

DOCKER

# Docker使用指南

Posted on 08月 9, 201508月 14, 2015 in docker

## Docker使用指南

### Docker常用指令

#### 容器使用顺序

- `docker create` 建立新的容器（通过docker ps显示容器名与ID）
- `docker run` 建立并运行新的容器
- `docker stop` 停止容器
- `docker start` 开始停止状态下的容器
- `docker restart` 重启容器
- `docker rm` 删除容器
- `docker kill` 发给容器SIGKILL信号
- `docker attach` 链接容器
- `docker wait` 等待直到容器停止

#### 管理镜像

##### 搜索镜像

指令：`docker search`

说明：搜索Docker Hub上的镜像文件

例：`docker search --starts=10 ubuntu`

说明：列出十颗星以上的含有ubuntu关键词的镜像

##### 下载镜像

指令：`docker pull`

例：`docker pull ubuntu`

说明：下载最新版本名为ubuntu的镜像。下载后通过docker images指令可以列出你电脑上现有的镜像文件

## 运行镜像（创建新的容器）

指令：`docker run`

例1：`docker run -i -t centos:latest /bin/bash`

说明：其中-i是表示interactive，-t是表示tty。

例2：`docker run -d ubuntu /bin/sh -c "while true; do echo hello world; sleep 1; done"`

说明：大多数应用程序是以后台驻留形式运行的。这里-d表示daemon。

例3：`docker run myimage "{CMD} -extraparams"`

说明：在Dockerfile里CMD命令用来定义需要运行的程序。通过{CMD}你可以在运行时覆盖这个Dockerfile里设置的命令。（参照：[Docker Reference](#)）

例4：`docker run -d -p 18080:8080 -v /data/logs:/opt/glassfish/logs -v /data/logs:/var/log tutm/glassfish`

说明：这里-p表示port用来设置端口映射（这里Boot2Docker端口：18080，容器端口：8080）。-v表示volume用来设置磁盘卷映射（这里把容器上的/opt/glassfish/logs和/var/log/同时映射到Boot2Docker上的/data/logs文件夹里）

## 删除镜像

指令：`docker rmi`

例1：`docker rmi ubuntu:latest`

说明：删除某一个镜像

例：`docker rmi $(docker images -a -q)`

说明：删除所有镜像

## 容器管理

### 容器列表

指令：`docker ps`

例：`docker ps -a`

说明：显示容器列表，包括容器的ID，运行状态等。其中a是表示显示所有的容器（包括未激活状态）。

### 停止容器

指令：`docker stop`

例1：`docker stop <容器ID>`

说明：停止某一个容器

例2：`docker stop $(docker ps -a -q)`

说明：停止所有容器

### 重启容器

指令：`docker restart`

例：`docker restart <容器ID>`

说明：重启容器

### 连接容器

指令：`docker attach`

例：`docker attach <容器ID>`

说明：重新连接容器

指令：`docker exec`

例：`docker exec -i -t <容器ID> bash`

说明：在容器上运行其他指令，比如在后台模式下建立bash链接

## 删除容器

指令：`docker rm`

例1：`docker rm <容器ID>`

说明：删除某一个容器，如果容器处于激活状态可以通过-f（force）强制删除。

例：`docker rm $(docker ps -a -q)`

说明：删除所有容器

## 显示日志

指令：`docker logs`

例：`docker logs -f <容器ID>`

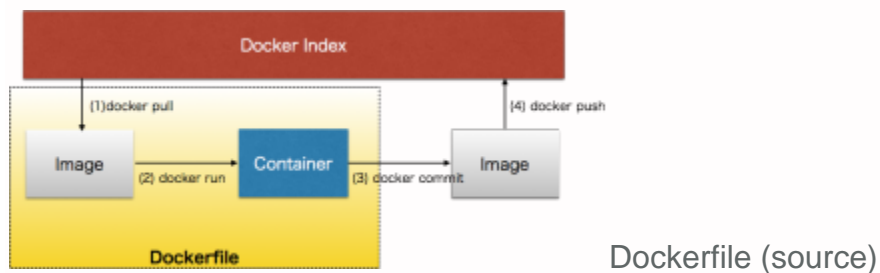
说明：支持json-file日志的容器。参数-f（follow）显示容器的STDOUT和STDERR(参照：[Docker Reference](#))

## *Dockerfile*

### 创建镜像

## Dockerfile创建容器步骤

1. 创建Dockerfile文件。在Dockerfile里：
  - a. 下载合适的基础系统镜像
  - b. 修改镜像
  - c. 设置命令行
2. 运行`docker build -t dev/dockertest:v20150810 .`（参数-t表示tag，这里镜像的repository为dev/dockertest，它的tag为v20150810）



Dockerfile (source)

例：一个在debian平台上运行run.sh的Dockerfile

```
FROM debian:latest

MAINTAINER devops@bogotobogo.com

# 1 - RUN

RUN apt-get update && DEBIAN_FRONTEND=noninteractive apt-get install -yq apt-utils

RUN DEBIAN_FRONTEND=noninteractive apt-get install -yq htop

RUN apt-get clean

# 2 - CMD
```

```
CMD ["htop"]
```

```
CMD ["ls", "-l"]
```

```
# 3 - WORKDIR and ENV
```

```
WORKDIR /root
```

```
ENV DZ version1
```

```
# 4 - ADD
```

```
ADD run.sh /root/run.sh
```

```
CMD ["/run.sh"]
```

```
# 5 - ENTRYPOINT (vs CMD)
```

```
ENTRYPOINT ["/run.sh"]
```

```
CMD ["arg1"]
```

---

## HTTP/HTTPS Proxy

在Dockerfile里添加HTTP/HTTPS代理：

- ENV http\_proxy http://server:port
- ENV https\_proxy http://server:port