**Cloud Computing**

|  |  |
| --- | --- |
| **审核人** |  |
| **重要性** | 中 |
| **紧迫性** | 中 |
| **拟制人** | 张包峰 |
| **提交日期** |  |

**作者：张包峰**

(版权所有,翻版必究)

**修改记录**

|  |  |  |
| --- | --- | --- |
| **更新时间** | **变更内容** | **变更理由** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

[**作者：张包峰** 1](#_Toc394676903)

[1.1 Docker设计 4](#_Toc394676904)

[1.2 LXC 4](#_Toc394676905)

[1.3 Cgroups 4](#_Toc394676906)

[1.4 aufs 4](#_Toc394676907)

[1.5 源码分析 5](#_Toc394676908)

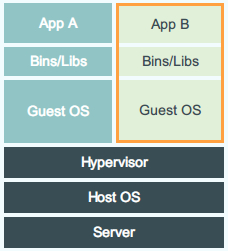
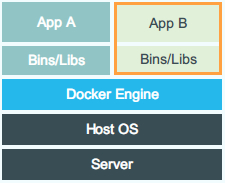
[1.6 with Spark 5](#_Toc394676909)

[1.7 参考资料 5](#_Toc394676910)

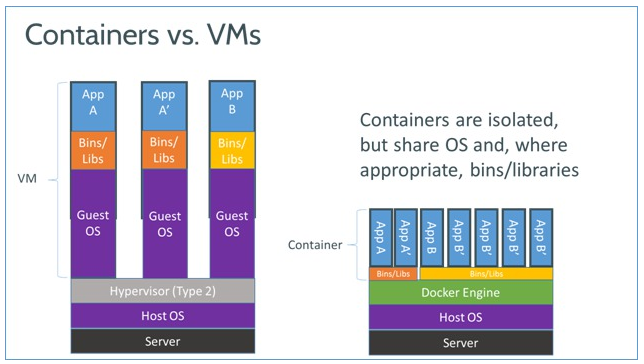
## Docker

先从**设计和功能**来理解Docker。

### 设计

[](http://cms.csdnimg.cn/article/201407/02/53b3c67030ae8.jpg) [[](http://cms.csdnimg.cn/article/201407/02/53b3c68844d34.jpg)](http://cms.csdnimg.cn/article/201407/02/53b3c68844d34.jpg)

Docker 几乎就没有什么虚拟化的东西，并且直接复用了 Host 主机的 OS，在 Docker Engine 层面实现了调度和隔离重量一下子就降低了好几个档次。 Docker 的容器利用了 LXC，管理利用了 namespaces 来做权限的控制和隔离， cgroups 来进行资源的配置，并且还通过 aufs 来进一步提高文件系统的资源利用率。



IaaS普遍使用虚拟机,开销较大。Docker明显更轻量。笔者认为IaaS和PaaS各有专攻，PaaS去专注于安全级别的隔离是没有意义的，IaaS也不应该去感知到应用。一个公有云中，可以让每个租户使用不同的虚拟机，虚拟网络来做到安全和资源上的隔离。再通过PaaS统一运维，管理计算资源。

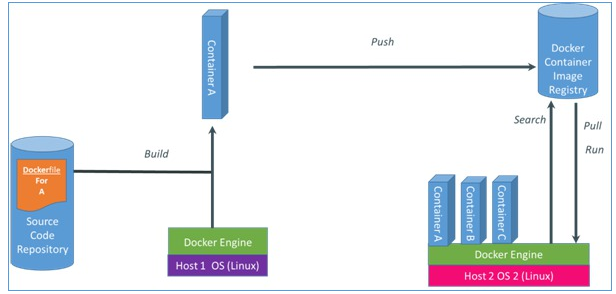
### 功能

Docker可以让开发和运维都变得简单。

开发者不必要像使用一般PaaS一样在充满着限制的条件下开发应用，可以就如同平常一样，自由的使用各种资源。老子说“太上下知有之 ，其次誉之，其次侮之”。

Dock对于开发者就是“下知有之”的状态。爽的人是运维。要使用Docker，需要在机器上安装Docker Engine，

1. 创建一个Container。实际上是一个Linux Container，Docker会将网络，存储这些事情都配好。
2. 下载应用并安装。比如可以用sudo apt-get install mysql-server 来安装一个MySQL。并配置一些参数什么的。
3. 打包上传Image。Docker可以将这个Linux Container打包成Image，启动脚本也在其中。并上传至Image Registry中。这个Image仅仅包括你修改的增量部分，所以体积比较小。
4. 一条命令跑起来。使用Docker Run 一条命令可以从Image Registry中下载Image 并跑起来。



需要重启的时候，只要重启Container。需要迁移的时候，只要迁移Container。一切干干净净

### LXC

### Cgroups

<https://www.kernel.org/doc/Documentation/cgroups/cgroups.txt>

### aufs

<http://en.wikipedia.org/wiki/Aufs>

### 源码分析

### with Spark

### 参考资料

5分钟弄懂Docker

<http://code.csdn.net/news/2820497>

深入浅出Docker(一)

<http://www.infoq.com/cn/articles/docker-core-technology-preview>

Docker能够运行任何应用的PaaS云

<http://www.yankay.com/docker-paas-for-any-application/>

## Heroku

公有云PaaS

## CloudFoundry

私有云PaaS

## KVM

虚拟机

## Xen

## OpenStack

## Amazon

## 阿里云