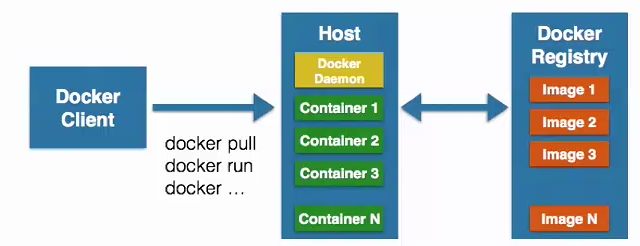
Docker学习笔记

# 1 docker的基本概念和框架

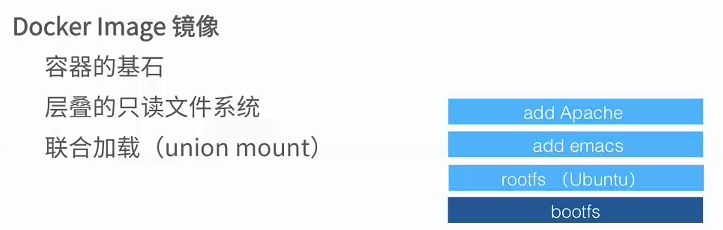
## 1.1 Docker组成



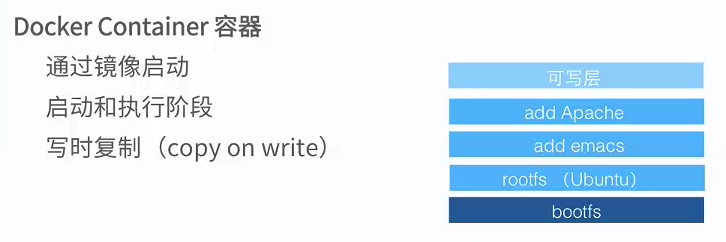
Docker Client客户端(C)

Docker Daemon守护进程(S)

Docker Image镜像



Docker Container容器



Docker Registry仓库

公有仓库(Docker Hub)

私有仓库

## 1.2 Docker容器相关技术

(1) Namespaces命名空间

对于编程语言来说：封装 ---> 代码隔离

对于操作系统来说：系统资源的隔离，系统资源包括进程、网络、文件系统......

|  |  |
| --- | --- |
| **Docker的五种命名空间** | |
| PID (Process ID) | 进程隔离 |
| NET (Network) | 管理网络接口 |
| IPC (InterProcess Communication) | 管理跨进进程通信的访问 |
| MNT (Mount) | 管理挂载点 |
| UTS (Unix Timesharing System) | 隔离内核和版本标识 |

这五种资源是怎么管理起来？答：Control groups 控制组，

(2) Control groups 控制组

控制组是可以限制、记录、隔离进程使用物理资源的机制，用来分配资源，就是为了Docker而生的。

|  |  |
| --- | --- |
| **控制组功能** | |
| 资源限制 | 例如为进程设置内存限制，当内存达到限制值时，禁止申请新进程 |
| 优先级设定 | 可以设定哪些进程组使用更多CPU、磁盘IO资源 |
| 资源计量 | 可以计算进程组使用了多少资源 |
| 资源控制 | 可以将进程组挂起和恢复 |

从Control groups 功能中可以知道有哪些Docker容器的能力

|  |  |
| --- | --- |
| **Docker容器的能力** | |
| 文件系统隔离 | 每个容器都有自己的root文件系统 |
| 进程隔离 | 每个容器都运行在自己的进程环境中 |
| 网络隔离 | 容器间的虚拟网络接口和IP地址都是分开的 |
| 资源隔离和分组 | 使用Control groups将CPU和内存之类的资源独立分配给每个Docker容器 |