

## 02 | 预习篇 · 小鲸鱼大事记（二）： 崭露头角

A close-up photograph of a white, star-shaped flower with yellow stamens, surrounded by green foliage and other buds. The flower is in sharp focus, while the background is a soft, out-of-focus green. Several green buds are visible on thin stems around the main flower.



讲述：张磊 大小：3.14M

在上一篇文章中，我说到，伴随着 PaaS 概念的逐步普及，以 Cloud Foundry 为代表的经典 PaaS 项目，开始进入基础设施领域的视野，平台化和 PaaS 化成了这个生态中的一个最为重要的进化趋势。

就在对开源 PaaS 项目落地的不断尝试中，这个领域的从业者们发现了 PaaS 中最为棘手也最亟待解决的问题：究竟如何给应用打包？

遗憾的是，无论是 Cloud Foundry、OpenShift，还是 Clodify，面对这个问题都没能给出一个完美的答案，反而在竞争中走向了碎片化的歧途。

而就在这时，一个并不引人瞩目的 PaaS 创业公司 dotCloud，却选择了开源自家的一个容器项目 Docker。更出人意料的是，**就是这样一个普通到不能再普通的技术，却开启了一个名为“Docker”的全新时代。**

你可能会疑问，Docker 项目的崛起，是不是偶然呢？

**事实上，这个以“鲸鱼”为注册商标的技术创业公司，最重要的战略之一就是：坚持把“开发者”群体放在至高无上的位置。**

相比于其他正在企业级市场里厮杀得头破血流的经典 PaaS 项目们，Docker 项目的推广策略从一开始就呈现出一副“憨态可掬”的亲民姿态，把每一位后端技术人员（而不是他们的老板）作为主要的传播对象。

简洁的 UI，有趣的 demo，“1 分钟部署一个 WordPress 网站”“3 分钟部署一个 Nginx 集群”，这种同开发者之间与生俱来的亲近关系，使 Docker 项目迅速成为了全世界 Meetup 上最受欢迎的一颗新星。

在过去的很长一段时间里，相较于前端和互联网技术社区，服务器端技术社区一直是一个相对沉闷而小众的圈子。在这里，从事 Linux 内核开发的极客们自带“不合群”的“光环”，后端开发者们啃着多年不变的 TCP/IP 发着牢骚，运维更是天生注定的幕后英雄。

而 Docker 项目，却给后端开发者提供了走向聚光灯的机会。就比如 Cgroups 和 Namespace 这种已经存在多年却很少被人们关心的特性，在 2014 年和 2015 年竟然频繁入选各大技术会议的分享议题，就因为听众们想要知道 Docker 这个东西到底是怎么一回事儿。

**而 Docker 项目之所以能取得如此高的关注，一方面正如前面我所说的那样，它解决了应用打包和发布这一困扰运维人员多年的技术难题；而另一方面，就是因为它第一次把一个纯后端的技术概念，通过非常友好的设计和封装，交到了最广大的开发者群体手里。**

在这种独特的氛围烘托下，你不需要精通 TCP/IP，也无需深谙 Linux 内核原理，哪怕只是一个前端或者网站的 PHP 工程师，都会对如何把自己的代码打包成一个随处可以运行的 Docker 镜像充满好奇和兴趣。

这种受众群体的变革，正是 Docker 这样一个后端开源项目取得巨大成功的关键。这也是经典 PaaS 项目想做却没有做好的一件事情：PaaS 的最终用户和受益者，一定是为这个 PaaS 编写应用的开发者们，而在 Docker 项目开源之前，PaaS 与开发者之间的关系却从未如此紧密过。

**解决了应用打包这个根本性的问题，同开发者与生俱来的亲密关系，再加上 PaaS 概念已经深入人心的完美契机，成为 Docker 这个技术上看似平淡无奇的项**

## 目一举走红的重要原因。

一时之间，“容器化”取代“PaaS 化”成为了基础设施领域最炙手可热的关键词，一个以“容器”为中心的、全新的云计算市场，正呼之欲出。而作为这个生态的一手缔造者，此时的 dotCloud 公司突然宣布将公司名称改为“Docker”。

这个举动，在当时颇受质疑。在大家印象中，Docker 只是一个开源项目的名字。可是现在，这个单词却成了 Docker 公司的注册商标，任何人在商业活动中使用这个单词，以及鲸鱼的 Logo，都会立刻受到法律警告。

那么，Docker 公司这个举动到底卖的什么药？这个问题，我不妨后面再做解读，因为相较于这件“小事儿”，Docker 公司在 2014 年发布 Swarm 项目才是真正的“大事儿”。

那么，Docker 公司为什么一定要发布 Swarm 项目呢？

通过我对 Docker 项目崛起背后原因的分析，你应该能发现这样一个有意思的事实：虽然通过“容器”这个概念完成了对经典 PaaS 项目的“降维打击”，但是 Docker 项目和 Docker 公司，兜兜转转了一年多，却还是回到了 PaaS 项目原本深耕了多年的那个战场：**如何让开发者把应用部署在我的项目上。**

没错，Docker 项目从发布之初就全面发力，从技术、社区、商业、市场全方位争取到的开发者群体，实际上是为此后吸引整个生态到自家“PaaS”上的一个铺垫。只不过这时，“PaaS”的定义已经全然不是 Cloud Foundry 描述的那个样子，而是变成了一套以 Docker 容器为技术核心，以 Docker 镜像为打包标准的、全新的“容器化”思路。

**这，正是 Docker 项目从一开始悉心运作“容器化”理念和经营整个 Docker 生态的主要目的。**

而 Swarm 项目，正是接下来承接 Docker 公司所有这些努力的关键所在。

## 总结

今天，我着重介绍了 Docker 项目在短时间内迅速崛起的三个重要原因：

1. Docker 镜像通过技术手段解决了 PaaS 的根本性问题；
2. Docker 容器同开发者之间有着与生俱来的密切关系；
3. PaaS 概念已经深入人心的完美契机。

崭露头角的 Docker 公司，也终于能够以一个更加强硬的姿态来面对这个曾经无比强势，但现在却完全不知所措的云计算市场。而 2014 年底的 DockerCon 欧洲峰会，则正式拉开了 Docker 公司扩张的序幕。

思考题

1. 你是否认同 dotCloud 公司改名并开启扩张道路的战略选择？
2. Docker 公司凭借“开源”和“开发者社群”这两个关键词完成崛起的过程，对你和你所在的团队有什么启发？

精选留言(25)

日拱一卒

1. 公司改名无可厚非，项目开源和商业牟利不应该是冲突的。
2. Docker带来的启示：任何项目或者技术都应该是以用户为中心，找准目标人群，深挖用户痛点，通过用户最能接受的方式，去解决问题。

作者回复: 说的一点没错，用户才是项目和技术的衣食父母，脱离群众要不得

2018-08-29

□

□ 35

龙坤

你好，张老师，读到这里，毕竟我没经历过旧PaaS时代的项目，但我试图去挖掘和感受那个没有docker项目的年代，在PaaS平台上，一个团队是如何把应用开发、应用部署、应用扩展、应用分布式、应用维护与监控等一系列操作表现与实施出来？暂且到读到这里，了解到Docker项目仅仅解决了应用打包的困境，能让开发者省去上云后带来的系统差异性所导致的种种问题，可对于应用来说，始终都要回到PaaS的基础设施上，然后去实现多主机通信、存储共享、权限控制等等这些Docker项目没出现前都要的基本技术实施。那么我的的问题是：docker项目的出现，带来应用打包优势同时，是否将来会增加关于网络、存储、权限等额外技术栈应用的复杂度？

2019-03-21

□

□ 10

Cloud\*

这个改名真是太正确了，一下品牌就火了。开源永远都是最好的选择，首先得让开发者认可你，才能走得长远。

2018-08-29

□

□ 9

cxyfreedom

如果按天时地利人和三方面来说，总结的第一点就是地利，第二点是人和，第三点就是天时。另外改名这事，我觉得这是作为一个运营公司从商业角度需要考虑的，没什么问题。

2018-09-01

□

□ 4

blackpiglet

对于 dotcloud 改名和扩张，我想当时他们可能没有其他选择，毕竟是创业公司，是有盈的压力的。docker 的开源不易带来直接的收入，所以必然要有下一步。

2018-08-29

□

□ 3

vimfun

我觉得更好一点的顺序为：

1. PaaS 概念已经深入人心的完美契机。（天时）
2. Docker 镜像通过技术手段解决了 PaaS 的根本性问题；（地利 自我创新出来的地利）
3. Docker 容器同开发者之间有着与生俱来的密切关系；（人和）

时势造英雄啊

2019-08-14

□

□ 1  
abs

请问下swarm和k8s有什么关联和区别？为什么我知道很多公司用k8s，却很少听说有公司用swarm的？

作者回复: 其实就是主流方案与非主流方案的区别。至于为啥kubernetes 变成主流了，咱们专栏前四篇讲的就是这个故事

2018-09-08

□

□ 1  
Dale

互联网公司的玩法，需要寻找商业化的点子，改名字肯定可以吸引更多的人关注，使用的人多了，流量上来了，就有机会变现。

2018-09-04

□

□ 1  
Geek\_49d8fa

我认为在“一统江山”之前，改名是个“自费武功”的操作。  
我认为：先实现简单易用，再去开源，人们愿意用开源，都是基于“成本”。

2019-08-07

□

□

心灵捕手

在技术的发展上，以用户为核心，并且能提高开发效率，提高用户体验的技术必将引领潮流，成为新时代的弄潮儿。当然，开源生态同样重要

2018-12-09

□

□

Geek

最有道理的莫过于 docker的镜像 以及打包  
解决了运维一遍又一遍的重复 无价值的工作  
让运维的工作从幕后走到前台

2018-11-22