



链滴

Docker 基本使用

作者: [QForever](#)

原文链接: <https://ld246.com/article/1577626156684>

来源网站: [链滴](#)

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

一、Docker基本命令

```
# 查看docker的版本信息
docker -v
# 查看docker的基本信息，例如多少镜像，多少个容器等
docker info
# 查看docker命令
docker help
# 查看本地docker镜像
docker images
# 启动
systemctl start docker
# 守护进程重启
sudo systemctl daemon-reload
# 重启docker服务
systemctl restart docker
# 重启docker服务
sudo service docker restart
# 关闭docker
service docker stop
# 关闭docker
systemctl stop docker
# 启动容器
docker run 镜像名:版本号
docker run hello-world:latest
# 查看正在运行的容器
docker ps
# 列出所有容器ID
docker ps -aq
# 停止所有容器
docker stop $(docker ps -aq)
# 停止单个容器
docker stop 要停止的容器名
docker stop 容器ID 平滑停止
docker kill 容器ID 暴力停止
# 删除单个容器
docker rm 要删除的容器名
docker rm [-f] 容器ID 强制删除,可以删除正在运行的容器
# 删除所有容器
docker rm $(docker ps -aq)
# 删除单个镜像
docker rmi 镜像名:版本号
docker rmi mysql:5.7.26
# 删除所有镜像
docker rmi $(docker images -q)
# 显示所有的容器，过滤出Exited状态的容器，取出这些容器的ID
sudo docker ps -a|grep Exited|awk '{print $1}'
# 查询所有的容器，过滤出Exited状态的容器，列出容器ID，删除这些容器
sudo docker rm `docker ps -a|grep Exited|awk '{print $1}'`
# 删除所有未运行的容器（已经运行的删除不了，未运行的就一起被删除了）
sudo docker rm $(sudo docker ps -a -q)
# 根据容器的状态，删除Exited状态的容器
sudo docker rm $(sudo docker ps -qf status=exited)
```

```
# 查看最近运行的两个容器
docker ps -n 2
# 以交互的模式启动容器
docker run -i -t centos
# 退出容器
exit      关闭容器并退出
ctrl+p+q  容器不关闭, 只是退出
# 后台启动容器
docker run -d centos
# 查看容器的日志
-t显示日志的时间, -f监视日志的增长
docker logs -t -f 容器ID
# 进入已退出的容器
docker attach 容器ID
# 不进入容器, 但是执行容器的操作
docker exec 容器ID ls
# 数据卷
/localData  宿主机里面的文件夹
/dockerData 容器里面的文件夹
docker run -it -v /localData:/dockerData centos:8.0
# 备份镜像
docker save 8.0:centos -o /home/qiang/docker/centos-8.0.tar.gz
# 导入镜像
docker load -i centos-8.0.tar.gz
# 修改镜像的仓库名和版本号
docker tag [IMAGE ID] [REPOSITORY名]:[TAG标识]
docker tag 0f3e07c0138f docker.io/centos:8.0.1959
# 获取docker容器中的root权限, 容器在执行状态
sudo docker exec -it -u root 容器ID bash
前提在run docker容器时候加上参数 --privileged=true
# 查看容器的信息
docker inspect 容器ID
# 进入容器
docker exec -it 容器ID /bin/bash
docker exec -it 容器ID /bin/sh
# 拉取镜像
docker pull 镜像名:版本号
docker pull mysql:5.7.26
# 拉取指定位置仓库的镜像
docker pull b3log/solo
```

二、Dockerfile

```
# 指定基础镜像
FROM centos:8.0
# 作者的信息
MAINTAINER 875667601@qq.com
# 执行什么命令
RUN yum install -y vim
# 容器对外暴露的端口
EXPOSE 80
# 进入到容器后进入到哪个目录
WORKDIR /usr/local/
```

```
# 配置环境变量
ENV JAVA_HOME=/datas/jdk1.8.0_60
ENV PATH=$JAVA_HOME/bin:$PATH
# 将文件拷贝到镜像中并解压
ADD jdk-8u60-linux-x64.tar.gz /datas/
# 将文件拷贝到镜像中
ADD jdk-8u60-linux-x64.tar.gz /datas/
# 配置数据卷
VOLUME ["/containerDataVolumn"]
# 容器启动时候执行的命令
CMD /bin/bash
# 容器启动时候执行的命令
ENTRYPOINT ["echo", "hello"]

# 构建镜像, 注意后面还有个点
docker build -t mycentos:8.0 .
```

三、修改Docker为阿里镜像

xxxxx具体去阿里云官网看

```
vim /etc/docker/daemon.json
{
  "registry-mirrors": ["https://xxxxxx.mirror.aliyuncs.com"]
}
#重新加载守护进程
sudo systemctl daemon-reload
#重启docker服务
sudo systemctl restart docker
```

四、启动和进入MySQL

```
docker run --name mysql -p 3306:3306 -e MYSQL_ROOT_PASSWORD=123456 -d mysql:5.7.2
docker exec -it mysql bash
```

五、启动参数说明

```
docker run
--detach或-d 后台运行
--rm 容器停止或退出就删除
--network=host 容器的端口都对应主机的端口, 不存在映射
--env或-e 设置系统的环境变量
--name 启动容器后的名字
--restart=always docker重启时容器也重启
```

六、从Docker容器复制文件

```
docker cp solo:/opt/solo/skins/nijigen /localData/solo/skins/nijigen
```