服务器托管业务的一部分。

David一直以来的经验是在托管行业做系统运维，有十多年的运维经验（前九年都在HostRocket.Com）。2011年的时候他加入Voxel/Internap，成为Zachary的下属。按照David的自述，2014年夏初Zachary去邀请他加入Packet的时候，他一开始觉得当时市场上的IaaS服务已经很多，没有必要再做一个。但是聊着聊着他就觉得市场上的IaaS的确不够好用，再加上他当时已经玩了一段时间的Docker，非常看好在裸金属基础架构上做一套基于容器技术的部署系统的价值，所以就觉得值得做。

Voxel的部署管理系统完全是由自己编写的。在Voxel合并入Internap的那一年（2012年），他们遇到的无数网络问题、大量硬件变更和系统变更给这套系统带来了很大压力，这些问题最终体现在了服务稳定性变差，当时在[WHT](#)上有用户表达了强烈的不满。

现在有机会重新搭建一套自动化安装管理系统，而且最好在几个月内就能在生产系统上给客户用，David开始考察OpenStack。在David看来，如果OpenStack在网络自动化、IP管理、安装流程、硬件生命周期管理这几个方面的基础足够扎实，那完全可以依赖这套基础快速搭建起他想要的东西。

David做了大量调研，读了当时大部分的IRC讨论，自己也尝试用DevStack做过安装，觉得不错。让他真正做出决定的是来自RackSpace的一篇博客，该博客讲述了[RackSpace是如何用Ironic组件实现他们的裸金属云服务体系的](#)。

于是他们拿着最新的Juno版本提枪上战场了。四个月的努力让他们收获了如下经验。

- Nova、Ironic驱动、Neutron是他们最需要关心的三个组件。Ironic是为了实现裸金属管理，而Neutron则是为了避开2层网络和VLAN，让每台主机直接连接3层网络。
- 大部分文档是过时的或者错误的，于是David不得不用大量时间debug以验证哪个文档是正确的。
- OpenStack社区的确很大，人很多，但几乎没人有Ironic组件的生产使用经验，就连项目的核心开发者有时候也解答不了他们的部署问题。
- 一个人同时搞透Ironic和Neutron是一件几乎不可能完成的任务。David决定自己去研究Ironic，让另一位同事去研究Neutron。
- 阅读了每一篇Ironic的文档和讨论后，David的结论是，OpenStack从整体设计上就是为虚拟机准备的。你一旦用Ironic，就只能用openvswitch和linuxbridge。
- Neutron能够支持的厂商交换机有限，而且能够支持的网络模型也有限。
- RackSpace能跑起来是因为他们做了超级多的定制。而他们定制的很多重要补丁是不公开的，你得自己写。而你要能写出来这样的补丁，必须非常非常懂OpenStack的核心代码！

发展到这个阶段，David开始觉得自己掉进了一个大坑。花了大量的时间，但项目进展缓慢。不过，他还是决定要把Neutron再仔细了解一下。为何？

“在物理交换机和物理服务器的体系下，安装系统不难。但是要做到可靠，这就超级难了。自动化是无止尽的持续投入，而最容易出问题的地方就是网络的自动化。”

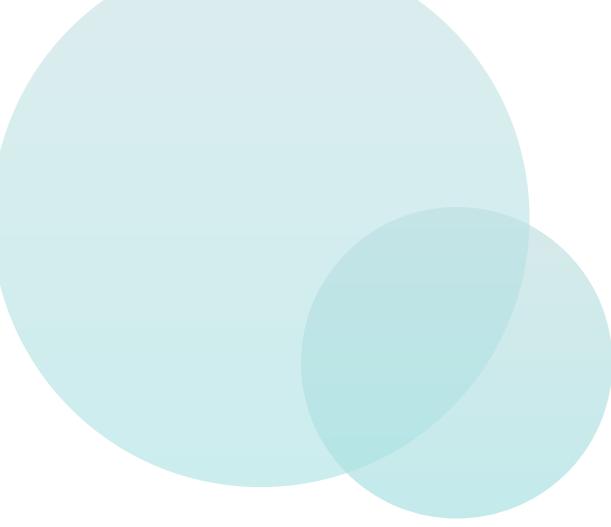
而Neutron宣称自己是：

“一套服务库，可提供按需使用、可扩展、支持多种技术的网络抽象。”

这样一套东西当然不可不看。但是看了之后，以下是David学到的东西。

- Neutron的大部分功能是针对虚拟机，而非物理交换机的。
- 他们用的是Juniper的交换机，而Juniper交换机的Neutron驱动已经老掉牙了，对Juno版也没提供多少新的支持。
- Neutron的IP管理体系自己做得很原始，又无法跟外部的IP资产管理体系对接。

最终，他们在圣诞前夕决定扔掉了OpenStack，然后用了三个星期开发了一套自己的IP管理系统。至于OpenStack，并不是不好，而是不合适他们的场景。之后他们还会继续关注OpenStack项目，并发布他们的一个Neutron驱动。



成功部署OpenStack 的十个技巧

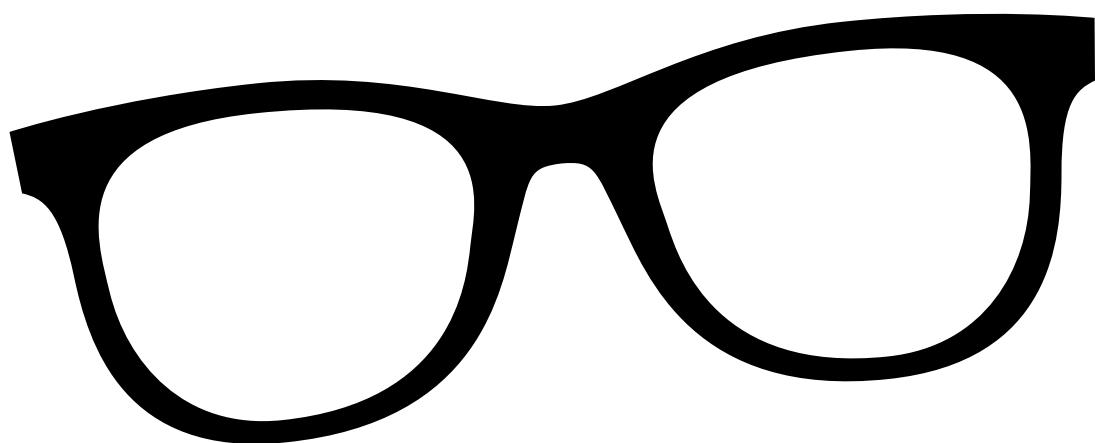
作者：谢丽

OpenStack为用户带来了诸多好处。使用免费开源工具构建自己的云对许多公司而言都非常有吸引力。但启动OpenStack项目之前，要有一个切实可行的目标。Rajiv Sodhi是OpenStack服务商[Mirantis](#)公司的一名区域总经理。近日，他在计算机世界英国站发表了一篇文章，给出了有助于OpenStack项目沿着正确方向前进的十个技巧。

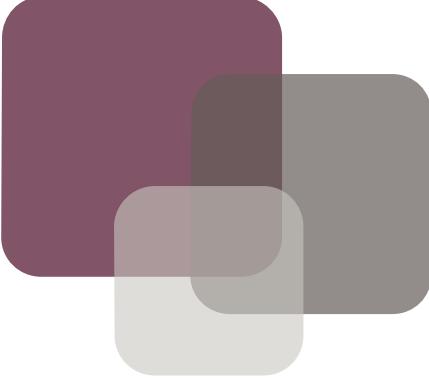
1. 资金准备——虽然部署OpenStack用的都是免费软件，不需要支付许可费，但开源软件并不是拿来就可以用的。与之相关的各种软件更新都非常快，而且软件的最新版本通常会不稳定，社区推出修复补丁的速度可能并不能满足项目需求。这时候就需要投入资金，雇佣人员进行Bug修复。因此，OpenStack项目需要预算和专用资源。
2. 人员准备——OpenStack项目通常比较大，需要许多人的参与。项目负责人必须了解其他所有人的需求，在文档中明确描述用例场景及项目建设目标。比如，是要创建一个公有云还是私有云，遗留应用如何处理等等。
3. 澄清术语——不要想当然地认为人们对某个术语有相同理解。要花时间了解每项工作：什么人、干什么、为什么、什么时间、在哪里、如何。
4. 接受遗留系统不会轻易消失的现实——我们可能无法将所有的系统都迁移到新构建的云上。一些遗留系统可能并没有做好往云上迁移的准备，尤其是那些业务规则没有在文档中完整描述的遗留系统。
5. 考虑迁移工作量——在大多数情况下，我们都无法仅仅通过克隆应用程序的所有元素就实现应用程序的扩展，有些服务可能需要从头搭建。
6. 与开发人员合作——应用程序在OpenStack中与在传统环境中不同。这使得运维人员和开发人员之间关系发生了变化。运维人员在完成云的构建之后，不仅要向开发人员提供足够的选择，还要提供更多的专业知识，以便他们可以恰当地架构和推进解决方案。
7. 不要假设团队成员具备所需的技能——OpenStack涉及IP网络、资源管理、源代码管理、存储冗余和优化、安全加密等众多技术，任何人在短时间内都很难全部掌握。

8. 出具方案——让每个人都知道他们需要知道的信息。比如，要让首席财务官知道云的好处及长远意义。那样他才会愿意投资。再就是需要有一个新的商业模式。我们经常看到，一些开始很小的公司借助Mirantis OpenStack Express实现了业务增长，因为它使他们的预算更具体、更可管理和预测。因此，实施OpenStack项目之前，要了解用户的经济状况和云的价值，然后提出相应地计划。
9. 针对程序崩溃制定恰当的方案——要有恰当的监控、冗余和预警，不能直到出现问题才知道。
10. 做好系统故障的准备——做好系统和应用程序出现故障的准备，确保系统在异常情况下可以不间断地运行。这样才能真正地体验到OpenStack的好处。

Docker 专栏



云生态专刊



etcd：从应用场景到实现原理的全方位解读

作者：孙健波

随着CoreOS和Kubernetes等项目在开源社区日益火热，它们项目中都用到的etcd组件作为一个高可用强一致性的服务发现存储仓库，渐渐为开发人员所关注。在云计算时代，如何让服务快速透明地接入到计算集群中，如何让共享配置信息快速被集群中的所有机器发现，更为重要的是，如何构建这样一套高可用、安全、易于部署以及响应快速的服务集群，已经成为了迫切需要解决的问题。etcd为解决这类问题带来了福音，本文将从etcd的应用场景开始，深入解读etcd的实现方式，以供开发者们更为充分地享用etcd所带来的便利。

经典应用场景

要问etcd是什么？很多人第一反应可能是一个键值存储仓库，却没有重视官方定义的后半句，用于配置共享和服务发现。

A highly-available key value store for shared configuration and service discovery.

实际上，etcd作为一个受到ZooKeeper与doozer启发而催生的项目，除了拥有与之类似的功能外，更专注于以下四点。

- 简单：基于HTTP+JSON的API让你用curl就可以轻松使用。
- 安全：可选SSL客户认证机制。
- 快速：每个实例每秒支持一千次写操作。
- 可信：使用Raft算法充分实现了分布式。

随着云计算的不断发展，分布式系统中涉及到的问题越来越受到人们重视。受阿里中间件团队对[ZooKeeper典型应用场景一览](#)一文的启发，笔者根据自己的理解也总结了一些etcd的经典使用场景。让我们来看看etcd这个基于Raft强一致性算法的分布式存储仓库能给我们带来哪些帮助。

值得注意的是，分布式系统中的数据分为控制数据和应用数据。使用etcd的场景默认处理的数据都是控制数据，对于应用数据，只推荐数据量很小，但是更新访问频繁的情况。

场景一：服务发现（Service Discovery）

服务发现要解决的也是分布式系统中最常见的问题之一，即在同一个分布式集群中的进程或服务，要如何才能找到对方并建立连接。本质上来说，服务发现就是想要了解集群中是否有

进程在监听udp或tcp端口，并且通过名字就可以查找和连接。要解决服务发现的问题，需要有下面三大支柱，缺一不可。

1. 一个强一致性、高可用的服务存储目录。基于Raft算法的etcd天生就是这样一个强一致性高可用的服务存储目录。
2. 一种注册服务和监控服务健康状态的机制。用户可以在etcd中注册服务，并且对注册的服务设置key TTL，定时保持服务的心跳以达到监控健康状态的效果。
3. 一种查找和连接服务的机制。通过在etcd指定的主题下注册的服务也能在对应的主题下查找到。为了确保连接，我们可以在每个服务机器人都部署一个Proxy模式的etcd，这样就可以确保能访问etcd集群的服务都能互相连接。

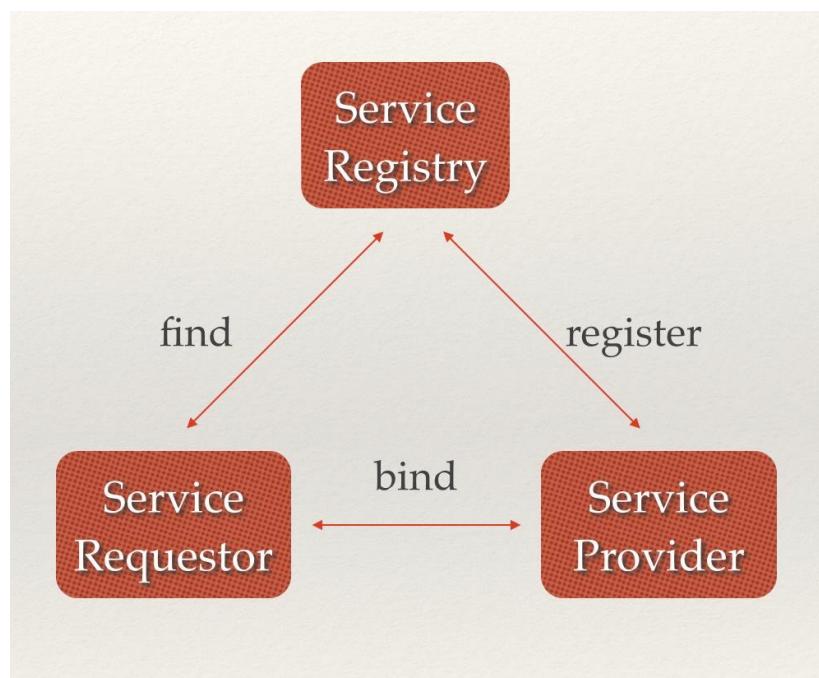


图1 服务发现示意图

下面我们来看服务发现对应的具体场景。

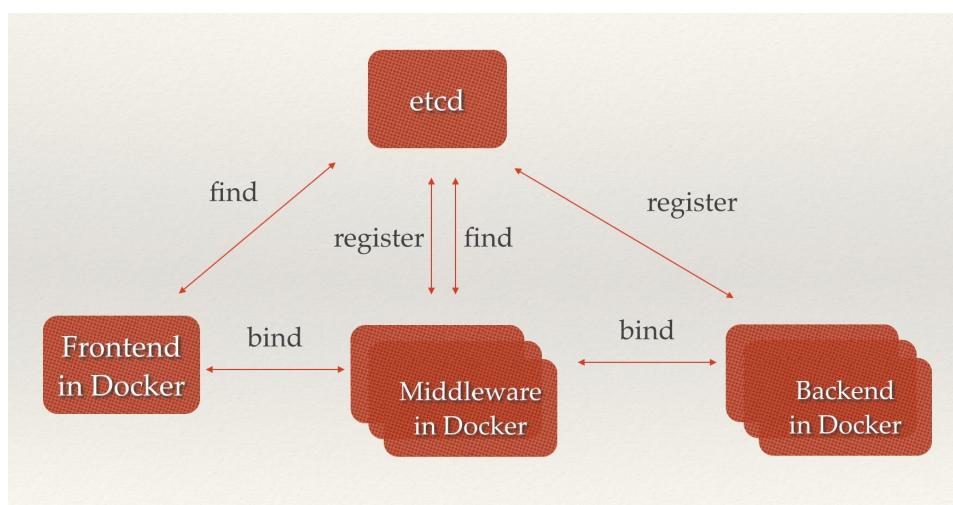


图2 微服务协同工作

- 微服务协同工作架构中，服务动态添加。随着Docker容器的流行，多种微服务共同协作，构成一个相对功能强大的架构的案例越来越多。透明化的动态添加这些服务的需求也日益强烈。通过服务发现机制，在etcd中注册某个服务名字的目录，在该目录下存储可用的服务节点的IP。在使用服务的过程中，只要从服务目录下查找可用的服务节点去使用即可。
- PaaS平台中应用多实例与实例故障重启透明化。PaaS平台中的应用一般都有多个实例，通过域名，不仅可以透明的对这多个实例进行访问，而且还可以做到负载均衡。但是应用的某个实例随时都有可能故障重启，这时就需要动态的配置域名解析（路由）中的信息。通过etcd的服务发现功能就可以轻松解决这个动态配置的问题。

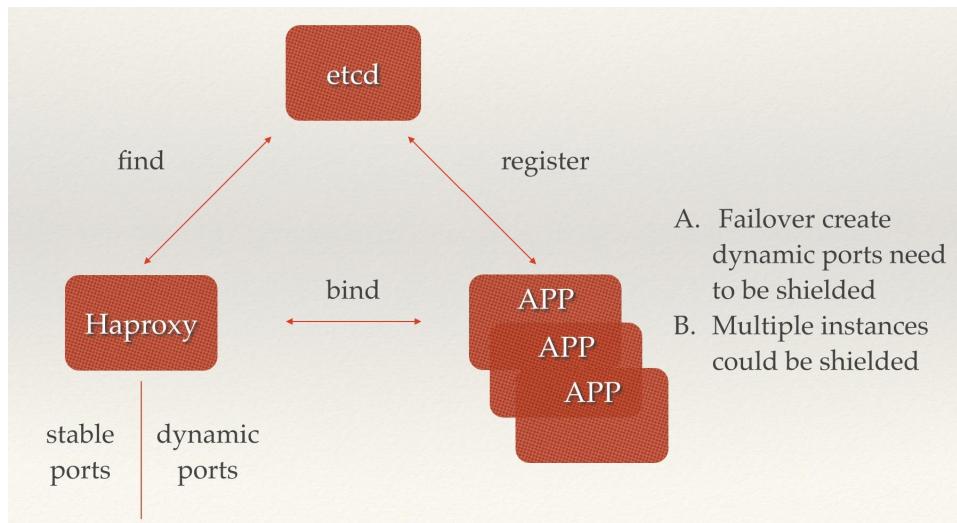


图3 云平台多实例透明化

场景二：消息发布与订阅

在分布式系统中，最适用的一种组件间通信方式就是消息发布与订阅。即构建一个配置共享中心，数据提供者在这个配置中心发布消息，而消息使用者则订阅他们关心的主题，一旦主题有消息发布，就会实时通知订阅者。通过这种方式可以做到分布式系统配置的集中式管理与动态更新。

- 应用中用到的一些配置信息放到etcd上进行集中管理。这类场景的使用方式通常是这样：应用在启动的时候主动从etcd获取一次配置信息，同时，在etcd节点上注册一个Watcher并等待，以后每次配置有更新的时候，etcd都会实时通知订阅者，以此达到获取最新配置信息的目的。
- 分布式搜索服务中，索引的元信息和服务器集群机器的节点状态存放在etcd中，供各个客户端订阅使用。使用etcd的key TTL功能可以确保机器状态是实时更新的。
- 分布式日志收集系统。这个系统的核心工作是收集分布在不同机器的日志。收集器通常是按照应用（或主题）来分配收集任务单元，因此可以在etcd上创建一个以应用（主题）命名的目录P，并将这个应用（主题相关）的所有机器ip，以子目录的形式存储到

目录P上，然后设置一个etcd递归的Watcher，递归式的监控应用（主题）目录下所有信息的变动。这样就实现了机器IP（消息）变动的时候，能够实时通知到收集器调整任务分配。

- 系统中信息需要动态自动获取与人工干预修改信息请求内容的情况。通常是暴露出接口，例如JMX接口，来获取一些运行时的信息。引入etcd之后，就不用自己实现一套方案了，只要将这些信息存放到指定的etcd目录中即可，etcd的这些目录就可以通过HTTP的接口在外部访问。

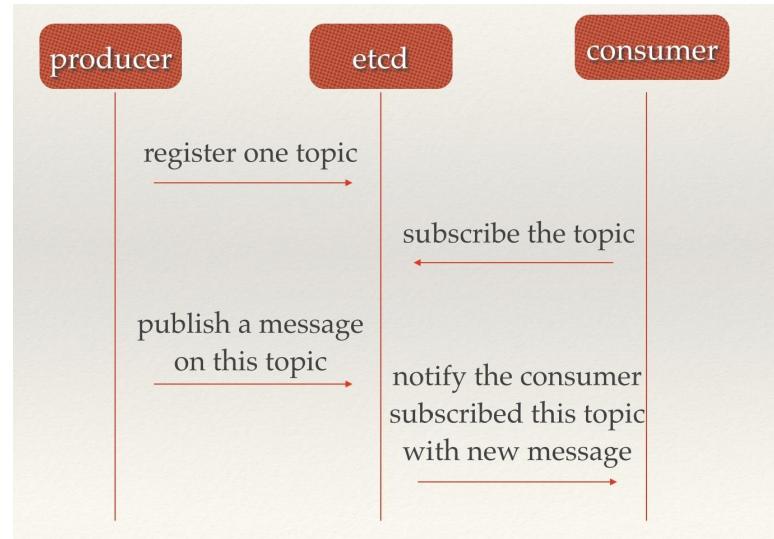


图4 消息发布与订阅

场景三：负载均衡

在场景一中也提到了负载均衡，本文所指的负载均衡均为软负载均衡。分布式系统中，为了保证服务的高可用以及数据的一致性，通常都会把数据和服务部署多份，以此达到对等服务，即使其中的某一个服务失效了，也不影响使用。由此带来的坏处是数据写入性能下降，而好处则是数据访问时的负载均衡。因为每个对等服务节点上都存有完整的数据，所以用户的访问流量就可以分流到不同的机器上。

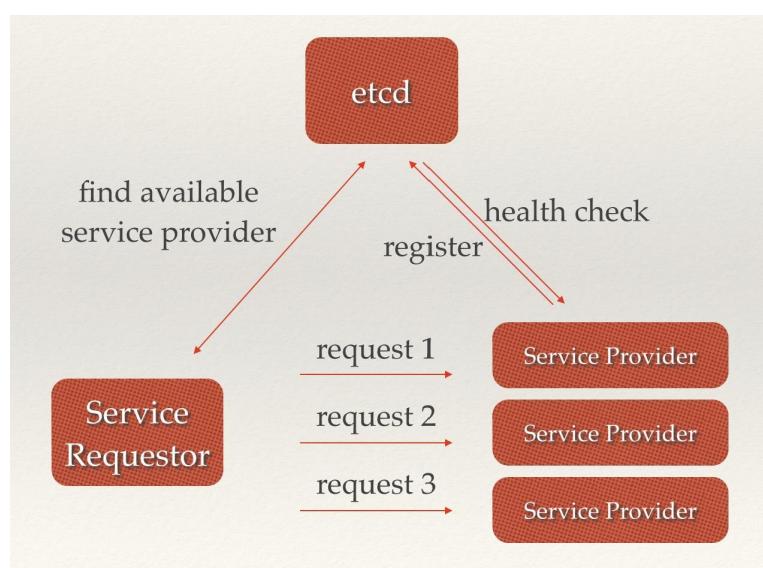


图5 负载均衡

- etcd本身分布式架构存储的信息访问支持负载均衡。etcd集群化以后，每个etcd的核心节点都可以处理用户的请求。所以，把数据量小但是访问频繁的消息数据直接存储到etcd中也是个不错的选择，如业务系统中常用的二级代码表（在表中存储代码，在etcd中存储代码所代表的具体含义，业务系统调用查表的过程，就需要查找表中代码的含义）。
- 利用etcd维护一个负载均衡节点表。etcd可以监控一个集群中多个节点的状态，当有一个请求发过来后，可以轮询式的把请求转发给存活着的多个状态。类似KafkaMQ，通过ZooKeeper来维护生产者和消费者的负载均衡。同样也可以用etcd来做ZooKeeper的工作。

场景四：分布式通知与协调

这里说到的分布式通知与协调，与消息发布和订阅有些相似。都用到了etcd中的Watcher机制，通过注册与异步通知机制，实现分布式环境下不同系统之间的通知与协调，从而对数据变更做到实时处理。实现方式通常是这样：不同系统都在etcd上对同一个目录进行注册，同时设置Watcher观测该目录的变化（如果对子目录的变化也有需要，可以设置递归模式），当某个系统更新了etcd的目录，那么设置了Watcher的系统就会收到通知，并作出相应处理。

- 通过etcd进行低耦合的心跳检测。检测系统和被检测系统通过etcd上某个目录关联而非直接关联起来，这样可以大大减少系统的耦合性。
- 通过etcd完成系统调度。某系统有控制台和推送系统两部分组成，控制台的职责是控制推送系统进行相应的推送工作。管理人员在控制台作的一些操作，实际上是修改了etcd上某些目录节点的状态，而etcd就把这些变化通知给注册了Watcher的推送系统客户端，推送系统再作出相应的推送任务。
- 通过etcd完成工作汇报。大部分类似的任务分发系统，子任务启动后，到etcd来注册一个临时工作目录，并且定时将自己的进度进行汇报（将进度写入到这个临时目录），这样任务管理者就能够实时知道任务进度。

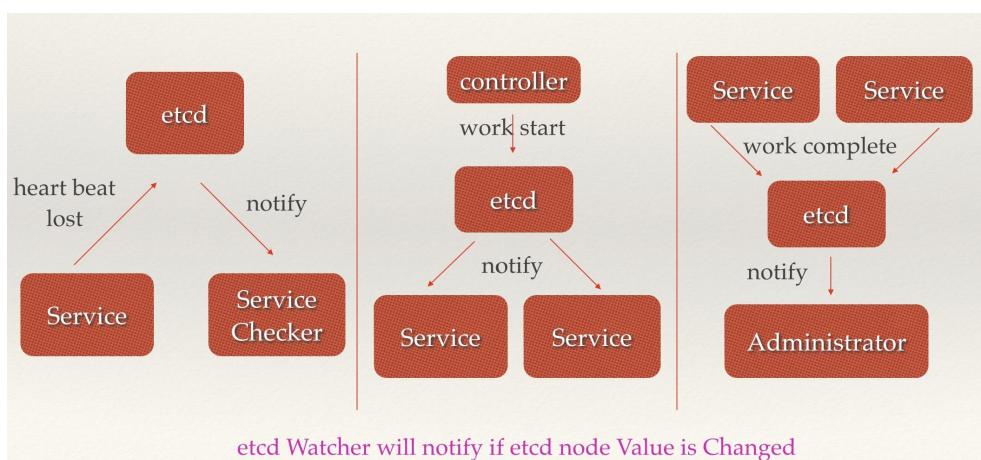


图6 分布式协同工作

场景五：分布式锁

因为etcd使用Raft算法保持了数据的强一致性，某次操作存储到集群中的值必然是全局一致的，所以很容易实现分布式锁。锁服务有两种使用方式，一是保持独占，二是控制时序。

- 保持独占即所有获取锁的用户最终只有一个可以得到。etcd为此提供了一套实现分布式锁原子操作CAS (CompareAndSwap) 的API。通过设置prevExist值，可以保证在多个节点同时去创建某个目录时，只有一个成功。而创建成功的用户就可以认为是获得了锁。
- 控制时序，即所有想要获得锁的用户都会被安排执行，但是获得锁的顺序也是全局唯一的，同时决定了执行顺序。etcd为此也提供了一套API（自动创建有序键），对一个目录建值时指定为POST动作，这样etcd会自动在目录下生成一个当前最大的值为键，存储这个新的值（客户端编号）。同时还可以使用API按顺序列出所有当前目录下的键值。此时这些键的值就是客户端的时序，而这些键中存储的值可以是代表客户端的编号。

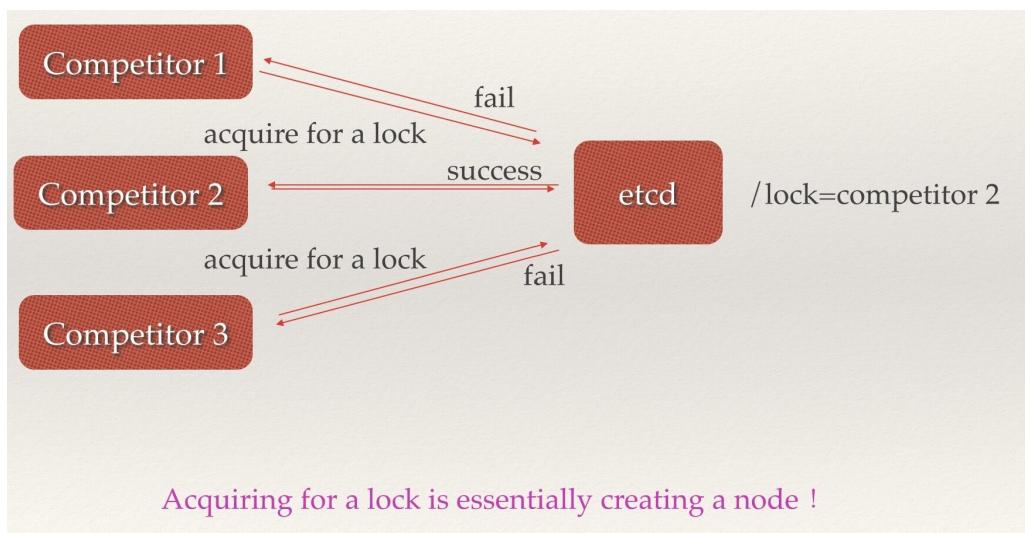


图7 分布式锁

场景六：分布式队列

分布式队列的常规用法与场景五中所描述的分布式锁的控制时序用法类似，即创建一个先进先出的队列，保证顺序。

另一种比较有意思的实现是在保证队列达到某个条件时再统一按顺序执行。这种方法的实现可以在/queue这个目录中另外建立一个/queue/condition节点。

- condition可以表示队列大小。比如一个大的任务需要很多小任务就绪的情况下才能执行，每次有一个小任务就绪，就给这个condition数字加1，直到达到大任务规定的数字，再开始执行队列里的一系列小任务，最终执行大任务。
- condition可以表示某个任务不在队列。这个任务可以是所有排序任务的首个执行程序，也可以是拓扑结构中没有依赖的点。通常，必须执行这些任务后才能执行队列中的其他任务。

- condition还可以表示其它的一类开始执行任务的通知。可以由控制程序指定，当condition出现变化时，开始执行队列任务。

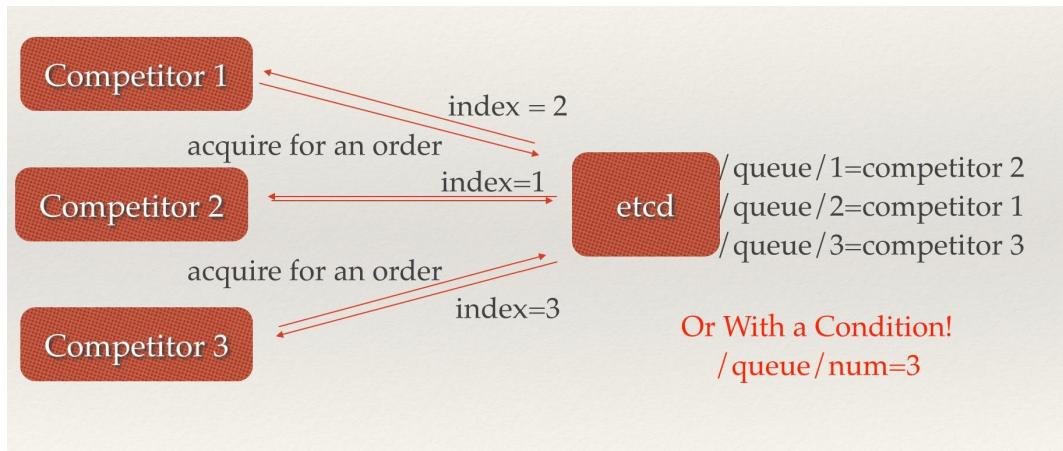


图8 分布式队列

场景七：集群监控与Leader竞选

通过etcd来进行监控实现起来非常简单并且实时性强。

- 前面几个场景已经提到Watcher机制，当某个节点消失或有变动时，Watcher会第一时间发现并告知用户。
- 节点可以设置TTL key，比如每隔30s发送一次心跳使代表该机器存活的节点继续存在，否则节点消失。

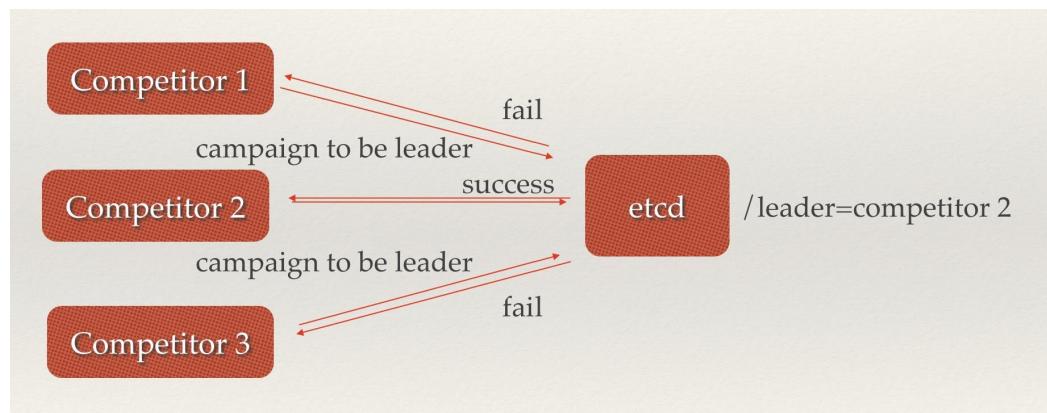


图9 Leader竞选

这样就可以第一时间检测到各节点的健康状态，以完成集群的监控要求。

另外，使用分布式锁，可以完成Leader竞选。这种场景通常是一些长时间CPU计算或者使用IO操作的机器，只需要竞选出的Leader计算或处理一次，就可以把结果复制给其他的Follower。从而避免重复劳动，节省计算资源。

这个的经典场景是搜索系统中建立全量索引。如果每个机器都进行一遍索引的建立，不但耗时而且建立索引的一致性不能保证。通过在etcd的CAS机制同时创建一个节点，创建成功的机器作为Leader，进行索引计算，然后把计算结果分发到其它节点。

场景八：为什么用etcd而不用ZooKeeper？

阅读了“[ZooKeeper典型应用场景一览](#)”一文的读者可能会发现，etcd实现的这些功能，ZooKeeper都能实现。那么为什么要用etcd而非直接使用ZooKeeper呢？

相较之下，ZooKeeper有如下缺点。

1. 复杂。ZooKeeper的部署维护复杂，管理员需要掌握一系列的知识和技能，而Paxos强一致性算法也是素来以复杂难懂而闻名于世。另外，ZooKeeper的使用也比较复杂，需要安装客户端，官方只提供了Java和C两种语言的接口。
2. Java编写。这里不是对Java有偏见，而是Java本身就偏向于重型应用，它会引入大量的依赖。而运维人员则普遍希望保持强一致、高可用的机器集群尽可能简单，维护起来也不易出错。
3. 发展缓慢。Apache基金会项目特有的“[Apache Way](#)”在开源界饱受争议，其中一大原因就是由于基金会庞大的结构以及松散的管理导致项目发展缓慢。

而etcd作为一个后起之秀，其优点也很明显。

1. 简单。使用Go语言编写部署简单；使用HTTP作为接口使用简单；使用Raft算法保证强一致性让用户易于理解。
2. 数据持久化。etcd默认数据一更新就进行持久化。
3. 安全。etcd支持SSL客户端安全认证。

最后，etcd作为一个年轻的项目，真正告诉迭代和开发中，这既是一个优点，也是一个缺点。优点是它的未来具有无限的可能性，缺点是无法得到大项目长时间使用的检验。然而，目前CoreOS、Kubernetes和CloudFoundry等知名项目均在生产环境中使用了etcd，所以总的来说，etcd值得你去尝试。

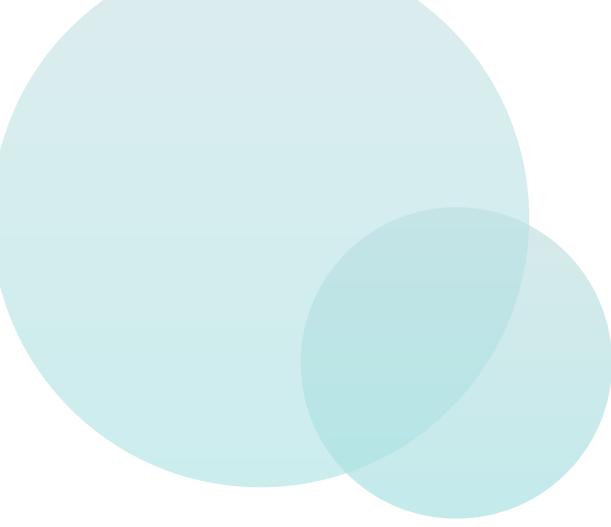
2 作者简介

孙健波，[浙江大学SEL实验室硕士研究生](#)，目前在云平台团队从事科研和开发工作。浙大团队对PaaS、Docker、大数据和主流开源云计算技术有深入的研究和二次开发经验，团队现将部分技术文章贡献出来，希望能对读者有所帮助。

原文链接：<http://www.infoq.com/cn/articles/etcd-interpretation-application-scenario-implement-principle>

观点&趋势

云生态专刊



BaaS服务的定义、发展以及未来

作者：郭蕾

BaaS (Backend as a Service) 是一种新型的云服务，旨在为移动和Web应用提供后端云服务，包括云端数据/文件存储、账户管理、消息推送、社交媒体整合等。BaaS是垂直领域的云服务，随着移动互联网的持续火热，BaaS也受到越来越多的开发者的亲睐。它作为应用开发的新模型，可以降低开发者成本，让开发者只需专注于具体的开发工作。

BaaS是[移动中间件](#)的替代品（或者说备选方案），它使用统一的API和SDK来连接移动应用到后端云存储，传统的移动中间件通过本地的物理服务把后端服务集成到应用中。而BaaS通过云来集成后端服务。中间件和BaaS的最大不同是它们是否包含或者提供云的服务，BaaS可以说是PaaS平台在移动垂直领域的延伸，更可以说是移动中间件和云的融合。而现在它们都在以不同的形式来存在，云的优势很明显，那就是简单、成本低廉，中间件的优势是数据安全、易于扩展。所以从现在的趋势来看，它们不存在明显的取代关系，只不过可能以后BaaS的体量会更大。移动中间件将更多的被有能力的企业使用，同时也会有越来越多的中小型企业、开发者选择使用BaaS。

虽然BaaS属于PaaS的范畴，但两者也有区别。[Quora](#)上有人简要描述了二者的不同，BaaS简化了应用开发流程，而PaaS简化了应用部署流程。PaaS是一个执行代码以及管理应用运行环境的开发平台，用户通过SVN或者Git之类的代码版本管理工具与平台交互，对于开发者来说，PaaS就像是一个容器，输入是代码和配置文件，输出是一个可访问应用的URL。而BaaS平台进一步将用户需求进行了抽象，比如用户管理，开发者希望创建用户数据库表（模型）后，客户端就可以通过Restful接口直接操作对应的模型，所有的操作都可以被抽象为CRUD。之前，开发者需要创建表、写接口、写校验，而在BaaS平台中，开发者只需要定义模型，平台就会自动生成对应的接口，这可以让开发者更加专注具体的客户端代码。专门针对手机端的BaaS服务称为MBaaS，目前大多的BaaS平台都属于这一类。

随着移动互联网的发展，移动行业的分工也会像其它行业一样逐渐细化，后端服务就是这样被抽象出来，它统一向开发者提供文件存储、数据存储、推送服务等实现难度较高的功能，以帮助开发者快速开发移动应用。在国外，BaaS服务已经受到巨头的重视，2013年4月，Facebook收购Parse；2014年6月，苹果发布了CloudKit；2014年10月，Google收购了Firebase。[Parse](#)、[CloudKit](#)、[Filrebase](#)都是国外知名的BaaS类产品，苹

果和谷歌通过BaaS服务可以更好的完善其生态圈，Parse也可以帮助Facebook建立它在移动端的地位，从巨头们在BaaS方面的布局也可以看出BaaS的价值。总体来说，[BaaS平台的优势](#)包括（来自搜狗百科）：

- 提高效率：减少移动APP开发中各个环节的成本，提高效率；
- 缩短上市时间：减少从构思到制作过程中的阻碍，并降低上线后的运营成本；
- 减少交付APP所需的资源：BaaS需要的开发者和IT资源更少；
- 针对手机和平板优化：BaaS供应商在优化移动APP数据和网络上花费了大量时间和资源，减少了跨平台和移动终端的碎片化的问题；
- 安全和弹性的基础设施：BaaS提供捆绑的基础设施，解决了弹性、安全性和性能等运营难题，让开发者专注开发；
- 大量的常用API资源：BaaS将常用和必要的第三方API资源汇总，省去开发者单独收集的麻烦。

在国内，提供BaaS服务的厂商也[有很多](#)，典型的代表有[APICloud](#)、[Bmob](#)、[友盟](#)，主要提供的功能包括社会化媒体集成、数据/文件存储、数据分析、消息推送、支付。以APICloud为例，它们主要提供的服务包括：

- 数据存储。用户可以通过可视化的界面设计数据库，包括创建Class、定义字段、录入数据等。同时，BaaS平台可以自动生成对应的Restful API，用户可以通过任何语言操作已有的API，另外，平台也内置用户系统、角色系统、文件系统、权限控制等模块。
- 数据推送。结合APP中的标签设置，针对不同属性的用户推送差异化信息，包括定时推送、离线推送等。
- 版本管理。支持iOS及Android版本的同步或异步管理，在控制台内流程化进行开发和版本管理。支持增量更新，终端用户可在应用内进行更新。
- 数据统计。平台可以查看应用的新增用户以及活跃用户数据，并支持自定义事件统计。

从功能上看，国内的BaaS厂商（特指能够提供完整的平台能力的厂商）提供的功能大同小异，大都集中在推送、存储、统计方面。值得注意的是，这几个重点功能又有相应的厂商在做，比如文件存储的七牛和又拍、推送服务的极光推送、统计服务的友盟、及时聊天的环信，所以随着这块市场的成熟，BaaS平台在功能方面的重心应该是整合其它垂直云服务的能力。

从盈利模式看，都是向少部分用户收费。纵观目前面向开发者的公司，它们的盈利模式大多是部分服务收费或者部分用户收费，现在的这几家BaaS厂商基本都是对部分高端用

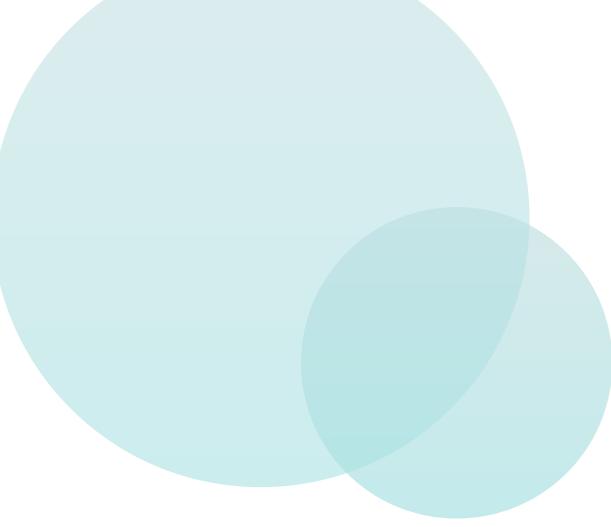
户收费。但是从云的发展趋势来看，接下来会有更多的中小型公司会使用BaaS服务，所以新一年BaaS平台也许会面向企业提供差异化的服务。

从竞争角度来看，由于BaaS在国内的整体份额都比较小，所以目前各个厂商都在全力扩展自己的用户基数，直接的竞争还谈不上。不过，目前市场的几家厂商侧重点也不一样，比如APICloud提供的是端和云的能力，用户可以通过SDK开发跨平台的应用。

分析机构 MarketsandMarkets 报告 BaaS 市场到 2017 年将会达到 77 亿美元，而 2012 年仅为 2.165 亿美元，年增长率达到了 104%。预计在2015年BaaS服务会受到更多用户的亲睐，BaaS的发展趋势总体来看可以总结为如下几个方面。

- 出现更多的垂直云服务：随着技术的发展与市场需求，整个移动互联网行业发展的特点是更加的垂直、细分和专业，所以也会出现更多的垂直领域的BaaS服务提供商。
- API云服务蓬勃发展：随着云和大数据的结合，业务层跟数据层结合的越来越紧密，移动APP更侧重界面的逻辑和表现，而APP所需的数据与服务都需要通过API的形式从云端获取，所以能够提供数据存储和App逻辑业务相关的API输出的数据云BaaS服务将会有更多的需求和发展。
- 满足自定义功能扩展：BaaS在提供标准服务的基础上，让开发者可以根据自己的产品和业务特点，通过在线配置和上传代码的功能来扩展自定义的功能，满足个性化需求。
- 成为行业移动化解决方案：随着移动互联网和越来越多的行业结合，BaaS服务以其简洁、高效、灵活、专业的特点，也会应用到各种行业的解决方案中，成为行业移动化解决方案中云端的支撑服务。

随着BaaS服务的成熟和稳定，基础服务功能使用专业的BaaS服务已经成为了移动应用开发中的常规选择，被越来越多的客户接受，2015年BaaS服务有更好发展。



谷歌关于容器的观点分享

作者 : João Miranda 译者 : 丛一

通过借鉴他们在这一技术领域十年的经验，谷歌云平台团队撰写了一系列的文章分享其对容器的看法。谷歌的前两篇文章提供了关于这个主题的总览。这两篇文章解释了容器集群和他们所定义特征背后的逻辑。之后又向读者展示了如何在Kubernetes上应用这一切。

容器能够带来一系列的好处。包括更简单的部署模型，快速可用性和微服务架构理想的基础设施层。像Docker之类的容器产品更关注于在单一计算机上操作。这就产生了协调多个容器的需求。或者用高级资深工程师和Kubernetes联合创始人Joe Beda更喜欢的说法，“即兴爵士乐表演”的管理需求，这个说法能够更好地反映出容器管理对“实时条件和输入”的反应。

像Kubernetes这样的容器集群提供了与集群管理、网络和命名有关的服务。他们“创建了一个抽象的层次，让开发人员和管理员可以将需要改进的服务作为一个整体，改进其行为和性能，而不是服务的某个容器组件或基础设施资源”。

随着容器数量的迅速增加，容器协调的需求也随之迅速地变得显而易见。谷歌就是个极端的例子，每秒要启动7000个容器，或每周要启动20亿个。这就是对容器集群的需求。

除了其他的要求之外，谷歌构建的容器集群还必须能够做到在保持可用的状态下完成更新，可扩展并且便于操控和监控。通过满足这些要求，谷歌收益匪浅。首先，可以构建有清晰的合约和边界的微服务。然后，清晰的边界让小型工程团队能够保持软件的可管理和可扩展。容器集群具有自愈能力和无摩擦的水平扩展能力，并且有着高效的资源利用率。“集群运维团队和应用运维团队角色”的专业化能力是另一个隐含的好处。Joe Beda举例提到，“GMail的运维和开发团队很少需要与集群运维团队直接对话”。开发人员可以将注意力集中在构建服务上，而不需要考虑底层的基础设施问题。

优秀容器集群的要素

Joe还分享了“优秀的容器集群管理器所需的要素”。包括动态的容器配置，以集合方式思考以及集群内的连接。

动态容器配置就是在考虑容器应该如何运行以及运行在何处的公开意向的前提下，找到一个可以运行给定载荷的位置。这是一个背包问题的实例。每个容器都有其自身的限制，如何在有限的可用计算资源内匹配各个容器呢？在安排给定的载荷时，集群必须要考虑可用容量（如

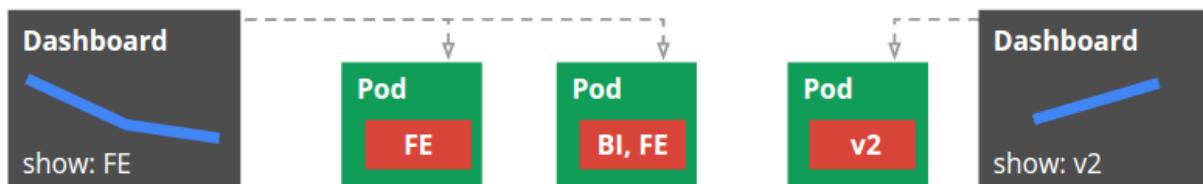
CPU、RAM等），特殊的硬件需求以及实时变化的条件（如故障、自动扩展等）。

Kubernetes用Pod解决了这个问题：

Pod（和一群鲸鱼（a pod of whales）或豌豆荚（pea pod）里的含义一样）由一组相互关联的共享容量的Docker容器构成。Pod模型化为容器化环境下，特定于应用程序的一个“逻辑宿主”。其中可能包含一个或多个相互关联并紧密耦合的容器——在前容器（pre-container）世界，他们会执行在同一个物理或虚拟主机上。

第二个要素是提供以集合方式思考的可能性。Kubernetes通过提供[标签](#)和[复制控制器](#)使其成为可能。每个Pod可以有一系列的标签，每个标签是一个键值对。Pod通过标签分组，例如根据应用层级或地理位置。之后标签的使用可以有多种方式，例如由[服务](#)或复制控制器使用。复制控制器，顾名思义，用于保证大量的Pod拷贝能够一直保持活动，从而为横向扩展活动提供帮助。

Kubernetes Labels



Behavior	Benefits
● Metadata with semantic meaning	→ Allow for intent of many users (e.g. dashboards)
● Membership identifier	→ Build higher level systems ...

Kubernetes标签，来自于文章“[是什么造就了容器集群？](#)”

成功的容器集群的第三个要素是通信。当有多个容器，进而有多个微服务存在时，通信就成为了一个关键的组件。为了在无需知道它们各自的容器实际运行位置的情况下，微服务之间也可以互相通信，好的容器集群应该提供一个命名解析系统。命名解析系统在容器启动或迁移之后能够立即了解到全部变化是十分重要的。Kubernetes在标签和Watch API模式的帮助下，提供了轻量级的服务发现方案。受[Google Chubby论文](#)的启发，该模式提供了一种从服务传递异步事件的方式。Kubernetes团队意识到并不是所有的客户端都会为了使用这个API而重写。因此Kubernetes提供服务代理的概念来处理这类场景。

[服务代理]是一个简单的网络负载均衡器/代理，以单一稳定的IP/端口形式暴露在网络中，可以帮助完成命名查询。

简而言之，谷歌云平台团队将容器集群定义为：

一个动态配置和管理容器的系统，以Pod的形式组合在一起，连同所有的相互连接和通信渠道，运行在节点之上。

谷歌向容器化的转移开始于将[cgroups](#)（控制群组的简写）加入Linux内核，以隔离一个进程集合的资源使用。通过简化容器技术，[Docker](#)让更大范围的社区采用了容器技术，从而使这种技术得以推广。

InfoQ一直在[跟进Kubernetes的发展](#)，其中包括一篇由[Carlos Sanchez](#)执笔的深度文章。



生态圈新闻

云生态专刊

七牛音视频服务价格下调80%

根据艾瑞数据显示，2013年9月，国内移动视频用户已达1.3亿，同比增长105.7%。随着4G网络进一步普及，音视频的分享与消费，将与移动化、碎片化的用户特征结合，取代单一的文字信息流，成为用户的新习惯。

作为音视频基础技术的专业服务提供者，七牛云存储整合多家CDN资源，并提供全套音视频的存储、加速、转码、水印、切片、拼接、取帧以及流媒体播放服务。在高达300亿的新市场开启之际，音视频相关技术的完善，以及产业链上下游的配合，将直接决定产业未来的成熟度。从2015年2月1日起，七牛决定将音视频服务价格整体下调80%，为即将到来的新时代开路。

APICloud发布模块Store 提供一站式解决方案

2月4日，移动应用云服务平台APICloud发布“模块Store”，模块Store是一个聚合的平台，提供了APP开发生命周期各个阶段的第三方解决方案，包括支付宝、微信、环信等。APICloud将第三方应用与自身平台进行了深度定制，开发者只需要点击一个按钮即可完成集成。通过这一产品，APP开发者无需再去寻找不同的第三方服务提供商并逐个进行沟通，所以官方也号称他们提供的是1+1（一站式和一键集成）的解决方案。

APICloud定位是一家云+端的移动云平台，发布四个月通过开发者邀请自传播的方式就获得了超过3万移动开发者（用户），目前已经进军欧美市场。近期也会发布正式版本，并开放注册。

IBM云计算体验周在京召开 带用户全方位体验IBM云布局

1月28日，“IBM云计算体验周”在北京召开，这是IBM在国内首次以体验周的形式展示IBM在混合云发展趋势下的全方位能力，IBM与众多用户共同分享了如何通过与IBM的合作在各自的领域通过云计算进行创新的实践。

IBM近两年在云计算领域的布局已经初见成效，根据IBM刚刚发布的2014年财报显示，2014年IBM云计算收入同比增长60%，达到70亿美金。目前IBM已经完成了很多布局，包括与苹果、SAP、微软、英特尔等全球顶级企业展开合作，同时也包括与国内像腾讯、世纪互联、华胜天成、讯鸟软件、南天电脑、威达云等企业之间的合作。

环信即时通讯云获红杉A+轮300万美元融资

1月21日消息，环信即时通讯云CEO刘俊彦宣布，环信A+轮融资资金到位。本轮投资机构为红杉资本，本轮融资金额为300万美元。至此，环信经过3轮融资已经拥有资金储备近1000万美元。其中天使轮为经纬中国、A轮为SIG、A+轮为红杉资本。

刘俊彦表示，环信的融资将主要用于后续研发和云服务运维。环信的愿景是为所有的App提供沟通和社交能力，这样一个市场规模将远远大于目前任何一个已知的社交平台，包括微信。他表示当前环信已经服务了1万3千多家APP，SDK覆盖的注册用户已经过亿。

青云QingCloud支持私有网络LB和IPsec加密隧道服务

青云日前宣布推出基于网络层面的两项重要功能，即负载均衡器的私有网络模式和路由器的IPsec加密隧道服务。前者是QingCloud在负载均衡器技术上的又一重大突破，私有网络LB不仅能让用户节省公网IP，还能通过私有网络LB构建出更加复杂的网络架构，让企业拥有与物理世界一样的IT环境。后者是通过使用加密的安全服务在不同的网络之间建立保密和安全的通讯隧道，能够确保混合云中的数据传输更加安全。

负载均衡器可以将来自多个公网地址的访问流量分发到多台主机上，并支持自动检测和隔离不可用的主机，从而提高业务的服务能力和可用性。目前，QingCloud的负载均衡器支持HTTP/HTTPS/TCP三种监听模式，并支持透明代理，可以让后端主机不做任何更改，直接获取客户端真实IP。

UnitedStack在公有云中实现共享存储

UnitedStack UOS公有云使用Ceph搭建高性能的块存储服务，提供了满足企业级存储要求的稳定、高性能、高可靠性、高可用性的存储服务，又避免传统DAS、NAS、SAN存储的容量限制与高成本。Ceph可以从容应对很多故障，能够轻松扩展，并使用它支撑公有云和托管云的云主机、云硬盘服务。任何依赖共享存储才能够运行的应用你都可以在UOS公有云上做测试，包括数据库高可用（Oracle RAC、MySQL HA）及共享文件系统。

AWS新推WorkMail主打兼容性与安全性

AWS于上月底宣布推出面向企业级电子邮件市场的新服务WorkMail，这是一项托管型商务电子邮件与日历编制服务，它能够良好兼容企业用户现有的桌面与移动电子邮件客户端，并与iOS、Android、Amazon Fire以及Windows Phone设备进行同步，同时WorkMail还允许企业用户自行控制加密数据的密钥以及用于存储数据的AWS区域位置，即便邮件被拦截，其内容也不会被泄露。WorkMail现在已经已推出预览版，每个注册账号可以免费试用30天，最多可供25人使用。正式版推出后，将采用每月/每用户4美元的方式进行收费，每位用户将拥有50GB的存储容量。

TechRepublic专栏作家、Java工程师James Sanders在博客上表示，WorkMail对比微软与谷歌的同类服务并不具备突出的优势，但该服务的推出对于整个企业级电子邮件服务市场的良性竞争来说，仍然具有一定的积极意义。

QCon

全球软件开发大会

2015.4.23-25 北京国际会议中心



9折
优惠

现在报名

(截至3月29日)

- 针对软件开发领域的热点与趋势，议定18个最具实践价值的专题，覆盖更广的开发者人群，为参会者提供更多选择

可扩展、高可用架构设计

移动开发最佳实践

新兴大数据处理技术与工具

安全、隐私与风控

自动化运维

互联网金融背后的技术架构

云计算平台构建与应用

微服务和响应式框架

团队建设

Java和新锐语言

软件质量

超越JavaScript

敏捷之后，是什么

思考开源

知名移动案例分析

基于大数据的机器学习和数据挖掘

挑战全栈开发

技术创业

抢票热线：010-64738142

会务咨询：qcon@cn.infoq.com

精彩内容策划中，讲师自荐/推荐，线索提供：speakers@cn.infoq.com

更多经典专题 精彩内容 敬请登录：www.qconbeijing.com

QClub

Docker专场——聚焦国内Docker创业以及企业实践

Docker显然已经成为2014年最火的开源项目之一，它受到越来越多的开发和运维人员的青睐。本期QClub聚焦国内Docker创业以及最佳实践案例，以及Docker创业公司将如何利用这一支点来撬动新的市场。



docker

时间：2015年3月29日（周日）9:00 ~ 17:30

地址：北京市海淀区知春路25号 丽亭华苑酒店

限量免费点击报名



促进软件开发领域知识与创新的传播



中文 | 英文 | 日文 | 葡文 |