

# 平台即服务（ PaaS ）实操指南

## 平台即服务 ( PaaS ) 实操指南

最初，平台即服务厂商根据自身支持的语言确立自己的市场差异性，比如支持 Java 或者 .NET，但是最终他们还是要不断发展支持多种语言，最后还要用基础架构即服务支持数据存储、消息服务、应用服务和可迁移性。市场上为开发者提供了各种 PaaS。虽然看起来 PaaS 厂商都极为相似，但是却也存在诸多不同。要跟上市场的变化，在选择 PaaS 平台时，有一些关键性能需要注意。

在这本技术指南中我们将重点介绍平台即服务的市场现状以及如何将遗留开发转移到 PaaS 之上，同时我们会给出最新的私有 PaaS 概念和产品选择指导，希望对您有所帮助！另外如果您有任何意见或者建议欢迎致信 [zhangpeiying@techtarget.com.cn](mailto:zhangpeiying@techtarget.com.cn)。

### PaaS 市场全观察

平台即服务在今年滋生出很多话题，而且 2014 年其采用可能持续攀升，但是从长期来看，这个分类将会越来越难在日渐模糊的云服务类型中定义。最近的云市场调查显示，IT 专业人士在新的一年里对于平台即服务 ( PaaS ) 的兴趣增加了。下面我们就来看看具体的市场情况，以及企业该如何根据自身的现状作出选择。

- ❖ PaaS 采用会在 2014 年持续攀升？
- ❖ 云市场观察：平台即服务厂商
- ❖ 当 IaaS 不再 迁移至 PaaS 云前途几何

### 遗留开发和 PaaS 真假

云计算已经刮起了流行旋风，随着其不断成熟，应用不断增长，新用户涌现。在 IT 业界软件即服务和基础架构即服务愈发如鱼得水。但是因为云洗白和市场不成熟，平台即服务仍旧处在黎明前的黑暗中。另外，相比于管理企业自己的发展基础设施，平台即服务提供了很多优势，可以有更多时间专注于设计和编码。然而对于新项目来说，PaaS 可能是首选，但是 PaaS 可能不适合现有的、遗留的开发工作。那么该如何抉择呢？

- ❖ 雾里看花 巧辨真假 PaaS

- ❖ [遗留开发工作入 PaaS 你考虑好了吗](#)
- ❖ [遗留开发进入 PaaS 五要素](#)
- ❖ [PaaS 应用可移植性：问题与解决方案](#)

## 私有 PaaS

对于很多企业来说，尽管对于云计算的兴趣并不是来源于公有云的使用，而是来自私有云的使用。这其中包括私有平台即服务（PaaS）的使用，相比较而言，这个概念比较新。在这部分中我们也将着重介绍私有 PaaS 的概念以及作用，同时介绍如何选择最佳的私有 PaaS 产品。

- ❖ [搏击私有 PaaS 概念](#)
- ❖ [私有 PaaS 拯救企业治理和云安全](#)
- ❖ [改善云体验：如何选择最佳私有 PaaS 产品](#)

## PaaS 采用会在 2014 年持续攀升？

平台即服务在今年滋生出很多话题，而且 2014 年其采用可能持续攀升，但是从长期来看，这个分类将会越来越难在日渐模糊的云服务类型中定义。

最近的云市场调查显示，IT 专业人士在新的一年中对于平台即服务（PaaS）的兴趣增加了。TheInfoPro 在 2013 年七月进行的调查有 97 位受访者参与，18%的受访者表示其 2014 年在 PaaS 上花费的费用将会比 2013 年高。只有 2%预测会回落。同时 9%表示会保持原来的态势。

TechTarget2013 年秋季的 Cloud Pulse 调查中有 119 为受访者参与，31%指出其基础架构的 1%-25% 都基于 PaaS。在更小的一组人群中，56 位受访者中的一半都预测其基础架构的 1%-25%在未来半年会运行在 PaaS 之上。

调查的结果显得内容对于很多厂商都比较有利，尽管他们可能更喜欢知道两份报告的不同之处。通常而言，IT 专家更喜欢微软的 Azure、Salesforce.com 的 Force.com 以及亚马逊的 Web 服务。

### PaaS 采用源自开发者生产力

从一种较高的水平来看，IT 专家寻求能够为开发者提供一种高度自动化的环境，这种环境中每一种元素，比如补丁、补丁包和服务器升级都可以抽离开，应用服务库易于朝如到新开发的应用中。

“管理多种服务器和管理一个数据库会产生大量的开销，削减了开发者编写软件的时间，” Dan Sullivan 说道，他是云计算咨询公司 DS Applied Technologies 的负责人，“在编码和管理服务器或者数据库之间做一个选择的话，据我所知大多数开发者更喜欢编码。”

移动开发将会是另一个巨大的 PaaS 驱动，Sullivan 说。比如，亚马逊、谷歌、微软和 Rackspace 都在推出面向移动开发的服务套件，具备了推送消息和地理编码的功能。“这些都是移动应用中常用的功能，但是并不能实际帮开发者区别其产品，” Sullivan 说，“这也使得面向移动的 PaaS 更具有活力。”

开发者的时间是衡量 PaaS 采用的一个通用指标，但是云咨询师表示财务节省也有是显著指标。“我们有个客户去年就具体的而应用在 Force.com 上进行了概念验证，”云咨询公司 Cloud Technology Partners 的

高级副总裁 John Treadway 说道。这个客户对比了使用硬件堆栈和本地软件开发一个新的 Java 应用和使用 Force.com 开发。“并非采用六个月和数十万的经费，而是在两周内实现，其成本为 8 万。”

### 不同模型的 PaaS 交付混乱

然而在任何关于 PaaS 的讨论中，双方协定术语定义都令人头痛。

在市场调研中，三分之二的顶级 PaaS 厂商都提到这个问题。Force.com 旨在快速的开发者，这些开发者用可视化工具工作，通过范式定义 workflow，尽可能少的进行编码。相反，微软的 Azure PaaS 产品则更加面向代码，目的在于让程序员使用 Visual Studio。“这还是一个不成熟的市场，” Treadway 说道，“没有标准。”

分析师表示，过去很多观点可能会说 PaaS 是个不错的注意，具体到实际的提供的功能和可以担负的工作复杂问题就变得更为复杂。“一般来说谈到 PaaS 都是很复杂的，而且实际的成本会花在哪里？” John Rymer 说道，他是 Forrester Research 的分析师，“选择大相径庭。”

### “IaaS-plus” 和 SaaS 让事情更糟糕

随着类似亚马逊弹性计算云( EC2 )这样的服务激化了这个领域的矛盾，关于如何定义 PaaS 和未来的 PaaS 采用都导致了不同的讨论。EC2 是基础架构即服务 ( IaaS )，但是可以提供自动扩展、负载均衡以及将很多通常和 PaaS 相关的服务。然而，客户还需要访问并且对底层服务器负责，这些内容又通常不是 PaaS 的特性。

Forrester Research 提出了这个术语：“IaaS-plus”，而且这个术语只是让围绕 PaaS 采用的讨论更加复杂化。实际上，一些产业观察者预测 PaaS 市场在明年会变得很困难，因为 IaaS-plus 即 IaaS 厂商会增加增值服务，更多的软件即服务 ( SaaS ) 厂商也在针对开发者提供应用平台。

“现在这些已经发生了，” Development Partners 软件公司的首席分析师，同时也是 Windows Azure MVP，Bill Wilder 说道，“云平台服务已经发展成为一种工具箱，你可以使用，满足 PaaS 或者 IaaS 需求，或者是 API 可用的 SaaS。”

今年，风险投资不断催化 PaaS 市场，但是也有 VC 在日渐模糊的云服务分类前退缩不前。

原文链接：[http://www.searchcloudcomputing.com.cn/showcontent\\_78877.htm](http://www.searchcloudcomputing.com.cn/showcontent_78877.htm)

(来源：TechTarget 中国 作者：Beth Pariseau 翻译：张培颖)

## 云市场观察：平台即服务厂商

最初，平台即服务厂商根据自身支持的语言确立自己的市场差异性，比如支持 Java 或者 .NET，但是最终他们还是要不断发展支持多种语言，最后还要用基础架构即服务支持数据存储、消息服务、应用服务和可迁移性。

市场上为开发者提供了各种 PaaS。虽然看起来 PaaS 厂商都极为相似，但是却也存在诸多不同。要考虑能控制的类型，如果有的话，开发者就可以在基础架构上进行配置了。理想上，PaaS 厂商管理了实施的所有细节，但是有时候开发者却需要更多的控制。因此考虑厂商支持的服务极为重要，比如关系型数据库和 NoSQL 数据库。如果关注厂商锁定，一旦准备在某一个厂商的 PaaS 产品上开发了应用，就要评估转移到其他的 PaaS 厂商是否困难。

要跟上市场的变化，在选择 PaaS 平台时，有一些关键性能需要注意。下面我们就来具体看一下。

### IaaS 厂商支持 PaaS

很多 IaaS 云厂商逐渐转向提供 PaaS。亚马逊 Web 服务( AWS )弹性 Beanstalk 就是混合了 IaaS 和 PaaS 模型的工具；提供了自动化扩展、负载均衡和应用监控。

AWS Elastic Beanstalk 可能对于那些喜欢 PaaS 优势的开发者有吸引力，这些开发者偶尔需要配置服务器、存储或者其应用使用的其他服务。不像那些把基础架构与客户隔离开的 PaaS 厂商，AWS 为 IT 提供了运行应用的资源的完全控制，包括弹性计算云( EC2 )实例 和简单对象存储服务( S3 )。AWS Beanstalk 支持 Java、.NET、Python、Ruby、Node.js 和 PHP。

Google App Engine 是为了在 Google 基础架构上运行 Python、Go 或者 Java 而设计。应用运行在一个受限制的沙盒中，因此现有的应用可能需要大量改变。比如，应用不能编写文件系统，且必须使用 AppEngine、缺失 NoSql 模式数据库，存储稳定数据。开发者也要访问基于 MySQL 5.5 设计的 Google Cloud SQL 和对象数据存储 Google Cloud Storage，Google Cloud SQL。

微软 Azure 最初作为 PaaS 厂商，随后提供 IaaS。除了支持多种语言，微软 Azure 还提供移动设备应用后端支持的专业服务。SQL 数据库和报告服务也包含在 PaaS 产品中。在思考大数据时，微软可能并不是我们

第一个想到的云厂商，但是它和 Hortonworks 是合作伙伴关系，在 Azure 中支持 Hadoop。Azure 的这种结合易于使用且微软对于非微软技术的支持也会让其 PaaS 吸引更多的 .NET 开发者。

## PaaS 厂商静观其变

除了 IaaS 厂商提供 PaaS 特性之外，也有各种各样单纯的 PaaS 厂商。Heroku 提供 PaaS，支持通用语言，比如 Java、Ruby、Node.js 和 Python，以及功能性编程语言 Scala 和 Clojure。Heroku 抽象虚拟机（VM）实例，成为一种结构，称之为 dynamo，dynamo 被描述为运行单一命令的容器。Heroku 基础架构管理你的应用的 dynamo 数量和分布。除了计算服务之外，Heroku 用户可以利用数据库、队列和性能监控服务。

红帽 OpenShift 是一种 PaaS 应用框架，支持通用语言，但是也实现了定制化的功能，允许用户安装其他语言。比如 AWS Elastic Beanstalk，OpenShift 为客户提供了更加明显的定制化控制。为了在 OpenShift 中运行应用，开着创建一个配置文件，指定使用的组件，比如 Python 和 MongoDB，以及应用代码位置。OpenShift 可以根据负载自动化扩展，或者系统管理员可以人工按需扩展。

CloudBees 的 PaaS 定位是支持敏捷开发者和 Java 虚拟机（VM）工作，比如 Java、Grails 和 Scala。Jenkins 用来构建、测试和监控应用，提供对于持续集成的支持。CloudBees 为关系型数据库提供可管理的 MySQL 服务支持，为使用其他服务厂商的合作伙伴提供 CloudBees “生态系统”，包括进行版本控制的 CloudForge、性能监控的 AppDynamics、兼容 CouchDB 云数据库的 Cloudant 以及支持 MongoDB 的 MongoHQ。

PiCloud 采取与其他厂商不同的 PaaS 方法。这家 PaaS 厂商关注计算密集的 Python 应用。开发者可以使用 PiCloud 库导入代码到其应用，这种功能可以将计算密集功能推入到云端，运行代码，而不是在本地设备上运行。这种放到最小化了开发者工作方法的改变，允许他们通过添加一些代码到其程序中，从而利用 PaaS。

原文链接：[http://www.searchcloudcomputing.com.cn/showcontent\\_72748.htm](http://www.searchcloudcomputing.com.cn/showcontent_72748.htm)

(来源：TechTarget 中国 作者：Dan Sullivan 译者：张培颖)



## 当 IaaS 不再 迁移至 PaaS 云前途几何

尽管几乎所有的云用户都是从使用基础设施即服务(IaaS)开始实施云计算的，但是那些在相同操作系统和中间件上运行大多数应用的用户则在不久之后认为平台即服务(PaaS)能够更好地满足他们的需求。PaaS 云为承诺在一个单一操作系统和中间件上使用云计算的用户提供了更低的软件使用许可和运行成本，因为云平台能够将这些组件都囊括其中。

已经完成 IaaS 至 PaaS 云转移的用户报告说，它需要用户在三个方面做好规划：操作系统和中间件的兼容性、部署和集成实践，以及运行与管理。使用者还建议，所有潜在的云计算用户都应当在他们规划的早期就审查 IaaS 至 PaaS 的迁移，甚至是在他们确定做出迁移决策之前。最后，这些早期用户表示，通常云计算服务类型变更的唯一选择就是使用多厂商的云计算，而这一点甚至比从一个云计算平台转为另一个更困难。

### 选择 IaaS 和 PaaS 厂商

一般来说，IaaS 对中间件选择的容忍度更为宽裕：任何你选择组成机器镜像的中间件都可以运行。一些 IaaS 云计算厂商可能会限制用户选择操作系统，或者至少根据你所选择的操作系统要求管理或部署选项的略有不同。与之相反，PaaS 一般会指定操作系统和中间件的大部分。对于希望保留迁移至 PaaS 选择的用户来说，首先选择 PaaS 云 PaaS 云厂商然后根据这个决定选择 IaaS 操作系统和中间件软件是非常关键的。否则，对于你所选 PaaS 平台，如果不重写或重新授权软件使用，就可能无法完成从 IaaS 迁移至 PaaS 的工作。

大多数 PaaS 云厂商将提供一个 IaaS 模式，这就意味着他们的“平台”将运行可托管其他操作系统和中间件的虚拟机。例如，微软公司的 Windows Azure 就是这样的。选择一个能够运行 IaaS 的 PaaS 厂商对于用户将是意义非凡的，要知道你仍然能够在 PaaS 厂商的 IaaS “容器”中运行操作系统和中间件，但这些操作系统和中间件在同一厂商的 PaaS 模式下则是无法运行的。托管 IaaS 的模式将有助于解决你在变更厂商时所遇到的问题，同时可能减少在迁移过程中的部署和运行差异，但它也有它自己的问题。

### IaaS 迁移至 PaaS 考虑云管理

基于云计算应用的部署和集成是运行云计算管理接口的问题。即便在 IaaS 厂商中，这些接口也是各有不同。这意味着，当进行 IaaS 至 PaaS 迁移时，部署/集成工具和实践都将必须有所改变。



在初步的云计算规划中，应寻找同时支持 IaaS 和 PaaS 厂商的工具。但在进行迁移时，应对部署和集成脚本程序做出一些改变。商业工具可能比开源工具更易于被接受，因为大多数 PaaS 平台使用了专用的操作系统和/或中间件，而众多开源产品并没与被更新以适应这类选择。

对于 PaaS 和 IaaS 来说，云应用的长期管理几乎肯定是不同的，因为操作系统和大多数中间件的管理都变成了 PaaS 下云厂商的责任。企业将不会更新这些组件，但是当 PaaS 软件有计划做出重大变更时他们可能仍然需要执行一些回归测试。这就意味着，运行和管理将必须进行变更以便于消除与平台软件更新升级相关的风险，同时应用生命周期管理任务可能需要一些来自于云厂商的外部协作以便于对新的软件版本进行测试和切换。

在部署和与运行/管理任务中，PaaS 可能会带来意想不到的好处，因为大多数公司都是基于在他们大多数内部服务器上已标准化的平台来选择 PaaS 的。在这种情况下，在数据中心的用于部署和管理的工具也同样适用于 PaaS 应用的部署与管理，而一些 PaaS 厂商还将为数据中心和 PaaS 云之间的混合云计算集成提供特殊的工具。通常也可能创建混合云应用以便于从数据中心向 PaaS 通过提供简易的“云计算爆发”以提供优化的性能和可用性。在 IaaS 中也提供了与之相同的功能，但是他们几乎总是需要用户进行更多的集成工作。

### 当迁移工作无法奏效时

不管你为 IaaS 至 PaaS 迁移工作做了多么充分的准备，有可能发生无法正常工作。在大多数情况下，这是因为一个在 IaaS 上运行的应用在软件平台层次上与你的 PaaS 选择不兼容，或者甚至与新的 PaaS 厂商的“IaaS 容器”设施也不兼容。在这种情况下，你必须接受从以下三个选项中做出抉择：寻找相同应用的一个 PaaS 平台版本，在云中放弃使用这个有问题的应用，或者最后一招，使用多个公共云厂商。

如果使用了套装软件，那么第一个选择就是它最适用的；与你的软件厂商核对一下它所能支持的操作系统和中间件清单，并确认你的软件使用许可证是否覆盖了新的配置，即使需要支持额外的成本。如果只有少数应用不与 PaaS 迁移兼容，那么把这些应用在内部进行托管是合理的。当没有应用能够正常运行时，使用多个厂商则成为最后的选择。有一个用户把使用多个云计算厂商描述为“需要不断花钱的礼物”，因为当使用多个厂商的方案时，与部署、集成和管理/运行相关的问题会源源不断的出现。出于这个原因，大多数用户都会把它当作他们最后的选择；所以当真正需要做出决策时，很难找到可用的部署与管理工具，并选择对各种云计算厂商具有最广泛支持的软件包。

原文链接：[http://www.searchcloudcomputing.com.cn/showcontent\\_75191.htm](http://www.searchcloudcomputing.com.cn/showcontent_75191.htm)

(来源：TechTarget 中国 作者：Tom Nolle 翻译：滕晓龙)

## 雾里看花 巧辨真假 PaaS

云计算已经刮起了流行旋风，随着其不断成熟，应用不断增长，新用户涌现。在 IT 业界软件即服务和基础架构即服务愈发如鱼得水。但是因为云洗白和市场不成熟，平台即服务仍旧处在黎明前的黑暗中。

对软件即服务 (SaaS) 这样特定类型的应用，以一种方式访问复杂应用轻而易举，而且无需大量现金支出，只需要很低的管理支出即可。同样，基础架构即服务 (IaaS) 吸引了越来越多的企业，提供了一种访问多种且大量计算、存储和带宽资源的能力，而且可以像本地基础架构那样受到控制，无需先期投入。

平台即服务 (PaaS) 则是完全另一幅光景。主要受到前瞻性的开发者的拥戴，PaaS 的主要价值定位在于增加了生产率以及更快的部署时间。PaaS 也提供了内置的自动扩展和故障恢复，如果开发者想要在其应用中增加这些功能，则不需要学习这些复杂的代码。

“结合了预制 OS 和开发平台，应用部署就变得轻而易举，” Roger Jennings 说道，他是 OakLeaf Systems 的首席咨询师，同时也是一位 .NET 开发者。尽管大多数 IT 人士自然而然的趋向于用 IaaS 满足自身需求，但是在微软的 Windows Azure PaaS 上只需要十分之一的时间即可搭建一个网站。



当今，PaaS 市场只是整个公有云市场的一小部分。但是如果 PaaS 飞黄腾达，很多专家也确信会是这样的结果，它将对 IT 人士产生广泛的影响，他们的角色和职责也会发生转变。

但是市场仍旧处于初期，对于企业 IT 而言，很难预测会有多少应用、什么类型的 PaaS 平台，以及 PaaS 内置的应用是他们可能需要落地的。

## PaaS 剖析

大多数 IT 部门需要做的第一件事就是了解真正的 PaaS 和假冒的 PaaS 之间的区别。

“你否记得我们过去看到的所有的云洗白都是来自基础架构提供商？”James Staten 说道，他是 Forrester Research 的以为分析师，“在 PaaS 领域其实更为糟糕。”

Staten 解释道，他经常看到一些厂商尝试将 IaaS 的上的老一套，在增加一点服务就变成了 PaaS，让开发者和运维人员甘傲困惑。

从核心观点将，真正的 PaaS 平台必须包含抽象的运行时环境、应用服务器、缓存层、整合开发工具，增加自动扩展以及故障恢复功能。用一个老一点的术语——中间件，它可以在公有 IaaS 之上运行，或者通过在本地硬件上运行实现交付。

真正的 PaaS 包括但不仅限于，比如微软 Azure、Engine Yard、Heroku、CloudBees 和 Google App Engine。亚马逊 Web 服务（AWS）的 Elastic BeanStalk，尽管经常说自己是 PaaS，却完全对不起自己的账单，不符合 PaaS 的规则。

“Elastic BeanStalk 就一个如何在 IaaS 上部署复杂应用的脚本，增加了一些故障恢复和可扩展的脚本，”Staten 说道。相反，真正的 PaaS 并不提供脚本，但是暴露出可以为应用所调用的组件。

真正的 PaaS 和假冒 PaaS 之间的区别不止是学术上的差异；对于开发团队会产生实际的影响。对于那些相信自己就是在 PaaS 上开发的开发者来说，期望是“我编写我的代码，我来部署，可以自动化扩展而且能够自动化实现故障恢复，”Staten 说道，而假冒的 PaaS 则是“应用无法真的扩展和实现故障恢复。”

### PaaS 平台来源

很多 PaaS 平台都起源于具体的编程语言。随着时间推移，大多数 PaaS 厂商都开始超越单一的语言，提倡多语言。然而，值得我们铭记的是，要找到适合你的环境的最适合的选择。下面是一些 PaaS 厂商以及他们的原始开发环境的不完全列表。

AppFog——PHP

CloudBees——Java

CloudFoundry——Ruby on Rails

Engine Yard——Ruby on Rails、PHP

Google App Engine——Python

Heroku——Ruby

Microsoft Windows Azure——.Net

原文链接：[http://www.searchcloudcomputing.com.cn/showcontent\\_75379.htm](http://www.searchcloudcomputing.com.cn/showcontent_75379.htm)

(来源：TechTarget 中国 作者：Alex Barrett 翻译：张培颖 )

## 遗留开发工作入 PaaS 你考虑好了吗

相比于管理你自己的发展基础设施，平台即服务提供了很多优势，你可以有更多时间专注于设计和编码。然而对于新项目来说，PaaS 可能是首选，但是 PaaS 可能不适合现有的、遗留的开发工作。

将遗留的开发项目转移到平台即服务（PaaS）之前，需要考虑五件事。

### 如何使用 PaaS？

不同的公司，PaaS 起到不同的作用，来适应他们的 IT 环境和目标。首先，你必须找出如何将 PaaS 融入到你的组织中。对于一些服务，你可以轻松地将计算移动到云，同时保持当地资源的其他功能。例如，Pi 云提供应用接口(API)，将你的本地 Python 代码复制到云，并在云中运行，而你的开发工具和代码存储库可以保留在本地。

另一种方法是开发本地资源与测试 PaaS 产品。当你运行大型测试套件或者需要单独的共享资源实例来进行测试时，这种方法很有用。你也可以通过使用云集成开发环境(IDE)来开发云。基于浏览器 IDEs 所支持的功能——一些功能无足轻重，并且可能不包括所有功能，你可能已经习以为常。

### 软件开发实践和工具与 PaaS 是否匹配？

如果你使用 Git、SVN、Ant 或者 Maven 等工具开发软件，想一想，在 PaaS 使用这些工具有多么容易。PaaS 的版本控制系统与本地资源的版本控制系统，没什么不同。然而，重写构建脚本的任务尤其重要。如果你的通常做法不那么正式，或者使用国产工具，那么就要考虑如何适应 PaaS 环境，并且审查你的代码和文档的访问控制策略。检查你的 PaaS 提供商的访问控制机制，以确保你可以控制你需要的访问方式。

### 需要整合本地资源吗？

应用通常需要与其他应用或共享资源整合，如企业数据库。在这种情况下，需要了解如何访问这些 PaaS 资源。如果你使用一个内部应用，该应用实现 Web 服务 API，并且服务于外部客户端应用，你就能转移到 PaaS。如果安全是一个关注的问题，并且只有虚拟私有网络(VPN)的客户端应用允许访问本地资源，就要尽早测试

PaaS 的 API。如果你需要实现 VPN 功能，并且你的 PaaS 提供商无法满足你的需要，你可以考虑用基础设施即服务(IaaS)云代替它。

### 开发堆栈是否完全支持？

PaaS 从单一语种的平台到支持一系列的语言、数据库和其他服务的平台，快速发展。找出一种 PaaS，能够支持你全部的应用堆栈，并且减少 PaaS 采用的阻碍。例如，如果你用 Java 开发，使用 Jenkins 来持续集成，那么，CloudBees 可能是一个不错的选择。如果你需要支持 Ruby 和 Node.js，可以考虑 Engine Yard。如果你正在构建 NoSQL 平台，Red Hat's PaaS，OpenShift，可能是个不错的选择。

### 开发阶段是什么？

你在开发进程中所处的位置，对本地开发转移到云开发的成本和效益有重大影响。通常，项目越到后期，在开发环境和工具上的投资也就越多。随着项目的进展，你需要创建更多的软件，必须移动到 PaaS 平台，该平台增加了打造一个开关的成本。

PaaS 的优势大于完成转移所需的成本和时间时，将正在进行的项目开发，转移到 PaaS 平台合乎情理。转移到 PaaS 后，你的管理服务器、操作系统和其他基础设施将如释重负。如果需要扩展应用，最大的回报在于可以避免耗时且具有挑战性的工程问题。

在云中运行你的应用，可以节省更多的时间和成本，转移到 PaaS 值得一试。

原文链接：[http://www.searchcloudcomputing.com.cn/showcontent\\_74576.htm](http://www.searchcloudcomputing.com.cn/showcontent_74576.htm)

(来源：TechTarget 中国 作者：Dan Sullivan 翻译：滕晓龙)



## 遗留开发进入 PaaS 五要素

平台即服务提供了管理企业自己的开发基础架构的优势，允许企业有更多时间关注于设计和编码。虽然 PaaS 可能是新项目的优先选择，但是对于现有、遗留开发工作可能并不合适。在将遗留开发项目转移到平台即服务（PaaS）之前，需要认真考虑如下五件事。

### 1、如何使用 PaaS？

不同的公司对于 PaaS 有不同的使用，来适用其 IT 环境和目标。首先必须了解如何将 PaaS 合并到企业中。一些服务可以将一些计算轻松转移到云端，同时维护本地资源上的其他功能。比如，Pi Cloud 提供了一种应用程序接口（API），可以复制本地的 Python 代码到云端，并在云端运行，同时开发工具和代码库则保留在本地。

另一种选择是用本地资源开发，用 PaaS 产品测试。在运行大型测试套件或者需要共享资源的单独实例进行适当的测试时，这种方法十分有用。也可以通过使用云集成开发环境（IDE）将所有的开发转移到云端。通过基于浏览器的 IDE（一些是轻量级的）审核性能支持，可能会不适应所有的性能。

### 2、软件开发实践和工具是否适合 PaaS？

如果使用 Git、SVN、Ant 或者 Maven 这样的工具开发软件，要考虑在 PaaS 中这些是否能够轻松使用。在一个 PaaS 版本控制系统中工作与本地资源工作并没有过多的区别。然而，重写构建脚本则是一种更加具体的工作。如果没有过多正式的实践或者使用自主研发的工具，要考虑这些工具如何适用于 PaaS 环境，且针对代码和文档，审核访问策略。检查 PaaS 提供商的访问控制机制，确保能够按照需要的方式控制访问。

### 3、是否需要集成本地资源？

应用通常需要整合其他的应用或者共享资源，比如企业数据库。在这样的例子中，理解如何从 PaaS 访问这些资源。如果使用通过 Web 服务 API 实现的内置应用，且已经服务于外部客户端应用，应该能够转移到 PaaS。如果安全是关注要点，且只有在虚拟私有网络（VPN）上的客户端应用能够访问本地应用，就要在开发的早期测试 PaaS API。如果需要考虑 VPN 功能且 PaaS 提供商并不符合需求，可能需要考虑基础架构即服务（IaaS）云来代替。

#### 4、开发堆栈是否完全支持？

从单一语言平台到支持大量语言、数据库和其他服务的平台，PaaS 产品快速成熟。要找到能够支持全部应用堆栈的 PaaS 选择，减少采用 PaaS 开发的障碍。比如，如果用 Java 开发，用 Jenkins 进行持续集成，可能 CloudBees 会是个不错的选择。如果需要使用 Ruby 和 Node.js，Engine Yard 更适合。如果在 NoSQL 平台上构建，红帽的 PaaS——OpenShift 就是上佳选择。

#### 5、开发期是怎样的？

处于开发期的那个阶段会严重影响从本地开发转移到云端开发的成本和好处。典型的，你的开发周期进行的越长久，就会在开发环境和工具中放入更多的投资。按照项目进度情况，需要创建更多软件且比如转移到一个 PaaS 平台上，增加了转换的成本。

把一个进行中的项目开发转移到 PaaS 平台，只有在 PaaS 成本权重优势高于迁移所需要的潜在时间时才有意义。在转移到 PaaS 之后，就可以释放服务器管理、操作系统和其他基础架构的管理了。如果需要扩展应用，避免时间消耗以及工程师问题，巨大的回报就出现了。

在云端运行应用，且以最小的技术介入进行扩展，就能节省时间，确实值得努力转移到 PaaS。

原文链接：[http://www.searchcloudcomputing.com.cn/showcontent\\_72127.htm](http://www.searchcloudcomputing.com.cn/showcontent_72127.htm)

(来源：TechTarget 中国 作者：Dan Sullivan 翻译：滕晓龙)

## PaaS 应用可移植性：问题与解决方案

很少企业能够忍受将自己的应用开发实践同单一厂商绑定，但是很多企业又在不知不觉中将应用开发和唯一云提供商绑定在一起。

随着云越来越具有竞争力以及一些提供商具有长期稳定性风险，开发者必须理解如下的事情：即平台即服务（PaaS）是圈套，因为平台特性支持不均衡，一些 PaaS 提供商比其他的提供商造成了更大的风险，所有的 PaaS 选择通过正确的项目步骤都可以变得对于可移植性更加友好。

即便是在云应用的初期，PaaS 服务用户提出了其应用的可移植性问题，而不是向 PaaS 提供商提问，而是在从第一个提供商转向不同的提供商时提问，或者甚至是迁回数据中心时才提问。在一些案例中，这种转换要求软件的主要改变，而且导致项目滞后，甚至生产力损失。主要是两个具体问题导致的，开发者必须在为 PaaS 创建可移植性应用时解决。

PaaS 可移植性的第一个问题是缺少 PaaS 提供商之间一致的平台定义。使用基础架构即服务（IaaS），开发者同裸机共事，可以提供应用需要的所有系统软件。这种平台的问题就是可移植性，因为从一个 IaaS 提供商迁移到另一个时，甚至会移植到本地内部的虚拟机。使用 PaaS，提供商选择自己支持的操作系统和中间件元素，如果提供商做出不同选择，随后应用在这些不同领域使用的性能就不能迁移。如果一些 PaaS 性能是提供商自定制的，将应用迁移会本地相同的平台甚至更难。

对于这个可移植性问题的最佳解决方案就是创建一个平台参照点。PaaS 服务通常都是围绕一个操作系统提供的（比如，Linux、Windows），一群中间件用于通信和数据库服务，还有管理和开发工具。同时多种云提供商可能提供相同的基础平台，变换了中间件和工具，因此绘制你优先的平台类型性能图表很重要，高亮标出哪些不统一支持的性能/工具。避免这些性能和工具，就能避免可移植性问题。

第二个问题就是缺少可靠平台替代提供商。当今 PaaS 服务通常提供两种形式。首先，平台“所有者”（微软的 Windows/Azure 为例）可能会对有效的服务器平台提供一个云版本。在这种案例中，第一类 PaaS 厂商的优势可能也会阻碍竞争力，尽管平台提供商考虑到了（微软最近变更了 Windows Server，促进非微软 Windows PaaS 产品。）

当一个支配性的提供商压制 PaaS 竞争力，云用户可用的唯一替代物就是使用 IaaS 服务，包括其机器映像中的 PaaS “平台” 部分。如果这样做，关键就是确保所有的 PaaS 性能对于本地服务器配置可用。主要平台提供商（比如微软、IBM、HP 或者甲骨文）的风险就可能仅仅避免小型 PaaS 业务，但是在这些地方小型提供商就是 PaaS 唯一的选择，计划 IaaS 退路是个谨慎的过程。

第二个问题的解决方案就是适配器设计模式。云端应用必须参照可能不是所有提供商都可用的服务时，封装参照到更高层的软件元素中（遵循适配器设计模式原则，共有面向对象设计）并转换通用需求到 PaaS 服务需要的具体形式。

例如，假设一个应用为亚马逊的 Redshift 仓储服务开发。然而 IaaS 服务和广泛使用的亚马逊 EC2 兼容，其他的 IaaS 提供商不提供 Redshift，且应用是为了 “miniPaaS” 编写，EC2/Redshift 在不变更所有的项目参照到 Redshift 的情况向下就不能迁移。一个开发者编写了一个小型的软件对象，称之为 “Warehouse”，用于代替 Redshift 应用程序接口（API）。Warehouse 内部，代码可能改变数据库操作参数，成为 Redshift 格式，随后调用 Redshift。如果应用随后转移到一个不支持 Redshift 的提供商，Warehouse 就要变更来执行正确的数据库访问 API 需求来模拟。Warehouse 对象单一的变更就可以进行应用迁移。

这种基于抽象的机制也适用于管理应用可移植性问题，即需要在一个平台性能上构建一个云应用，竞争分析显示并没有广泛的支持。关键在于确保至少要有合理的方式在多种平台上提供同样的性能/功能，即便相同的 API 不能跨平台工作，比如上面提到的 Warehouse 例子，在一个或者更多的替代平台上根本没有可比较的服务，然后适配器设计模式机制在它们之中并不能轻松的支持可移植性。

久而久之，PaaS 最大的可移植性风险并不是正常的 PaaS 平台，比如 Azure，但是随着 IaaS 服务通过增加一些性能发展成 PaaS 服务，比如亚马逊的 Redshift 或者缓存服务。这些平台的很多用户永远不会把自己看作是 PaaS 用户，可能在他们第一次尝试将应用转移到另一个提供商时就会被不可移植性攻其不备。少量的预先计划可以防止主要的问题，经验也能够协助云专家要理解可能让 PaaS 可移植性成为性能区别持久压力。

原文链接：[http://www.searchcloudcomputing.com.cn/showcontent\\_71941.htm](http://www.searchcloudcomputing.com.cn/showcontent_71941.htm)

(来源：TechTarget 中国 作者：Tom Nolle• 译者：张培颖)

## 搏击私有 PaaS 概念

公有云计算的核心好处在于可以在众多客户之间共享大量硬件和软件。这些 IT 资产可以按需配置，业务部门只需支付使用的服务。云计算系统也可以按需扩展，因此企业可以在需要时扩展大量计算能力。

对于很多企业来说，尽管对于云计算的兴趣并不是来源于公有云的使用，而是来自私有云的使用。这其中包括私有平台即服务（PaaS）的使用，这也是我一直致力解决的概念，多年前就已经出现在我的技术雷达图中。

在对比私有 PaaS 和公有 PaaS 之前，理解平台即服务作为一种技术的概念至关重要。PaaS，不管是公有的还是私有的，都提供了基础开发、部署和运营能力，支持设计、编码、测试和基于云的系统的管理。这项技术的价值在于其为企业提供了一种替代物，来支持当时流行的任何开发工具，包括购买运行它们的服务器。

使用公有 PaaS，企业 IT 可以简单的选择一个公有 PaaS 提供商，订阅服务并开始工作。使用私有 PaaS，会有更多的步骤，而且要购买更多的东西。企业 IT 在现有硬件或者新硬件上，安装私有 PaaS 软件，内部开发者将私有 PaaS 像公有 PaaS 服务那样使用。

很多企业 IT 专家退回使用公有云，包括公有 PaaS，这其中原因多种多样，控制、安全、法规遵从等等，在一些案例中，他们都有各自的道理所在。

在过去几年中，大多数公有 PaaS 来源于影子 IT。PaaS 为开发者提供了访问他们所需技术的能力，在构建核心业务应用的同时，能够进入 IT 技术雷达圈。很多企业 IT 正在尝试在其控制下运行那些“劣质云”应用。

基于上面这些原因，私有 PaaS 市场在过去几年中得以发展。市面上有很多种私有 PaaS 技术，包括 Apprenda、ActiveState Stackato、红帽 OpenShift、CumuloLogic 和 CloudBees。所有的这些技术都采用不同的方法来提供开发技术，包括语言、部署模型、测试和操作。然而，在提及私有 PaaS 的优势时，他们又都大同小异。

**私有 PaaS 的显著优势如下：**

- 迁移至公有云，或者假定能够创建混合云。企业可以在需要额外容量的时候转移到基于公有云的平台，或者可以在私有云和公有云之间分离流程，成为混合云。
- 提供单一企业云开发环境的能力。这意味着在开发操作之上可以进行核心控制，包括开发和部署工具的使用治理。对于标准化企业云环境开发行之有效。
- 能够创建公有云中不存在的企业应用。尽管很多企业 IT 都有点过快地传播了使用公有云的恐惧、不确定和怀疑，但在很多案例中，确实因为一些好的业务、法律、技术的原因，导致企业应用和数据不适合放入公有云中。因此私有替代物的使用有时候是更好的方法，在某种情况下，也是通向公有云的一条途径。

就像所有的技术一样，私有 PaaS 也有缺点。

采用私有 PaaS 的劣势在于无法获得和公有 PaaS 一样的成本优势，主要是考虑到需要购买、配置和操作硬件和软件，这些都要支持私有 PaaS 平台。因此，根本上来讲和传统方法没什么差别。实际上，私有 PaaS 可能会让这件事情变得更加复杂，而且难以管理，因为现有的开发技术没有彻底舍弃。

尽管存在这些缺点，私有 PaaS 仍旧开始扎根。然而，私有 PaaS 技术成功的关键在于来年是否能够证明其自身价值，或者只是作为现有内部应用开发方法的替代物。私有 PaaS 还需要证明其对于公有 PaaS 的替代成本效益。

原文链接：[http://www.searchcloudcomputing.com.cn/showcontent\\_72245.htm](http://www.searchcloudcomputing.com.cn/showcontent_72245.htm)

(来源：TechTarget 中国 作者：David S. Linthicum 译者：滕晓龙)



## 私有 PaaS 拯救企业治理和云安全

围绕安全和治理的担忧刺激一些平台即服务厂商开始提供私有基础架构上的产品，而不仅仅是公有云上的。

“不管是处于数据主权还是治理原因，一些客户告诉我们 PaaS 必须放在他们的防火墙之内，” ActiveState 的 Copeland 说道，“他们说，别想把他们的数据放在公有云上。”

他解释道，实现内部的平台即服务（PaaS）也为未来公有 PaaS 的使用奠定了基础。惠普在企业内部使用 ActiveState Stackato，并决定将其作为运行在 OpenStack 之上的惠普云应用平台即服务的一部分发放许可，目前仍是测试版本。“这个产品为我们提供了一种通用的基础架构堆栈，我们可以在私有云和公有云平台上进行部署，” Dan Baigent 说道，他是惠普云服务业务开发高级总监。

就算公有 PaaS 并不是企业的一个选择，提供了私有版本，IT 运维团队可以为开发者提供更好的敏捷性以及自服务，而且要比管理传统开发环境简单很多，Chris Turra 说道，他是 Mozilla（火狐浏览器）的网络操作工程师。

在对开源 Cloud Foundry 进行试验后，Mozilla 选择了 Stackato。同管理非 PaaS 开发环境对比后，Turra 说这种更为简单。“你无须管理独立的结点，而是管理其他结点可以设置的服务，头脑中最初都有一个概念，”他说，一旦你克服了，“设置就不是那么琐碎了，变得简单。”

总而言之，实现 PaaS 支撑了操作团队任务，Turra 说道，“我们需要平台工程师。我们想要开发者可以工作效率更快。”

Engine Yard 是一家提供私有 PaaS 实现的 PaaS 厂商。

构建内置 PaaS 不仅仅可以加速开发，还可以服务其他的使用者，Forrester 的 Staten 说道，将公有云部署应用带回到企业内部，操作团队也要管理公有云和私有数据中心之间的集成点。从这一点上看，“我们并不期望大量的私有 PaaS，大多数都是混合的，” Staten 说道。

即便提供的 PaaS 服务对于你的企业有意，但是也别卷入舆论的宣传中。基础架构即服务（IaaS）也好，PaaS 也好，市场总归是要不断发展的。可以对企业计划私有和公有云的组合，但是千万别单纯的取代了基础架构即服务，相反相互补充才是正确的做法。

原文链接：[http://www.searchcloudcomputing.com.cn/showcontent\\_75381.htm](http://www.searchcloudcomputing.com.cn/showcontent_75381.htm)

(来源：TechTarget 中国 作者：Bill Claybrook 译者：滕晓龙)

## 改善云体验：如何选择最佳私有 PaaS 产品

基础架构即服务 (IaaS) 产品通常更易于满足企业的计算需求，因为只提供了虚拟机，也可能增加了一些数据库特性。不幸的是，IaaS 用户发现自身在部署和集成其云应用和其内部 IT 时，正面临着新的运营挑战，极少数企业期望将一切都转移到云端。

现在新的私有平台即服务 (PaaS) 工具可以有效解决这个问题。这些工具不仅旨在提供所有操作系统和中间件所需的应用，而且提供了一种“运营件 (operationsware)”。正确的私有 PaaS 产品可以让企业的云体验更为积极，而且帮助企业更好地符合其业务目标。

过去，术语 PaaS 通常涉及到操作系统和中间件。然而现在，这个术语适用于安装在私有云或者公有云之上的任何形式的工具，可以通过软件应用程序接口 (API) 提供应用服务。现在 PaaS 平台提供了安全、移动化、法规遵从、治理、运营、编制、数据库，甚至是应用开发服务。他们可以在企业自己的数据中心运营，而且大多数情况下，也可以在公有云或者私有云中运营，这些都让 PaaS 成为一种新奇且具有价值的传统中间件和云的混合体。

### 选择正确的 PaaS 产品

做出最佳的私有 PaaS 选择的第一步就是确定每一个竞争者和企业的其余部分的技术的兼容性。并不是所有硬件、软件和云环境都可以为任何既定的 PaaS 产品所支持，因此企业应该详细清理其 IT 环境，并且确定将要运行什么，以及将要在哪里应用新的 PaaS 选择。IT 团队要消除不支持其 IT 组建混合的包，或者是不够成熟，无法支持一些重要系统或者软件的部分。理想上，企业应该寻求客观的范畴，用来验证其提供商的稳定性和支持程度。

使用 PaaS 工具通常要求应用、应用生命周期管理和软件实践和流程做出最低限度的改变。出于这种原因，企业应该小心找出并应用最全面的可用工具；用这种方法会导致他们浪费时间整合无价值或者是浪费成本的特性。最佳的做法是：关注当下的痛点（大多数企业的列表都是与安全或者法规遵从和治理），并识别出能够最有效解决这些问题的 PaaS 产品。

另一方面，IT 团队应该要着眼于当下的需求之外，确保其 PaaS 选择拥有适用于未来需求的特性。否则，他们会发现在一年后可能就要被迫改变其方法。

在回顾痛点列表以及 PaaS 产品时，IT 团队可能发现他们的问题在运营和开发中间分开。这意味着他们需要为应用创建一致且有效的运营框架。并且同时支持应用开发来解决移动和其他问题。PaaS 工具倾向于按照功能线划分，决策制定者可能发现最佳的 PaaS 选择实际上就是数个不同的 PaaS 平台，每一个符合一套需求支持。最终，能够轻松整合其他的 PaaS 元素成为选择正确的私有 PaaS 产品的高度重要因素。

大多数企业发现只有三或者四个 PaaS 产品可以通过筛选的初级阶段。这个简短的列表随后要经受审计/测试，可以通过两个步骤进行处理：现场复查使用以及应用引导测试。

### 缩小 PaaS 选择范围

几乎所有的 PaaS 提供商都乐于提及以前的购买过其产品的客户，现场复查包的安装以及运行了一个或者多个账户的客户更是有帮助。当 PaaS 的目标之一变成系统化安全或者治理时尤为如此。在这些领域通过简化复查文档，几乎不可能回顾特性和功能；相反，需要同用户对话，看看他们已经整合了多少 PaaS 特性到自己的运营当中，以及具体的 PaaS 选择工作的如何。在现场复查中，IT 团队应该记录下情况以作进一步检查。这些对于企业的引导应用都是聚焦点。

令人意外的一大部分企业没能引导测试 PaaS 工具。一些越过了同厂商的试验阶段，但是即便有这些也没能揭露平台的实际问题，因为试验阶段并没有将软件提供给全范围的需求使用。构建引导测试计划需要从需求开始，而且假设在评估阶段就要记录起来，最后增加到现场复查中。内部复查应该确立引导测试运行多久可以却信心所有相关的问题都被发现了。

引导测试是一个学习新的 PaaS 部署的理想场所。一个好的产品可以互相协调并且在其特性间协作；大多数情况下可以将特性组成一个功能分类，可以作为一个单元进行部署，通过不断改变程序避免迷惑操作人员。安全和治理控制通常都最先部署，随后是集成和管理/编制。引导测试将帮助发现依赖性导致的问题。

PaaS 产品创造了一个有效的超级中间件层，其优势和限制性将会任何运行在其上的应用或者服务。最佳的建议：认真制定 PaaS 选择；重新选择一个替代方案艰难且昂贵。

原文链接：[http://www.searchcloudcomputing.com.cn/showcontent\\_79506.htm](http://www.searchcloudcomputing.com.cn/showcontent_79506.htm)

(来源：TechTarget 中国 作者：Tom Nolle 翻译：张培颖 )

本期电子书由 TechTarget 出品

