docker基本命令

参考: http://my.oschina.net/2xixi/blog/516951

**Dockerfile**

 [FROM](http://my.oschina.net/2xixi/blog/516951#OSC_h4_4)

 [RUN](http://my.oschina.net/2xixi/blog/516951#OSC_h4_5)

 [MAINTAINER](http://my.oschina.net/2xixi/blog/516951#OSC_h4_6)

 [CMD](http://my.oschina.net/2xixi/blog/516951#OSC_h4_7)

 [ENTRYPOINT](http://my.oschina.net/2xixi/blog/516951#OSC_h4_8)

 [USER](http://my.oschina.net/2xixi/blog/516951#OSC_h4_9)

 [EXPOSE](http://my.oschina.net/2xixi/blog/516951#OSC_h4_10)

 [ENV](http://my.oschina.net/2xixi/blog/516951#OSC_h4_11)

 [ADD](http://my.oschina.net/2xixi/blog/516951#OSC_h4_12)

 [VOLUME](http://my.oschina.net/2xixi/blog/516951#OSC_h4_13)

 [WORKDIR](http://my.oschina.net/2xixi/blog/516951#OSC_h4_14)

 [ONBUILD](http://my.oschina.net/2xixi/blog/516951#OSC_h4_15)

FROM

基于哪个镜像

RUN

安装软件用

MAINTAINER

镜像创建者

CMD

container启动时执行的命令，但是一个Dockerfile中只能有一条CMD命令，多条则只执行最后一条CMD.

CMD主要用于container时启动指定的服务，当docker run command的命令匹配到CMD command时，会替换CMD执行的命令。如:  
Dockerfile:

CMD echo hello world

运行一下试试:

edwardsbean@ed-pc:~/software/docker-image/centos-add-test$ docker run centos-cmd

hello world

一旦命令匹配：

edwardsbean@ed-pc:~/software/docker-image/centos-add-test$ docker run centos-cmd echo hello edwardsbean

hello edwardsbean

ENTRYPOINT

container启动时执行的命令，但是一个Dockerfile中只能有一条ENTRYPOINT命令，如果多条，则只执行最后一条

ENTRYPOINT没有CMD的可替换特性

USER

使用哪个用户跑container  
如：

ENTRYPOINT ["memcached"]

USER daemon

EXPOSE

container内部服务开启的端口。主机上要用还得在启动container时，做host-container的端口映射：

docker run -d -p 127.0.0.1:33301:22 centos6-ssh

container ssh服务的22端口被映射到主机的33301端口

ENV

用来设置环境变量，比如：

ENV LANG en\_US.UTF-8 ENV LC\_ALL en\_US.UTF-8

ADD

将文件<src>拷贝到container的文件系统对应的路径<dest>  
所有拷贝到container中的文件和文件夹权限为0755,uid和gid为0  
如果文件是可识别的压缩格式，则docker会帮忙解压缩

* 如果要ADD本地文件，则本地文件必须在 docker build <PATH>，指定的<PATH>目录下
* 如果要ADD远程文件，则远程文件必须在 docker build <PATH>，指定的<PATH>目录下。比如:

docker build github.com/creack/docker-firefox

docker-firefox目录下必须有Dockerfile和要ADD的文件

注意:使用docker build - < somefile方式进行build，是不能直接将本地文件ADD到container中。只能ADD url file.

ADD只有在build镜像的时候运行一次，后面运行container的时候不会再重新加载了。

VOLUME

可以将本地文件夹或者其他container的文件夹挂载到container中。

WORKDIR

切换目录用，可以多次切换(相当于cd命令)，对RUN,CMD,ENTRYPOINT生效

ONBUILD

ONBUILD 指定的命令在构建镜像时并不执行，而是在它的子镜像中执行

Docker镜像导出及载入

docker save -o saveim.tar dn/test:testfile

docker load --input saveim.tar

移除本地镜像

docker rmi

docker rm(删除容器)

Docker 容器

docker run ubuntu:14.04 /bin/echo 'Hello world'

docker run -t -i ubuntu:14.04 /bin/bash

-t 选项让Docker分配一个伪终端（ pseudo-tty） 并绑定到容器的标准输入上， -i 则让容器的标  
准输入保持打开。

docker run

检查本地是否存在指定的镜像， 不存在就从公有仓库下载  
利用镜像创建并启动一个容器  
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  
从宿主主机配置的网桥接口中桥接一个虚拟接口到容器中去  
从地址池配置一个 ip 地址给容器  
执行用户指定的应用程序  
执行完毕后容器被终止

守护进程

do docker run -d ubuntu:14.04

容器输出信息

docker logs dn/centos

终止容器

docker stop

docker ps -a 查看包括终止状态的容器

进入容器

docker attach

或者

nsenter

导出和导入容器

docker export 7691a814370e > ubuntu.tar

这样将导出容器快照到本地文件。

cat ubuntu.tar | sudo docker import - test/buntu:v1.0

此外， 也可以通过指定 URL 或者某个目录来导入， 例如

docker import http://example.com/exampleimage.tgz example/imagerepo

\*注：用户既可以使用 docker load 来导入镜像存储文件到本地镜像库， 也可以使用 docker import 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信息  
（ 即仅保存容器当时的快照状态） ， 而镜像存储文件将保存完整记录， 体积也要大。 此外， 从容器快照文  
件导入时可以重新指定标签等元数据信息。

删除容器

docker rm trusting\_newton

仓库

仓库地址 dl.dockerpool.com/ubuntu 来说， dl.dockerpool.com 是注册服务器地  
址， ubuntu 是仓库名。

docker数据管理

数据卷

-v

数据卷容器

--volume-from

网络

外部访问容器

-P Docker 会随机映射一个 49000~49900 的端口到内部容器开放的网络端口。

-p （ 小写的） 则可以指定要映射的端口， 并且， 在一个指定端口上只可以绑定一个容器。 支持的格式有  
ip:hostPort:containerPort | ip::containerPort | hostPort:containerPort 。

实时查看输出信息

docker logs -f nostalgic\_morse

查看容器内部变量

docker inspect