# Docker

Docker是一个开源的应用容器引擎，它可以解决运行环境和配置问题，方便做持续集成并且有助于整体发布的容器虚拟化技术。

可以让我们打包他们的应用以及依赖包到一个轻量级、可移植的容器中，然后发布到任何流行的Linux机器上，也可以实现虚拟化，容器是完全使用沙箱机制，相互之间不会有任何接口，重要的是容器性能开销极低，docker的主要目标就是：一次封装，到处运行。

Docker主要有三部分镜像，容器，仓库

**镜像**是一种轻量级、可执行的独立软件包，用来打包软件运行环境和基于运行环境开发的软件，它包含运行某个软件所需的所有内容，包括代码、库、环境变量，配置文件等等，

docker的镜像实际上由一层一层的文件系统组成，这种层级的文件系统UnionFS。这种分层文件最大的一个好处就是：共享资源。比如：有多个镜像都从相同的 base 镜像构建而来，那么宿主机只需在磁盘上保存一份base镜像，同时内存中也只需加载一份 base 镜像，就可以为所有容器服务了。而且镜像的每一层都可以被共享。

在Docker中镜像是一种完全相同的副本，镜像是只读的。我们能写入的只有容器**。**在使用镜像的时候我们可以理解成就是一个模板，它里面帮我们安装好了一些应用程序并且配置好了程序的运行环境和配置，比如我们获取了一个centos的镜像，我们在获取到镜像后通过docker run -it --name centostest centos 命令就可以生成一个叫centostest的容器并运行这个容器，这时这个叫centostest的容器就相当于一个新的centos系统，在docker中容器，就是一个运行时环境，运行容器，然后在容器中运行程序，至于仓库就是我们存放镜像的地方

在linux上面安装docker 之后可以通过docker version来查看docker是否安装成功以及版本号

## docker 的一些操作

docker images 列出本机的所有镜像 镜像名 版本号 id 创建时间 大小

docker search 镜像名 搜索镜像 收藏数

docker pull mysql:5.6(版本号) 下载一个centos 的一个镜像

docker pull centos 下载镜像不指定版本 （下载最新的）

docker rmi 镜像名(id) 注意正在被容器使用的镜像是无法删除

docker rmi -f 镜像名(id) 可以删除正在被容器使用的镜像

**容器的操作命令（有镜像才能创建容器）**

docker run -it --name 容器名称（唯一的） 镜像（创建容器并进入终端一旦退出容器结束）

docker run -itd --name 容器名称（唯一的） 镜像（创建容器进入后台运行不会进行终端 （exec 进入终端）终端退出 容器继续运行）

容器端口映射：

docker run -itd --name (名称) -p (服务端口):(映射端口) /bin/bash

退出容器 ctrl+p+q 不会停止容器 exit 会停止容器

进入后台运行容器的终端 docker exec -it 容器名称(容器id) /bin/bash

查看运行中的容器 docker ps

查看所有的容器 docker ps -a

删除容器 docker rm 容器id（名称）（运行中的容器不能删）

删除容器 docker rm -f 容器id（名称）（可以删除运行中的容器）

停止容器 docker stop 容器id（名称）

启动容器 docker start 容器id（名称）

**拷贝 宿主机向容器中拷贝文件**

docker cp 宿主机的文件 容器id:目录

**拷贝 容器中的文件 拷贝到宿主机上**

docker cp 容器id:目录 宿主机的目录

## 数据卷怎么创建 (创建一个容器 并和宿主机共享目录)

docker run -itd --name 容器名 -v 宿主目录(/opt/docker/gx):/容器目录(/gx) 镜像 /bin/bash

数据卷容器（多个容器共享一个目录 （修改一个目录 多个容器跟着改变））

先创建一个共享的容器

docker run -v 宿主目录:容器目录 --name 容器名称 镜像 /bin/bash

创建容器指定共享的容器 实现共享目录的

docker run -itd --volumes-from 共享容器名 --name容器名称 镜像 /bin/bash

## Docker 容器配置mysql

docker run -id --name memysqla -p 4000:3306 -e MYSQL\_ROOT\_PASSWORD=root mysql:5.6

memysqla 容器名

4000:3306 将容器的 3306 端口映射到宿主机的 4000端口

root mysql密码

mysql:5.6 mysql版本号

## 制作镜像

创建DockerFile文件

内容如下：

FROM centos

MAINTAINER liuwenhao

RUN mkdir -p /docker/doc

ADD jdk1.8.0\_45 /docker/doc/jdk

ADD springcloud\_product\_eureka\_server-1.0-SNAPSHOT.jar /docker/doc/

ENV JAVA\_HOME /docker/doc/jdk

ENV PATH $PATH:$JAVA\_HOME/bin

EXPOSE 3400

ENTRYPOINT ["java","-Djava.security.egd=file:/dev/./urandom","-jar","/docker/doc/springcloud\_product\_eureka\_server-1.0-SNAPSHOT.jar"]

生成镜像：docker build -f DockerFile文件 -t 生成的镜像名称:版本号 .

启动容器：docker run -itd --name (名称) -p (服务端口):(映射端口) 镜像id /bin/bash

## 私有仓库

1、拉取私有仓库镜像

docker pull registry

2、启动私有仓库容器

docker run -id --name=registry -p 5000:5000 registry

3、打开浏览器 输入地址

http://私有仓库服务器ip:5000/v2/\_catalog，看到{"repositories":[]} 表示私有仓库搭建成功

4、修改daemon.json vim /etc/docker/daemon.json

在上述文件中添加一个key，保存退出。此步用于让 docker 信任私有仓库地址；注意将私有仓库服务器ip修改为自己私有仓库服务器真实ip

{"insecure-registries":["私有仓库服务器ip:5000"]}

5、重启docker 服务

systemctl restart docker

docker start registry

把镜像推到私有仓库

1. 标记镜像为私有仓库的镜像

docker tag 本地镜像 私有仓库服务器IP:5000/放到仓库的镜像名称

1. 上传标记的镜像

docker push 私有仓库服务器IP:5000/放到仓库的镜像名称

从私有仓库拉镜像

docker pull 私有仓库服务器IP:5000/放到仓库的镜像名称

Docker rabbitmq

拉取 管理版本 进行

docker pull rabbitmq:management

运行容器

docker run -dit --name Myrabbitmq -e RABBITMQ\_DEFAULT\_USER=admin -e RABBITMQ\_DEFAULT\_PASS=admin -p 15672:15672 -p 5672:5672 rabbitmq:management