## 列出镜像

docker image：列出已经下载下来的镜像

docker system df：查看镜像、容器、数据卷所占用的空间

## 删除镜像

docker image rm [选项] <镜像1> [<镜像2> ...]

## Dockerfile指令详解

1. FROM 指定基础镜像

FROM 就是指定基础镜像，一个Dockerfile中FROM是必备的指令，并且必须是第一条指令。

1. RUN 执行命令

有两种格式：

1. shell格式：RUN <shell命令>如：

RUN echo '<h1>Hello, Docker!</h1>' > /usr/share/nginx/html/index.html

（2）exec格式：RUN ["可执行文件", "参数1", "参数2"]

错误用法：

FROM debian:stretch

RUN apt-get update

RUN apt-get install -y gcc libc6-dev make wget

RUN wget -O redis.tar.gz "http://download.redis.io/releases/redis-5.0.3.tar.gz"

RUN mkdir -p /usr/src/redis

RUN tar -xzf redis.tar.gz -C /usr/src/redis --strip-components=1

RUN make -C /usr/src/redis

RUN make -C /usr/src/redis install

正确用法：

FROM debian:stretch

RUN buildDeps='gcc libc6-dev make wget' \

&& apt-get update \

&& apt-get install -y $buildDeps \

&& wget -O redis.tar.gz "http://download.redis.io/releases/redis-5.0.3.tar.gz" \

&& mkdir -p /usr/src/redis \

&& tar -xzf redis.tar.gz -C /usr/src/redis --strip-components=1 \

&& make -C /usr/src/redis \

&& make -C /usr/src/redis install \

&& rm -rf /var/lib/apt/lists/\* \

&& rm redis.tar.gz \

&& rm -r /usr/src/redis \

&& apt-get purge -y --auto-remove $buildDeps

注意：

a.应该将相关的shell命令合成一个，避免产生多层镜像

b.写 Dockerfile 的时候，要经常提醒自己，这并不是在写 Shell 脚本，而是在定义每一层该如何构建

c. 每一层构建的最后一定要清理掉无关文件

3.COPY复制文件

格式：

COPY [--chown=<user>:<group>] <源路径>... <目标路径>

COPY [--chown=<user>:<group>] ["<源路径1>",... "<目标路径>"]

4.CMD 容器启动命令

两种格式：

shell 格式：CMD <命令>

exec 格式：CMD ["可执行文件", "参数1", "参数2"...]

注意：

1. 在指令格式上，一般推荐使用 exec 格式，这类格式在解析时会被解析为 JSON 数组，因此一定要使用双引号 "，而不要使用单引号
2. 容器中应用在前台执行和后台执行的问题

5.ENV 设置环境变量

格式有两种：

ENV <key> <value>

ENV <key1>=<value1> <key2>=<value2>...

注意：

1. 定义了环境变量，那么在后续的指令中，就可以使用这个环境变量

ENV NODE\_VERSION 7.2.0

RUN curl -SLO "https://nodejs.org/dist/v$NODE\_VERSION/node-v$NODE\_VERSION-linux-x64.tar.xz" \

&& curl -SLO "https://nodejs.org/dist/v$NODE\_VERSION/SHASUMS256.txt.asc" \

&& gpg --batch --decrypt --output SHASUMS256.txt SHASUMS256.txt.asc \

&& grep " node-v$NODE\_VERSION-linux-x64.tar.xz\$" SHASUMS256.txt | sha256sum -c - \

&& tar -xJf "node-v$NODE\_VERSION-linux-x64.tar.xz" -C /usr/local --strip-components=1\

&& rm "node-v$NODE\_VERSION-linux-x64.tar.xz" SHASUMS256.txt.asc SHASUMS256.txt \

&& ln -s /usr/local/bin/node /usr/local/bin/nodejs

1. 下列指令可以支持环境变量展开： ADD、COPY、ENV、EXPOSE、LABEL、USER、WORKDIR、VOLUME、STOPSIGNAL、ONBUILD

## 操作Docker容器

### 启动

#### 新建并启动

docker run 命令

如：docker run ubuntu:18.04 /bin/echo 'Hello world'

当利用 docker run 来创建容器时，Docker 在后台运行的标准操作包括：

1.检查本地是否存在指定的镜像，不存在就从公有仓库下载

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桥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.执行完毕后容器被终止

#### 启动已终止容器

docker container start <images>

### 后台运行

让 Docker 在后台运行而不是直接把执行命令的结果输出在当前宿主机下。此时，可以通过添加 -d 参数来实现。

如：docker run ubuntu:18.04 /bin/sh -c "while true; do echo hello world; sleep 1; done"

注： 容器是否会长久运行，是和 docker run 指定的命令有关，和 -d 参数无关。

要获取容器的输出信息，可以通过 docker container logs 命令。

docker container logs [container ID or NAMES]

### 终止容器

docker container stop <containerId or name>

docker container ls ：查看正在运行的容器

docker container ls –a ：查看正在运行和已经终止的容器

### 进入容器

1. attach 命令

如：docker –itd attach 243c

1. exec 命令

如：docker exec -it 69d1 bash

注意：使用attach进入容器后，用exit退出，会导致容器停止；而exec则不会。推荐使用exec。

### 删除容器

docker container rm <containerId or name> :删除一个终止状态的容器

docker container rm –f <containerId or name>:删除一个运行中的容器

docker container prune:删除所有处于终止状态的容器

window下：

image说明文件中的命令是linux下的命令，在window下执行会出错：

```

$ docker run -it --rm -v "$PWD":"$PWD" -w "$PWD" -u "$(id -u)" ebiven/vue-cli vue

docker: Error response from daemon: the working directory '$PWD' is invalid, it needs to be an absolute path.

See 'docker run --help'.

```

在window下需要使用 `%cd%` 输出当前目录，如下：

```

docker run -it --rm -v "%cd%":/code -w /code -p 8080:8080 ebiven/vue-cli vue

// -it: 进行交互式操作

// --rm：关闭该 container 后删除文件

// -v %cd%:/code: 将执行命令的目录映射到container的/code目录

// -w /code: 设置命令执行目录为/code

// -p 8080:8080：将本机的8080端口映射为container的8080端口

docker run –d –p 8080:8080 –w /root/vueproject ubuntu-vue npm run start

–w /root/vueproject设置npm run start命令的执行目录为/root/vueproject