

从Docker到Kubernetes 第1周

DATAGURU专业数据分析社区

【声明】 本视频和幻灯片为炼数成金网络课程的教学资料，所有资料只能在课程内使用，不得在课程以外范围散播，违者将可能被追究法律和经济责任。

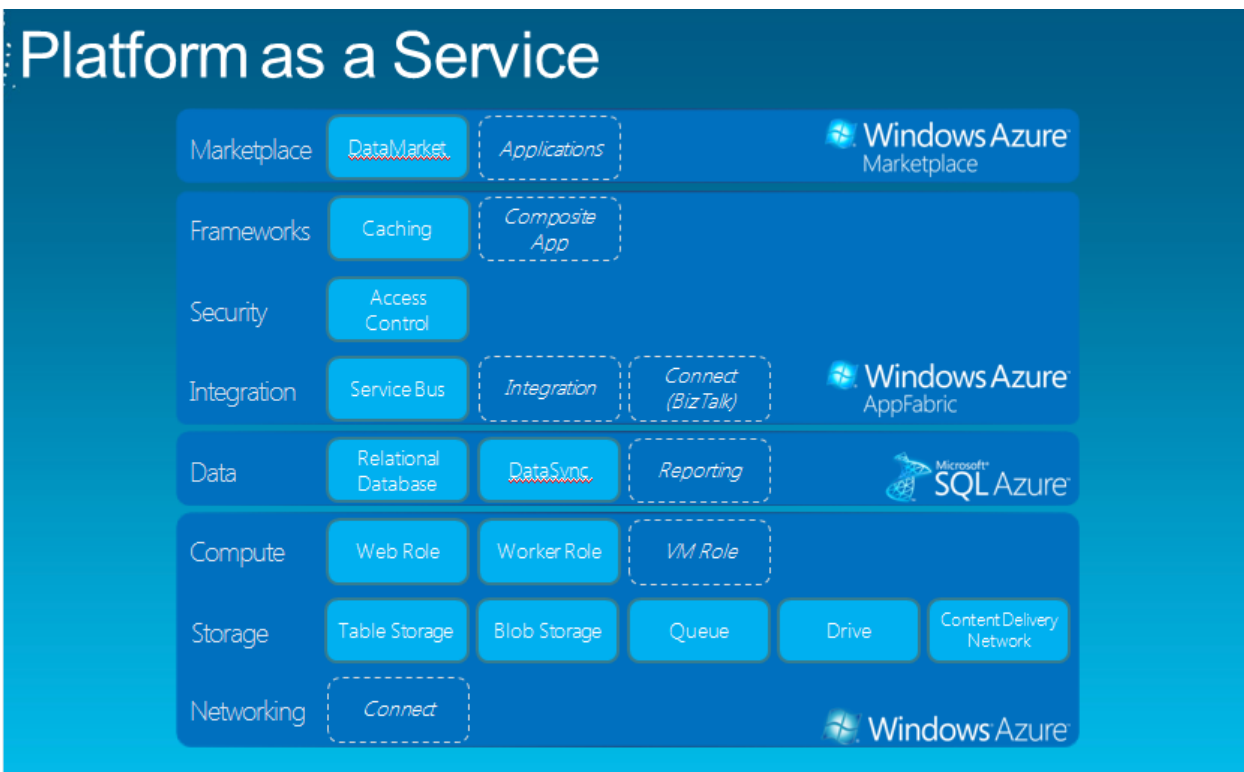
课程详情访问炼数成金培训网站

<http://edu.dataguru.cn>

Docker的历史与现状分析
Docker的技术原理介绍
Docker的基本概念
Docker学习必备基础技能

Docker的历史与现状分析

曾经我们都在做一个称之为PAAS的平台



PAAS平台的范围和内容

1. 确定产品定位和需求，确定首次迭代的范围。
2. 制作界面原型。
3. 技术选型，然后根据技术选型为每个开发者搭建开发环境和技术栈，例如 Java 环境、Python 环境、Ruby 环境、数据库、中间件等等。
4. 构建基础技术框架和服务，包括日志、存储、消息、缓存、搜索、数据源、集群扩展等等。
5. 模拟用户容量，构建测试环境。
6. 开始编写真正的业务代码，实现产品功能。
7. 迭代开发/测试，生生不息，周而复始，直到头发掉光为止……

PaaS平台的范畴

老一代的PAAS平台的局限性和困境

- 主要提供应用的部署和托管
- 针对应用开发者
- 仅支持特定的IaaS基础技术
- 支持单种开发语言和框架
- 支持特定的服务，比如自定义的数据存储APIs
- 没有很好的解决常用中间件的部署问题
- 难以解决应用和资源的隔离问题

Docker的历史与现状分析

新一代的PAAS平台

新一代的云应用平台技术则实现全方位的应用生命周期管理，关注开放性、应用的可移植性和云间相互操作性，其代表者包括**Cloud Foundry, OpenShift, Docker, Heroku, MoPaaS**等PaaS技术或服务，除了在第一代PaaS技术对用户为实现应用交付的加速所提供的功能外，具备以下大多数特征：

多语言和框架：支持多语言和框架以及语言框架的扩展机制

多服务：开放的核心服务以及服务的扩展机制

多云和多IaaS技术：支持多种IaaS技术和多云的部署，包括公有云和私有云



Docker究竟是什么？

Docker也被称之为第三代Paas平台

有人说Docker公司是最贵的卖T恤的公司

dotCloud 不仅支持诸如 PHP、MySQL 等传统技术框架，还包括 Node.js、MongoDB 等新兴技术。基于 dotCloud 提供的开发工具和技术框架，你可以直接使用 dotCloud 的 SDK 编写代码和构建业务服务，并在联网的时候把这些代码推送到云端，实现自动部署和测试。

Docker的历史与现状分析

Docker的诞生和幕后的公司

2010年，几个大胡子年轻人在旧金山成立了一家做 PaaS 平台的公司，起名为「dotCloud」，dotCloud主要是基于PaaS平台为开发者或开发商提供技术服务。

Docker于2013.03.27 正式作为public项目发布

DotCloud公司2013年10月改名为Docker Inc，转型专注于Docker引擎和Docker生态系统。

2014.1 被BLACK DUCK 评选为2013年10大开源新项目“TOP 10 OPEN SOURCE ROOKIE OF THE YEAR”

2014.9获取4000万美元融资，此时已经累计融资6600万美元

2015.4月获取了9500万美元融资，已经确立了在第三代PaaS市场的主导地位。

2015.6月DockerCon 2015大会上，Linux 基金会与行业巨头联手打造开放容器技术项目Open Container Project

落腮胡、皮夹克和摩托车，是Docker的灵魂人物CTO Solomon Hykes的标准记号，外型看起来像是推动社会运动的老大。



Docker的历史与现状分析

Docker生态圈



Docker已经收购了一系列创业公司，包括KiteMatic、Koality和Socketplane。完成新一轮投资后，Docker计划加快产品推进计划，在本季度推出Docker Hub企业版，并且在接下来几个月在存储、网络和安全三个产品方向发力。

红帽在新的RHEL 7版本中增添了支持Docker的功能，IBM公开拥抱Docker和容器，亚马逊推出了EC2容器服务，就连公认的竞争对手VMware也宣布支持Docker。

国内，不仅腾讯、阿里、百度、Ucloud、青云等云计算服务提供商已经将Docker用到实践，还有数家Docker初创企业在获得了天使投资后正在辛勤的开发，准备将产品尽快发布到市场上。

腾讯云计算公司对外宣布成为中国首家支持Docker Machine的云计算厂商，并将自身定位于Docker基础设施的服务商，迈出构建Docker“航运”基础设施的第一步。同时，在支持Docker Machine前提下，腾讯云也推出了常用系统的标准版Docker镜像，以方便用户能够一键便捷创建容器。

Docker的历史与现状分析

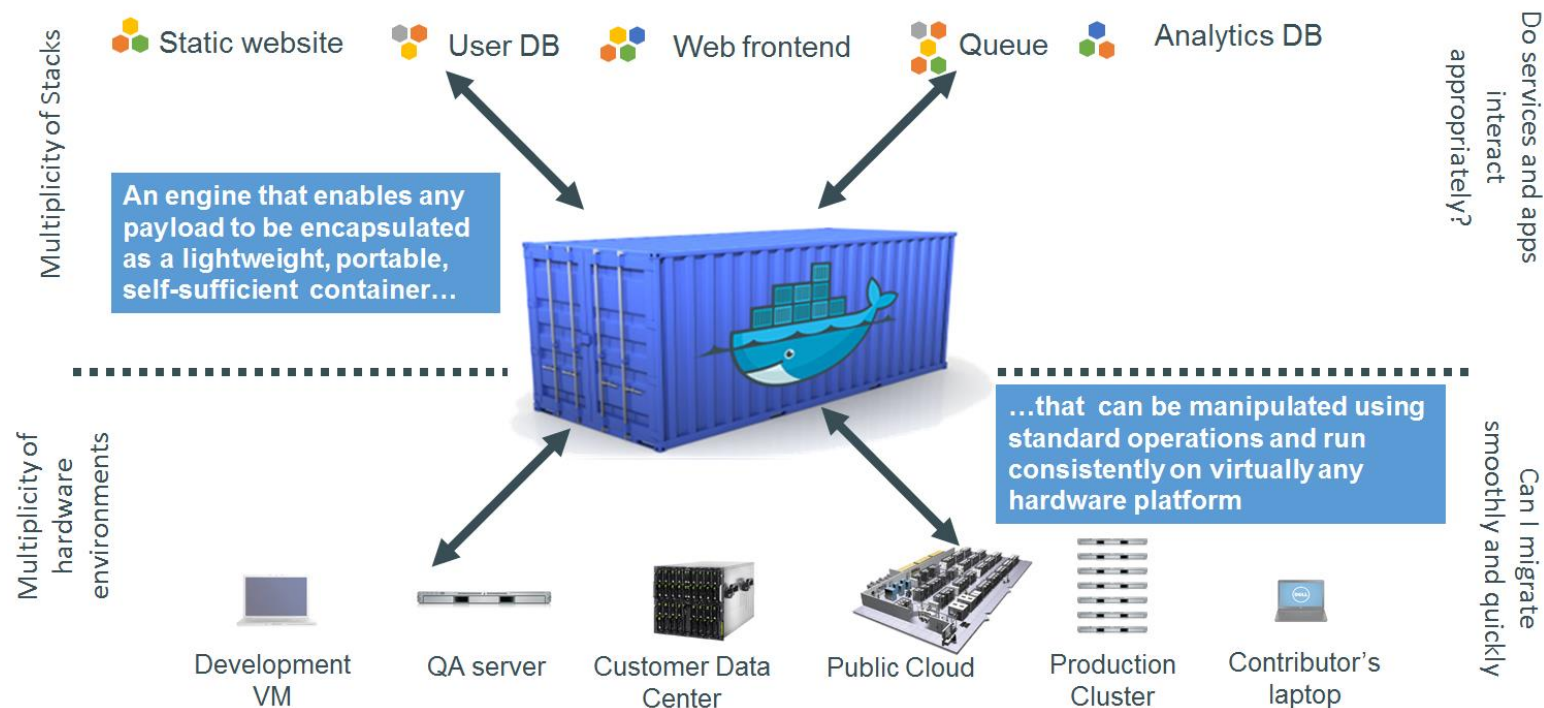
Docker生态圈，2014-2015发展

- 贡献者增长了183%;
- GitHub上关于Docker的项目增长了515%;
- Docker提供的工作机会增长了1720%;
- 使用Docker构建的应用程序增长了934%(Boot2Docker下载量增加了1456%);
- 容器的下载量增加了18082%(这个是根据DockerHub上镜像下载量来统计的)。
- Docker Hub业务的进展：用户数量增长至24万，Repo增长至15万，Pull的数量超过5亿次**

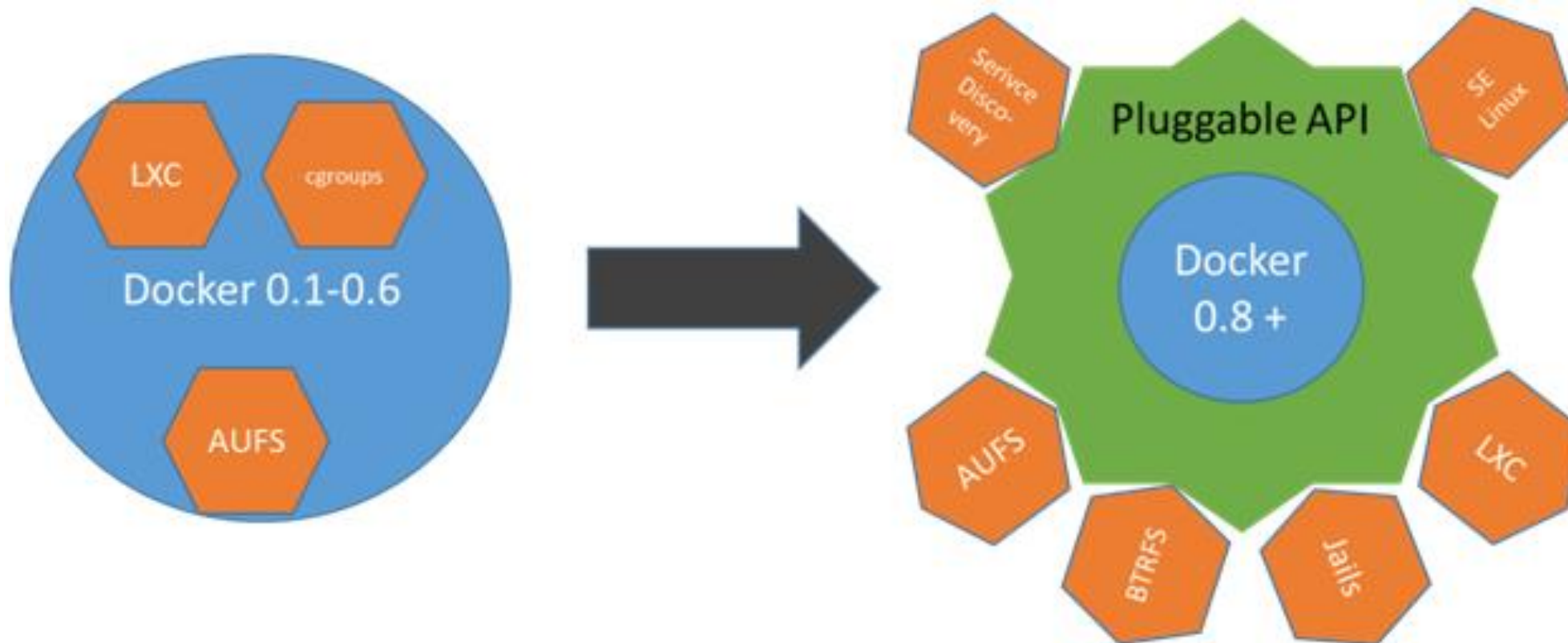
Docker的技术原理介绍

Docker就是虚拟化的一种轻量级替代技术。Docker的容器技术不依赖任何语言、框架或系统，可以将App变成一种标准化的、可移植的、自管理的组件，并脱离服务器硬件在任何主流系统中开发、调试和运行

简单的说就是，在 Linux 系统上迅速创建一个容器（类似虚拟机）并在容器上部署和运行应用程序，并通过配置文件可以轻松实现应用程序的自动化安装、部署和升级，非常方便。因为使用了容器，所以可以很方便的把生产环境和开发环境分开，互不影响，这是 docker 最普遍的一个玩法。



Docker相关的核心技术



Docker相关的核心技术之cgroups

Linux系统中经常有个需求就是希望能限制某个或者某些进程的分配资源。于是就出现了cgroups的概念，cgroup就是controller group，在这个group中，有分配好的特定比例的cpu时间，IO时间，可用内存大小等。cgroups是将任意进程进行分组化管理的Linux内核功能。最初由google的工程师提出，后来被整合进Linux内核中。

cgroups中的重要概念是“子系统”，也就是资源控制器，每种子系统就是一个资源的分配器，比如cpu子系统是控制cpu时间分配的。首先挂载子系统，然后才有control group的。比如先挂载memory子系统，然后在memory子系统中创建一个cgroup节点，在这个节点中，将需要控制的进程id写入，并且将控制的属性写入，这就完成了内存的资源限制。

cgroups 被Linux内核支持，有得天独厚的性能优势，发展势头迅猛。在很多领域可以取代虚拟化技术分割资源。cgroup默认有诸多资源组，可以限制几乎所有服务器上的资源：cpu mem iops,iobandwide,net,device acess等

Docker相关的核心技术之LXC

LXC是Linux containers的简称，是一种基于容器的操作系统层级的虚拟化技术。借助于namespace的隔离机制和cgroup限额功能，LXC提供了一套统一的API和工具来建立和管理container。LXC跟其他操作系统层次的虚拟化技术相比，最大的优势在于LXC被整合进内核，不用单独为内核打补丁

LXC 旨在提供一个共享kernel的 OS 级虚拟化方法，在执行时不用重复加载Kernel，且container的kernel与host共享，因此可以大大加快container的启动过程，并显著减少内存消耗，容器在提供隔离的同时，还通过共享这些资源节省开销，这意味着容器比真正的虚拟化的开销要小得多。在实际测试中，基于LXC的虚拟化方法的IO和CPU性能几乎接近 baremetal 的性能。

虽然容器所使用的这种类型的隔离总的来说非常强大，然而是不是像运行在hypervisor上的虚拟机那么强壮仍具有争议性。如果内核停止，那么所有的容器就会停止运行。

- 性能方面：LXC>>KVM>>XEN
- 内存利用率：LXC>>KVM>>XEN
- 隔离程度：XEN>>KVM>>LXC

Docker相关的核心技术之AUFS

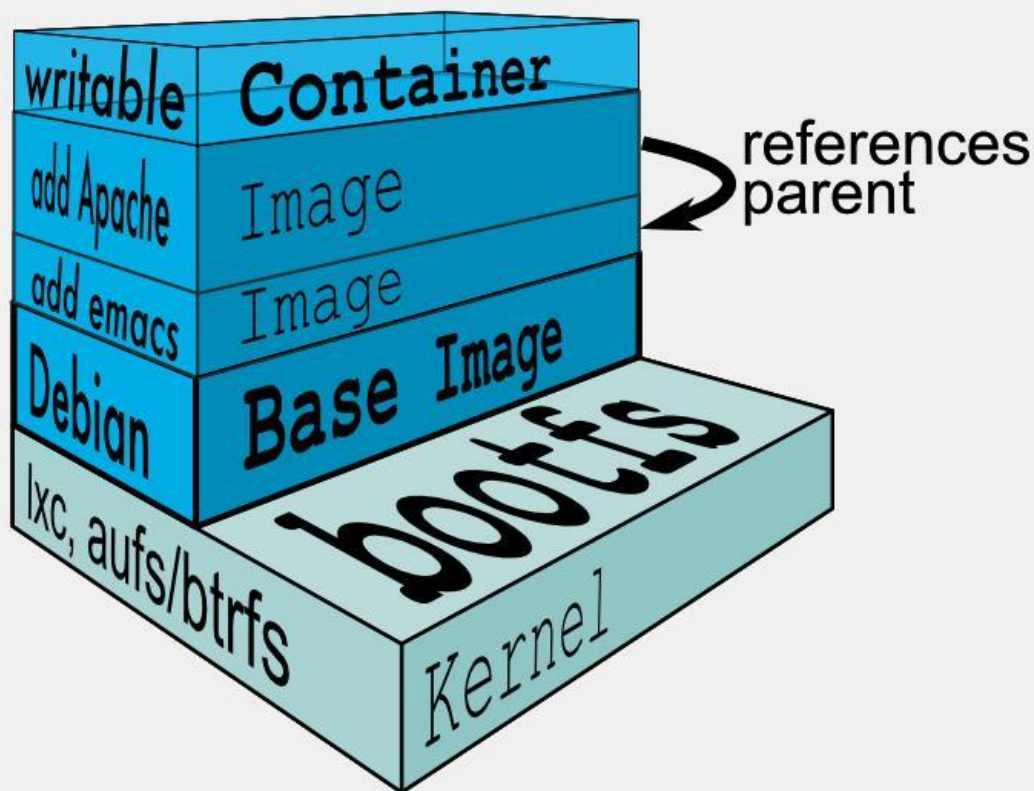
什么是AUFS? AuFS是一个能透明覆盖一或多个现有文件系统的层状文件系统。支持将不同目录挂载到同一个虚拟文件系统下,可以把不同的目录联合在一起,组成一个单一的目录。这种是一种虚拟的文件系统,文件系统不用格式化,直接挂载即可。

Docker一直在用AuFS作为容器的文件系统。当一个进程需要修改一个文件时,AuFS创建该文件的一个副本。AuFS可以把多层合并成文件系统的单层表示。这个过程称为写入复制(copy on write)。

AuFS允许Docker把某些镜像作为容器的基础。例如,你可能有一个可以作为很多不同容器的基础的CentOS系统镜像。多亏AuFS,只要一个CentOS镜像的副本就够了,这样既节省了存储和内存,也保证更快速的容器部署。

使用AuFS的另一个好处是Docker的版本容器镜像能力。每个新版本都是一个与之前版本的简单差异改动,有效地保持镜像文件最小化。但,这也意味着你总是要有一个记录该容器从一个版本到另一个版本改动的审计跟踪。

Docker原理之App打包



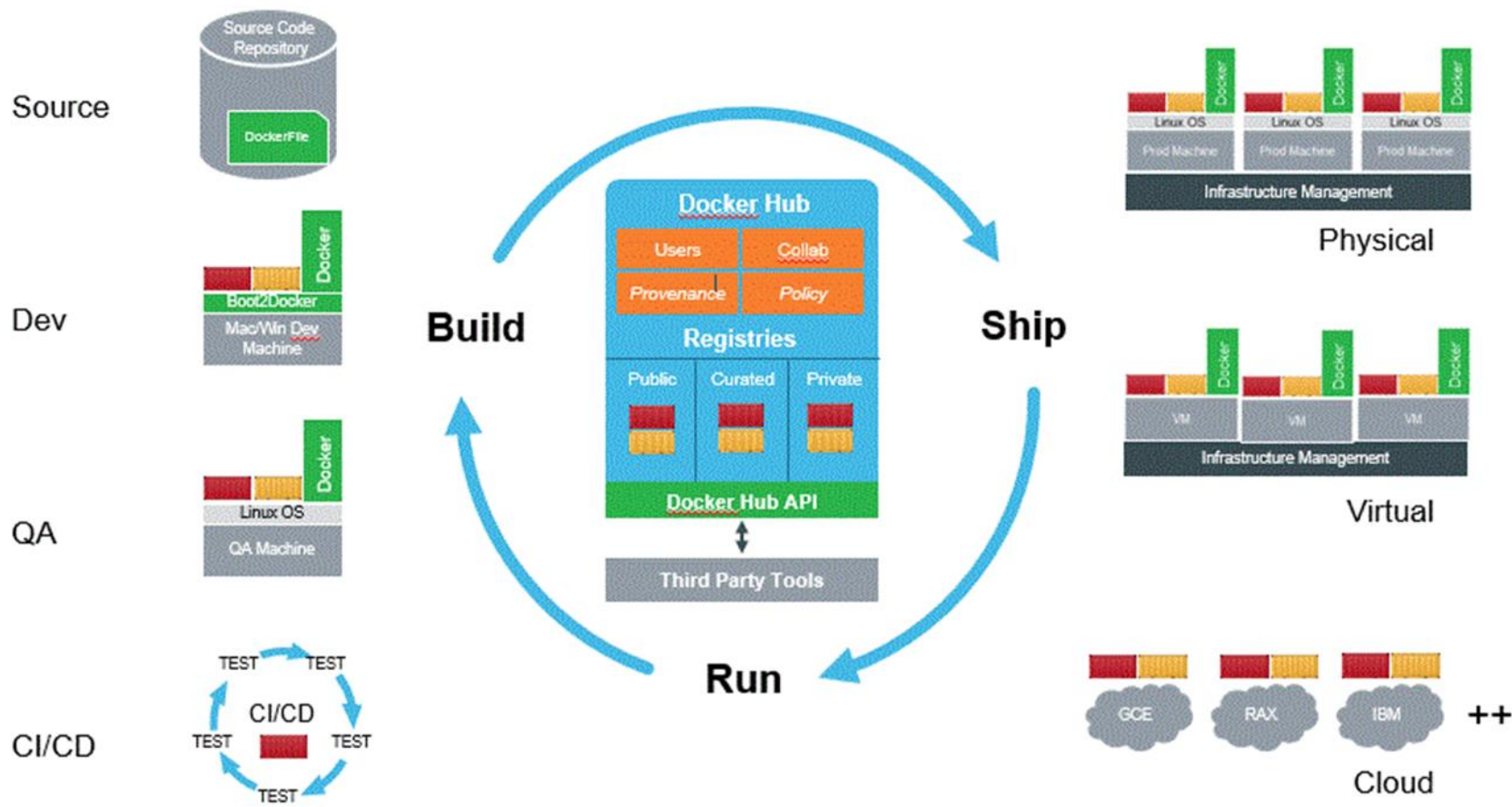
LXC的基础上, Docker额外提供的Feature包括: 标准统一的打包部署运行方案

为了最大化重用Image, 加快运行速度, 减少内存和磁盘 footprint, Docker container运行时所构造的运行环境, 实际上是由具有依赖关系的多个Layer组成的。例如一个apache的运行环境可能是在基础的rootfs image的基础上, 叠加了包含例如Emacs等各种工具的image, 再叠加包含apache及其相关依赖library的image, 这些image由AUFS文件系统加载合并到统一路径中, 以只读的方式存在, 最后再叠加加载一层可写的空白的Layer用作记录对当前运行环境所作的修改。

有了层级化的Image做基础, 理想中, 不同的APP就可以既可能的共用底层文件系统, 相关依赖工具等, 同一个APP的不同实例也可以实现共用绝大多数数据, 进而以copy on write的形式维护自己的那一份修改过的数据等

Docker的技术原理介绍

Docker全生命周期开发模式



Docker正在迅速改变云计算领域的运作规则，并彻底颠覆云技术的发展前景。从持续集成/持续交付到微服务、开源协作乃至DevOps，Docker一路走来已经给应用程序开发生命周期以及云工程技术实践带来了巨大变革。

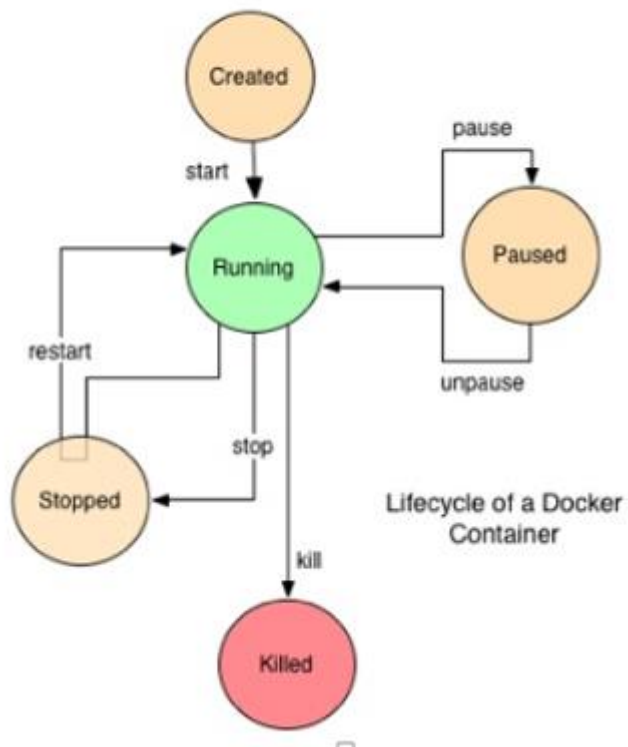
Docker Image

- Docker Image是一个极度精简版的Linux程序运行环境，比如vi这种基本的工具没有，官网的Java镜像包括的东西更少，除非是镜像叠加方式的，如Centos+Java7
- Docker Image是需要定制化Build的一个“安装包”，包括基础镜像+应用的二进制部署包
- Docker Image内不建议有运行期需要修改的配置文件
- Dockerfile用来创建一个自定义的image,包含了用户指定的软件依赖等。当前目录下包含Dockerfile,使用命令build来创建新的image
- Docker Image的最佳实践之一是尽量重用和使用网上公开的基础镜像

Docker Container

- Docker Container是Image的实例，共享内核
- Docker Container里可以运行不同Os的Image，比如Ubuntu的或者Centos
- Docker Container不建议内部开启一个SSHD服务，1.3版本后新增了docker exec命令进入容器排查问题。
- Docker Container没有IP地址，通常不会有服务端口暴露，是一个封闭的“盒子/沙箱”

Docker Container的生命周期



Docker Daemon

- Docker Daemon是创建和运行Container的Linux守护进程，也是Docker最主要的核心组件
- Docker Daemon 可以理解为Docker Container的Container
- Docker Daemon可以绑定本地端口并提供Rest API服务，用来远程访问和控制

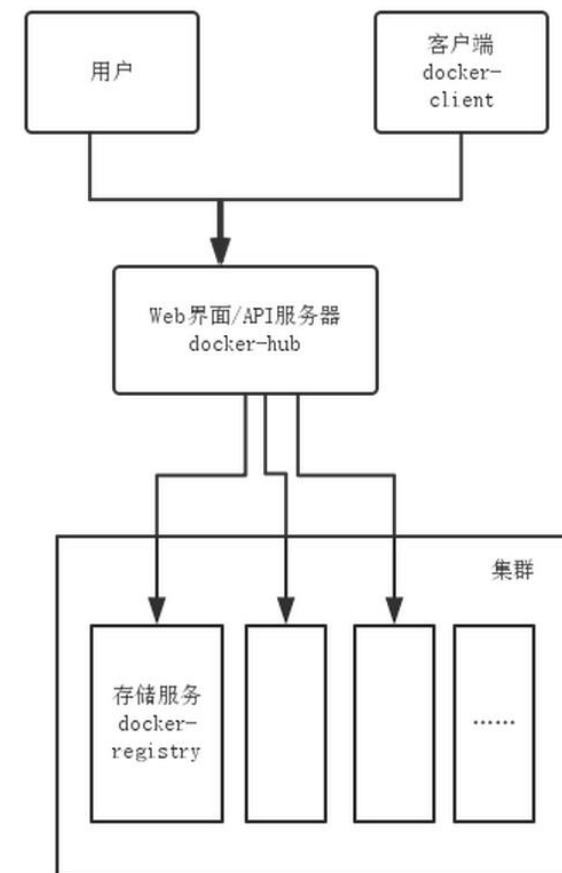
Docker Registry/Hub

Docker之所以这么吸引人，除了它的新颖的技术外，围绕官方Registry（ Docker Hub ）的生态圈也是相当吸引人眼球的地方。在Docker Hub上你可以很轻松下载到大量已经容器化好的应用镜像，即拉即用。这些镜像中，有些是Docker官方维护的，更多的是众多开发者自发上传分享的。而且你还可以在Docker Hub中绑定你的代码托管系统（目前支持Github和Bitbucket）配置自动生成镜像功能，这样Docker Hub会在你代码更新时自动生成对应的Docker镜像。

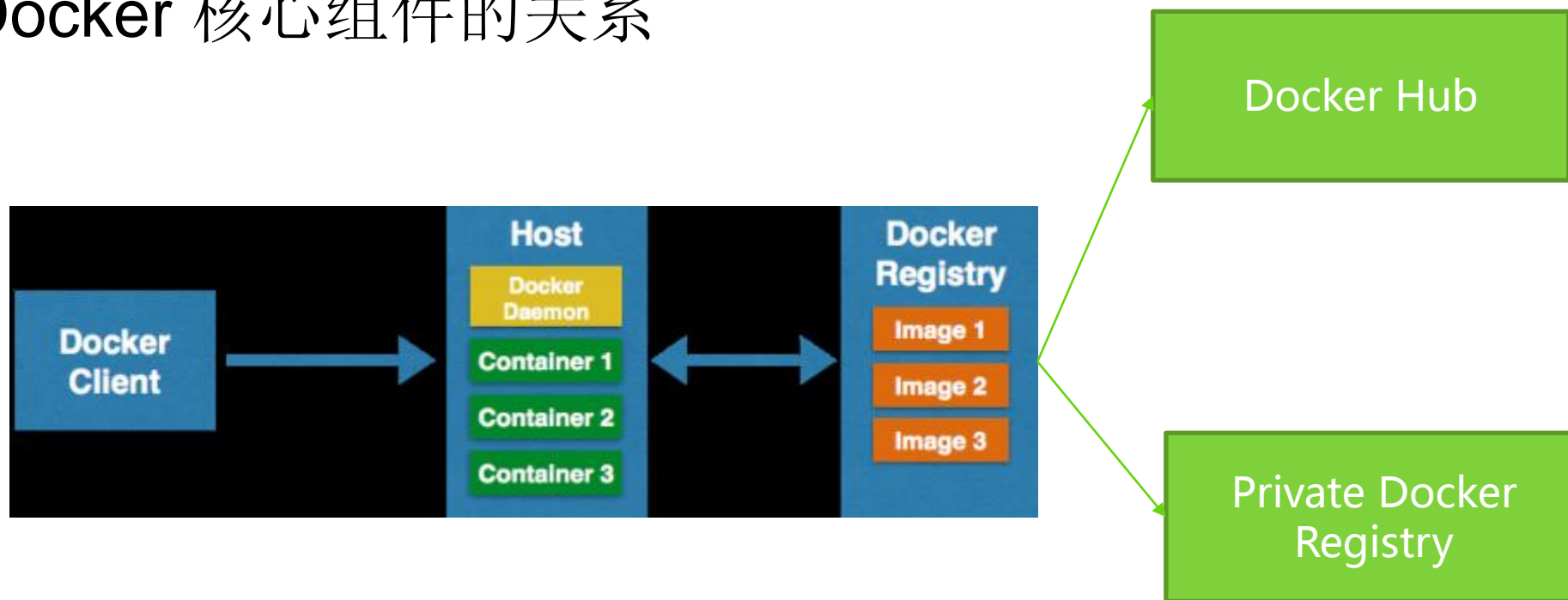
问题点：

Docker Hub是dotCloud公司私有的
国内曾有公司试图提供镜像服务，但被禁止
目前国内只有一个DaoCloud提供代理缓存服务

• [Docker Hub](#) - our SaaS service for sharing and managing your application stacks.



Docker 核心组件的关系



- Linux基本操作和基本知识：磁盘、文件、日志、用户、权限、安全、网络，建议centos/redhat
- 虚拟机相关技能：vmware workstation/virtbox熟练使用，虚拟机clone，组网，host-only网络，nat网络等熟练操作

Thanks

FAQ时间