Docker命令手册-笔记

作者：杨鹏飞

时间：2016-01-13

目录

[1 序言 1](#_Toc440489488)

[2 docker 2](#_Toc440489489)

[3 docker run子命令详解 3](#_Toc440489490)

[4 FAQ 5](#_Toc440489491)

[5 更新时间 6](#_Toc440489492)

# 序言

# Docker

-D 默认false 允许调试模式(debugmode)

-H 默认是unix:///var/run/docker.sock tcp://[host[:port]]来绑定或者

unix://[/path/to/socket]来使用(二进制文件的时候)，当主机ip host=[0.0.0.0],(端口)port=[4243] 或者 path=[/var/run/docker.sock]是缺省值，做为默认值来使用

--api-enable-cors默认flase允许CORS header远程api

-b 默认是空，附加在已存在的网桥上，如果是用'none'参数，就禁用了容器的网络

--bip默认是空，使用提供的CIDR

（ClasslessInter-Domain Routing-无类型域间选路）标记地址动态创建网桥(dcoker0),和-b参数冲突

-d 默认false 允许进程模式(daemonmode)

--dns默认是空，使ducker使用指定的DNS服务器

-g 默认是"/var/lib/docker":作为docker使用的根路径

--icc默认true，允许inter-container来通信

--ip默认"0.0.0.0"：绑定容器端口的默认Ip地址

--iptables默认true 禁用docker添加iptables规则

--mtu默认1500 : 设置容器网络传输的最大单元(mtu)

-p 默认是/var/run/docker.pid进程pid使用的文件路径

-r 默认是true 重启之前运行的容器

-s 默认是空，这个是docker运行是使用一个指定的存储驱动器

-v 默认false 打印版本信息和退出

## docker run子命令详解

    Usage: docker run [OPTIONS] IMAGE[:TAG] [COMMAND] [ARG...]

a=map[]: 附加标准输入、输出或者错误输出

-c=0: 共享CPU格式（相对重要）

-cidfile="": 将容器的ID标识写入文件

-d=false: 分离模式，在后台运行容器，并且打印出容器ID

-e=[]:设置环境变量

-h="": 容器的主机名称

-i=false: 保持输入流开放即使没有附加输入流

--privileged=false: 给容器扩展的权限

-m="": 内存限制 (格式:<number><optional unit>, unit单位 = b, k, m or g)

-n=true: 允许镜像使用网络

-p=[]: 匹配镜像内的网络端口号

--rm=false:当容器退出时自动删除容器 (不能跟 -d一起使用)

-t=false: 分配一个伪造的终端输入

-u="": 用户名或者ID

--dns=[]: 自定义容器的DNS服务器

-v=[]: 创建一个挂载绑定：

[host-dir]:[container-dir]:[rw|ro].如果容器目录丢失，docker会创建一个新的卷

--volumes-from="": 挂载容器所有的卷

--entrypoint="": 覆盖镜像设置默认的入口点

-w="": 工作目录内的容器

--lxc-conf=[]: 添加自定义

-lxc-conf="lxc.cgroup.cpuset.cpus = 0,1"

--sig-proxy=true: 代理接收所有进程信号(even in non-tty mode)

--expose=[]: 让你主机没有开放的端口

--link="": 连接到另一个容器(name:alias)

--name="": 分配容器的名称，如果没有指定就会随机生成一个

-P=false: Publish all exposed ports to thehost

### docker –run –v【很实用】

用-v挂载主机数据卷到容器内(还未测试)

docker run -v /path/to/hostdir:/mnt $container

## docker build

### 语法

docker build -t <镜像名> <Dockerfile路径>

### 示例

如Dockerfile在当前路径：

docker build -t xx/gitlab .

## docker cp【很实用】

### 语法

docker cp <containerId>:/file/path/within/container /host/path/target

### 示例

从container中拷贝文件出来，从容器a69e15b2034b的/opt/test.txt复制到当前目录

[root@master ~]# docker cp a69e15b2034b:/opt/test.txt .

[root@t2t showdoc]# docker cp 9146f086b5b3:/var/www/html/Sqlite/showdoc.db.php /root/

## docker快捷命令

### 删除所有镜像

docker rmi $(docker images | grep none | awk '{print $3}' | sort -r)

### 删除所有正在运行的容器

docker rm $(docker ps -a -q)

### 停止、启动、杀死一个容器

docker stop <容器名or ID>

docker start <容器名or ID>

docker kill <容器名or ID>

### 查看所有镜像

docker images

# Dockerfile

总结一下，基本常用的命令是: FROM, MAINTAINER, RUN, USER, PORT, ADD，EXPOSE，ENTRYPOINT，CMD

## EXPOSE 命令

EXPOSE 命令可以设置一个端口在运行的镜像中暴露在外

EXPOSE <port> [<port>...]

## ENV 命令

用于设置环境变量

ENV <key> <value>

## WORKDIR 命令

WORKDIR /path/to/workdir

配置RUN, CMD, ENTRYPOINT 命令设置当前工作路径

可以设置多次，如果是相对路径，则相对前一个 WORKDIR 命令

比如:

WORKDIR /a WORKDIR b WORKDIR c RUN pwd

其实是在 /a/b/c 下执行 pwd

## CMD 命令

有三种格式:

CMD ["executable","param1","param2"] (like an exec, preferred form)

CMD ["param1","param2"] (as default parameters to ENTRYPOINT)

CMD command param1 param2 (as a shell)

## ADD

ADD的另外一个特性是有能力自动解压文件。如果<src>参数是一个可识别的压缩格式（tar, gzip, bzip2, etc）的本地文件（所以实现不了同时下载并解压），就会被解压到指定容器文件系统的路径<dest>

# FAQ

## RUN、CMD、ENTRYPOINT的区别？

RUN是在Build时运行的，先于CMD和ENTRYPOINT。Build完成了，RUN也运行完成后，再运行CMD或者ENTRYPOINT。

ENTRYPOINT和CMD的不同点在于执行docker run时参数传递方式，CMD指定的命令可以被docker run传递的命令覆盖，例如，如果用CMD指定：

CMD ["echo"]

然后运行

docker run CONTAINER\_NAME echo foo

# 更新时间

2016-01-13 周三，整理docker命令参数；