

# 轻量级虚拟机应用容器 Docker

作者: [jairwx](#)

原文链接: <https://ld246.com/article/1543030058280>

来源网站: [链滴](#)

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



## docker 简介

**定义：**轻量级虚拟机 应用容器

**优点：**

- 1.摆脱依赖库以及繁琐的配置
- 2.数据和内存控件相互隔离，保证一定的安全性

**与VM区别**

性能开销少，资源利用率高，启动速度快

VM在Hypervisor将硬件设备虚拟化，每个容器需要安装自己的os。

## docker安装

这个可以参考

### Windows安装方法

<http://www.runoob.com/docker/windows-docker-install.html>

### Ubuntu Docker 安装

<http://www.runoob.com/docker/ubuntu-docker-install.html>

### CentOS Docker 安装

<http://www.runoob.com/docker/centos-docker-install.html>

# Windows Docker 安装

<http://www.runoob.com/docker/windows-docker-install.html>

## docker命令

### docker容器的相关命令

1.查看运行的docker容器 `docker ps`

查看所有容器(包括已经停止运行的)

```
docker ps -a
```

2.查看某个容器的日志 `docker logs 镜像名称/id`

3.删除容器 `docker rm`

```
docker rm $(docker ps -a | awk '/training\/webapp/ {print $1}')
docker attach
```

4.在容器里执行命令 `docker exec`

```
docker exec -it <image id> /bin/bash
```

5.进入容器进行命令操作 `docker attach <容器id>`

使用docker attach命名进入docker容器后:

- 如果要正常退出不关闭容器, 请按Ctrl+P+Q进行退出容器。
- 如果使用exit退出, 那么在退出容器后会关闭容器

6.查看容器详细情况 `docker inspect`

查看容器ip

```
docker inspect --format='{{.Path}}{{.NetworkSettings.IPAddress}}' $(docker ps -a -q)
```

查看容器的挂载情况

```
docker inspect --format='{{.Path}}{{.Mounts}}' $(docker ps -a -q)
```

7.容器打包成镜像 `docker commit`

命令把一个正在运行的容器保存为镜像

```
docker commit [OPTIONS] CONTAINER [REPOSITORY[:TAG]]
```

```
docker commit <CONTAIN-ID> <IMAGE-NAME>
```

参数说明:

- a :提交的镜像作者;
- c :使用Dockerfile指令来创建镜像;

-m :提交时的说明文字;

-p :在commit时, 将容器暂停。

## 8.启动已存在的容器

`docker start`

9.查看docker状态 `docker stats {容器名称}`

10.docker复制文件操作 `docker cp`

示例

`docker cp /www/runoob 96f7f14e99ab:/www/`

11.导出 `docker export {docker容器}`

使用容器nginx, 现在要将这个容器保存为一个文件 nginx.tar

`docker export nginx > /home/nginx.tar`

12.导入 `docker import`

`docker import {tar文件} {镜像名称}`

`docker import hello.tar.zg hello`

13.查看端口映射 `docker port {容器名称}`

`docker port php  
9000/tcp => 0.0.0.0:9000`

14.更改映射

`iptables -t nat -A DOCKER -p tcp --dport host_port -j DNAT --to-destination docker_ip:$dock  
r_port`

比如 php-fpm 的ip 172.17.0.2

`sudo iptables -t nat -A DOCKER -p tcp --dport 2021 -j DNAT --to-destination 172.17.0.2:22`

## docker镜像相关命令

1.查看镜像列表 `docker images`

2.删除镜像 `docker rmi`

移除所有的容器和镜像

`docker kill $(docker ps -q)  
docker rm $(docker ps -a -q)  
docker rmi $(docker images -q -a)`

3.搜索镜像 `docker search`

4.获取镜像 `docker pull {镜像名称}`

如果拉取时间过慢则需要增加中国源

## 6.创建容器并启动 docker run

```
docker run -p {容器的IP}:{主机的IP} --name {容器的名称} -v {主机目录}:{容器目录} -d {镜像名称:本}
```

参数:

-p 端口映射

-v 目录映射

-it 打开stdin, 终端登录

-d 前台还是后台

win10使用挂载

```
docker run -v f:/user:/data alpine ls /data
```

## 运行镜像 直接进入容器

```
docker run -t -i xxx /bin/bash
```

## 附加: supervisord

```
docker run .... /usr/bin/supervisord
```

注: supervisor 是用 Python 开发的一套通用进程管理程序, 能将一个普通的命令行进程变为后台 daemon, 并监控进程状态, 异常退出时能自动重启。可通过supervisorctl status 查看supervisord当前管理的所有进程的状态

## 7.存出 docker save {镜像}

```
docker save -o quay.io-calico-node-1.tar quay.io/calico/node
```

有一个**基础镜像**centos,现在要将这个镜像保存为一个文件centos-save-1204.tar

```
docker save centos > /home/centos-save-1204.tar
```

## 8.载入 dokcer load

```
dokcer load -i XXX.tar
```

# 导出 export 与 保存 save 的区别

1 .export导出的镜像文件大小 小于 save保存的镜像

2 .export 导出 (import导入) 是根据容器拿到的镜像, 再导入时会丢失镜像所有的历史, 所以无法行回滚操作 (docker tag <LAYER ID> <IMAGE NAME>);而save保存 (load加载) 的镜像, 没有失镜像的历史, 可以回滚到之前的层 (layer)。(查看方式: docker images --tree)

# Dockerfile

## 1.使用Dockerfile

文件名: Dockerfile

