



链滴

# 制作自己的 Docker 镜像

作者: [qq314872695](#)

原文链接: <https://ld246.com/article/1587043192521>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

# 容器转为镜像

- 使用 `docker commit` 命令将容器转换成镜像

`docker commit` 容器id 镜像名称:版本号

- 需要转移镜像时，将该镜像打成一个包

`docker save -o 压缩文件名称 镜像名称:版本号`

- 在另一台电脑加载这个镜像时，加载这个包

`docker load -i 压缩文件名称`

## 使用dockerfile创建镜像（推荐）

dockerfile是一个文本文件，包含了一条条指令，每条指令构建一层，基于基础镜像，最终构建出新的镜像。

## dockerfile用到的关键字

| 关键字   | 作用           | 备注   |
|---|--------------|--|
| FROM<br>基于那个image构建   | 指定父镜像        | 指定dockerfile                                 |
| MAINTAINER<br>这个dockerfile谁写的   | 作者信息         | 用来标  |
| LABEL<br>标签 可以使用Label代替Maintainer   | 标签           | 用来标明dockerfile<br>最终都是在docker image基本信息中可以查看 |
| RUN<br>是/bin/sh 格式: RUN command 或者 RUN ["command", "param1", "param2"]                                | 执行命令         | 执行一段命令 默                                     |
| CMD<br>时候的默认命令 和ENTRYPOINT配合使用.格式 CMD command param1 param2 或者 CMD ["co<br>mand", "param1", "param2"] | 容器启动命令       | 提供启动容  |
| ENTRYPOINT<br>些执行就关闭的容器中会使用   | 入口           | 一般在制作  |
| COPY<br>文件到image中   | 复制文件         | build的时候复                                    |
| ADD<br>件到image中 不仅仅局限于当前build上下文 可以来源于远程服务  | 添加文件         | build的时候添加                                   |
| ENV<br>境变量 可以在启动的容器的时候 通过-e覆盖 格式ENV name=value  | 环境变量         | 指定build时候的                                   |
| ARG<br>的时候使用的参数 如果有ENV 那么ENV的相同名字的值始终覆盖arg的参数   | 构建参数         | 构建参数 只在构                                     |
| VOLUME<br>定build的image那些目录可以启动的时候挂载到文件系统中 启动容器的时候使用 -v 绑定 格式 VOL<br>ME ["目录"]                         | 定义外部可以挂载的数据卷 |  |

|  |              |            |
|--|--------------|------------|
| EXPOSE   | 暴露端口         | 定义容器运行     |
| 时候监听的端口 启动容器的使用-p来绑定暴露端口 格式: EXPOSE 8080 或者 EXPOSE 8080/udp    |              |            |
| WORKDIR  | 工作目录         | 指定容器       |
| 部的工作目录 如果没有创建则自动创建 如果指定/ 使用的是绝对地址 如果不是/开头那么是在上一级orkdir的路径的相对路径 |              |            |
| USER   | 指定执行用户       | 指定build    |
| 者启动的时候 用户 在RUN CMD ENTRYPOINT执行的时候的用户                          |              |            |
| HEALTHCHECK  | 健康检查         | 指定         |
| 测当前容器的健康监测的命令 基本上没用 因为很多时候 应用本身有健康监测机制                         |              |            |
| ONBUILD  | 触发器          | 当存在ONBUILD |
| D关键字的镜像作为基础镜像的时候 当执行FROM完成之后 会执行 ONBUILD的命令 但是不影响当镜像 用处也不怎么大   |              |            |
| STOPSIGNAL   | 发送信号量到宿主机    |            |
| STOPSIGNAL指令设置将发送到容器的系统调用信号以退出。                                |              |            |
| SHELL  | 指定执行脚本的shell | 指定         |
| UN CMD ENTRYPOINT 执行命令的时候 使用的shell                             |              |            |

## 自定义centos镜像dockerfile案例

要求:

- 默认登录路径为/usr
- 可以使用vim

实现步骤:

1. **vi centos\_dockerfile** 在文件中输入以下内容保存并退出:

FROM centos:7 # 定义父镜像

MAINTAINER itheima<itheima@itcast.cn> # 定义作者信息

RUN yum install -y vim # 执行安装vim命令

WORKDIR /usr # 定义默认的工作目录

CMD /bin/bash # 定义容器启动执行的命令

2. 通过 **centos\_dockerfile**构建镜像:

**docker bulid -f ./centos\_dockerfile -t 镜像名称:版本 .**

(注意最后还有个点, 表示指定镜像构建过程中的上下文环境的目录) , 由于网络的原因安装vim过程能会失败, 多执行几次该命令就好了。

## 部署Spring boot项目

需求:

- 定义dockerfile发布Spring boot项目

实现：

1. 新建 `springboot_dockerfile`文件, jar包和 `dockerfile`文件需要在同一个目录下

FROM java:8

MAINTAINER itheima<itheima@itcast.cn>

ADD springboot-hello-0.0.1-SNAPSHOT.jar app.jar # 把springboot项目的jar包添加到镜像中并  
个简短的名字app.jar

CMD java -jar app.jar # 运行jar包

2. 通过 `springboot_dockerfile`构建镜像

docker build -f ./springboot\_dockerfile -t app . # 新的镜像名称为app