
Docker容器技术介绍



Docker特性

一个轻量级容器



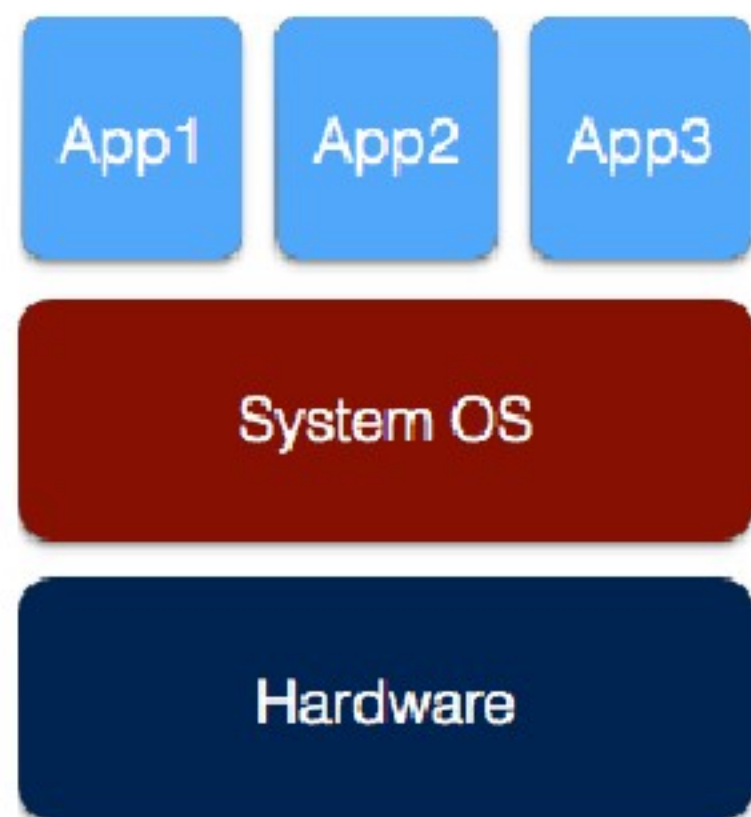
特性	容器	虚拟机
启动速度	秒级	分钟级
硬盘使用	一般为MB	一般为GB
性能	接近原生	弱于
系统支持量	单机支持上千个容器	一般几十个
隔离性	安全隔离	完全隔离

概述

Linux容器虚拟技术（LXC, Linux Container）是一种轻量级的虚拟化手段，它利用内核虚拟化技术提供轻量级的虚拟化，来隔离进程和资源。

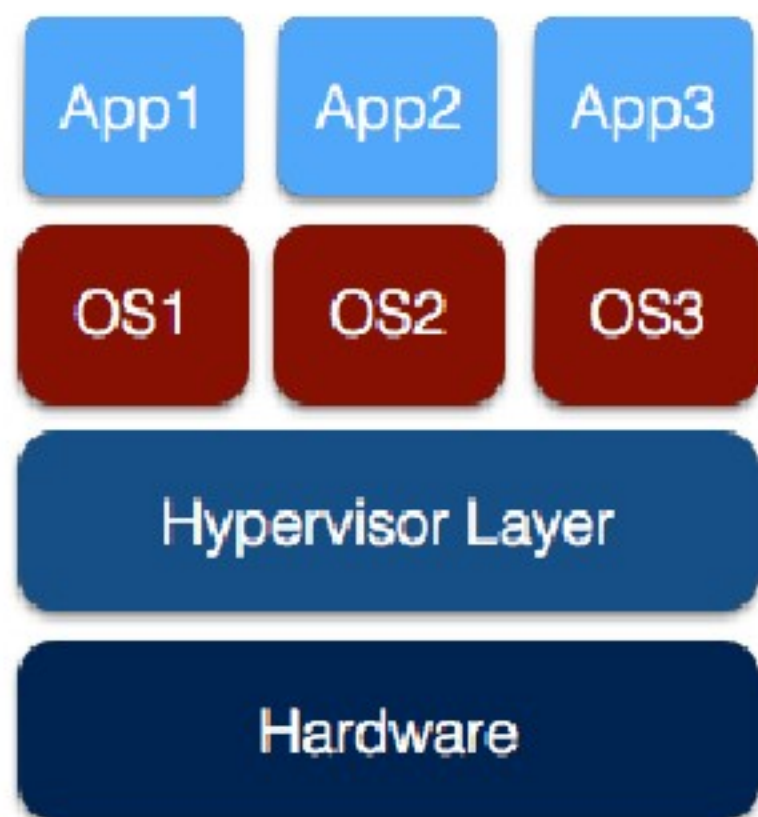
LXC有效地将由单个操作系统管理的资源划分到孤立的组中，以更好地在孤立的组之间平衡有冲突的资源使用需求。

各方案的实现区别



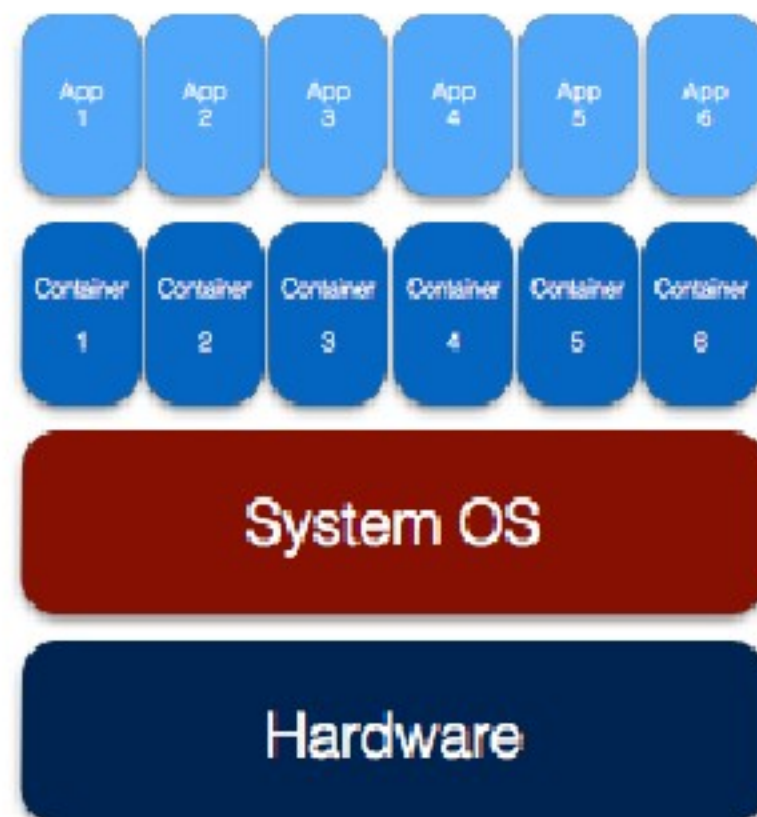
传统方案

- **优势**：成熟稳定，容易上手，技术牢靠，资料丰富
- **劣势**：方案陈旧，资源浪费，死板不灵活



虚拟化方案

- **优势**：风头正劲，相对成熟，厂商支持，体系完善
- **劣势**：复杂度高，成本高昂



容器方案

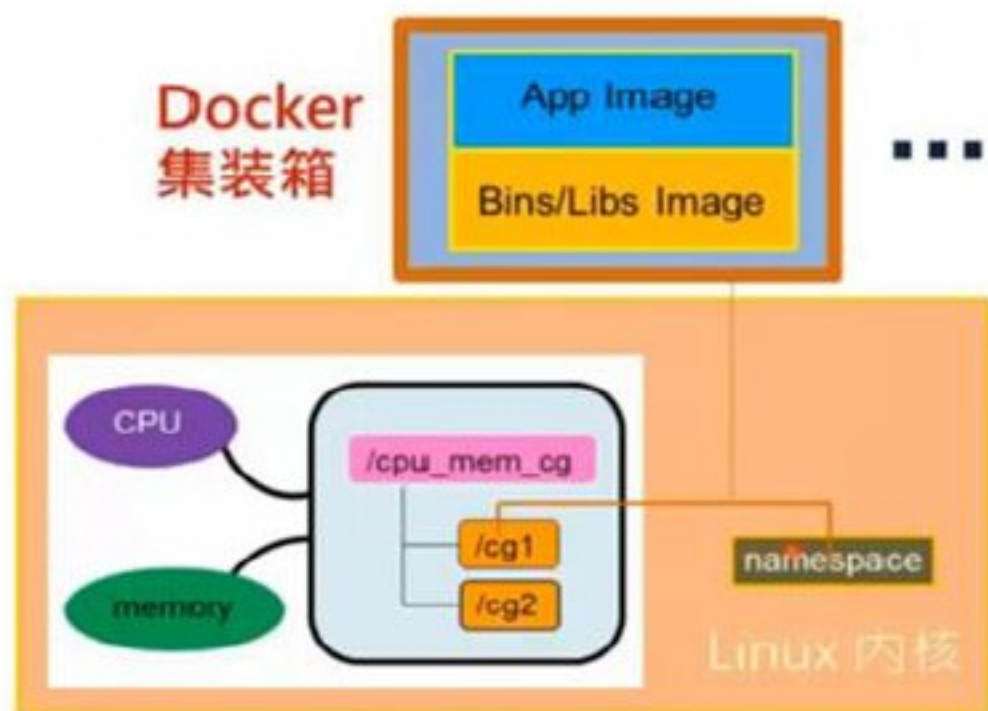
- **优势**：灵活度高，性能强悍，成长强势，免费
- **劣势**：方案新颖，案例太少，资料欠缺，很不成熟

Docker的原理与组成



每个container依赖namespace来实现隔离性，每个容器都有自己的命名空间，确保容器之间不会相互影响，每个容器拥有自己的进程和网络空间的虚拟环境，成为一个独立运行的单位。

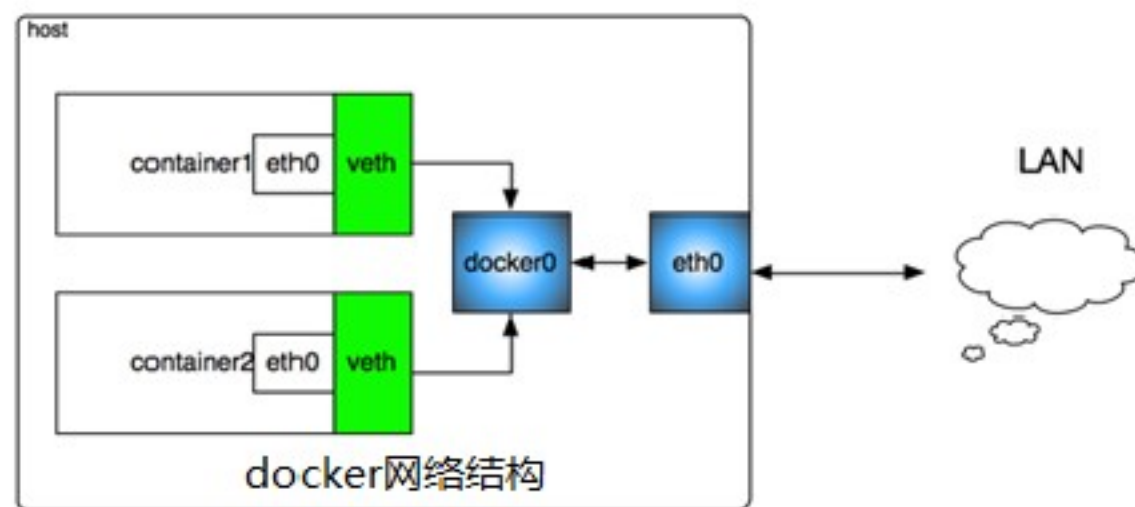
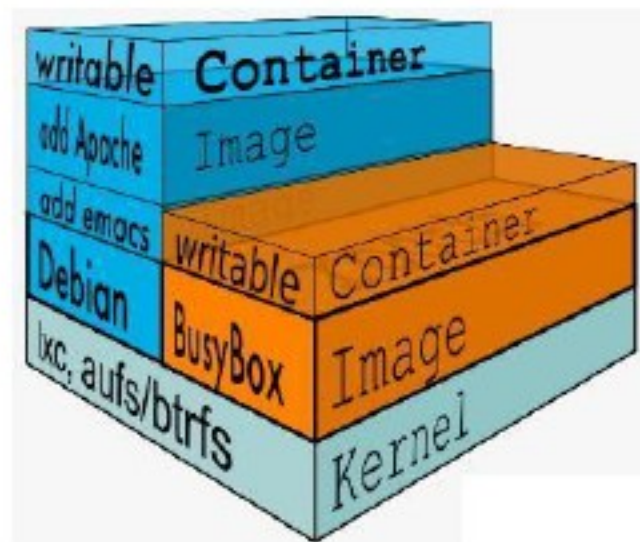
Container由内核的cgroup来对各个容器（进程）使用的系统资源做严格限制。



Docker一直在用AuFS（高级多层次统一文件系统）

作为容器的文件系统。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。

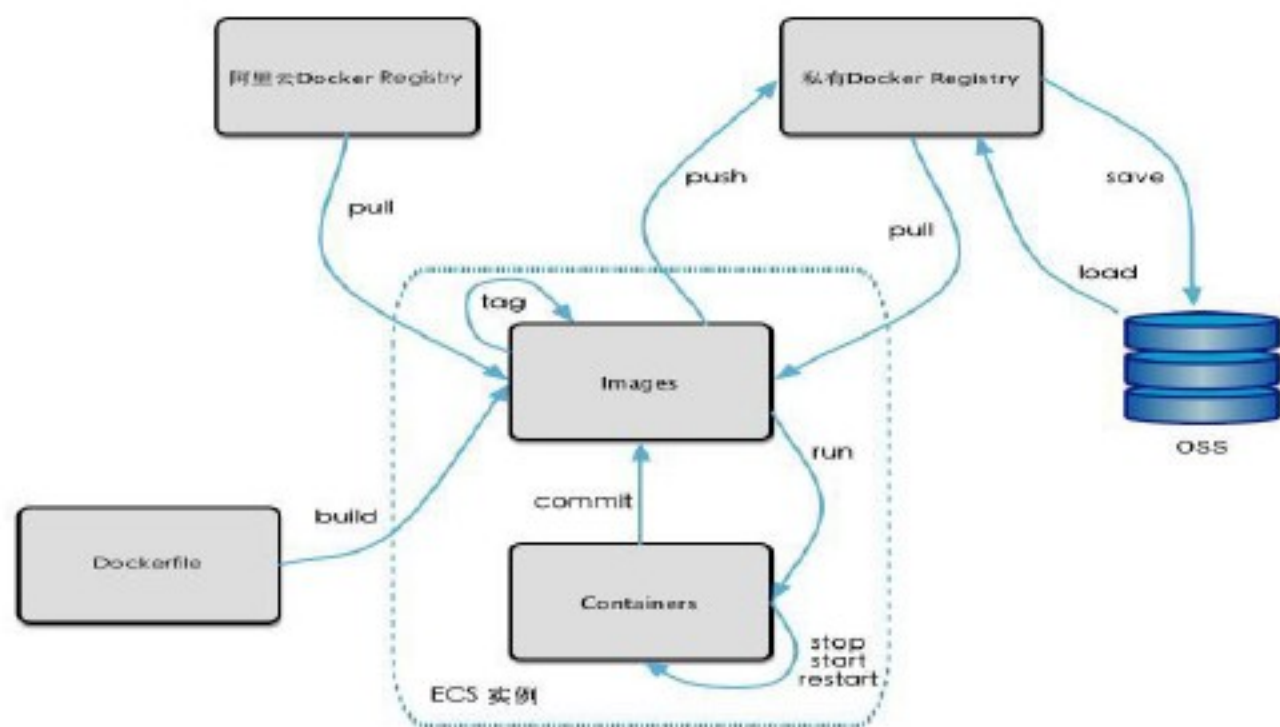
当一个进程需要修改一个文件时，AuFS创建该文件的一个副本



应用案例



- 百度在2013年10月份就已经成功使用Docker支持其BAE平台的Paas服务



阿里云ECS Docker生态图

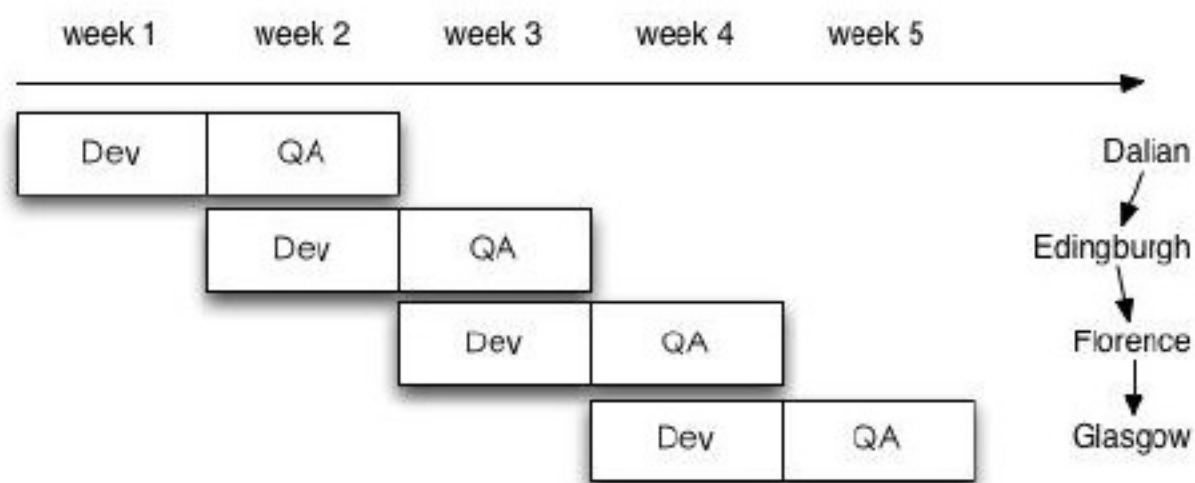
- 安全: ECS对于容器来说是一个sandbox, 把威胁都限制在实例级别。
- 简单: 运维只负责创建需要的实例交付给开发。
- 容错: 一旦实例环境出问题, 只要回滚快照就行。
- 弹性伸缩: 不论是垂直还是水平伸缩, 对于实例来说都轻而易举。
- 故障迁移: 一旦实例发生故障, 其上的容器会在线迁移到其他实例, 更能保证应用无中断。

Docker的企业级应用场景

•快速迭代开发与持续集成

Docker对开发环境极度适用，相对于虚拟机级别的环境，VM的创建动作很大，不便于频繁更新，启动销毁速度还很慢，对于并行开发的场景，尤其是快速迭代的开发周期，支持起来还是不够敏捷。

使用Linux container，使得“VM”的创建和销毁在秒级就能完成，由于只是做了网络和进程的隔离，“VM”的运行几乎没有开销，打个包就完成了更新，和git一样方便。



•精细化颗粒级服务

由于Docker的构建可以达到秒级，而且十分轻量，销毁方便，可以高度精细化服务颗粒度，蚂蚁群兵式的实现海量级别按需云服务 and 弹性扩展。

•资源优化绿色节能

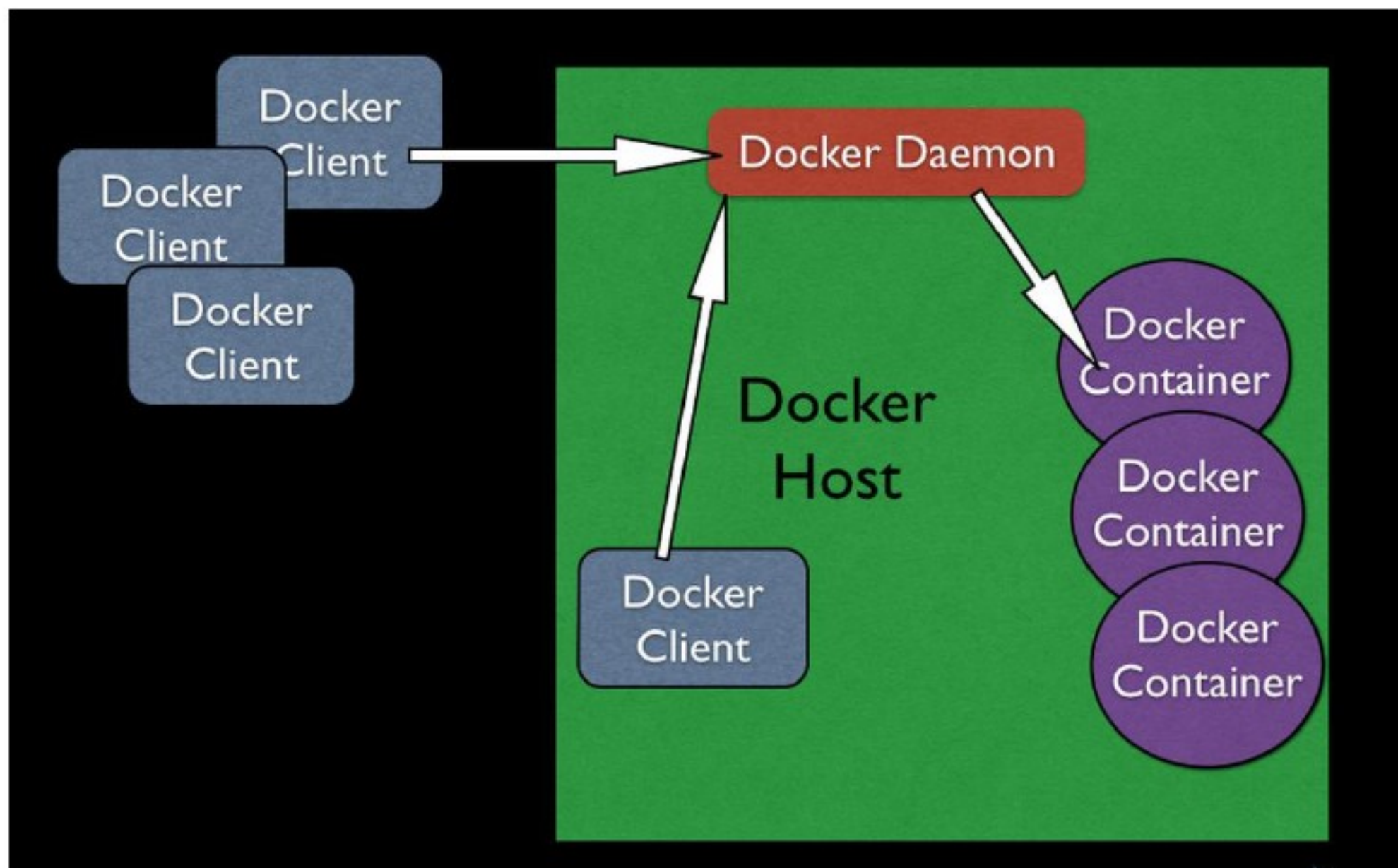
Container相对VM而言，直接调用本地操作系统的资源，也无需维护guestos，基本没有性能损耗，而且极其轻量化，进程级别的运行环境隔离，可以轻易承载海量的服务，对实现绿色节能数据中心有着极强的价值。



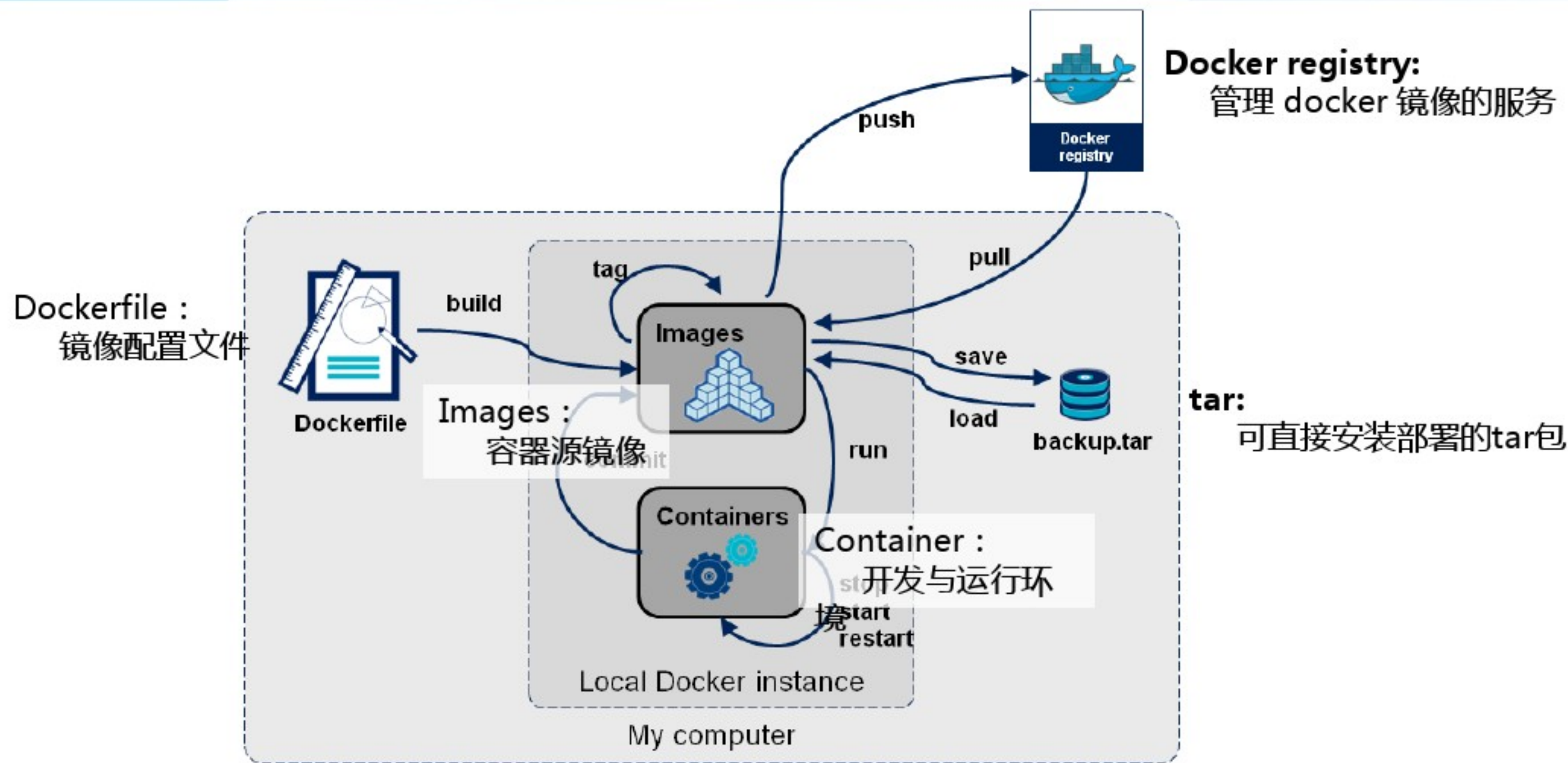
Docker的部署架构

Docker的部署架构十分简单，内网部署只需要一个私有镜像仓库，和可以跑linux的服务器即可实现，开发环境与部署环境几乎一致。

Docker通常的实现是CS结构，Docker守护进程运行在主机上，通过socket连接从客户端访问，客户端也可以运行同一主机上，但不是必须的。Docker命令行客户端也是类似的工作方式，但它通常通过Unix域套接字而不是TCP套接字连接。

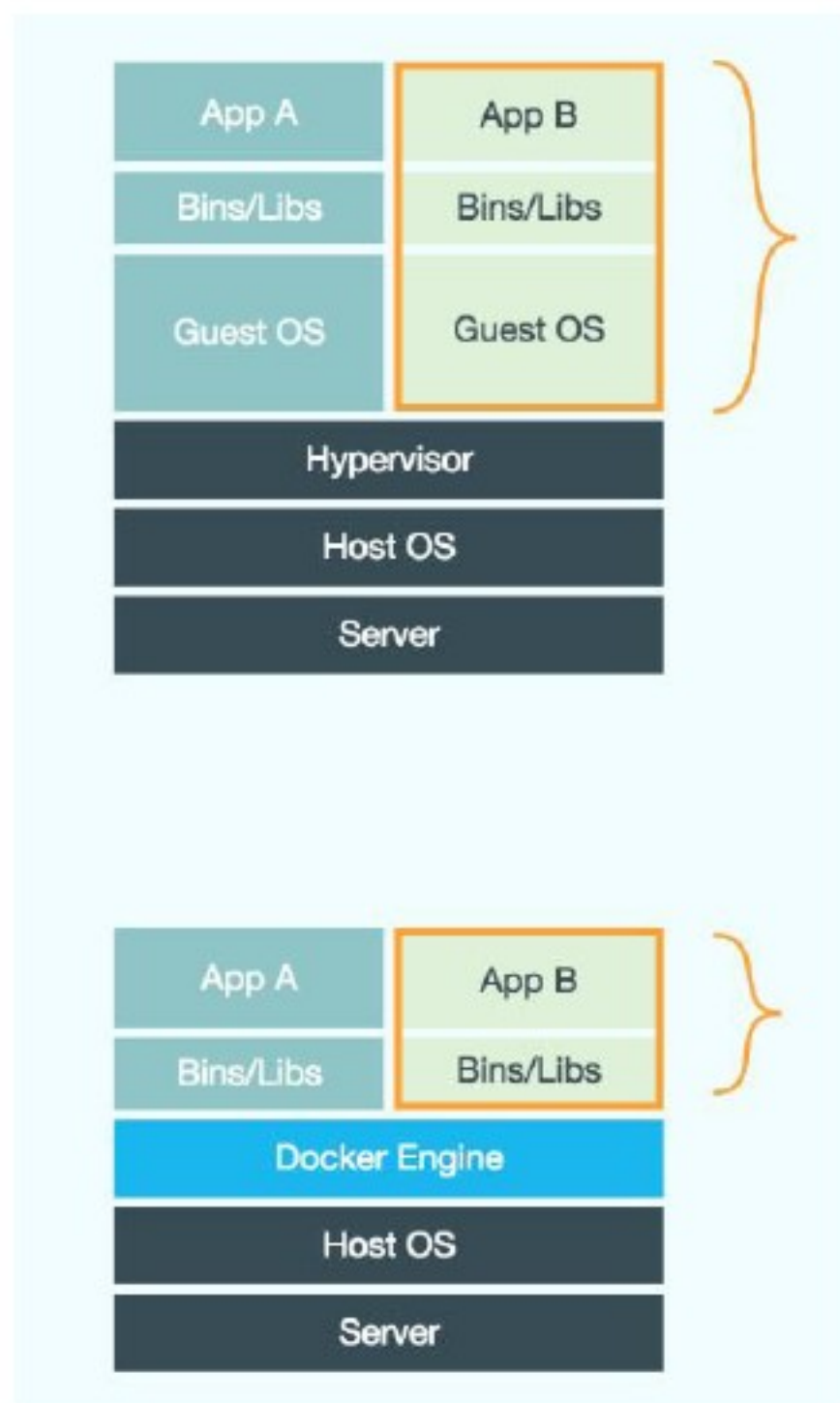


Docker中组件的概念和使用



Docker的部署架构十分简单，内网部署只需要一个私有镜像仓库，和可以跑linux的服务器即可实现，开发环境与部署环境几乎一致。

虚拟化和docker的对比



虚拟机

每个需要被虚拟化的应用不仅包括应用程序，哪怕只有几十MB和必要的二进制文件和库，还要包含一个可能至少几十GB的客户操作系统。

Docker

Docker容器只包含应用程序及其依赖关系。它作为一个独立进程在主机的操作系统的用户空间运行，与其它容器共享内核。因此，它不但具有虚拟机的资源隔离和分配利益，而且更轻便高效。

虚拟化和docker的选择



VS

Docker



Windows Based

生产环境，核心业务

成熟产品

要求较强的厂商服务

与现有虚拟化环境集成

对IT成本不特别敏感

应用服务无需经常变动

Linux Based

开发测试环境，非核心

试水体验

自有一定开发能力

无需和现有虚拟化集成

对IT成本敏感

DevOps持续集成

Note: Docker不是一个全功能的VM, 它有很多严重的缺陷, 比如安全、Windows支持, 因此不能完全替代虚拟机。现在Docker社区一直在弥补这些缺陷, 当然这会带来一定的性能损耗, 但随着Docker的发展, 未来容器虚拟化有可能会逐渐替代目前的服务器虚拟化方案。