

# docker 学习-day1(docker安装)

## 1.简介

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。

与传统虚拟机技术的区别：

传统虚拟机技术是虚拟出一套 硬件 后，在其上运行一个完整操作系统，在该系统上再运行所需应用进程；而容器内的应用进程直接运行于 宿主的内核，容器内没有自己的内核，而且也没有进行硬件虚拟。因此容器要比传统虚拟机更为轻便。

优势：

1. 更高效的利用系统资源。

2. 更快速的启动时间。

3. 一致的运行环境。

4. 持续交付和部署。

5. 更轻松的迁移。

6. 更轻松的维护和扩展。

## 2.基本概念

### 镜像（Image）

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。

分层存储: 镜像构建时，会一层层构建，前一层是后一层的基础. 每一层构建完就不会再发生改变，后一层上的任何改变只发生在自己这一层.

### 容器（Container）

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.容器可以拥有自己的 root 文件系统、自己的网络配置、自己的进程空间，甚至自己的用户 ID 空间;容器不应该向其存储层内写入任何数据，容器存储 层要保持无状态化。所有的文件写入操作，都应该使用 数据卷（Volume）、或者绑定宿主目录，在这些位置的读写会跳过容器存储层，直接对宿主(或网络存储)发 生读写，其性能和稳定性更高。

### 仓库(Repository)

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.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.如 `ubuntu:14.04`

## 3.安装docker

```
# install the latest docker
sudo yum -y install docker

# start docker service
sudo service docker start

# add ec2-user to the docker group
sudo usermod -a -G docker ec2-user
```

## 4.安装docker-compose与docker-machine

```
# docker-compose
curl -L https://github.com/docker/compose/releases/download/1.5.2/docker-compose-`uname -s`-`uname -m` > /usr/local/bin/docker-compose
# 授权
chmod +x /usr/local/bin/docker-compose

# docker-machine
curl -L https://github.com/docker/machine/releases/download/v0.6.0/docker-machine-`uname -s`-`uname -m` > /usr/local/bin/docker-machine
# 授权
chmod +x /usr/local/bin/docker-machine
```