

<https://hub.docker.com>
Docker 官方提供公共镜像仓库

下载镜像

```
]#docker help pull  
]#docker pull docker.io/busybox
```

上传镜像

```
]#docker help push  
]#docker push docker.io/busybox
```

导出镜像

```
]#docker save docker.io/busybox:latest -o busybox.tar
```

导入镜像

```
]# scp busybox.tar root@ip 地址:/root  
]#docker load -i busybox.tar  
]#docker images
```

启动镜像

```
]#docker images  
]#docker run -it docker.io/镜像名:latest /bin/bash  
放在后台运行可以 -itd
```

在另一终端查看容器信息

```
]#docker run -it 运行容器  
]#docker ps 查看容器信息列表  
]#docker top 查看容器进程列表  
]#docker images //查看镜像列表  
]#docker history //查看镜像制作历史  
]#docker inspect //查看镜像底层信息
```

docker attach 进入容器 exit 会导致容器关闭
]#docker ps (-aq -q) 显示容器全部 ID 或 ID
]#docker attach b8f(容器列表 ID 前三位)
要想退出不关闭容器可以按快捷键 **ctrl+p+q**
]#docker restart|stop|start b8f 启动 关闭 重启容器

docker exec 进入容器 退出不会退出容器
docker exec -it b8f bash

docker rmi //删除本地镜像 docker rm \$(docker ps -aq)将后面执行结果交给前面执行
docker rm \$(docker stop \$(docker ps -qa))
docker save //镜像另存为 tar 包
docker load //使用 tar 包导入镜像
docker search //搜索镜像
docker tag //修改镜像名称和标签

二 创建自定义镜像

```
]#docker run -it docker.io/centos
]# docker ps
]#docker commit 8d07 docker.io/myos:latest
```

1.创建一个 Apache 镜像

```
]#mkdir build
]#cd build
]#touch Dockerfile
]#vi Dockerfile
FROM docker.io/myos:latest      //基础镜像
RUN yum -y install httpd        //装包
WORKDIR /var/www/html           //定义容器默认工作目录
ADD index.html index.html       //复制文件到镜像
ENV LANG=C                      //定义环境变量/etc/sysconfig/httpd
ENV PS1="[ \u@ \h \W] \ $"       // /etc/bashrc 搜索 PS1
EXPOSE 80                       //设置开放端口号
EXPOSE 443
CMD ["httpd","-DFOREGROUND"]    //容器启动时执行的命令,仅可以有一条 CMD
]# dockdocker build -t docker.io/myos:http .    //生产镜像
]#docker run -d  docker.io/myos:http           //启动容器
```

三 创建 自定义镜像仓库

1.定义一个私有仓库

```
]#yum install docker-distribution
]#systemctl start docker-distribution
]# systemctl enable docker-distribution
```

2.修改配置文件

```
vim /etc/sysconfig/docker
INSECURE_REGISTRY='--insecure-registry 仓库 ip:5000'
ADD_REGISTRY='--add-registry 仓库 ip:5000'
```

```
]# systemctl restart docker
```

上传镜像:

1.为镜像创建标签

```
]#docker tag myos:latest 仓库 ip:5000/myos:latest
]#docker images
```

2.上传镜像

```
]#docker push 仓库 ip:5000/myos:latest
```

3.验证 远程启动镜像

```
]#docker run -it 仓库 ip:5000/myos:latest
```

查看私有镜像仓库中的镜像名称

```
]#curl http://仓库 ip:5000/v2/_catalog
```

查看某一仓库的标签

```
curl http://仓库 ip:5000/v2/myos/tags/list
```

私有仓库数据存储目录

```
/var/lib/registry
```

四 持久化存储

docker 容器不保存任何数据

重要数据请使用外部卷存储(数据持久化)

容器可以挂载真实机目录或共享存储为卷

1.将真实机目录挂载到容器中提供持久化存储

```
]#mkdir /var/data
```

```
]#docker run -v /var/data:/data -it myos:latest
```

(真机目录) : (容器目录自动创建)

五 创建 **docker** 自定义网桥

1.查看网络

```
]#docker network list
```

```
]# ip a s docker0
```

```
]#brctl show docker0 //启动容器会绑定该网桥
```

2.新建网桥

```
]#docker network create --subnet=10.10.10.0/24 docker1
```

```
]#docker network list
```

```
]#ip a s
```

```
]#docker network inspect docker1
```

3.使用自定义网桥

```
]#docker run --network=docker1 -itd docker.io/myos
```

重要核心

默认容器可以访问外网,但外部网络不可以访问容器内的资源,容器的特征是可以把宿主机变成对应的服务

1.我们可以使用 **-p** 参数把容器端口和宿主机端口绑定

-p 宿主机端口:容器端口

```
]#docker run -itd -p 80:80 docker.io/myos:httpd
```

```
]#docker run -itd -p 80:80 docker.io/nginx:latest
```