镜像

镜像就是打包好的环境和应用。镜像就是一个模板。

容器

就是运行镜像的实例，镜像是静态的概念，容器是动态的。

仓库

存放多个镜像的一个仓库。仓库可以是远程的也可以是本地。

附加解释：仓库是储存所有软件的原文件。

我们将仓库中的软件为材料，组合成的图纸就是镜像。

运行镜像的载体，就是一个容器。

docker的操作指令

【】用来代指容器名和id

查询：

docker images：查看本地的镜像

docker ps：查看所有运行中的容器

-a：查看所有容器，包括已经关闭的容器

容器：

docker create【镜像名称/id】： 创建一个新的容器，不启动

docker run【镜像名/id】：通过镜像创建一个容器并运行

-d：在后台创建一个容器并运行

docker rm【】：删除容器

docker stop【】：停止容器

docker start【】：开启已经创建的容器

docker restart【】：重启容器

docker logs【】：查看容器的运行的返回结果

docker inspect【】：查看容器的具体信息

容器的交互模式

docker attach【】：进入到容器，进行操作

运行是指定交互模式i----interat交互、t----tty终端、-d在后台进行运行

创建并运行容器，并进入到交互界面，-it和/bin/bash几乎是绑定使用的

docker run -it --name【指定容器名称】【指定镜像名】 /bin/bash

进入到交互式界面：

exit退出交互界面，但是容器也会退出

start重启容器，就可以用attach再次进入到容器，此时进入的是可以交互的容器

对于使用交互界面修改过得容可以使用commit【容器名称】【版本标签名】，把容器打包成新的镜像。

交互模式非启动时：

docker exec -it 【容器名称】 /bin/bash

run的参数：

-d后台运行

-p端口号ip:hostPort:containerPort -v挂载目录

-e创建容器时，规定特定参数

--link与其他容器进行容器名称的网络映射

-v 容器目录挂载

镜像：

docker search：查找命令。

docker pull：下载镜像【镜像名】：【镜像版本】

docker rmi：删除镜像，但是要在rm容器后才能删除镜像

docker push：提交自己的镜像到官方仓库或自己的仓库

docker attach:进入到已经运行的容器当中，退出时会关掉容器。已过时

docker commit：将修改过的镜像提交到容器当中。

docker export：将只读的系统导出到一个tar包中

docker import：将tar包导入到docker容器中

docker save：导出镜像

docker load：导入镜像

DockerFile

docker build –t 镜像名 DockerFile目录

FROM tomcat:latest //基于基准镜像 scarath表示不依赖于任何基准镜像

//说明信息

MAINTRAINER 维护容器的机构或作者

//描述标签

LABEL version = “1.0”

LABEL description = “马士兵”

//设置工作路径

WORKDIR /usr/local/tomcat/webapps

//复制文件

ADD 宿主机路径 容器路径 //支持解压缩功能

COPY

//设置环境常量

ENV JAVA\_HOME /usr/local/openjdk1.8

//运行指令

RUN //构建镜像时执行，镜像被构建后成为只读文件

//容器启动时执行，对镜像没有影响

ENTRYPOINT //入口点，容器启动时执行，多个ENTRYPOINT只有最后一个会执行

CMD //容器启动时，默认执行命令，多个CMD只有最后一个会执行，运行容器时，可以通过添加执行语句覆盖CMD的默认命令

{

指令格式shell指令，exec指令[“yum”,”install”,”-y”,”vim”]推荐使用Exec格式

//ENTRYPOINT和CMD同时存在时，命令会合并执行

}

EXPOSE //默认暴露出的端口号

Docker网络通讯

docker的虚拟ip

每一个docker容器都会创建一个虚拟ip，无法从外部直接访问，他只是docker内部通讯的一个标识

docker两个虚拟ip之间彼此是互通互联的

可以给docker进行命名，docker就可以自动完成网络的转发工作 –link建立容器间的单项通讯

bridge网桥进行容器的双向通信

每单创建一个网桥，docker就会在宿主机上创建一个虚拟网卡，虚拟网卡承担一个网关的作用

docker network create 桥名 //创建一个网桥

docker network connect 桥名 容器名 //给容器添加一个桥连接

Docker数据共享

volume数据卷，将数据放到宿主机上

-v 宿主机路径：容器路径

只创建一个容器，但是不运行他，并设置挂载点

docker create –name 名字1 –v 宿主机挂载点：容器挂载点 镜像名 /bin/true

共享容器挂载点

docker run –volums-from 名字1 –name名字2 -d 镜像名

这样如果需要修改挂载点，只需要修改挂载容器的路径就可以了

DockerCompose容器编排工具