初步了解Docker

Docker的主要目标是：build,ship and run any App,Anywhere。

通过封装，分发，部署，运行等声明周期的管理，达到应用级别的一次封装，到处运行。运用组件可以是web应用，数据库服务，甚至是一个操作系统或者编译器。

可以简单的将docker容器理解为一种沙盒，每个容器内运行一个应用，不同的容器相互隔离，容器之间也可以建立通信机制。很多时候，直接把容器当做应用本身也没有问题。

Docker的优势：

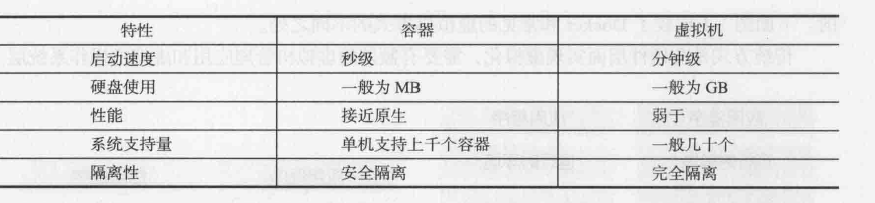
更快速的交付和部署。

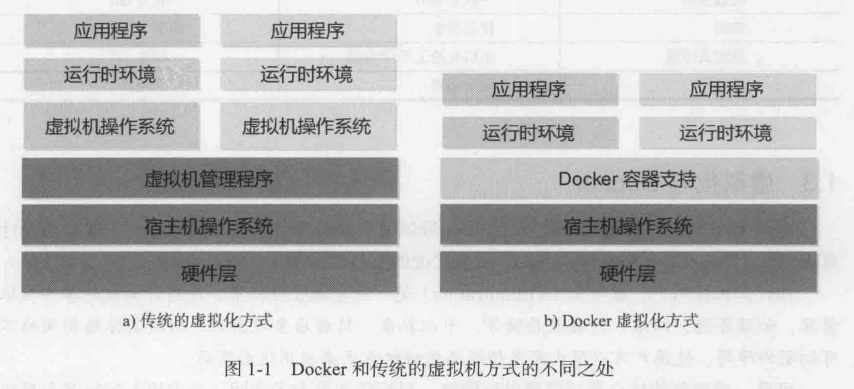
更高效的资源利用

更轻松的迁移和扩展

更简单的更新管理

Docker和虚拟机的比较：





Docker的核心概念和安装

**镜像**：面向docker引擎的只读模板，包含了文件系统。镜像是创建Docker容器的基础。

**Docker容器**：类似一个轻量级的沙箱，docker利用容器来运行和隔离应用。容器是从镜像创建的应用运行实例，可以将其启动，开始，停止，删除。而这些容器都是相互隔离，互不可见的。

可以将容器看作是简易版的linux系统环境。包括root用户权限，进程空间，用户空间和网络空间等，以及运行在其中的应用程序打包而成的应用盒子。

Docker仓库：类似于代码仓库，是docker几种存放镜像文件的场所。注册服务器是存放仓库的地方，往往存放多个仓库。每个仓库几种放一类镜像，往往包含多个镜像文件，通过不同的标签（tag）来区分。

根据所存储的镜像公开与否，docker仓库可以分为公开仓库和私有仓库两种形式。Docker利用仓库管理惊醒的设计理念和git非常相似。

镜像

Docker运行容器前需要本地存在对应的镜像，如果镜像不存在本地，docker会尝试先从默认镜像仓库下载(默认使用docker hub公共注册服务器中的仓库)，用户也可以通过配置，使用自定义的镜像仓库。

镜像是docker运行容器的前提。

3.1 获取镜像

镜像是Docker运行的前提，

使用docker pull命令从网络上下载镜像。

docker pul NAME:[tag] : 默认是最新版本的系统。

如：docker pull Ubuntu. 就是docker pull Ubuntu:latest

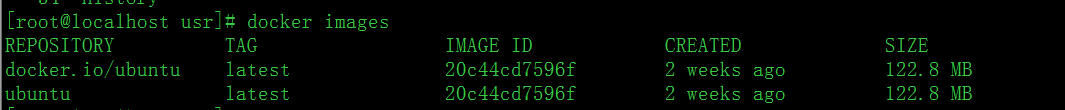
docker pull Ubuntu:14.4

使用下载的镜像 就可以创建一个容器：container。

docker run –t –i ubuntu /bin/bash

3.2 查看镜像信息

使用docker images 命令可以列出本地主机上已有的镜像



REPOSITORY: 来此哪个仓库，

TAG：镜像的标签信息。

IMAGE ID：镜像的ID

CREATE：创建时间

SIZE：镜像大小

使用docker tag NAME1 NAME2可以为镜像提一个别名。

标签在起到了引用或者快捷方式的作用。

使用docker inspect命令可以获取该镜像的详细作用。

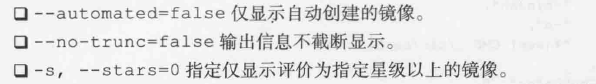
Docker inspect 20c44cd7596f // 返回一个json字符串的数据。

使用 docker inspect –f 来指定镜像的某个参数信息。

Docker inspect –f Container

3.3 搜寻镜像

使用docker search命令可以搜寻远程仓库中共享的镜像。默认搜索docker Hub官方仓库中的镜像。用法为docker search TERM ，支持的参数有：



3.4 删除镜像：

Docker rmi image[……],其中Iamge 可以为标签或ID

3.5 创建镜像：

1.基于已有镜像的容器创建

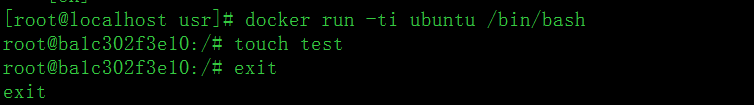
Docker commit [options] CONTRAINER [REPOSITORY[:tag]]

-a ,--author=”’ : 作者信息

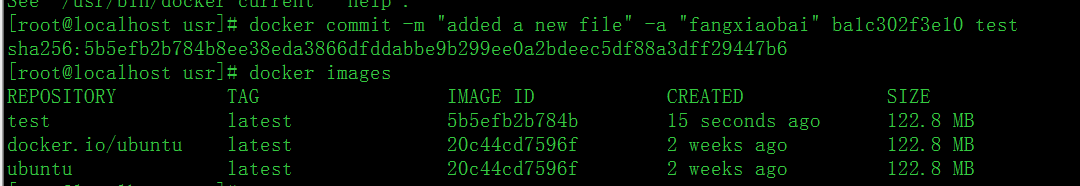
-m,--message=””, 提交信息

-p,-pause=true 提交是暂停容器运行

1）.进入docker容器中,创建一个文件test，退出



2）使用docker commit 创建一个新的镜像，重新查看image



2. 基于本地模板导入。

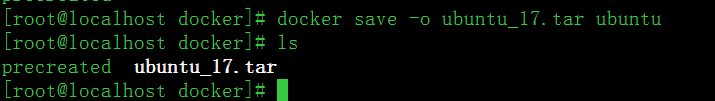
也可以直接从一个操作系统模板文件导入一个镜像。

首先下载一个模板压缩包。并使用以下命令导入：

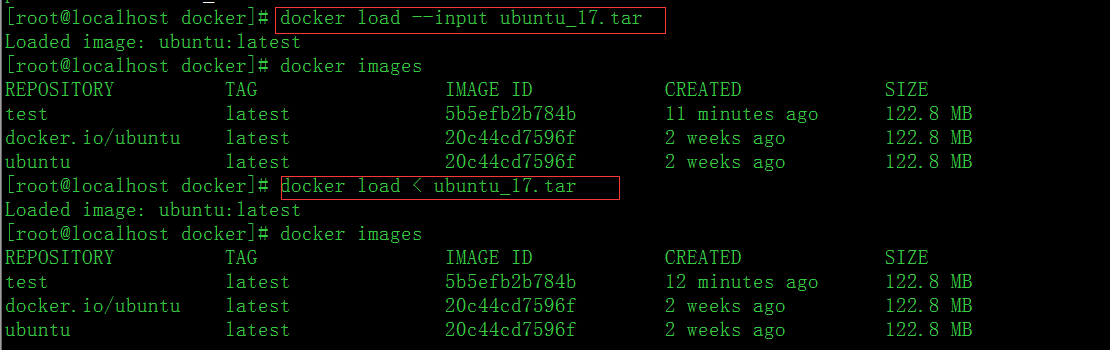
cat Ubuntu-14.04-xxx.tar.gz | docker import – Ubuntu:14.4

3.6 存出和载入镜像

将现有镜像存储为tar。



载入镜像：



3.7 使用docker push 命令上传镜像到仓库。

docker push NAME[:TAG]

3容器

可以使用docker create命令创建一个容器。

使用docker create新建的容器处于停止状态，可以使用docker start命令来启动。

3.1 新建并启动容器

docker create –it Ubuntu 新建的容器处于停止状态

docker start Ubuntu 启动它。

以上两条命令相当于一下：

Docker run –idt Ubuntu

上面的这条命令是docker在后台运行一下操作：

1. 检查本地是否存在指定的镜像。
2. 利用镜像创建并启动一个容器
3. 分配一个文件系统，并在只读的镜像层外面挂载一层可读写层。
4. 从宿主主机配置的网桥接口中接一个虚拟接口到容器中去
5. 从地址池配置一个IP地址给容器
6. 执行用户指定的应用程序
7. 执行完毕后容器被中止。

使用 docker run –t –i test /bin/bash 启动一个bash终端，允许用户进行交互。

-t 让docker打开一个伪终端，并绑定到容器的标准输入上，

-i 让容器的标准输入保持打开。

3.2 守护态运行

docker run –d

获取容器的信息，使用docker logs命令

3.3 终止容器

docker stop 来终止一个运行中的容器。

docker stop –t 10 dockerID 它会首先向容器中发送SIGTERM信号，等待一段时间后，在发送SIGKILL信号终止容器。

可以使用docker ps –a –q 命令查看处于终止状态的容器的ID信息。

处于终止状态的容器，可以用docker start重新启动

重新启动容器：docker restart <dockerId>。

3.4 进入容器

1.docker attach <dockerID>

当多个窗口同时attach到同一个容器中的时候，所有窗口会同步显示。当某个窗口因命令阻塞时，其他窗口也无法执行操作了。

2.自docker 1.3版本起，提供了一个简单的工具exec，可以直接在容器内运行命令。

Docker exec –ti <dockerID> /bin/bash

3.nsenter 工具

3.5 删除容器

docker rm [OPTIONS] container [container…..] 支持的选项同删除镜像一样(docker rmi IMAGES)

3.6 导入导出容器

docker export <containerId> > EXPORT\_NAME 导出容器

cat IMPORT\_NAME | docker import – test/Ubuntu 导入容器

实际上既可以使用docker load命令来导入镜像存储文件到本地的镜像库。又可以使用docker import 命令来导入一个容器快照到本地镜像库。这两者的区别在于容器快照文件将丢弃所有的历史记录和元数据记录(仅保存容器当时的快照状态)，而镜像存储文件将保存完成记录，体积要大。从容器快照文件导入时可以重新指定标签等元数据。

4.仓库

仓库是存放镜像的地方。

注册服务是存放仓库的集体服务器。每个服务器上可以有多个仓库，而每个仓库下面有多个镜像。

4.1 docker hub

公共仓库：<https://hub.docker.com>

1. 登录：

通过执行docker login命令输入用户名，密码，邮箱来完成注册和登录

1. 基本操作：

用户无需登录可以用docker search命令来查找官方仓库中镜像。并利用docker pull命令将其下载到本地。

用户还可以在登录后将自己的镜像推送到Docker hub 使用的命令为：docker push.

1. 自动创建

自动创建功能使得用户通过Docker Hub指定跟踪一个目标网站上的项目，一旦发现项目有新的提交，则自动执行创建。

要配置自动创建，包括一下步骤：

1. 创建并登录Docker Hub，以及目标网站，在目标网站中连接到账户Docker Hub
2. 在Docker Hub中配置一个自动创建。
3. 选取一个目标网站中的项目（需要含docker file）和分支。
4. 指定Dockerfile的位置，并提交创建。

之后，可以在docker hub的“自动创建”页面中跟踪每次创建的状态。

4.2 docker Pool 简介

**1.查看镜像**

<http://www.dockerpool.com/downloads> 可以看现有的仓库和存储的镜像。

**2.下载镜像**

docker pull 但是要在镜像名称前，添加注册服务器的具体地址dl.dockerpool.com:5000

docker pull dl.dockerpool.com:5000/Ubuntu Ubuntu:12.04

docker tag dl.dockerpool.com:5000/Ubuntu:12.04 ubuntu

4.3 创建和使用私有仓库

**使用registry镜像创建私有仓库**。

安装docker后，可以通过官方提供的registry惊险过来搭建一套本地私有仓库环境：

docker run –d –p 5000:5000 registry

这将自动下载并启动一个registry容器，创建本地的私有仓库服务。

通过 –v 指定位置，默认：/tmp/registry/

管理私有仓库镜像