

09 | 什么是PaaS？怎样深入理解和评估PaaS？

2020-03-23 何恺铎 来自北京

《深入浅出云计算》



你好，我是何恺铎。

欢迎你来到我们《深入浅出云计算》课程的第 9 讲，这也是我们 PaaS 篇的第 1 讲。让我们继续精彩的云计算之旅。

PaaS，对你来说也许不是一个陌生的词汇，你可能早已从业界大咖或身边同事的高谈阔论中屡次听到这个字眼。不过，很多人对于 PaaS 服务的评价，可是既有“真香快来”的赞赏，也不乏“大坑勿入”的批评，面对如此两极分化的评价，你估计也有点拿不定主意。这些如雷贯耳的 PaaS 服务们，究竟靠不靠谱、好不好用呢？

作为极客时间的一名“极客”，咱们人云亦云可不行，必须要建立起对 PaaS 的系统认知。从今天开始，我们就来好好地研究一下 PaaS。

让我们先从它的定义说起。

什么是 PaaS?

在 IaaS 篇中，我们主要是侧重于基础设施类的云服务，尤其是虚拟机、云磁盘、云网络等服务。它们的特点是，和传统 IT 基础设施往往有一个对应关系，所以被称为基础设施即服务 (Infrastructure-as-a-Service) 。

今天我们的主角 **PaaS** (Platform-as-a-Service)，则是指云计算提供的平台类服务，在这些平台的基础上，用户可以直接开发、运行、管理应用程序，而无需构建和维护底层的基础设施。

用更通俗的话来说，**PaaS 是在 IaaS 的基础上又做了许多工作，构建了很多关键抽象和可复用的单元，让我们用户能够在更上层进行应用的构建，把更多精力放在业务逻辑上。**

拿房子装修来打个比方的话，IaaS 就好像空空如也的毛坯房，我们还需要操心墙面、地板等基础性工作；而 PaaS 就好比精装修的房子，我们只要搬入自己喜欢的家具（业务逻辑），再适当装饰就可以“拎包入住”，开始美好生活了。

小提示：PaaS 本身也是基于底层 IaaS 构建出来的，使用了云上的各种基础设施。只是这个步骤云服务提供商代替我们用户完成了，还进行了一定程度的封装。

当然，随着 PaaS 服务形态种类的增多、边界的不断扩展，除了那些包含语言运行环境、可编程和可扩展的经典 PaaS 服务之外，还有更多的在云上用来辅助应用构建，或帮助运维的服务，也归入了广义上 PaaS 的范畴。这也是有道理的，因为它们同样是完整的现代应用程序生态的一部分。

PaaS 服务的核心优势是什么？

如果你去回顾云计算的历史，可能会惊奇地发现，PaaS 并不是在 IaaS 已经非常丰富和完善之后才出现的，它们甚至可以说是“同龄人”。因为在云计算发展的初期，不同公司选取了不同的发展路线，有的侧重 IaaS，有的则先押宝了 PaaS 路线。

拓展：不论是 IaaS 还是 PaaS，想要做好都不容易，需要云厂商很大的投入。如果你对于相关的早期历史有兴趣，可以参考我在 InfoQ 上发表的文章 [《激荡十年：云计算的过去、现在](#)

与未来》。

从某种角度讲，PaaS 其实更符合云的初衷，它代表了一种完全托管的理想主义，也更能代表人们对于研发生产力的极致追求。

所以 PaaS 服务的优势，就在于生产力，在于效率，尤其是在搭建和运维层面。比如我们课程后面会讲到大数据类的 PaaS 服务，你可以很方便地一键启动规模庞大的大数据集群，即刻开始运行分布式计算任务。想一想，如果是由你自己基于虚拟机来进行搭建的话，肯定得花上不少功夫。

注：进一步地来说，云上的各种 PaaS 服务是可以互相配合叠加的。运用得当的话，它们联合起来爆发出来的能力会非常强，效率优势会更加凸显出来。

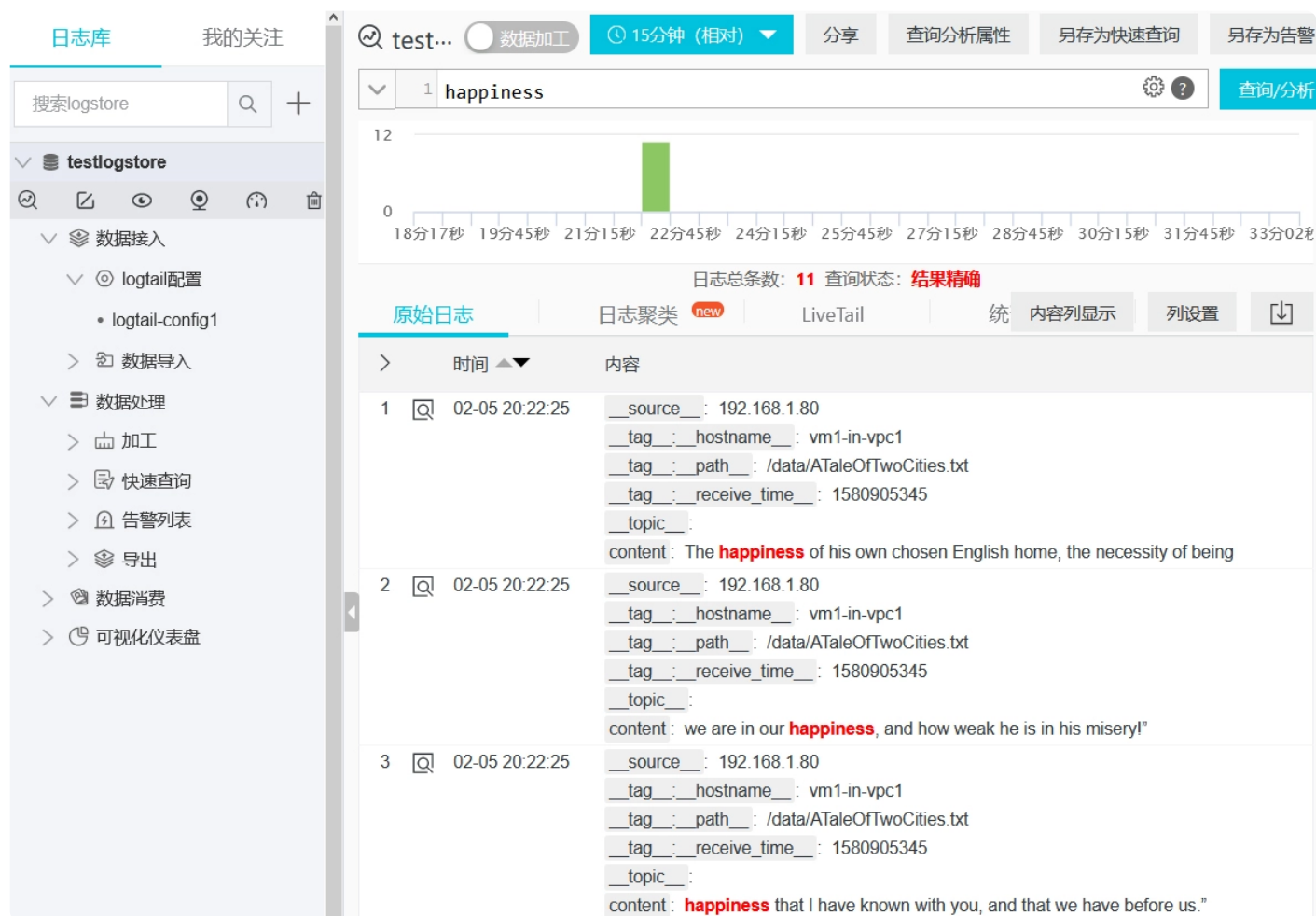
这里我给你举一个例子，来说明一下 PaaS 服务的优势。

日志服务是我们应用程序后端不可或缺的一个组件，通常会组合使用 ELK (Elasticsearch+Logstash+Kibana) 技术栈来自行搭建一个日志存储和分析系统。

而在云上，你可以轻松地找到 PaaS 服务来为你代劳。比如阿里云日志服务，就提供了一个端到端的日志收集、查询、分析和可视化的解决方案。在这个过程中，你不需要搭建和维护任何基础设施，只要按照产品提示进行设置就可以了。

利用阿里云的日志服务，我大概花了 1 分钟的时间，就建立了一个日志服务实例，并让它收集某个虚拟机 / data 目录下的日志文件。随后，我在目录中放置了一本小说《双城记》，很快这个文本文件就被自动传送到了日志服务，并索引起来。然后我就可以利用 PaaS 的功能，来进行各种查询分析了。

下图为我搜索单词 “happiness” 的效果示例：



阿里云日志服务的简单示例

怎样入手学习研究 PaaS?

由于软件构造的复杂性，用户对于可复用组件的需求是非常多的。所以经过多年的发展下来，云上的 PaaS 已经是琳琅满目、种类繁多。我们后面的课程也会陆续地讲解各种不同形式、服务不同目的的 PaaS 服务。

但在那之前，我想告诉你观察和认知 PaaS 服务的方法。这里有几个重要的维度值得你探寻和了解，让你能在清楚了它本身的业务用途之外，还可以洞察这个服务在产品设计和内部实现方面的一些信息。

第一个维度，就是服务是否带有内生的运行环境。

我个人把它称为“承载性”，即服务有没有运行时或执行环境，来承载我们具体业务逻辑的代码或配置。如果有，那么你需要去熟悉它的运行环境，了解它支持的语法，探寻各种参数设置。比如说，Web 服务可能带有 Java、.NET 等的运行时，数据库服务可能会包含 SQL 的执行引擎。

如果没有内含的运行环境，那就说明这个 PaaS 属于“开箱即用”的工具类型，也就是直接依靠自身内置功能来向你提供支持或帮助。这时它功能的完善程度，以及和你需求的匹配程度，就比较关键了。

第二个维度，是 PaaS 服务存在的位置和范围，以及给予你的控制粒度。

这个怎么理解呢？其实就是当你新建一个 PaaS 服务的实例，你一般会需要告诉系统部署的目标位置在哪里。请你注意，这个目标位置的选项是值得玩味的。比如你要仔细看看，这个服务是只能粗放地允许你指定区域，还是可以细化到可用区，以及是否能够设置为部署在具体某个私有网络之内等等。

这个维度的信息，一方面潜在地体现了 PaaS 服务的规模和可用性。比如云存储类服务一般只能让你选择区域，因为它本身冗余性方面的多可用区架构要求，决定了它无法支持指定更精细的位置。

另一方面，这个维度也反映了你对这个服务的掌控程度，你会知道它是否能够和你现有的架构进行深度集成。比如说，你很可能要求数据库 PaaS 服务必须位于你指定的 VPC 内，这样查询流量就能走内网通信，避免对公网暴露数据库。

第三个维度，在于服务是否是“有状态”的，也就是指服务是否具有较强的数据属性。

有些 PaaS 服务本身是无状态的，比如无服务器函数，这意味着它们比较容易扩展和提升规模；有些 PaaS 服务则会保存状态，或者说建立的初衷就是为了维护各种复杂的状态和数据。这对应着 PaaS 在计算存储能力输出上的不同角色和分工。

第四个维度，体现为支撑 PaaS 的虚拟机是否对外暴露，也就是会不会显示在 ECS、EC2 等虚拟机服务的门户列表中。

这是一个很有趣的视角。因为作为 PaaS 实现者，云厂商既可以选择开放，也可以不开放。有时针对同一类的服务，不同的云也可能采用不同的做法，这体现了云厂商在规划产品上的不同思路，也和它们各自的实现原理有关。

通常来说，暴露虚拟机的 PaaS 服务，拥有更高的开放程度，和 IaaS 的结合也更加紧密，甚至能够和其他 IaaS 服务配合联动。在成本方面，这种形式还可以和预付费的虚拟机兼容，让我们享受折扣。

而不暴露虚拟机的 PaaS 服务呢，往往意味着更好的独立性和封装性，说明它不希望你绕开机制来访问虚拟机，比如大多数的数据库服务。还有一种常见的可能是，这个服务需要专用硬件的配合，并非纯粹依赖虚拟机。

好了，有了上面的这些视角，相信你即便是对于一个新的 PaaS 服务，在快速研究之后，也能迅速地把握好要点并进行归类，同时形成清晰的高层次认识。对于它是否适合在你的架构中担任角色，你也会有一个大致的判断。

衡量评估 PaaS 的局限

我们都知道，软件工程的领域没有银弹。强大的 PaaS 也不例外，也有自己的局限。

PaaS 的核心理念在于封装，封装既带来了效率的优势，也同时带来了灵活性上的牺牲。我们需要在内置的设定和选项中开展工作，不能天马行空、随心所欲。PaaS 的应变能力也会差一些，比如当它出现一些 Bug 或者运营事故时，你无法自己动手去解决它，而是需要等待厂商进行修复。

这是 PaaS 诞生以来就伴随着质疑的原因，你的身边可能就有 PaaS 的反对者。有些以前只做 PaaS 的公有云公司也不得不向市场妥协，陆续开始了 IaaS 产品的研发。这和早期云市场的接受程度有关，也和当时 PaaS 自身的成熟度有关。

当然，这里我讲的局限性，不是为了奉劝你远离 PaaS，而是让你能更加客观地看待 PaaS 这个产品形态，更好地评估某项 PaaS 服务是否适用于你的场景。因为 PaaS 在带来巨大效率提升的同时，也的确要牺牲一点“自由”。

这里，我要给你介绍一些检查 PaaS 限制的方法，也是考察评估 PaaS 服务成熟度的重要思路，你需要好好参考和把握。

功能屏蔽：和自建服务相比，你需要研究 PaaS 的封装是否带来了某项功能、部分选项，还有扩展机制的屏蔽或者缺失，以及这些功能对你而言是否重要。

版本选择：你需要检查 PaaS 所提供的软件或运行环境的版本是否丰富，最早和最新的版本各是什么，还有版本粒度是否足够细致等等。我就曾经遇到过，因为所需数据库版本在 PaaS 上不存在，只能选择虚拟机进行部署的情况。

性能极限：确认 PaaS 服务所能够提供的性能极值，包括算力和存储的上限。你要和自己的需求量预测结合起来，避免“上车”后骑虎难下。

更新频率：查看 PaaS 服务的更新日志，了解云厂商和相应团队在这个 PaaS 服务上，是否还在继续做投入，是否在跟进一些最新的技术趋势。

成本陷阱：实际地通过 POC 实验，对 PaaS 服务进行试运行，注意要达到一定的量级，然后仔细查看它对应的账单，看看相关支出是否合理，你能否长期承受。

注：所以对于 PaaS 来说，其实设置界面选项越多往往越好，这也不失为一个甄别产品成熟度的简单办法。你不应该担心产品学习曲线陡峭的问题，这些不起眼的选项很可能在某个时刻被派上用场，发挥关键的作用。

我还是要再次强调，你应当理性地看待 PaaS。它肯定不是无所不能，但也绝非一无是处。更客观地学习了解它，有助于建立你对 PaaS 的理解和信任，在合适场景下，最大化地发挥它的优势和价值。

我个人对于 PaaS 还是非常看好的，它近年来日新月异的发展，已经极大地提升了竞争力。随着大量用户的不断实践和反馈，这些产品也越来越开放，突破了过去很多限制。有时即便 PaaS 相对自建会稍微贵一些，我也会优先选择 PaaS，因为它带来的效率提升，和时间人力的节省，远远超出了贵出的那点价格。

最后我想再补充一点，当云上官方的 PaaS 不足以满足你的需求时，还有第三方 PaaS 是值得考虑的选择，你通常能够在云厂商的各种云应用市场中找到它们。比如说，大数据领域中，炙

手可热的 Databricks 公司，就分别在 AWS 和 Azure 云都上架了自家的 PaaS 服务，比起内置大数据的云服务来说，也毫不逊色。

课堂总结与思考

作为 PaaS 篇的第一讲，我就先和你讨论到这里了。希望通过今天对 PaaS 的讲解，能够给你建立起一个对 PaaS 宏观层面的正确认识。同时，我今天介绍的几个观察评估要点，的确是你研究 PaaS 时值得参考的良好视角。后面在跟随课程讲到具体的各个 PaaS 服务的时候，也请你记得时不时地回看这一讲的内容，相互印证。

我自己是一个 PaaS 的乐观主义者。如果把你要构建的应用比作高楼大厦，那么 PaaS 作为大厦的基石和支柱，它是当之无愧、值得信赖的。在充分客观了解 PaaS 局限的前提下，你不妨积极大胆地拥抱 PaaS 吧。

好了，今天我留给你的思考题是：你目前接触使用最多的 PaaS 服务是哪个？它给你带来了怎样的效率提升？同时它有没有什么局限让你伤脑筋呢？

欢迎你在下方留言。如果你觉得这篇文章有帮助，欢迎你把它分享给你的朋友。我是何恺铎，感谢阅读，我们下期再见。

© 版权归极客邦科技所有，未经许可不得传播售卖。页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

精选留言 (16)



何恺铎 置顶

2020-03-23

[上讲问题参考回答]

1. “Cloud Shell”是云厂商为你提供的Shell交互环境（通常是免费的），默认安装了官方的CLI工具。你可以直接在上面很方便地执行云资源管理等脚本操作，免去了自己安装维护一个虚拟机的麻烦。
2. 资源组是用来管理账户中各类云资源的一个逻辑上的集合。它有两个特点，一是能够囊括各种不同类型的资源，二是一个资源只能属于一个资源组。一般可以用资源组来表达和标记整

个系统中具备一定规模的“模块”或“组件”，以便你对账户中的资源进行分类管理和成本归属的计算。



13



Helios

2020-03-23

这个问题答不上来了，因为公司业务限制导致我们没有使用公有云的任何paas相关服务，我们的业务是出包到客户场内由交付工程师去部署，都是一些对客户极其敏感的客户，所以暂时用不上公有云。

但是我能说一下我们没用paas的极低的效率~

我们的产品是基于k8s的，日志服务、监控服务、kafka服务，es服务，数据库服务.....当然也包括底层k8s的运维，都是我能搞，这还不是重点，重点是每次有人申请一套环境，我们还有从创建虚拟机到部署出产品给他们整出一套来，这就有了n套环境，每个环境出了问题都要我们解决，最多的时候一周有一半的时间花在这个上面，周报都不知道咋写~

如果把数据库、监控、k8s这些让运营商提供，一是可靠性有了保障，二是使用更加方便了，不用自己部署相关服务，简单配置即可~

作者回复: 是的，这正是PaaS的意义所在。

即便在私有云层面，虽然节奏比公有云慢一些，但也会有越来越多成熟的PaaS服务出现，同样能为私有部署提供便利。

共 2 条评论 >

12



mrtwenty

2020-03-26

用过阿里云的oss，rds、高防、web防火墙，

- 1、oss 文件独立存储、可以加cdn，节省ecs的带宽，独立存储，安全、负载均衡也不用考虑图片单独存储，几乎无限的空间，不用考虑很多的问题
- 2、rds数据库，由于公司没有专业的dba，数据库维护，直接交给了阿里云，升级硬件配置也非常方便，兼容原生的mysql，很好，就是价格有点贵
- 3、web防火墙，高防这些，只能交给专业的第三方或者阿里云，自己实现，几乎不可能，根本扛不住流量的冲击。

作者回复: 很棒，最经典的几个服务都用上了



7



leslie

2020-03-23

PaaS其实对于某个领域研究颇深的技术从业者：个人DB领域多年，接手的就是云厂商的RDB，初期操作策略相对简单还好；中后期2.0架构设计就发现对比实际需求在存储引擎、版本、读写分离、性能参数调整方面操作空间蛮有限的。

就像课程中的例子：装修好的房子你直接可以用，但是你发现装修中的许多不合理性你就没办法调整；无法对于数据系统做到真正的扬长避短；尤其当系统越来越大需要各种特性化优化时，根本发挥不出其真正的版本优势所在。厂商的PaaS的架构或内核版本其实相对于主流市场要晚5-10年。

有力使不上这大概对于专业人员接触此类系统最大的感觉。

谢谢老师今天的分享：期待后续的课程。

作者回复：分析得挺好，这就是我们提过的局限性，是要慎重的。不过晚5-10年这个有点夸张了，甚至在有些方面，其实云上数据库是在引领潮流的，我们后面会谈到的。



5



胖子

2020-03-24

"如果没有内含的运行环境，那就说明这个 PaaS 属于“开箱即用”的工具类型，也就是直接依靠自身内置功能来向你提供支持或帮助。这时它功能的完善程度，以及和你需求的匹配程度，就比较关键了。"，这段话不好理解，请问那些场景适用内含运行环境哪些场景适用不含运行环境？

作者回复：这里主要说的是PaaS的分类。通俗地说，有运行环境的就是那些允许你自己的代码在上面跑的承载性服务，没有运行环境的就是那些通过自有功能体现价值的非承载性服务。



3



李宇嘉(Yujia Li)

2020-03-23

之前项目上使用到了MongoDB，然后上云的时候选择了aws号称兼容MongoDB的DocumentDB，然后应用上去跑的时候就各种报错，太坑了，后来只能自己搭建MongoDB了。

作者回复：是的，有时是会有部分兼容性问题。不过，也不要一棍子打死，要看得细一些，具体是哪个feature、两边版本是不是对应、是哪一年的事情、现在是否已经解决等等。在我的实践中，也曾有

过某家云的MongoDB兼容服务来代替MongoDB，没有太大的问题。



👍 3



开心果源 ~ 老余

2020-03-26

希望老师说说PaaS涵盖微服务，容器，devops等服务，PaaS到底还能承载什么应用，讲得太泛泛了。

作者回复：这一讲是PaaS概述，后面就会进入到一个个具体的PaaS服务了。不过也不要小看这一讲，学习了后面的具体服务以后，可以来再回味一下这一讲里的理念和方法论。

共 2 条评论 >

👍 3



lennonHe

2021-04-29

PaaS 本身也是基于底层 IaaS 构建出来的，使用了云上的各种基础设施。只是这个步骤云服务提供商代替我们用户完成了，还进行了一定程度的封装。

这个部分有点疑问，请教下老师。逻辑上，PaaS本身是基于底层IaaS层构建出来的，但是在实际应用中是否PaaS的服务都是基于IaaS虚拟化出来的资源进行搭建？

目前了解到的，很多情况下大数据服务为了性能考虑，都是直接基于裸机进行部署的，然后通过云管理平台向用户提供服务。这种情况下怎么理解PaaS是基于IaaS进行构建的？



👍 1



Luke

2022-08-08 来自中国香港

主要使用了NAS和OSS，这两个倒没有什么自由度的问题，效率 and 安全性上很好。如果自己部署开源的方案，确实挺麻烦的，还有数据同步等等都需要考虑。



👍



kitsdk

2022-07-15

不知所云，用张磊大神的话说：经典paas是应用托管服务。



👍 1



我来也
2020-03-28

不知道阿里云的日志服务算不算是PaaS.

最近在研究阿里云的k8s,勾选了日志服务后,会自动创建相关的日志.

可以在上面看到很多个人的操作记录, 以及Ingress的日志.

目前还只是初步接触,需要慢慢学习怎么用.

但是感觉默认的配置很强大, 比起自己在es中创建规则, 要方便的太多太多.

也可以看看别人能玩到什么地步.

目前来看,除了费钱,没什么不好的.(目前来看,这个日志费用也非常低)



戴斌
2020-03-23

NAS、OSS等存储还是提升了效率的



小狼
2020-03-23

腾讯的RDS 阿里的日志服务和redis。



一步
2020-03-23

使用最多的PaaS服务当然是数据库了RDS



Christopher
2020-03-23

目前接触的是基于k8s搭建的一个paas平台, 一套部署应用的脚手架, 部署应用相对来说要方便一点, 缺点就是一些部署选项太多, 不太了解k8s可能不太理解, 需要一定的学习成本



俊采
2020-03-23

目前接触最多的是k8s, 对其中的网络访问配置和认证权限配置比较头疼。

