

## 深入浅出Docker (三) : Docker开源之路

### 1. 背景

Docker从一开始的概念阶段就致力于使用开源驱动的方式来发展，它的成功缘于国外成熟的开源文化氛围，以及可借鉴的社区运营经验。通过本文详细的介绍，让大家可以全面了解一个项目亦或者一项技术是如何通过开源的方式发展起来的。为了更准确的描述Docker的社区状况，请先看一份来自 Docker官方的数据：

### Docker, the community

- >500 contributors
- ~20 core maintainers
- >8,000 Dockerized projects on GitHub
- >20,000 repositories on Docker Hub
- >250 meetups in >90 cities in >30 countries
- >500,000 downloads of boot2docker

图中数据的看点有：

1. 超过500个代码贡献者。代码的贡献者在社区发展过程中是非常重要的催化剂，它会不断加快产品迭代的速度，让项目更快的交付到最终用户的手里。
2. 20个全职开发。一般的开源项目一般都不会有如此多的全职开发人员，但是Docker却有20个全职开发人员来驱动一个社区项目。这也从侧面证明了国外公司对开源项目的支持力度，以至于一家初创公司都舍得投入20个全职开发来推动开源技术的发展。
3. 超过8000个创建在GitHub上的Docker相关项目。通过这个规模可以看到围绕Docker可以使用的项目已经非常丰富。
4. Docker技术聚会。30个国家超过90个城市举办超过250个Docker技术聚会，这样的技术聚会还在不断增加，Docker技术爱好者可以在线申请承办此类技术聚会。
5. 50万次的boot2docker下载。boot2docker为Docker官方推荐客户端，50万次也代表当前潜在的用户群体。

### Docker Inc, the company

- Headcount: ~50
- Led by Open Source veteran Ben Golub (GlusterFS)
- Revenue:
  - t-shirts and stickers featuring the cool blue whale
  - SAAS delivered through Docker Hub
  - Support & Training

Docker公司目前正式员工50多名，由开源老手Ben Golub(前GlusterFS CEO)主持运营。现在主要有以下几个赢利点：

1. 印有蓝鲸的T恤和贴纸的品牌价值
2. 通过Docker Hub服务提供SaaS的分发服务
3. 提供Docker技术支持和培训

通过了解Docker的社区现状和公司的运营状况，可以发现其在运营的方式上并没有什么特别的地方。除了支付公司员工的正常开支外，它并没有像一般商业公司那样在推广上投入很多资金。Docker在推广上主要是将开源社区和社交网络作为基础推广平台，结合全球范围的Docker技术聚会，形成了良好的良性的客户互动和口口相传的品牌效应。在Docker的开源历程中，通过分析观察用户社区、源代码管理、合作伙伴的生态圈这三种形式，梳理出一套实践经验的案例，方便大家参考学习。

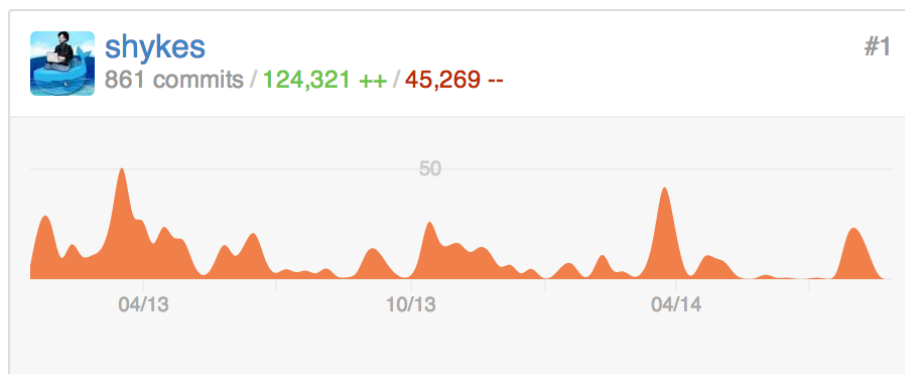
## 2. 用户社区维护

Docker技术首先考虑的是在技术社区里通过各种渠道来找到它的用户，而当前最流行的技术社交网络不是Twitter，而是GitHub。所以 Docker第一步是向GitHub上提交自己的代码开始吸引自己的用户的。我们总说万事开头难，那么Docker的第一个Commit应该是什么，它是否需要包括测试、用户文档、用户开发指南、设计理念等一系列的文档和代码呢？如果参照常规的开源项目，它们都考虑的都很周到，完善的代码文档结构会让用户 第一眼就知道这是一个成熟的项目，我们只要用就可以了。但Docker公司却不按套路出牌，第一个Commit仅包括6个主文件：

```
$ git log
commit a27b4b8cb8e838d03a99b6d2b30f76bdaf2f9e5d
Author: Andrea Luzzardi <aluzzardi@gmail.com>
Date:   Fri Jan 18 16:13:39 2013 -0800

    Initial commit
dxiao at dhcp-141-62 in ~/Documents/Code/go/docker on test
$ ll
total 88
drwxr-xr-x  2 dxiao  staff   68 Aug 21 17:36 bundles
-rw-r--r--  1 dxiao  staff  4573 Aug 27 15:41 container.go
-rw-r--r--  1 dxiao  staff  3530 Aug 27 15:41 container_test.go
-rw-r--r--  1 dxiao  staff  2509 Aug 27 15:41 docker.go
-rw-r--r--  1 dxiao  staff  3856 Aug 27 15:41 docker_test.go
-rw-r--r--  1 dxiao  staff  1068 Aug 27 15:41 filesystem.go
-rw-r--r--  1 dxiao  staff   766 Aug 27 15:41 filesystem_test.go
-rw-r--r--  1 dxiao  staff  2520 Aug 27 15:41 lxc_template.go
-rw-r--r--  1 dxiao  staff   738 Aug 27 15:41 state.go
-rw-r--r--  1 dxiao  staff  2041 Aug 27 15:41 utils.go
-rw-r--r--  1 dxiao  staff  2901 Aug 27 15:41 utils_test.go
```

没有README，没有开发环境指南，开始阶段用户无法有效的了解Docker项目。但是，这其实也是对的，因为在一个小众的开源项目的初期，很难吸引到社区用户来为它贡献代码。在接下来的很长一段时间，Docker主要是由Docker之父Solomon Hykes开发维护。在没有社区用户参与的情况下，他每天不分昼夜地提交代码，也许是为了生活，也许是为了爱好，Solomon Hykes在拼命的实现心中那个目标：让LXC创建的容器更容易使用。用当下时髦的那句话讲就是不忘初心。



从 Solomon Hykes 贡献代码的趋势图，我们可以看到只有保证项目的活跃度（持续贡献代码）那么项目才有可能获得用户的认可和关注。试想，“三天打鱼，两天晒网”的开源项目会 给用户怎么样的感觉？作者都不专注，用户怎么可能忠实？这种投入，并不是偶然性的，这是国外业界的开源文化，非常值得我们学习。

在代码贡献的同时，Docker.io的主站也在2013年的5月30日第一次提交到GitHub。这距离第一行代码提交到GitHub已经有5个月。总体来看，Docker从一开始仅仅是一个很酷的想法，到真正完成并可以对外发布版本，也是经历了小半年之久。开源的意义并不像国内大多数厂商发布开源项目一样，一定要在内部应用的很成功才发布到技术社区。我们通过以上数据，可以很容易的看出来，它一定是一个长期的、持续的过程。你越专注的投入，你的项目越有可能成功。

当然，如果仅仅Docker只在GitHub上提供代码来吸引用户，那也不会是最佳的实践。在网站建立后的下一步，它开始在全球各大城市的技术聚会上推广介绍Docker。Docker技术在全球推广的方法是通过线下的技术聚会，外加社交网络媒体传播才得到流行的。Docker官网通过发布“[创建Docker技术聚会指南](#)”，引导Docker技术的爱好者自发申请承办技术聚会。Docker官网主要通过类似Twitter、Docker周报、用户论坛等渠道，及时的把每个城市即将开始的技术聚会时间和联系人发布出来，就可以把Docker的技术推广做下去。并且这种形式的最大好处是口碑相传，其长尾效应的结果是Docker技术开始在当地的技术圈得到关注。像这样的技术聚会，一般都在30到200人之间，对于商业项目的推广是无法参照这样推广的。但这种形式就像一颗种子散落在城市中间，一旦有人介绍Docker技术到这个城市的技术圈，就会有更多的用户去GitHub上关注Docker的项目进展。

### 3. 源代码管理

Docker的源码是按照模块划分的，每一个模块都会单独放在一个目录里。并且每个目录都会包含一个MAINTAINERS.md。当你提交一个Pull Request的时候，子模块的维护者就会站出来回答“LGTM (Looks Good To Me)”表示已经看了你的代码并接受你提交的代码。

每一个想参与Docker开发的开发者，你需要进入一个特殊的目录hack，这个目录包括了所有你想知道的开发相关信息。它把开发者分为4类人，

1. 代码贡献者，详细指南看CONTRIBUTORS.md。
2. 代码维护者，详细指南看MAINTAINERS.md。
3. 程序打包者，详细指南看PACKAGERS.md。
4. 负责发布版本的维护者，详细指南看RELEASE-CHECKLIST.md。

所有的指南文档都非常详细完整，可以做到看完此文档就可以开始相应角色的任务。除了这些文档之外，hack目录还包括了发布，开发，测试等环节需要的辅助脚本，让开发者可以得到很多开发上的便利。

Docker从一开始就是围绕LXC开发的，但在版本稳定后，使用Go重写了一套类LXC接口实现。也就是[libcontainer](#)项目。它的代码管理方式与上面的Docker源代码管理方式一样，开发者可以很容易的导入到子项目libcontainer的开发。一致的管理方法可以提高流程的复用，这种管理方式希望能得到大家的借鉴参考。

### 4. 创建合作伙伴生态圈

首先，Docker从一开始使用Ubuntu版本作为开发环境，主要是Ubuntu上支持aufs(advanced multi layered unification filesystem)文件系统。所以，Docker对Ubuntu的支持是最好的。



还有就是大家非常熟悉的Google的GCE，在没有Docker之前，就已经使用自己开发的类LXC容器。在有了Docker之后，Google开发了kubernetes来管理Docker。



商业版本Linux的领导者RedHat，也是Docker生态圈非常重要的合作伙伴。并且它支持的Fedora社区、CentOS社区会对推广Docker技术起到关键性作用。



还有就是OpenStack的建立者Rackspace，在混合云解决方案上具有全球领导地位。Docker与它的合作，可以帮助Docker技术在混合云的解决方案中得到推广。



最后就是开源PaaS项目DEIS，基于Docker和CoreOS技术构建的类如Heroku发布流程的PaaS应用。与它的合作，可以让更多的企业看到一个使用Docker技术的范本。



## 5. 结论



Docker的开源之路可以说是开源项目的最佳实践，它从屌丝瞬间变为高富帅的历程并不是偶然。Docker的开源之路还在进行中，从它的模式中我们可以学习到的东西并不仅仅局限于以上罗列的几点，读者可以在参与开源项目的过程中多体会，找到自己的“痛点”，然后参考Docker等开源项目的做法，相信可以很快解决你的问题。

## 6. 作者简介

肖德时, Red Hat Engineering Service/HSS 内部工具组Team Leader. Nodejs开源项目nodejs-cantas Lead Developer。擅长企业内部工具的设计以及实现。开源课程Rails Starter的发起人。rubygem: lazy\_high\_charts的Maintainer。twitter账号：xds2000，邮箱：xiaods@gmail.com

## 7. 下期预告

Docker技术基本信息已经介绍的差不多了，接下来的系列将进入实战篇。下一篇将重点讲解Docker的集成测试部署方法，并结合众场景（数据库集成、测试、审查、部署）给出参考解决方案，敬请期待！