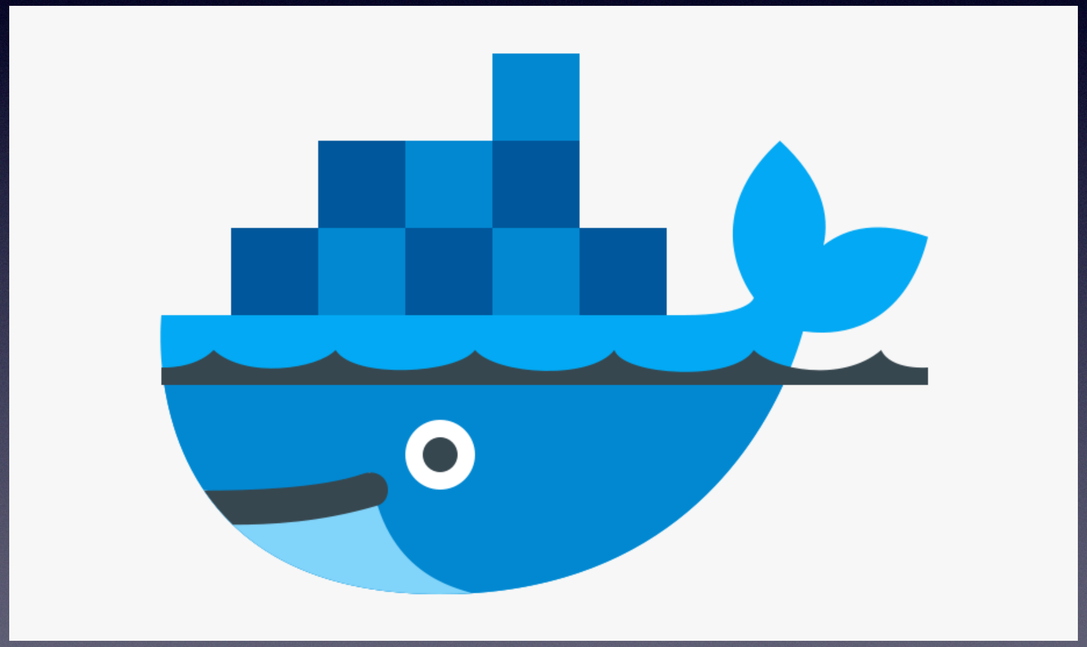
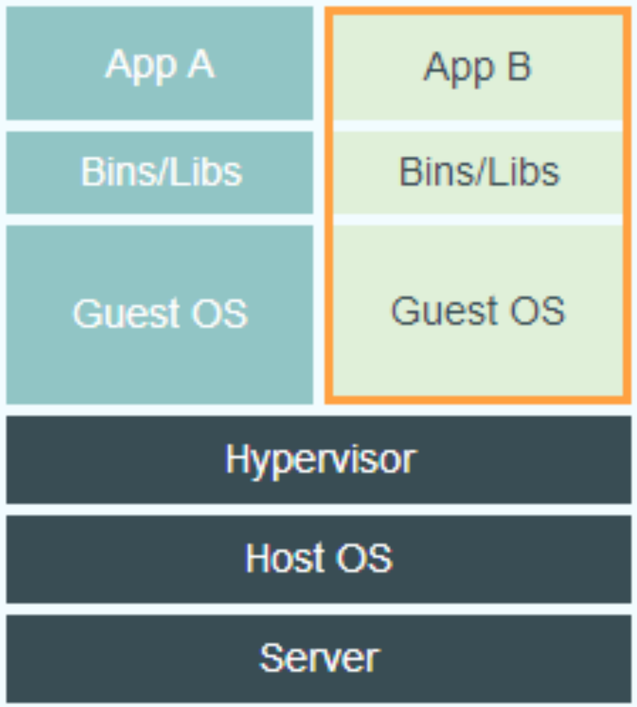
Docker 简介

- 2013 年 3 月，dotCloud 公司开源了自己的内部项目 —— Docker
- 2013 年底，dotCloud 公司改名为 Docker

什么是 Docker

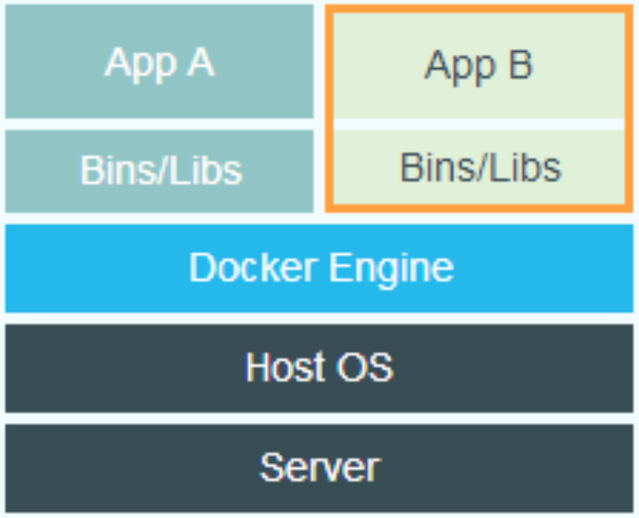
Docker 使用 Google 公司推出的 Go 语言 进行开发实现，基于 Linux 内核的 cgroup，namespace，以及 AUFS 类的 Union FS 等技术，对进程进行封装隔离，属于操作系统层面的虚拟化技术。由于隔离的进程独立于宿主和其它的隔离的进程，因此也称其为**容器**。





虚拟机

每个虚拟机不仅包含应用本身，还包含了必需的二进制文件、类库，甚至整个操作系统。



Docker

Docker 容器之包含了应用及其依赖。它在用户空间作为一个隔离的进程运行在操作系统上。因此，它有着虚拟机资源隔离和可分配的优点的同时，更加便捷有效。

为什么要使用 Docker?

- 更高效的利用系统资源
  无论是应用执行速度、内存损耗或者文件存储速度
- 更快速的启动时间
  直接运行于宿主内核，无需启动完整的操作系统，因此可以做到秒级、甚至毫秒级的启动时间
- 一致的运行环境
  通过镜像提供除内核外的完整的运行时环境，确保应用运行环境一致性
- 持续交付和部署
  通过 Dockerfile 来进行镜像构建，一次创建或配置，最大化再任意地方正常运行的可能性
- 更轻松的迁移
  确保了执行环境的一致性，使得应用的迁移更加容易
- 更轻松的维护和扩展
  使用的分层存储以及镜像的技术，使得应用重复部分的复用更为容易，也使得应用的维护更新更加简单

对比传统虚拟机总结

特性	容器	虚拟机
启动	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

基本概念

- 镜像（Image）
- 容器（Container）
- 仓库（Repository）

理解了这三个概念，就理解了 *Docker* 的整个生命周期。

基本概念——Docker 镜像

- Docker 镜像是一个特殊的文件系统，除了提供容器运行时所需的程序、库、资源、配置等文件外，还包含了一些为运行时准备的一些配置参数（如匿名卷、环境变量、用户等）。镜像不包含任何动态数据，其内容在构建之后也不会被改变。

基本概念——Docker 容器

- 镜像（Image）和容器（Container）的关系，就像是面向对象程序设计中的 类 和 实例 一样，镜像是静态的定义，容器是镜像运行时的实体。容器可以被创建、启动、停止、删除、暂停等。
- 容器的实质是进程，但与直接在宿主执行的进程不同，容器进程运行于属于自己的独立的 命名空间。

基本概念——Docker Registry

- 一个 Docker Registry 中可以包含多个 仓库（Repository）；每个仓库可以包含多个 标签（Tag）；每个标签对应一个镜像。
- 一个仓库会包含同一个软件不同版本的镜像，而标签就常用于对应该软件的各个版本。我们可以通过 <仓库名>:<标签> 的格式来指定具体是这个软件哪个版本的镜像。

demo

下期预告：容器编排