# 程序

docker程序分成docker daemon（dockerd）和docker client（docker）。

守护进程dockerd始终在后台运行，用于管理docker容器，处理docker客户端请求。docker是客户端程序，给用户提供了命令行操作接口。

# 存储

## AUFS

分层文件系统，实现了Union mount（多个目录mount到同一个挂载点），是docker存储的底层实现。

## Layer

AUFS是由若干层组成，叠加成一个完整的文件系统，每层包含这个文件系统中的部分文件。默认情况下，只有最上一层是可写的，其他层都是只读的。层组成了image和container的静态部分。

## 存储位置

**/var/lib/docker/aufs/diff**：存储image各层的内容（目录树结构），及2个container层的内容

**/var/lib/docker/aufs/layers**：存储各layer所依赖的其他layer。其中的xxx-init是某container的最上层的read-only层，包含本container的原始内容：而xxx则在xxx-init之上，为read-write层，初始时无内容，在container活动过程中生成或者修改的文件存放于此

# ID

Image、Container以及以及组成它们的各Layer的ID，都是通过65个字节的SHA256哈希值表示的，并且某个Image（或Container）的ID和组成它们的Layer的ID是没有关系的（之前是有的）。

# Image

镜像可以视为某一个标准的运行环境（文件系统），比如Nginx镜像，包含了运行Nginx的完整文件系统。

## 存储位置

**/var/lib/docker/aufs/diff**：组成Image的read-only layer存放于此（同container）

## 命令

**docker image ls**：显示所有镜像

**docker image rm IMAGE**：删除镜像

**docker image pull NAME[:**[**TAG|@DIGEST**](mailto:TAG|@DIGEST)**]**：从registry抓取镜像到本地

**docker image push NAME[:TAG]**：将本地镜像推到registry

**docker image build PATH**：通过dockerfile制作出一个镜像

**docker image history IMAGE**：显示镜像的历史

# Container

容器，其静态部分基于某个镜像，包括image各layer ，以及2个container layer，一个read-only和一个read-write，其中read-only层的名字是read-write层名字（哈希值）后加上“-init”。

read-only层中存放的是和容器运行环境相关的一些内容，又不适合作为image的部分存储，因此存储在这里。read-write层则存放的是容器运行之后对文件系统做的所有改动。

container停止之后，这两个layer也不会删除，只有当container被删除之后才会被删除。

read-only层的具体内容如下：

xxx-init/

├── dev

│   ├── console

│   ├── pts

│   └── shm

└── etc

├── hostname

├── hosts

├── mtab -> /proc/mounts

└── resolv.conf

## 存储位置

**/var/lib/docker/containers**：存放了container的metadata和配置

**/var/lib/docker/aufs/diff**：存储了container的各层（同image）

## 命令

**docker container ls**：显示当前所有运行的container，加上“-a”参数可显示已经停止的。

**docker container run IMAGE [COMMAND]**：启动一个image成一个container，加上“-i -t”参数表示需要控制台交互，等同于docker run IMAGE。

**docker container start CONTAINER**： 启动一个停止的container

**docker container restart CONTAINER**：重启一个container（无论其是否在运行）

**docker container attach CONTAINER**：连接到一个运行的container，连接其控制台，所有控制台输出一致

**docker container exec CONTAINER COMMAND**：在正在运行的container中执行命令，“-it”表示需要控制台交互，可以用此命令在已运行container上打开一个终端

**docker container commit CONTAINER [REPOSITORY[:TAG]]**：将一个容器创建为一个镜像

# dockerfile

类似于生成镜像的Makefile，描述了镜像的生成步骤。

# Registry

Registry是一个docker镜像的仓库，docker客户端通过pull和push来下载上载镜像到registry。registry可以是公共的或者私有的，两个主要的公共registry是Docker Hub和Docker Cloud，其中Docker Hub是docker缺省的registry。

# Service

# Swarm