# Doker

## 常用命令

Docker run –p 888:8080 –rm images 运行容器，并在退出时清除历史数据

Docker ps |grep tomcat 产看当前运行的容器的状态

Docker kill containerID 杀死某容器

Docker rmi –f imageid 强制删除某镜像

Docker build –t image:tag /usr 新建镜像，默认加载/usr下的Dockerfile构建文件

Docker save /load tar.tar imageid containerid 可组合打包多个镜像，但不包含文本文件

Docker export/import tar.tar imageid 只能为imagesid 打包整个文件系统，引入后原样恢复

## Docker的安全问题

安全性问题主要体现在是否会危害到宿主host的安全

两方面的安全漏洞

镜像的可信性

Docker daemon 的安全性：确保daemon的命令是可信的，如restapi，cli发送的操作命令

原因：

由于docker与宿主host公用内核，某个容器就有可能导致整个系统的奔溃

## 制作镜像

手动修改容器提交安装jdk

Docker run –itd imageName bash //运行某个镜像，-itd 命令行 交互 守护的方式运行，-d 保证了命令行执行exit时容器不停止，否则会停止容器，也可以通过 ctrl+P +q 。bash 表示运行使用默认bash命令 image为基础镜像文件，这里为centos镜像

Docker exec –it containerid bash 进入容器交互命令行模式

使用 yum install -y java-1.8.0-openjdk.x86\_6 //容器内执行命令安装jdk

Exit //退出容器，但不停止容器

Docker build containerid resposi：tag //提交容器生成镜像，docker commit [OPTIONS] CONTAINER [REPOSITORY[:TAG]]

DockerFile

注释通过#号

主要分为四部分：

基础镜像信息、维护者信息、镜像操作指令、容器启动指令

From Ubuntu #名为Ubuntu的基础镜像

RUN 命令

执行shell命令

执行linux上的可执行脚本

一般情况下一层镜像配置一个Run，因为每一个Run都会新建一层镜像

Run 为构建镜像时执行的命令

CMD

容器启动时执行的命令，构建时并不执行

**例子**  通过Dockerfile文件构建一个jdk环境的镜像

Docker search | grep centos //搜索centos基础镜像

Docker pull centos //拉去最新镜像文件到本地

编写dockerfile

FROM centos //指定基础镜像

RUN yum install -y java-1.8.0-openjdk.x86\_6 //使用yum安装jdk 注意-y一定在install后面 否则会报错

Copy /usr/local/tomcat /usr/local/tomcat

Volume /data //语法volume[“dir”,”dir”] 在docker run时不指定目录的话，会默认在/var/lib/docker下持久化数据,卷细节参见：<http://dockone.io/article/128>

可以通过 docker inspect 查看挂载信息

CMD java –version //启动容器时打印java版本

docker build -f /usr/local/Docker\_file/DockerFile . -t jdk\_centos:v1.3

//指定dockerfile文件 指定生成镜像的仓库和版本

// docker build [OPTIONS] PATH | URL path 表示docker执行上下文

**例子**：通过centos基础镜像制作jdk tomcat 运行环境，并绑定webapps、conf卷

**Dockerfile**：

FROM centos

RUN yum -y install java java-1.8.0-openjdk.x86\_64

COPY apache-tomcat-8.0.52\_64 /usr/local/tomcat

VOLUME ["/usr/local/tomcat/conf","/usr/local/tomcat/webapps"]

CMD java -version

CMD /usr/local/tomcat/bin/start.sh

**构建命令**

docker build -f /usr/local/Docker\_file/DockerFile -t tomcat\_centos:base /usr/local

指定Dockerfile文件， 指定仓库标签 指定build 上下文路径（dockerfile中文件会根据该路径找源文件，上面标红部分）

**运行命令**

docker run -it -v /usr/local/Docker\_volume/conf:/usr/local/tomcat/conf -v /usr/local/Docker\_volume/webapps:/usr/local/tomcat/webapps tomcat\_centos:base bash