Docker 入门实践笔记

# 基础

两个重要的概念：

## 1.镜像

创建方法：

1）拿一个已经存在的镜像，对它进行修改来创建子镜像

实现方法：

a)在一个文件中指定一个基础镜像及需要完成的修改

b)通过运行一个镜像，对其进行修改并提交

查看镜像

docker images

## 2.容器

1)运行容器

docker run -p 80:80 –name mynginx -v $PWD/www:/www -v $PWD/conf/nginx.conf:/etc/nginx/nginx.conf -v $PWD/logs:/www/logs -d nginx

-p 80:80 ：将容器的80端口映射到主机的80端口

--name mynginx :将容器命名为mynginx

-v $PWD/www:/www :将主机中当前目录下的www挂载到容器的/www

-v $PWD/logs:/var/log/nginx :将nginx的access.log 和 error.log放到$PWD/logs目录下

2)查看容器启动情况

docker ps

## 3.数据卷

## 4. 镜像和容器常用操作指令

1). 删除一个容器

docker rm id(容器id)

docker rm $(docker ps -a -q)

-a all :所有容器

-q quite: 只显示容器id

注意：删除容器之前要先Stop 容器

2).停止一个容器

 docker stop id

3). 删除一个镜像

docker rmi id(镜像id)

4).查看某个容器的日志

docker logs ID

5).查看本地当前所有镜像

docker images

6).查看镜像历史版本

docker history image\_name

7).获取镜像

docker pull image\_name:tag

8).运行通过镜像来启动容器

docker run -it image\_name /bin/bash

-i 交互模式

-t 伪终端

9).启动、进入一个已有的容器

docker start containerId

docker attach containerId (old)

docker exec -it containerId /bin/bash (new)

10).宿主机与docker 的容器实现文件共享

docker run -it -v /宿主路径:/容器内路径 image\_name:tag /bin/bash

更高级的用法是数据卷

11).将容器保存为镜像

docker commit 容器id image\_name:tag

## 5.Dockerfile

约定：默认指令应该为大写

例如

FROM ubuntu

1).FROM 从FROM 开始，表示指定基础镜像

FROM ubuntu

2).MAINTAINER,指定作者

MAINTAINER <author\_name>

3).RUN 在shell或exec的环境下执行的命令

4).ADD 复制文件指令，有两个参数source 和destination

destination为容器内的路径，source可以是URL或启动配置上下文的一个文件

ADD source destination

5).CMD,提供容器默认的执行命令，只允许使用一次CMD指令，使用多个会抵消之前的所有指令，只有最后一次生效，CMD的三种形式：

CMD [‘executable’, ‘param1’, ‘param2’]

CMD [‘param1’, ‘param2’]

CMD command param1 param2

6).EXPOSE 指定容器在运行时监听的端口，语法如下:

EXPOSE <port>;

7).ENTRYPOINT配置给容器一个可执行命令，意味着该镜像每次被调用时仅能运行指定的应用

8).WORKDIR 指定RUN、CMD、ENTRYPOINT的工作目录

9).VOLUME 授权访问从容器内到主机上的目录

VOLUME [‘/data’]