讨论课 03 – 复用云技术

范亮 1352899

侧重于容器技术，讨论各种方案以及挑战

1、容器技术的理解

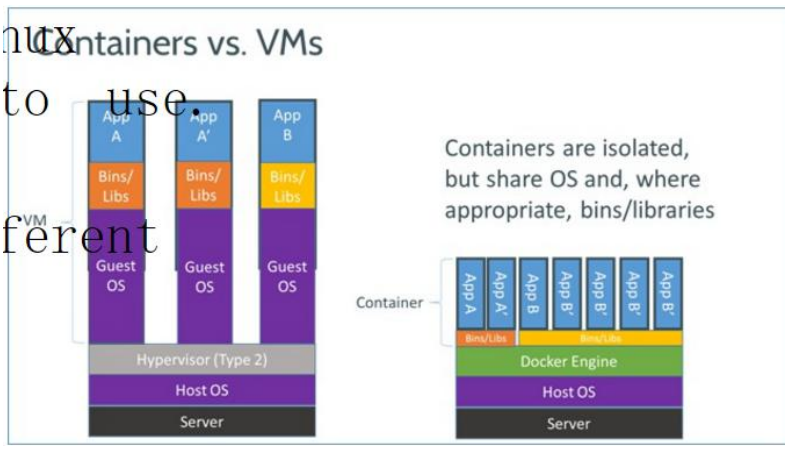
1.1 关键技术

> 隔离

容器为应用程序提供了隔离的运行空间：每个容器内都包含一个独享的完整用户环境空间，并且一个容器内的变动不会影响其他容器的运行环境。为了能达 到这种效果，容器技术使用了一系列的系统级别的机制诸如利用 Linux namespaces 来进行空间隔离，通过文件系统的挂载点来决定容器可以访问哪些文件，通过 cgroups 来确定每个容器可以利用多少资源。此外容器之 间共享同一个系统内核，这样当同一个库被多个容器使用时，内存的使用效率会得到提升。

系统虚拟化技术，虚拟层为用户提供了一个完整的虚拟机：包括内核在内的一个完整的系统镜像。CPU 虚拟化技术可以为每个用户提供一个独享且和其他用户隔离的系统环境，虚拟层可以为每个用户分配虚拟化后的 CPU、内存和 IO 设备资源。

VM 与 Dockers 对比



> 安全

对于容器卷和数据，除了标准文件权限和配置"只读"或者"读写"访问，没有其他别的安全性问题。这意味着用户在容器上的文件访问权限需要匹配主机设置。

> 数据完整性

使用卷和数据容器共享数据，能够保护数据的完整性。如文件锁定需要容器本身的管理功能。这是一个额外的开销，必须添加到应用程序中。

容器没有提供数据保护设施，如快照或复制，因此数据管理必须由主机或容器来处理。

外部存储也缺乏支持。除了系统操作系统中提供的功能外，Docker 并没有在外部存储中提供特定的支持。

容器卷默认存储在/var/lib/dockerdirectory 目录中，这可能会成为一个性能瓶颈。然而，在 Docker 启动进程中转换容器卷默认存储位置是可行的。

1.2 挑战

集群管理、资源管理、存储

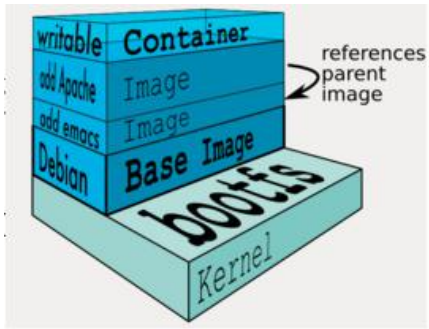
Docker Engine 目前还有单点失效问题，Engine 失效或更新会导致一个节点上所有容器终止；另外 Docker 容器目前还无法再节点之间方便的迁移，这阻碍了对应用进行动态调度和可用性的需求。

2、复用方面

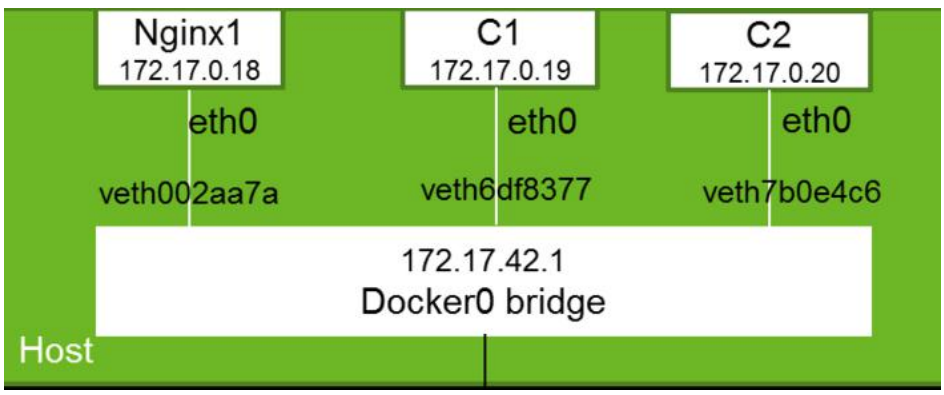
docker 为例：

为很多应用提供了很轻的服务，可以直接通过 docker engine 轻量化使用很多应用。这些应用以云存储，且是复用形式的。

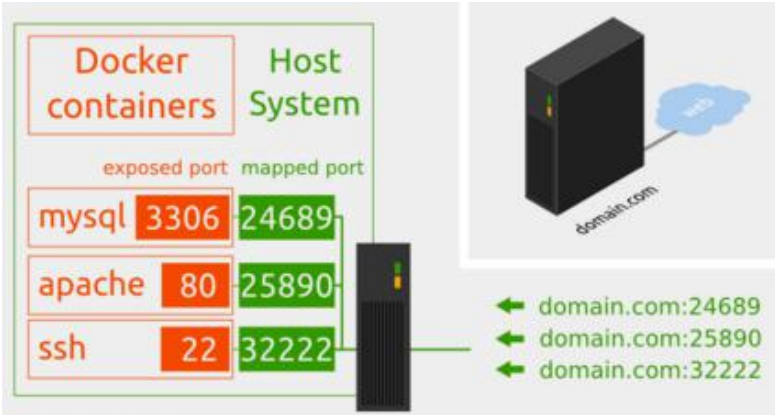
> image & layer



>Network:



> Docker service:



Build image -DockerFile

引用：

1. http://www.csdn.net/article/2015-07-17/2825242 容器创业公司盘点

2. https://yq.aliyun.com/articles/4124

阿里 Docker 服务开发中的 5 大挑战与经验沉淀

3. https://dockerbook.com/