PaaS调研报告（2021.7）

摘要

本文旨在承接2012年林琳博士撰写的PaaS调研报告（以下简称“前文”）的内容，对截至2021年7月PaaS的基本情况、主流产品、支持技术和实际应用进行调研与综述。本文跟踪调研了前文中记载的PaaS的基本情况、分类法和主流产品，继续细化PaaS的分类法并为每项分类增加了用例与实例，深入研究了主流、新兴PaaS平台的功能、特性、架构、使用流程与运行环境，并对前文写作完成时到2021年7月之间涌现的新的PaaS的概念、分类法和产品进行记录调查。

一、PaaS概述

PaaS（Platform as a Service，平台即服务）是云计算服务的一个类别，这类云计算服务允许客户在不用关心与开发相关的底层基础设施的情况下提供、实例化、运行、测试和管理他们生产的应用（[Wikipedia](https://en.wikipedia.org/wiki/Platform_as_a_service)）。一般的应用开发者通常不愿意关心服务器、存储设备和数据备份，他们只想写代码、测试应用、启动应用和持续修复bug，PaaS应此需求而生，PaaS的提供商自动且透明地处理一切与服务器相关的后台事务（[Sacha Labourey, 2013](https://www.networkworld.com/article/2163430/paas-primer--what-is-platform-as-a-service-and-why-does-it-matter-.html)）。

1. PaaS平台的特性

PaaS是将一个完整的软件研发和部署平台，包括应用设计、应用开发、应用测试和应用托管，都作为一种服务提供给客户。同时还是支撑SaaS实质落地的核心应用环境与工具，对于想进入或已经进入SaaS领域的提供商，不仅大大降低了他们服务的门槛，而且可以帮助他们进行产品多元化和产品定制化服务，从而大幅提高软件的交付效率。

三、PaaS分类法

前文分别在功能、部署模式和计算资源管理三个方面对当时的PaaS进行分类。

3.1 功能

3.2 部署模式

从PaaS的计算资源部署的角度，PaaS可以分为公有PaaS、私有PaaS、社区PaaS和混合PaaS，前两者的显著区别在于对数据的掌控，后两者是由前两者结合、衍生而来的。这一部分与前文相比没有显著的更新，只对应每种部署模式更新了一些应用实例。

公有PaaS的客户将自己生产的应用和数据托管于公有PaaS提供商的服务器，客户对数据的实际掌握力度相对较弱。一旦服务器因自然灾害、法律因素或其他原因导致数据丢失，将对向外提供应用服务的PaaS客户造成较大损失。公有PaaS的提供商当然也具备一定的数据安全服务和数据备份能力，但客户对此的控制能力和选择权很有限。相比之下，私有PaaS需要一定建设和维护成本，但在数据的安全和备份等方面有更多选择空间。

公有PaaS和私有PaaS大概有如下对比：

|  |  |  |
| --- | --- | --- |
| 维度 | 公有PaaS | 私有PaaS |
| 用户 | 基于个体产品和服务的创业公司、个人 | 政府、基于平台的大型企业 |
| 业务场景 | 对外互联网业务 | 内部业务 |
| 技术架构 | 自研架构、关注分布式、大集群 | OpenStack开源架构，关注灵活性和适配性 |
| 兼容性 | 根据客户要求，在原有基础上实现较小成度的定制化以适配 | 主动兼容和适配内部业务 |
| 安全 | 主机层实现安全隔离 | 网络层实现安全隔离 |
| 定制 | 一般难以定制 | 灵活定制 |
| 成本 | 初期成本相对低，随着业务量增大成本提高 | 初期成本高，随着业务量增加成本被摊薄 |
| 运维 | 用户无法自主运维，由公有PaaS提供商统一运维 | 自主运维或托管第三方运维 |

除了公有PaaS和私有PaaS，还有社区PaaS和混合PaaS。社区PaaS搭建在一组业务相似或互补（具有连通性）的公司之间，组织中的成员共享一套基础设施、都可以登录使用PaaS。企业组通过这种方式分摊建设成本、提高交互效率。社区PaaS的实质为所有单位不唯一但仍于外部互联网物理隔离的私有PaaS。混合PaaS是两种或两种以上的上述PaaS的混合体。不同形式的PaaS的实例相互独立，但在混合PaaS整体的内部相互结合交互，能够发挥出多种PaaS部署模式的优势。

公有PaaS的实际应用有Apprenda, VMware- and EMC-owned Pivotal, and Red Hat OpenShift（[Apprenda, 2021](https://apprenda.com/library/paas/paas-examples-platformasaservice-options-to-consider/)）。Apprenda云平台是一个支持Kubernetes的平台，向基于Java和.NET的开发者提供云本地和传统应用的支持（[Apprenda, 2021](https://apprenda.com/platform/)）；Pivotal CF™是VMware和EMC等多个公司共同拥有的基于Cloud Foundry的大数据和应用PaaS，有Google Cloud和AWS提供的便于在他们的IaaS设施上部署的PaaS的模板（[Sumo logic, 2021](https://www.sumologic.com/glossary/pcf/)），和互联网头部客户、大型政企等有较多合作（[邹大斌, 2019](http://soft.zhiding.cn/software_zone/2019/0821/3120571.shtml)）。

[](https://apprenda.com/platform/)

图 1Apprenda概述

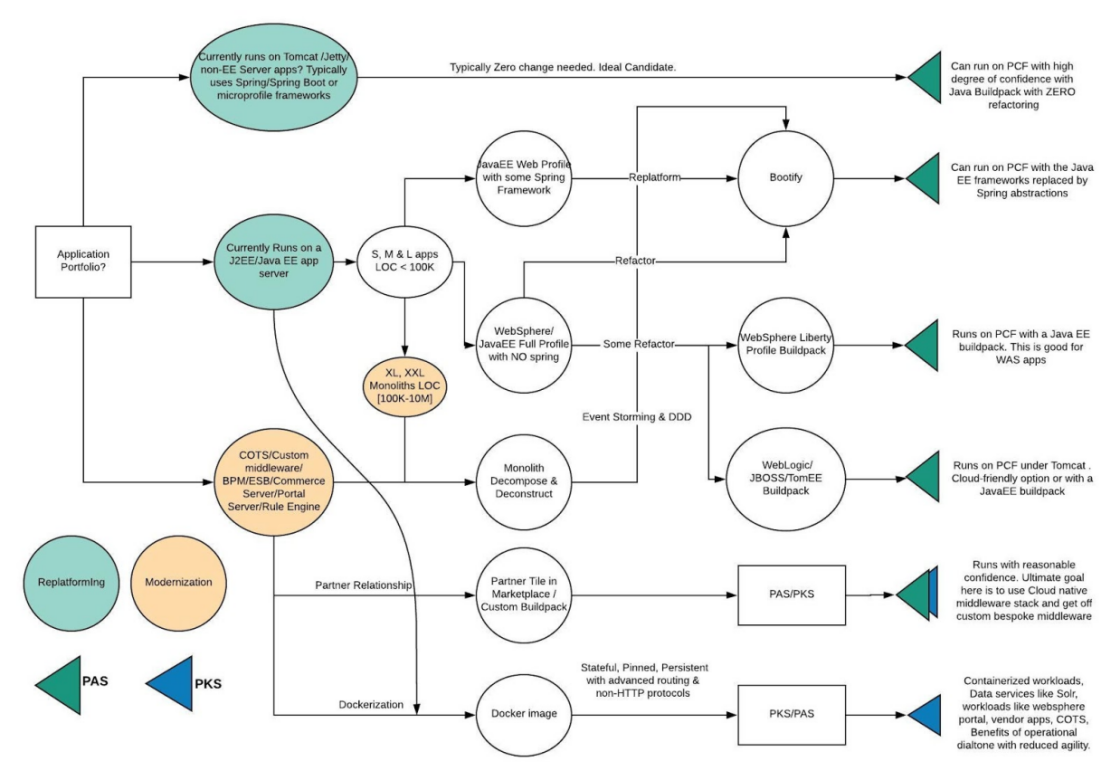
[](https://cloud.rohitkelapure.com/2018/11/when-to-choose-pas-pks.html)

图 2关于如何使用Pivotal产品的决策树

私有PaaS主要有两个应用方向，一个是企业管理部分硬件设备，一个是政企单位管理敏感的信息和数据（[BBVA API Market, 2016](https://www.bbvaapimarket.com/en/api-world/differences-between-public-and-private-paas/)）。私有PaaS值得一提的例子有国产的ZStack和StarVCenter。ZStack与中国政府在身份认证管理、数据交换、中间件、工作流、报表数据挖掘和开发环境测试等方面有广泛深入的PaaS合作（[ZStack, 2021](https://www.zstack.io/thesolution/government/)），该企业还入围2021年中央国家机关虚拟化及虚拟化管理软件采购项目（[中央政府采购网, 2021](http://www.zycg.gov.cn/freecms/site/zygjjgzfcgzx/ggxx/info/2021/6f9d31d30d1b109e7daf80ef0ffcb0a5-A.html?id=4626b2fb-da5a-11eb-9548-fa163e9acaa1)）； StarVCenter具有轻量级、操作简便的特点，适合中小企业采用（[starvcs, 2021](http://www.starvcs.com/case/case1.pdf)），可以直接下载使用，单机部署非常快速简便（[Huboss, 2021](https://blog.csdn.net/HUBOSS/article/details/114369275)）。

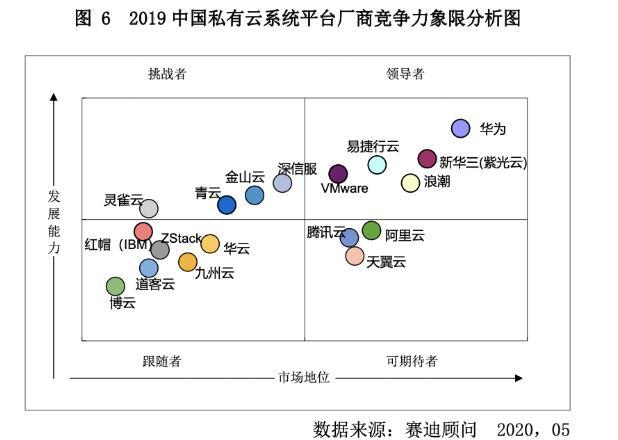


图 32019中国私有云系统平台厂商竞争力象限分析图

社区PaaS暂时没有好例子【TODO, low priority】。

混合PaaS的实例有Red Hat OpenShift（以下简称“红帽OS”）。红帽OS是专为开放混合云策略构建的企业级 Kubernetes 平台。凭借红帽 OS的全堆栈自动化运维、跨所有环境的一致体验以及面向开发人员的自助服务置备，团队可以紧密携手合作，更有效地从构思想法过渡到生产阶段。红帽 OS 既可作为领先公共云中的全托管式云服务提供，也可作为自我管理软件提供给需要更高定制化程度的企业（[Red Hat, 2021](https://www.redhat.com/zh/technologies/cloud-computing/openshift)）。

3.3 计算资源管理

四、世界知名PaaS平台调研

本节对如下PaaS平台和解决方案进行调研，包括前文调研的Google Cloud Provider（在前文中为“Google App Engine”）、Force.com、WSO2 Stratos、IBM SmartCloud、Oracle、Microsoft Azure（在前文中为“Windows Azure”）、Engine Yard等国外的 PaaS 产品，和本文新囊括的Heroku、azure、IBM Cloudpak 红帽等产品。

为了把握各种PaaS产品的特点，本节主要从如下几个角度来分析一款PaaS产品：

|  |  |
| --- | --- |
| 角度 | 说明 |
| 产品架构 |  |
| 使用流程 | 客户使用产品的一般流程。 |
| 开发支持 | 产品支持何种代码、怎样方便客户编辑、上传和管理代码；产品提供何种配套软硬件和服务（例如数据库）。 |
| 角色活动 | 主要从服务供应方和服务合作方的角度讨论可以和需要进行的活动。 |
| 应用场景 | 对产品的主要优势、特点和客户群体进行总结。 |

4.1 Google Cloud Provider

在前文中，Google的PaaS产品Google App Engine（GAE）是当时最成熟、功能最全面的 PaaS 平台，其主要用户是个人开发者。时至今日，Google整合了多种已有架构、自研的和收购第三方的产品，提供更综合的PaaS服务Google Cloud Provider（GCP）（[Google, 2021](https://cloud.google.com/docs/overview)）。

4.1.1 产品架构

经过十数年的发展，GCP的架构非常复杂。本节主要介绍GCP的两种基本示例服务架构和GCP的重要特色组件架构。

GCP针对现代Web应用和移动后端这两种基本使用场景分别提供了一种示例服务架构（[Google, 2021](https://cloud.google.com/appengine" \l "section-8)）。对于现代Web应用，GCP允许客户在其提供的 App Engine 上快速部署应用（详见4.1.2）；对于移动后端，GCP同样使用App Engine部署应用。

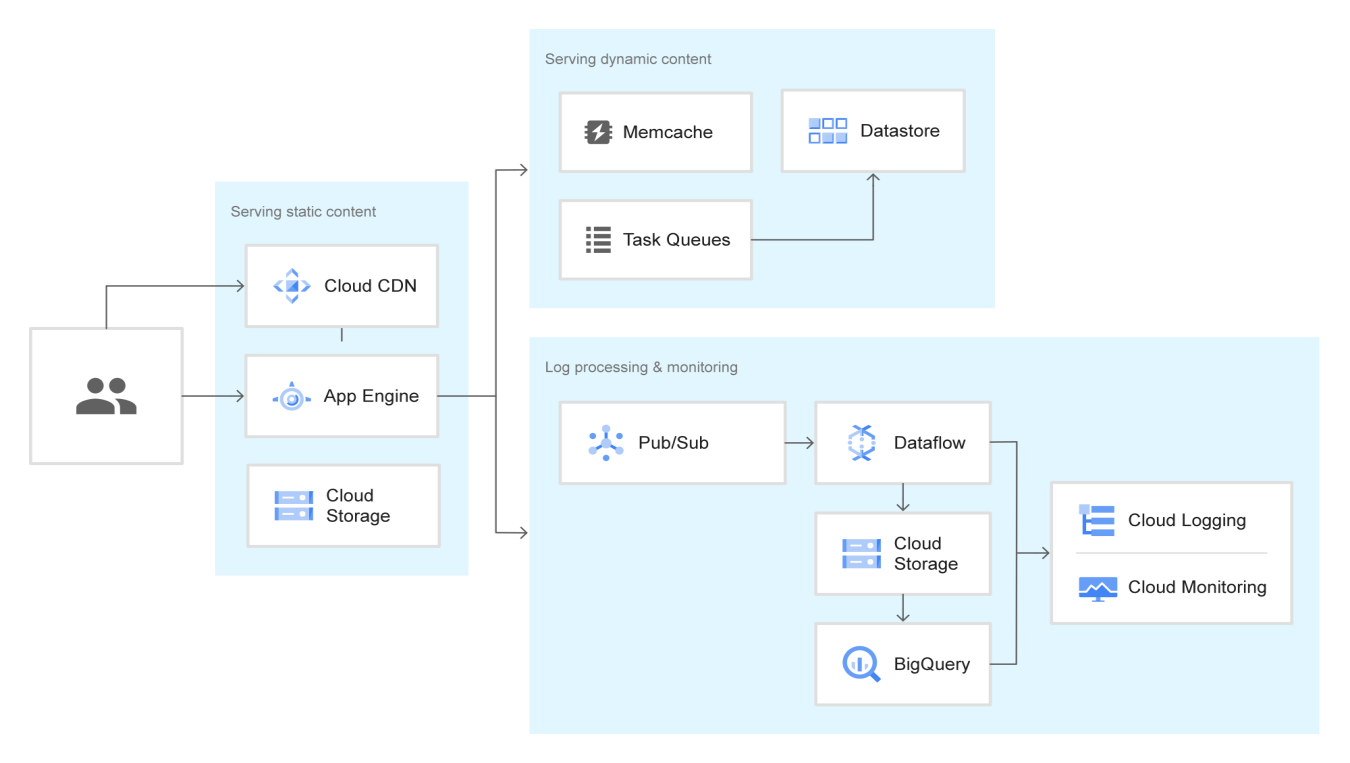


图 4GCP支持Web应用的基础架构图

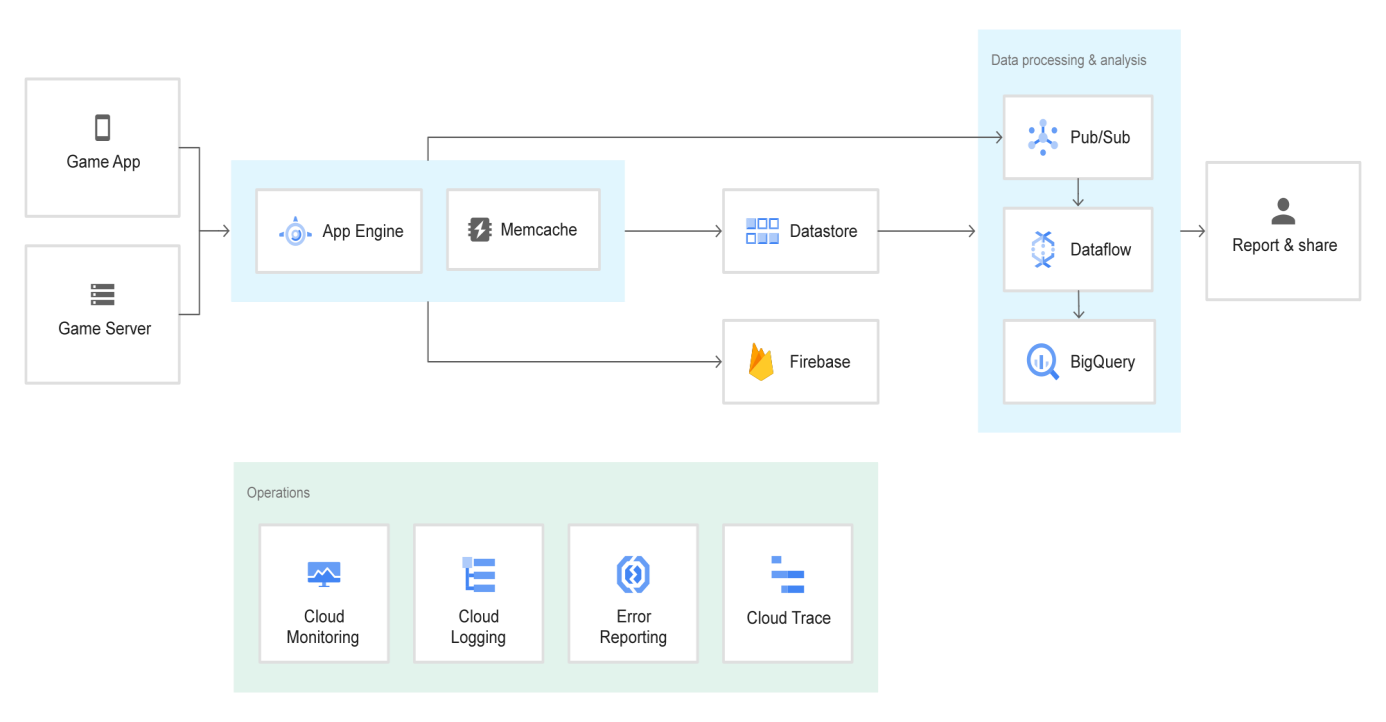


图 5GCP支持移动后端的基础架构图

GCP的特色组件包括但不限于GAE（Google App Engine）、

GAE是GCP的核心组件，是帮助客户在零服务器管理（Serverless）平台搭建高度可伸缩应用的服务（[Google, 2020](https://www.c-sharpcorner.com/article/google-app-engine-what-is-it-its-advantages-and-why-you-should-use-it/)）。

具有零配置部署和零服务器管理的特点，一般情况下客户只需提供基本功能代码即可实现应用。此外，App Engine还支持一定程度的Serverless，应对突发的流量变化可以自动规划物理硬件等的使用。

还创造性地提供Firebase服务用于帮助客户进行产品分析、提升产品质量甚至构建A/B测试。

4.1.2 使用流程

GCP的简要使用流程为：评估需求环境和团队能力→规划云基础架构与移动方式→设计部署过程→充分利用云原生技术和功能对应用进行优化。



GCP首先要求客户评估对环境的需求：标准环境（Standard）或柔性环境（Flexible）。他们的区别大概如下表（[Google, 2021](https://cloud.google.com/appengine/docs/the-appengine-environments)）：

|  |  |  |
| --- | --- | --- |
| 功能 | 标准环境 | 柔性环境 |
| 应用基本运行 | 应用在支持多种语言（详见4.1.3）的沙盒中运行 | 应用在Compute Engine虚拟机上的Docker容器中运行 |
| 部署启动时间 | 快（数秒） | 较慢（数分钟） |
| 支持后台处理 | 否 | 是 |
| 支持SSH调试 | 否 | 是 |
| 支持对Runtime更改 | 否 | 是 |
| 支持WebSocket | 否 | 是 |
| 支持第三方二进制文件 | 是，但仅对于支持语言的特定版本支持 | 是，广泛支持 |
| 缩减至零 | 是 | 否，至少1个实例 |

客户决定好目标环境后，需要参考Google Cloud 采用框架评估所在组织在云技术采用方面的成熟度（[Google, 2021](https://cloud.google.com/adoption-framework?hl=zh_cn)），并根据反映团队云技术能力的测量结果决定迁移类型（原样迁移、改进后迁移或淘汰后重写）。

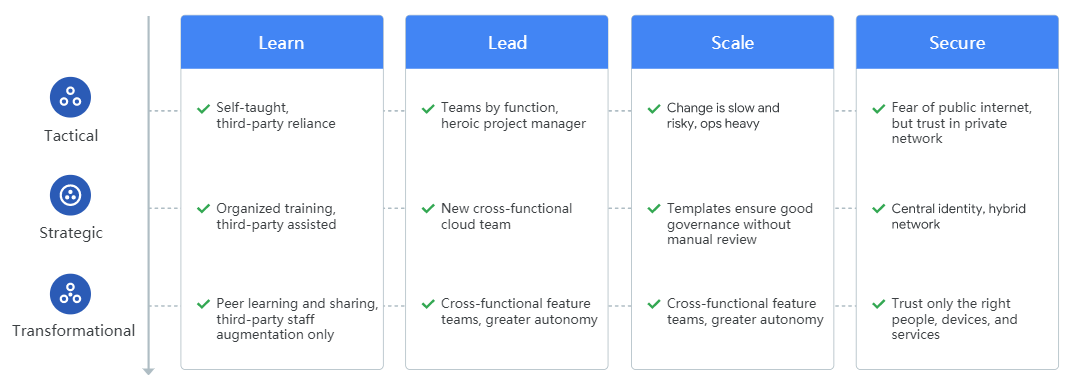


图 6Google Cloud 采用框架

除了评估团队能力，应用迁移前客户还需要评估应用本身、迁移工作负载和费用等。应用本身需要评估的类型有包括但不限于数据库、消息代理、网络设备和其他应用依赖项，同时提供所有相关的硬件规格、系统环境和许可证，根据任务的关键性和任务的依赖项与从属项构建目录矩阵。

规划阶段，GCP提供多种身份类型和对应的资源组织、群组和角色供客户依据团队架构来选用。

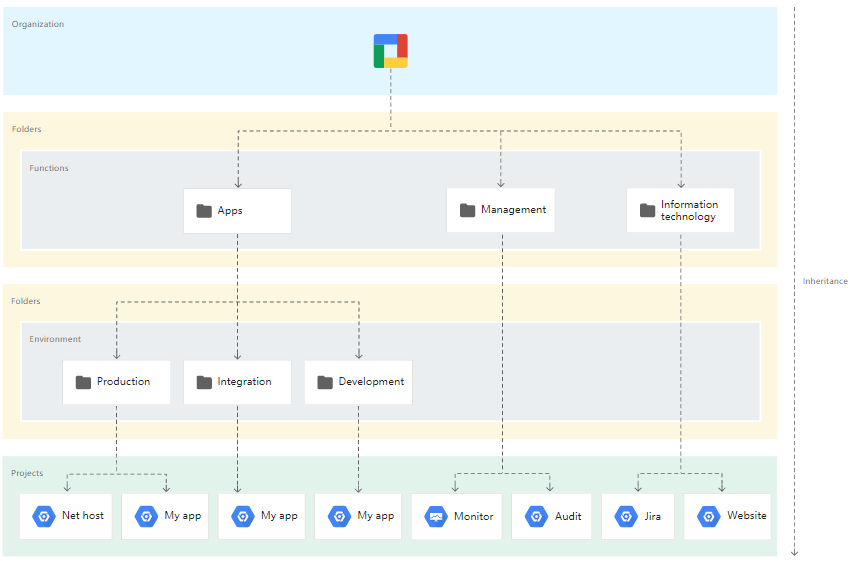


图 7一种面向功能的层次结构

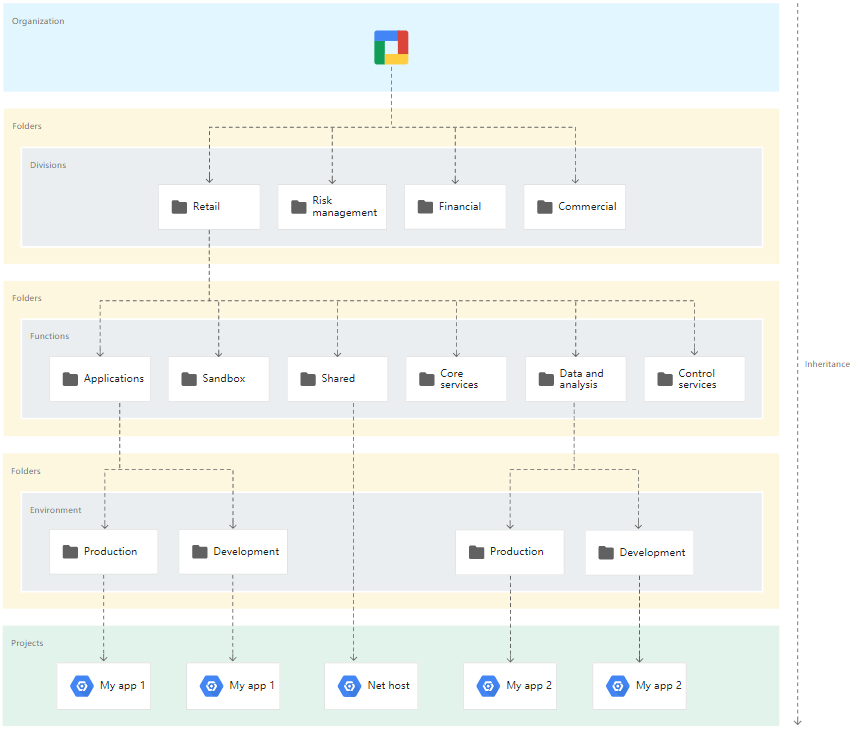


图 8一种面向精细访问权限的层次结构

部署方面，GCP推荐客户使用配置管理（CM）工具自动化地配置环境，对已经在容器化方面有投资的客户可以使用Google Kubernetes Engine（GKE）等服务来编排工作负载，也可以借助Cloud Console纯手动部署。

4.1.3 开发支持

根据谷歌云官网文档提供的信息，GCP支持Java、Node.js、Python、C# / .NET、Go、Ruby和PHP（[Google, 2021](https://cloud.google.com/docs)）。谷歌云没有Cloud development kit，只有Google Cloud Go CDK（[Document](https://gocloud.dev/)/[Github link](https://github.com/google/go-cloud)）。有人使用Pulumi这个第三方CDK

GCP提供三种服务和资源交互的方式，分别是云控制台、云命令行和客户端库。云控制台提供基于Web的图形界面；云命令行提供预装的Cloud SDK、内置编辑器、对上述所有支持语言的语法支持、临时的Compute Engine虚拟机实例和Web预览功能；客户端库提供对服务的访问权限、有针对支持语言的优化、在Admin API提供资源管理功能以及与谷歌地图、谷歌云端硬盘和Youtube等流行产品良好适配（[Google, 2021](https://cloud.google.com/products/management/)）。

另外值得一提的是，GCP的Project Management是一个争议颇多的服务：一个GCP账户下可以创建多个项目，并且可以针对属于一个账户的不同项目进行针对性的管理（尤其是权限管理），还支持将一个项目的资源配置直接克隆到另一个项目，而其他PaaS提供商往往采用一个账号绑定一个项目的模式（[Google, 2021](https://cloud.google.com/docs/overview)）。

4.1.4 关联配套

4.1.5 运维管理

4.1.6 应用场景

根据ITNEXT上的一篇评论，谷歌云服务（尤其是和Amazon的AWS云服务进行比较时）具有包括但不限于以下十个重大优势（[Javier Ramos, 2020](https://itnext.io/google-cloud-advantages-over-aws-28751469e570)）：

1. 网络协议栈（Network Stack）：谷歌利用其极低延迟的内部网络提供强大的网络传输性能，特别是对于有低延迟要求的微服务和大数据处理；同时，GCP的虚拟私有云VPC（Virtual Private Cloud）是全球化的，GCP服务的整个网络都可由软件需求进行定制，具有很大的灵活性；GCP的负载均衡器在网络边缘位置工作，提供全球负载均衡和自动扩展。一言以蔽之，借助GCP，开发者可以非常轻松地构建全球性地基础架构，这对于其他云提供商来说都相当困难。
2. 开发者体验

GCP的控制台使用起来非常方便，用户可以从浏览器直接获得终端并无需SSH密钥就能安全地访问虚拟机；GCP的身份管理工作也非常高效，它与G Suite集成并提供SSO（单点登录）；最后，GCP的大多数服务都提供模拟器，使开发者无需第三方工具或者复杂的集成就能用便携笔记本测试所有应用程序。

1. Pub/Sub

GCP使用单一的Pub/Sub（发布订阅模式）用于消息传递，高效、成本低廉、易于集成和使用、支持多种客户端与协议甚至还不需要服务器。

1. 数据库

Google特别擅长数据方面的工作，比如管理和扩展大数据或者为各种用例提供灵活的解决方案。GCP提供的如下三种大数据解决方案在当前是无可匹敌的：

* 1. Big Table

Big Table是一个完全托管的千万亿字节级的NoSQL数据库，提供一致的10ms以下的延迟，非常快速可靠、易于扩展并且经济划算。

* 1. BigQuery

BigQuery是GCP的主打产品之一，是“一个无服务器、高度可扩展且经济划算的云数据仓库，专为业务敏捷设计（[Google, 2021](https://cloud.google.com/bigquery)）”，利用Google基础架构的强大功能使用户可以轻易储存、提取和分析PB乃至EB级的数据。

* 1. Cloud Spanner

Cloud Spanner是针对区域和全球应用程序数据的完全托管、可扩展的关系数据库服务。相比于Big Table和BigQuery，Cloud Spanner是GCP真正特色的解决方案，其他云提供商中几乎没有与之类似的数据库。Cloud Spanner庞大的关系型数据库，它允许客户使用常规的SQL处理巨大体量的工作还能同时保证强一致性。Cloud Spanner的缺点是价格并不便宜。

1. 机器学习与人工智能

Google拥有最好的机器学习平台，为所有类型的用户和用例提供了工具（从专用于深度学习的虚拟机到高级API）。

1. Kubernetes

Kubernetes方面，GCP比其他云提供商更便宜、更新、更快、更易于使用。由于其灵活性和价格优势，GKE（Google Kubernetes Engine）可能是世界上最好的Kubernetes托管云服务。GKE允许客户轻松地将业务从本地迁移到云，它安全并且易于设置，提供出色的自动缩放，很容易监视。不仅如此，Google还将Autoscaler和Kubernetes集成，实现了k8s节点的自动伸缩机制，从某种意义上说几乎做到了Serverless的理念。

1. 价格

一般来说，无论是使用Kubernetes还是加购计算和存储设施，GCP的价格都相当实惠。GCP偶尔推出的秒杀抢购的虚拟机和GKE群集的价格更是十分低廉。

4.2 Salesforce App Cloud

前文中，Salesforce面向企业的云计算PaaS平台Force.com被认为是当时最成功的公有PaaS平台之一。2010年，Salesforce公司以2.12亿美元的价格收购了另一个优秀的PaaS平台Heroku（[Brian Heater, 2010](https://techcrunch.com/2010/12/08/breaking-salesforce-buys-heroku-for-212-million-in-cash/)），并随后逐步将Force.com和Heroku整合为更全面的PaaS平台Salesforce App Cloud（[Faizi Fakhruddin, 2015](https://www.appcloud101.com/archives/7221)）。

Force.com是

Heroku是一个基于容器的PaaS平台，

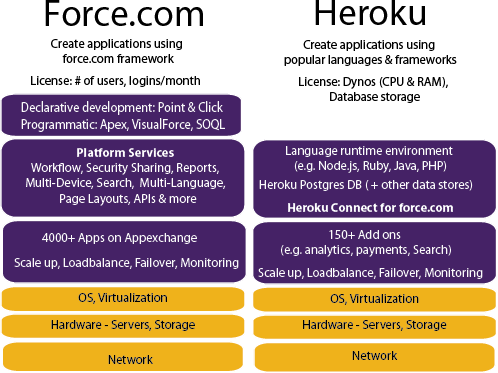


图 9Force.com和Heroku的简明对比

4.2.1 产品架构

本节将分别介绍Force.com与Heroku两个PaaS产品的架构，并引入一种将两个产品连通的架构设计Heroku Connect以便于把握整个Salesforce App Cloud的架构特点。

4.2.1.1 Force.com架构

4.2.1.2 Heroku架构

Heroku使用一种简洁但强大的开源架构。如下图所示，Heroku的架构自外到内大致由反向代理服务器、HTTP缓存、动态路由处理层、动态计算单元（Dynamometer，一般简称为Dyno）网格层、数据库和内存缓存组成。

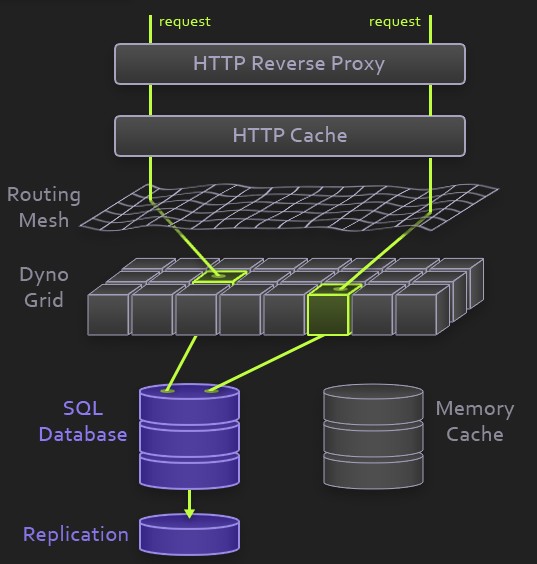


图 10Heroku架构示意图

1. 反向代理服务器

尽管Heroku允许客户的应用直接通过HTTP被访问，但多数Heroku应用会采用Nginx进行反向代理（[Greg Nokes, 2020](https://dev.to/heroku/nginx-routing-on-heroku-567j)），以解决包括但不限于处理SSL、HTTP的请求中转和Gzip的压缩传输等问题。Heroku提供的用例中，客户常常同时部署多个前端Nginx 服务器来解决DNS及负载均衡的问题（[Heroku, 2021](https://devcenter.heroku.com/articles/http-routing)）。

1. HTTP缓存

有效的HTTP缓存可以让应用有更好的应答效率，同时能减轻动态计算单元上的负载（从而为客户降低成本）。Heroku不提供默认的HTTP缓存服务，推荐使用Amazon CloudFront和CloudFlare等内容分发网络CDN（[Heroku, 2020](https://devcenter.heroku.com/articles/http-caching)）或Rack::Cache和JAX-RS等HTTP缓存服务（[Heroku, 2020](https://devcenter.heroku.com/articles/increasing-application-performance-with-http-cache-headers)）实现HTTP缓存。

1. 动态路由处理层

每当应用被访问并且HTTP缓存中找不到请求对象，访问请求就会被传递进动态路由处理层，由动态路由处理层应用轮询调度算法决定分配哪些动态计算单元去处理这一请求（[Anonymous, 2014](https://www.oreilly.com/library/view/heroku-cloud-application/9781783550975/ch02s02.html)）。这个层面原理上是一个分布式的动态HTTP请求的路由池，大大增强了应用的可扩展性和容错性，可以根据下一层服务的负载容量来合理进行路由的选择。

1. 动态计算单元网格层

这一层面是Heroku架构设计的核心。Heroku使用一种叫做Dyno（动态计算单元）的分离的虚拟化Linux容器来运行应用——Dyno可以被看作粒度更细的服务器集群环境；客户可以根据应用运行需要的资源量来指定使用的Dyno数量，Heroku也会根据Dyno的使用情况衡量服务用量并收取费用（[Heroku, 2021](https://www.heroku.com/dynos)）。

Dyno根据工作任务的不同分为三种类型：Web Dyno，Worker Dyno和One-off Dyno。最常用的Dyno类型是Web Dyno，相当于一个Web服务器实例，负责接收HTTP请求并提供服务，应用可以通过使用更多的Web Dyno以增强其请求处理能力；Worker Dyno负责执行后台作业或其他辅助任务以及由Web Dyno转交的耗时间且复杂的计算任务；One-off Dyno则承担客户偶尔需要的临时工作，例如一些周期性任务、临时备份或数据迁移等（[NagaSai, 2017](https://absyz.com/heroku-architecture/)）。

1. 数据库和内存缓存

Heroku采用PostgreSQL作为默认数据库，默认支持Redis提供内存缓存服务，

————————————————————————————

Heroku提供了大量的插件和扩展,可用于数据库、电子邮件支持和其他很多服务。

平台的灵活性极高且支持多种编程语言。若想把程序部署到Heroku上,开发者要使用Git把程序推送到Heroku的Git服务器上。在服务器上,git push命令会自动触发安装、配置和部署程序。

Salesforce App Cloud的实质就是整合了Force.com和Heroku两个组件的优势和特点形成的定制化PaaS平台。一般而言，客户可以自主选择依靠Force.com或者Heroku搭建自己的应用；但是，Salesforce更推荐使用一种架构上最优的设计Heroku Connect。

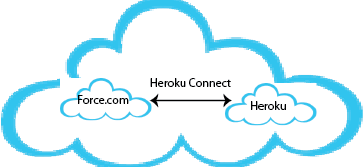


图 11Heroku Connect

在Salesforce App Cloud中，除了基本的API调用、CSV文件和ETL工具，基于Force.com和Heroku的应用相互之间还可以通过Heroku Connect来交换数据。Heroku Connect是一种保持Heroku中的Postgres数据库与Force.com的对象同步的复制技术，将Heroku和Force.com紧密结合在一起。

1. Compute Intensive Work on Large Data Volumes

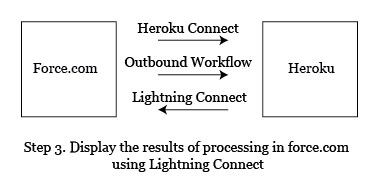
An application which needs complex workflows & fine grained user access can be quickly created on Force.com. At those points in your application where you need compute intensive resources – you can send the computation request to Heroku through outbound workflow and sync back the results of the computation with Heroku Connect.



use-case-1\_1



use-case-1\_2

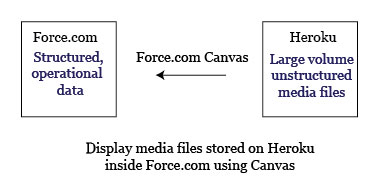


use-case-1\_3

2. When Your App Requires Large Volume of Media Files

Instead of storing large volume of unstructured media files – videos, images – in force.com, store them on Heroku. The display the images/videos in context in your force.com app using Canvas. Canvas is a tool from Force.com which allows you to embed external content securely and in-context in Force.com user interface.

To store large volume of unstructured files using your Heroku application, you will also need storage on AWS.

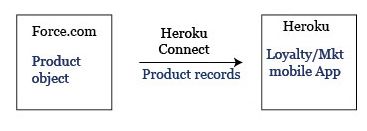


use-case-2\_1

3. Employee App + B2C App: An Integrated Approach

e.g. Macy’s Mobile App

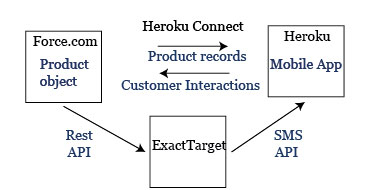
A visitor on a mobile app (served by Heroku), browses product information (served by Force.com through Heroku Connect).



use-case-3\_1

The visitor/customer then logs using social media (say Facebook) and “Like” a product item (& shares it on his Facebook page). The customer’s interactions are captured by the Mobile Heroku App and synced back to Force.com Sales Cloud using Heroku Connect. The sales rep views customer’s interactions on Account page. The customer is a record in the Contact related list for that Account.

The sales rep rewards the customer by sending a discount code, to the customer’s mobile app through SMS. The SMS message is delivered via ExactTarget Marketing Automation API. ExactTarget also sends the customer surveys which the customer can complete to earn more points.

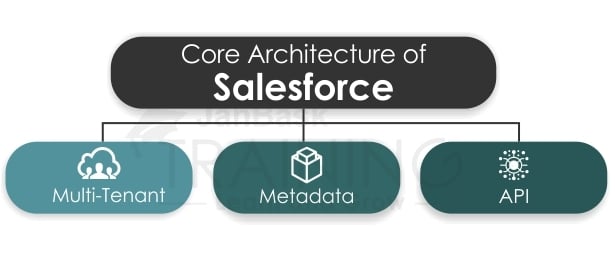


use-case-3\_2

The customer can also submit ideas for new products to the Force.com Customer Community through External Identity feature in Force.com. SSO between Heroku app and Force.com ensures that the customer only needs to login once.

You can find this last mentioned starter app for Heroku on https://www.heroku.com/cx

I do want to emphasize that each framework – Force.com and Heroku – are full functional enterprise application frameworks. While together, they are a potent combination, in a vast majority of applications each by itself can be a complete solution for your business requirements too. Ultimately, your skill as an architect shines in crafting the right architecture pattern and choosing what’s right for your business.



1). Multi-Tenant

Multitenancy is an incredible word for influencing you to sound like a geek at supper parties, all things considered, all it implies is that you're sharing your assets. Salesforce gives a central arrangement of administrations to every one of its clients in the multitenant cloud. Regardless of the span of your business, you gain admittance to a similar registering power, information stockpiling, and central features. Whereas a customary single-inhabitant application requires a committed arrangement of assets to satisfy the necessities of only one association, a multitenant application can fulfil the requirements of numerous occupants (organizations or divisions inside an organization, and so forth.) utilizing the equipment assets and staff expected to oversee only a solitary programming case.

2). Metadata

In Metadata-Driven Architectures multitenancy is viable just when it can support applications that are dependable, adaptable, upgradeable, secure, and quick. It's hard to make a statically ordered application executable that can meet these and another one of kind, difficulties of multitenancy. Characteristically, a multitenant application must be dynamic in nature, or polymorphic, to satisfy the individual desires of different inhabitants and their clients. Thus, multitenant application plans have developed to utilize a runtime motor that produces application parts from metadata—information about the application itself. In an all-around characterized metadata-driven engineering, there is an unmistakable division of the ordered runtime engine (kernel), application information, the metadata that portrays the base usefulness of an application, and the metadata that relates to each occupant's information and customizations.

3). API

On a very basic level, APIs enable diverse bits of programming to interface with each other and trade data. On the off chance that it sounds sort of theoretical, just take a quick glance at the PC you're working on right at this moment. You can presumably discover a progression of connections of different shapes and sizes that help various types of associations. These resemble the equipment form of APIs. You don't need to know how the USB port functions. You should simply understand that when you connect your telephone to a USB port, it passes data to your PC.

APIs are comparable. Without knowing the subtle elements, you can interface your applications with different applications or programming frameworks. The fundamental innovation deals with the specifics of how data goes all through the framework.

Conclusion

Salesforce architecture is not a surprising result of any random series of hit and trial experiments. Every feature of its architecture has been carefully planned and placed right where it requires to be. If you get a hand at its architecture you can understand most of its functionality.

4.2.2 使用流程

4.2.3 开发支持

Force.com允许客户定制Salesforce应用或者使用force.com提供的框架制作自己的应用。Force.com提供的应用框架是图形化、轻代码的，便于技术缺乏的小微企业或个人快速建立应用，同时force.com也提供一种面向对象语言APEX和UI设计语言Visual Force供客户应对更高需求（[Faizi Fakhruddin, 2015](https://www.appcloud101.com/archives/7221)）。

根据Heroku的官方网站，Heroku支持Python、Java、PHP、Ruby、Node.js、Go、Scala和Clojure（[Heroku, 2021](https://devcenter.heroku.com/start)），也可以通过使用Heroku社区已有或根据开放API自定义的构建包（Buildpacks）支持另一种语言（[Heroku, 2021](https://www.heroku.com/languages)）。

4.2.4 关联配套

The strength of Force.com lies not in the easy development tools, but in the integrated nature of the platform. Database, user directory, security access controls and application logic are tightly integrated in an enterprise grade framework. The applications you create can exchange data with any external system – ERP apps, databases and more – through a variety of ways including APIs, CSV files, ODBC connections.

4.2.5 运维管理

4.2.6 应用场景

Heroku与使用十二因子应用（Twelve Factor App）方法构建的应用程序协同运行时效果最好。还可以使用第三方应用程序，它们作为Heroku平台里面的服务。

功能特性：

一旦你推送了应用程序的源代码，Heroku就会获取你在应用程序中使用的框架和语言所特有的必要依赖项，让源代码准备好执行。

Heroku让你可以实现针对特定环境的配置（比如支持服务的登录信息）的管理和源代码的管理相互独立，以提高安全性和移植性。

你可以快速启用应用程序环境的一次性实例，以运行临时命令。

（James Somers, 2013）【重要材料】

4.3 WSO2 Stratos

4.3.1 产品架构

4.3.2 使用流程

4.3.3 开发支持

4.3.4 关联配套

4.3.5 运维管理

4.3.6 应用场景

4.4 IBM SmartCloud

4.4.1 产品架构

4.4.2 使用流程

4.4.3 开发支持

4.4.4 关联配套

4.4.5 运维管理

4.4.6 应用场景

4.5 Oracle

4.5.1 产品架构

4.5.2 使用流程

4.5.3 开发支持

4.5.4 关联配套

4.5.5 运维管理

4.5.6 应用场景

4.6 Microsoft Azure

Microsoft Azure 是Microsoft的公有 PaaS 产品，它的服务器和相关的网络基础设施是由微软的数据中心通过公共的 Internet 提供的。Windows Azure平台包括具有高度弹性和伸缩性的云计算操作系统，数据存储设备以及由Windows Server 2008 实例（物理或虚拟的）提供的相关服务。Windows Azure 软件开发包（SDK）提供了一个云服务、工具和 API 的开发版本，用于开发、部署和管理 Windows Azure 中可伸缩的服务。SDK 中包括了一组标准化的 Azure应用程序的 Visual Studio 2008 和 2010 模板，同时，Visual Studio 中的控件也对Windows Azure 应用程序提供了相应的支持。Windows Azure 具有如下的功能：

1) 向已有的应用程序包中添加 Web 服务能力；

2) 通过创建、修改和发布基于 Web 的应用程序，最小化用户内部的资源；

3) 远程提供大规模存储、批处理、大计算量的计算等服务；

4) 快速廉价的创建、测试、调试和发布 Web 服务；

5) 降低建设本地计算资源的花费和风险；

6) 减少 IT 管理的投入。

Azure可以说既是某种基础设施即服务（IaaS），又是某种PaaS，所以有理由认为它其实不属于这份名单。然而，Azure又是被Gartner评为是IaaS和PaaS这两个领域的唯一主要的云平台。托管服务和非托管服务的这个强大组合让你可以以自己喜欢的任何方式，构建、部署和管理应用程序，以获得无可比拟的工作效率。

功能特性：

可以在短短几分钟内配置Windows和Linux虚拟机和应用程序。可以使用Azure中与你在本地使用的同样的虚拟机和管理工具。

构建和部署面向安卓、iOS和Windows的一系列广泛的现代应用程序，它们充分利用了云计算――包括Web、移动、媒体和业务解决方案。可以自动增加或减少资源，以满足任何要求。

Azure提供了托管SQL和NoSQL数据服务，并提供了内置支持，以便从数据获得洞察力。可以充分发挥云端SQL Server的功能，并使用HDInsight来构建Hadoop集群，以便分析数据。

托管用户帐户，与现有的本地目录同步，并且跨Azure、Office 365和数百个流行的软件即服务应用程序（包括Salesforce、DocuSign、Google Apps、Box、Dropbox及更多）获得单点登录功能。

4.6.1 产品架构

4.6.2 使用流程

4.6.3 开发支持

操作系统同时支持Windows和各种版本的Linux，开发语言同时支持.NET、Java、Python、PHP、Node.js和Ruby。

4.6.4 关联配套

视频服务Windows Azure Media Services到其Azure库、Azure还突出了SQL Azure数据库服务，用于存储，业务分析，缓存，身份安全通信和内容分发等服务。

4.6.5 运维管理

4.6.6 应用场景

4.7 Engine Yard

Engine Yard在去2018年收购了PHP应用开发平台Orchestra，为PHP开发者提供了部署PHP应用到云中的服务。开发者可以使用流行的PHP框架构建应用程序，并可以通过Git或Subversion进行部署。

4.7.1 产品架构

4.7.2 使用流程

4.7.3 开发支持

4.7.4 关联配套

4.7.5 运维管理

4.7.6 应用场景

4.8 AppFog

AppFog是一种多语言、多框架的PaaS，它是构建多个私有云的一种不错选择。它支持Java、Ruby、PHP、Python、Node、Scala和Erlang，并提供MySQL、PostgreSQL、Redis、RabbitMQ以及第三方附件。AppFog基于开源的Cloud Foundry平台，支持Git、SVN和Mercurial等代码管理系统。

功能特性：

Varnish Cache和操作码缓存运行应用程序，减轻了服务器负载，提升了性能。

不用再配置服务器、防火墙、Apache和安全系统，也不用再安装框架。

与Git、SVN和Mercurial之类的代码管理系统兼容。

可以跨许多地区、数据中心和基础设施运行。

只访问你所需要的服务器，并只为你使用的资源支付费用。

Web应用程序技术包括PHP、Node、Ruby、Python和Java。

可以将任何流行的服务添加到你的应用程序中，包括MySQL、PostgreSQL、Redis和RabbitMQ。

4.8.1 产品架构

4.8.2 使用流程

4.8.3 开发支持

4.8.4 关联配套

4.8.5 运维管理

4.8.6 应用场景

四、国内知名PaaS平台调研

本节对如下PaaS平台和解决方案进行调研，包括前文调研的Google App Engine、Force.com、WSO2 Stratos、IBM SmartCloud、Oracle、Windows Azure、Engine Yard等国外的 PaaS 产品，和本文新囊括的Heroku、azure、IBM Cloudpak 红帽等产品。

4.10 百度云

4.1.1 产品架构

4.1.2 使用流程

4.1.3 开发支持

它的开发环境有Node.js、PHP、Python、Java、Static等；

云数据库：MySQL、MongoDB、Redis等；

其他服务：语音识别、人脸识别、百度翻译、百度地图、云推送。

4.1.4 关联配套

4.1.5 运维管理

4.1.6 应用场景

4.11 阿里云

4.1.1 产品架构

4.1.2 使用流程

4.1.3 开发支持

它的开发环境有 PHP、Java等；

云数据库：MySQL、SQL Serve等；

其他服务：阿里系应用的良好对接。

4.1.4 关联配套

4.1.5 运维管理

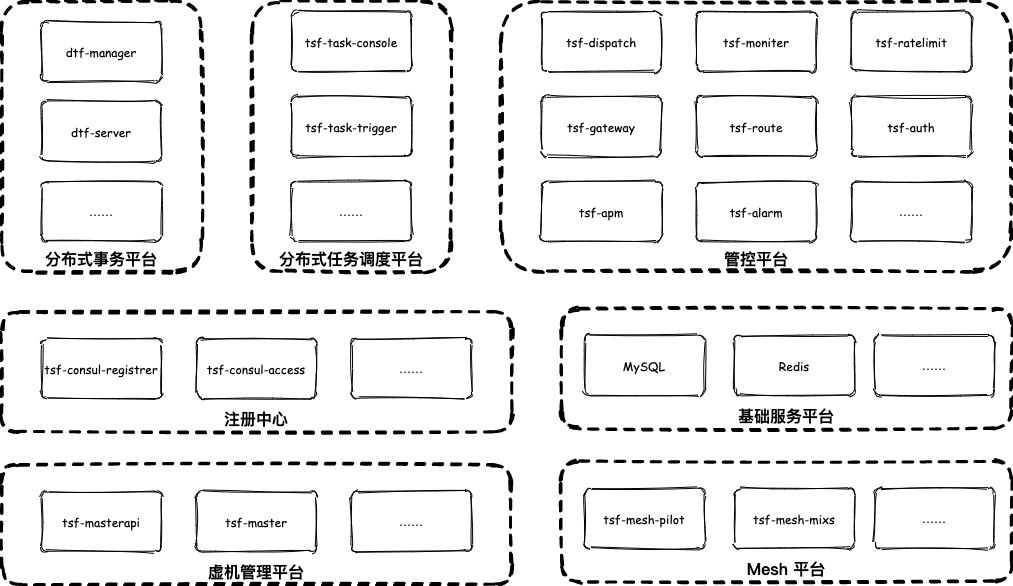
4.1.6 应用场景

4.12 腾讯微服务平台（TSF）

腾讯微服务平台（Tencent Service Framework，TSF）是一个围绕着应用和微服务的PaaS技术平台。平台旨在为客户提供应用全生命周期管理、数据化运营、立体化监控和服务治理等功能，帮助企业客户解决传统集中式架构转型的困难，打造大规模高可用的分布式系统架构，实现业务、产品的快速落地。TSF以腾讯云中间件团队多款分布式产品为核心基础组件，提供秒级推送的分布式配置服务、链路追踪等高可用稳定性组件。此外，TSF与腾讯云API网关和消息队列打通，让企业轻松构建大型分布式系统。

4.1.1 产品架构

相比于大型系统架构为了应对高并发、大存储等因素而使用的复杂架构，TSF规模小且服务类目偏少，属于中小型的微服务架构。TSF架构设计主要侧重于产品的稳定、内聚和高可用等方面。  
 从组件分布上看，TSF整体功能主要分布在管控平台、注册中心、分布式任务调度平台、分布式事务平台、Mesh平台等各个子平台中，每个子平台又包含若干个数量不等的服务。



4.1.2 使用流程

4.1.3 开发支持

开发环境：PHP、Java等；

云数据库：MySQL；

其他服务：腾讯系应用的良好对接。

4.1.4 关联配套

4.1.5 运维管理

4.1.6 应用场景

4.11 阿里云

4.1.1 产品架构

4.1.2 使用流程

4.1.3 开发支持

它的开发环境有 PHP、Java等；

云数据库：MySQL、SQL Serve等；

其他服务：阿里系应用的良好对接。

4.1.4 关联配套

4.1.5 运维管理

4.1.6 应用场景

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| PaaS产品 | 主要特点 | 面向客户 | 部署方式 | 开发支持 | 关联配套 | 其他 |
| 新浪云(http://sae.sina.com.cn) |  | 主要适用于网站、博客、论坛等小型应用。 |  | 现阶段，SAE仅支持Web开发语言PHP和关系数据库MySQL |  |  |
| 百度应用引擎(http://developer.baidu.com/service) |  |  |  | 用户可基于BAE平台进行PHP、Java应用的开发、编译、调试、发布。 | 同时BAE平台也提供了若干云服务，包括fetchURL、task queue、SQL。 | 目前，BAE尚处于公测阶段，许多性能和服务还亟待完善。 |
| 阿里云（http://www.aliyun.com） | 帮助应用开发者简化网络应用程序构建和维护，并根据应用访问量和数据存储增长进行扩展。 |  |  | ACE支持PHP语言编写的应用程序；支持在线创建MYSQL远程数据库应用。 |  |  |
| 腾讯开放平台(http://open.qq.com/reg) | 腾讯基于其拥有的各大社交平台推出的应用开放平台。用户可将开发好的游戏、网站等应用一次性同时接入QQ空间、朋友网、腾讯微博、Q+平台，让开发和运营流程更简单、更安全。 |  |  |  |  |  |
| 厦门吉鼎J2Paas低代码平台(https://bbs.jikaiyun.com/index) | 低代码PaaS云平台，既能为软件应用提供安全、稳定、高效的运行环境，同时还提供了功能全面的JJ Studio开发工具包，方便用户迅速开发出自己的SaaS化应用！ |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

VMware Cloud Foundry

Cloud Foundry是VMware的一项开源PaaS计划，使用各种开源开发框架和中介软件，来提供PaaS服务。开发者可以通过这个平台来建设自己的SaaS的服务，不用自行建设和维护硬体服务器和中介软件。由于Cloud Foundry采用开源的网站平台技术，所以开发者的应用程序也可以任意转移到其他平台上而不受限于PaaS的平台。 目前Cloud Foundry可以支多种开发框架，包括Spring for Java、Ruby on Rails、Node.js以及多种JVM等。

Force.com

Force.com是企业云计算公司Salesforce.com的社会化企业应用平台，允许开发者构建具有社交和移动特性的应用程序。另外，它还提供了有助于在云上更快建立及运行业务应用程序的所有功能，包括数据库、强劲分析、实时工作流程及审批、可编程云逻辑及网站功能等。Force.com支持Apex编程语言，开发者可以基于UI层面编写数据库触发器和程序控制器。

在这里插入图片描述

Amazon Elastic Beanstalk

Elastic Beanstalk为在Amazon Web Services云中部署和管理应用提供了一种方法。该平台建立如面向PHP的Apache HTTP Server和面向Java的Apache Tomcat这样的软件栈。开发人员保留对AWS资源的控制权，并可以部署新的应用程序版本、运行环境或回滚到以前的版本。CloudWatch提供监测指标，如CPU利用率、请求计数、平均延迟等。

Engine Yard Orchestra

Engine Yard在去2018年收购了PHP应用开发平台Orchestra，为PHP开发者提供了部署PHP应用到云中的服务。开发者可以使用流行的PHP框架构建应用程序，并可以通过Git或Subversion进行部署。

SaaS简介及国内外SaaS软件厂商

SaaS（软件即服务）继续在商业世界拥有强大的吸引力，这是有原因的。SaaS也被称为按需软件、托管软件或基于Web的软件，它避开了传统的软件需要数据团队安装、维护和管理的流程，而倾向于通过Internet交付基于云的应用程序。有了SaaS，软件厂商将全程负责维持软件的安全、可用性和高性能，非常适合那些打算削减IT工作量和部署成本的企业。

国内外SaaS厂商盘点

\*\*1.微软公司：\*\*微软是全球有名的OS与应用软件提供商，虽然介入这个领域比较迟，但是在云服务方面，其云计算Azure为众多的Windows平台开发者提供支持的同时开始涉足非 Windows平台，希望借此Amazon展开竞争，有传闻说Azure将很快支持Linux系统。

在这里插入图片描述

\*\*2.北京金山：\*\*金山的WPS提供的是一站式云端办公解决方案，它不仅包含Office专业套件，还提供了专业的在线文档管理平台、企业内部即时通讯、企业邮箱等现代化办公所需的一整套基础云服务。有了WPS+云办公，无需IT基础架构投入，一键即可将办公室搬入云端。

\*\*3.厦门吉鼎：\*\*吉鼎科技是国内低代码Paas云平台的领航者及Saas化软件应用服务商，一直专注软件技术的深度研发，在Saas化软件应用、APP开发、技能培训等方面形成专业的产品服务，业务全面涉及政务、商贸、餐饮、金融、教育、旅游、电商、医疗等各行各业。

\*\*4.Vmvare：\*\*Vmvare是全球最大的虚拟化软件提供商，而虚拟化技术是云计算发展最关键的技术之一，VMWare自己不提供云服务，但是其vCloud 专为企业和服务提供商设计一组通用云计算服务，它支持所有应用程序或操作系统，并能够选择应用程序的运行位置。

\*\*5.Salesforce：\*\*Salesforce是全球按需 CRM 解决方案的领导者，通过以2.12亿美元收购云计算创业公司Heroku后，Salesforce在云计算领域极具发展优势，势头一直都比较猛。

\*\*6.Citrix：\*\*它是一家应用服务软件方案提供商，1年前以2亿多美元收购创业公司Cloud.com后，迅速进入云计算服务领域。其CloudStack平台已经成为vCloud平台的强大竞争对手。

[PaaS领域的五大玩家 - 知乎 (zhihu.com)](https://zhuanlan.zhihu.com/p/49506901)

为了表明PaaS在实际环境下如何工作，我列出并介绍了本人认为目前市面上的五大PaaS提供商，以及各自的独特优势。

红帽OpenShift

红帽OpenShift基于开源应用程序，提供一系列广泛的语言、数据库和组件。PaaS可高度定制，提供了三种形式：

OpenShift Online（一种基于云的托管服务）。

OpenShift Enterprise（一种在你的数据中心中运行的私有PaaS）。

OpenShift Origin（开源应用程序托管平台）。

OpenShift可以自动化处理诸多系统管理任务，比如虚拟服务器配置、设置和扩展，并支持用于管理代码的Gi代码库。

功能特性：

OpenShift Online

让你可以专注于代码和创新，而不是专注于基础设施的配置和管理，因而缩短了构建和部署应用程序所需的时间。

为你提供了种类多样的编程语言、框架和运行时环境，包括Java EE6及JBoss EAP。

充分利用开源平台和基于标准的组件，确保应用程序可移植，并消除了厂商锁定现象。

集成的开发工具和直观的界面让你能够迅速上手。没有新的编程模型，没有应用程序变动，也没有云锁定。

OpenShift Enterprise

这是一种本地的私有平台即服务（PaaS）解决方案，让你可以更快速地交付应用程序，并满足贵企业不断增长的应用程序需求。借助红帽公司屡获殊荣的私有平台即服务：OpenShift Enterprise，你可以改进开发人员的工作效率，提升运营效率，并提高硬件使用率。

功能特性：

Web控制台、命令行或集成开发环境（IDE）。

Java（EE6）、Ruby、PHP、Python和Perl。

公共云、私有云或混合云。

需要时，可实现应用程序自动扩展。

OpenShift Origin

功能特性：

构建自己的PaaS

支持众多语言运行时环境和数据层，包括Java EE6、Ruby、PHP、Python、Perl、MongoDB、MySQL和PostgreSQL。

值得注意的其他云平台

我觉得写一篇云计算方面的文章不能不提到AWS。与Azure一样，AWS可能也被归类为既是IaaS，又是PaaS。但是由于亚马逊的服务范围已大大拓宽，远远超出了PaaS的范畴，我决定本文中不提到它。

我可以为大家推荐的其他服务包括Engine Yard和Caspio。

Engine Yard在AWS上运行其平台，所以其PaaS价值更多地体现在编排和管理上，而不是体现在提供软件组件上。

Caspio与其他PaaS提供商有点不一样，原因在于它并不提供一个功能完备的软件开发环境，而是致力于将类似数据库的桌面功能引入到云端。Caspio是为构建基本数据库，提供数据录入表单和报表生成而设计的。

2. 百度应用引擎(http://developer.baidu.com/service)

Baidu App Enginee（BAE），是百度推出的网络应用开发平台。用户可基于BAE平台进行PHP、Java应用的开发、编译、调试、发布。同时BAE平台也提供了若干云服务，包括fetchURL、task queue、SQL。目前，BAE尚处于公测阶段，许多性能和服务还亟待完善。

3. 阿里云（http://www.aliyun.com）

Aliyun Cloud Enginee（ACE），是阿里推出的一个基于云计算基础架构的网络应用程序托管环境，帮助应用开发者简化网络应用程序构建和维护，并根据应用访问量和数据存储增长进行扩展。ACE支持PHP语言编写的应用程序；支持在线创建MYSQL远程数据库应用。

4. 腾讯开放平台(http://open.qq.com/reg)

腾讯开放平台，是腾讯基于其拥有的各大社交平台推出的应用开放平台。用户可将开发好的游戏、网站等应用一次性同时接入QQ空间、朋友网、腾讯微博、Q+平台，让开发和运营流程更简单、更安全。

5.J2Paas低代码平台(https://bbs.jikaiyun.com/index)

云计算服务不仅是互联网巨头的盛宴，也势必涌现一批快速成长的IT新贵，J2Paas云平台即是深具潜力的PaaS后起之秀。J2Paas低代码平台，是厦门吉鼎科技于2016年正式发布的一款业界领先的低代码PaaS云平台。既能为软件应用提供安全、稳定、高效的运行环境，同时还提供了功能全面的JJ Studio开发工具包，方便用户迅速开发出自己的SaaS化应用！

6. Microsoft Windows Azure

Windows Azure是微软的云计算平台，其主要目标是帮助开发者开发可运行在云服务器、数据中心、Web和PC上的应用程序。开发者能使用微软全球数据中心的储存、计算能力和网络基础服务。 Azure服务平台包括了以下主要组件：Windows Azure;Microsoft SQL数据库服务、Microsoft .Net服务;用于分享、储存和同步文件的Live服务等。

7. VMware Cloud Foundry

Cloud Foundry是VMware的一项开源PaaS计划，使用各种开源开发框架和中介软件，来提供PaaS服务。开发者可以通过这个平台来建设自己的SaaS的服务，不用自行建设和维护硬体服务器和中介软件。由于Cloud Foundry采用开源的网站平台技术，所以开发者的应用程序也可以任意转移到其他平台上而不受限于PaaS的平台。 目前Cloud Foundry可以支多种开发框架，包括Spring for Java、Ruby on Rails、Node.js以及多种JVM等。

8. http://Force.com

http://Force.com是企业云计算公司http://Salesforce.com的社会化企业应用平台，允许开发者构建具有社交和移动特性的应用程序。另外，它还提供了有助于在云上更快建立及运行业务应用程序的所有功能，包括数据库、强劲分析、实时工作流程及审批、可编程云逻辑及网站功能等。http://Force.com支持Apex编程语言，开发者可以基于UI层面编写数据库触发器和程序控制器。

9. Amazon Elastic Beanstalk

Elastic Beanstalk为在Amazon Web Services云中部署和管理应用提供了一种方法。该平台建立如面向PHP的Apache HTTP Server和面向Java的Apache Tomcat这样的软件栈。开发人员保留对AWS资源的控制权，并可以部署新的应用程序版本、运行环境或回滚到以前的版本。CloudWatch提供监测指标，如CPU利用率、请求计数、平均延迟等。

可以关注一下K2 BPM PaaS，是以PaaS架构为基础的企业BPM平台解决方案。基于微服务架构对BPM平台类应用进行充分解耦，将BPM全生命周期的管理落地、流程应用开发&运维的最佳实践，以一组通用组件服务的方式呈现给用户，以便能够更快更好更简单地搭建业务流程。云原生理念的BPMS重构基于BPM PaaD流程平台，实现流程全生命周期敏捷管理，助力企业流程管理及优化。微服务架构设计基于微服务架构对应用进行充分解耦，实现业务能力服务化共享，服务沉淀和中台设计。容器化部署标准化应用部署和交付，有效保障性能需求的同时，提升资源使用效率，助力企业业务互联网化。DevOps集成DevOps工具，开发、测试和运维一体化，助力企业IT敏捷化转型。

基于代码的还是容器云