**dock**

环境:

dock1 192.168.1.31

dock2 192.168.1.32

**docker命令:**

docker images

docker htlp pull

下载镜像 doc ker pull docker.io/busybox

上传镜像 docker push docker.io/busybox

导入镜像 docker save docker.io/busybox:latest -o busybox.tar

导出镜像 docker load -i busybox.tar

启动镜像 docker run -it docker.io/centos:latest /bin/bash

查看容器信息 docker ps

查看镜像列表 docker images

查看镜像制作历史 docker history

查看镜像底层信息 docker inspect

删除本地镜像 docker rmi

镜像另存为tar包 docker save

使用tar包导入镜像 docker load

搜索镜像 docker search

修改镜像名称和标签 docker tag

安装dock

配置yum源

在真机 RHEL7-extras.iso

[student@room9pc01 ~]$ cd /linux-soft/04/openstack/

[student@room9pc01 openstack]$ ls

cirros.qcow2 RHEL7-extras.iso RHEL7OSP-10.iso small.qcow2

[student@room9pc01 ~]$mkdir /var/ftp/extras

[student@room9pc01 ~]$ cd /linux-soft/04/openstack/

[student@room9pc01 ~]$mount RHEL7-extras.iso /var/ftp/extras

[root@dock1 ~]# vim /etc/yum.repos.d/local.repo

[local\_repo]

name=CentOS-$releasever - Base

baseurl="ftp://192.168.1.254/centos-1804"

enabled=1

gpgcheck=0

[local]

name=CentOS

baseurl="ftp://192.168.1.254/extras"

enabled=1

gpgcheck=0

[root@dock1 ~]# yum -y install docker

[root@dock1 ~]# systemctl start docker

[root@dock1 ~]# systemctl enable docker

[root@dock1 ~]# ifconfig

....................................

docker0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500

inet 172.17.0.1 netmask 255.255.0.0 broadcast 0.0.0.0

ether 02:42:b7:93:7f:bc txqueuelen 0 (Ethernet)

[root@dock1 ~]# vim /etc/hosts

docker1 192.168.1.31

docker2 192.168.1.32

[root@dock1 ~]# for i in dock2

> do

> scp /etc/hosts $i:/etc/hosts

> scp /etc/yum.repos.d/local.repo $i:/etc/yum.repos.d/local.repo

> done

[root@dock1 ~]# docker images

官方镜像 : <http://hub.docker.com>

查找有关mysql的镜像

[root@dock1 ~]# docker search mysql

INDEX NAME DESCRIPTION STARS OFFICIAL AUTOMATED

docker.io docker.io/mysql MySQL is a widely used, open-source relati... 8512 [OK]

docker.io docker.io/mariadb MariaDB is a community-developed fork of M... 2954 [OK]

docker.io docker.io/mysql/mysql-server Optimized MySQL Server Docker images. Crea... 629 [OK]

OFFICIAL 下面如果是 yes 就是官方提供的

下载镜像busybox

[root@dock1 ~]# docker pull docker.io/busybox

**名称**+**标签**=**唯一的ID**  分开可以重复

[root@dock1 ~]# docker images

REPOSITORY TAG IMAGE ID CREATED SIZE

仓库的名称 标记/标签

192.168.1.31:5000/busyboxlatest db8ee88ad75f 5 weeks ago 1.224 MB

备份镜像

[root@dock1 ~]# docker save docker.io/busybox:latest -o

把busybox.tar镜像传给docker2

[root@docker1 ~]# rsync -av busybox.tar root@192.168.1.32:/root/

在docker2主机上恢复镜像

[root@docker2 ~]# docker load -i busybox.tar

查看docker1和docker2的busybox镜像的ID一样

IMAGE ID

db8ee88ad75f

真机操作:

cd /linux-soft/04/docker/

[student@room9pc01 docker]$ cp nginx.tar redis.tar ubuntu.tar centos.tar /var/ftp/

下载nginx.tar redis.tar ubuntu.tar centos.tar 四个镜像

[root@docker1 ~]# lftp 192.168.1.254

lftp 192.168.1.254:~> ls

lftp 192.168.1.254:/> mget \*.tar

恢复所有镜像

[root@docker1 ~]# for i in \*.tar

> do

> docker load -i $i

> done

启动centos镜像

[root@docker1 ~]# docker run -it docker.io/centos:latest /bin/bash

[root@37f920883d31 /]#

在另一台终端查看启动的镜像

[root@docker1 ~]# docker ps

37f920883d31 docker.io/centos:latest "/bin/bash" 36 seconds ago Up 33 seconds backstabbing\_morse

docker ps 相当于 virsh list

docker run 相当于 virsh define start console

latest 不写 系统默认会补齐

配置yum源

[root@37f920883d31 /]# cd /etc/yum.repos.d/

[root@37f920883d31 yum.repos.d]# rm -rf \*

[root@37f920883d31 yum.repos.d]# vi local.repo

[local\_repo]

name=CentOS-$releasever - Base

baseurl="ftp://192.168.1.254/centos-1804"

enabled=1

gpgcheck=0

[local]

name=CentOS

baseurl="ftp://192.168.1.254/extras"

enabled=1

gpgcheck=0

查找ifconfig命令是那个软件包

[root@37f920883d31 yum.repos.d]# yum provides ifconfig

..............................

**net-tools**-2.0-0.22.20131004git.el7.x86\_64 : Basic networking tools

起四种镜像,

centos不用加解释权(/bin/bash)可以起来

[root@docker1 ~]# docker run -it docker.io/centos

[root@398486f9ccfa /]#

ubuntu 不用加解释权(/bin/bash)可以起来

[root@docker1 ~]# docker run -it docker.io/ubuntu

root@087790b1693e:/#

busybox不用加解释权(/bin/bash)可以起来

[root@docker1 ~]# docker run -it docker.io/busybox

/ #

nginx是服务要加解释器(/bin/bash)

[root@docker1 ~]# docker run -it docker.io/nginx /bin/bash

root@80eccc457aba:/#

redis 是服务要加解释器(/bin/bash)

[root@docker1 ~]# docker run -it docker.io/redis /bin/bash

root@40c27d558aa6:/data#

dpg

apt-get 用法

查看镜像制作历史 docker history

[root@docker1 ~]# docker history docker.io/nginx:latest

查看镜像底层信息 docker inspect

[root@docker1 ~]# docker inspect docker.io/centos:latest

**镜像里有容器删不了镜像**

多个文件可以使用同一个镜像

修改镜像名称和标签 docker tag

[root@docker1 ~]# docker tag docker.io/busybox:latest oooxxx:latest

两个id一样

docker.io/busybox latest db8ee88ad75f 5 weeks ago 1.224 MB

oooxxx latest db8ee88ad75f 5 weeks ago 1.224 MB

**容器常用命令**

运行容器 docker run

查看容器列表 docker ps

关闭容器 docker stop

启动容器 docker start

重启容器 docker restart

进入容器 docker attach | exec

删除容器 docker rm

查看容器底层信息 docker inspect

查看容器进程列表 docker top

运行容器 docker run 相当于创建新的容器,启动容器,连接进去三个命令的组合体

[root@docker1 ~]# docker run **-it** docker.io/centos

启动容器放入后台运行

[root@docker1 ~]# docker run **-itd** docker.io/centos:latest

只显示容器ID

[root@docker1 ~]# docker ps -aq

[root@docker1 ~]# docker run -it docker.io/centos

[root@67ce3687c4a8 /]# sleep 12345

[root@docker1 ~]# docker ps //查看容器ID

[root@docker1 ~]# docker top 67ce3687c4a8 //查看容器中进程信息

删除容器 docker rm 不能删除正在运行的容器

[root@docker1 ~]# docker ps -qa

[root@docker1 ~]# docker rm 2bb2a1eeab4c

删除正在运行的容器

[root@docker1 ~]# docker rm $(docker stop 2bb2a1eeab4c)

**进入容器 docker attach | exec**

[root@docker1 ~]# docker run -itd docker.io/centos:latest //后台起一个容器

[root@docker1 ~]# docker ps //查看容器ID

7c0e7c2139d2

[root@docker1 ~]# docker exec -it 7c0e7c2139d2 /bin/bash //用进程ID进入容器

[root@7c0e7c2139d2 /]# cd /etc/yum.repos.d/

[root@7c0e7c2139d2 yum.repos.d]# rm -rf \*

[root@7c0e7c2139d2 yum.repos.d]# vi local.repo

[local\_repo]

name=CentOS-$releasever - Base

baseurl="ftp://192.168.1.254/centos-1804"

enabled=1

gpgcheck=0

[root@7c0e7c2139d2 ~]# yum -y install vim net-toold iproute lftp psmisc bash-completion

**用run -itd后台起一个容器,再用exec+容器ID进入容器,如果不删,容器一直可以使用**

[root@docker1 ~]# docker run -itd docker.io/centos:latest //后台起一个容器

[root@docker1 ~]# docker ps //查看容器ID

7c0e7c2139d2

[root@docker1 ~]# docker exec -it 7c0e7c2139d2 /bin/bash //用进程ID进入容器

[root@7c0e7c2139d2 ~]# exit

exit

[root@docker1 ~]# docker stop 7c

7c

[root@docker1 ~]# docker start 7c

7c

[root@docker1 ~]# docker exec -it 7c /bin/bash

[root@7c0e7c2139d2 /]#

[root@docker1 ~]# rpm -qa | grep bash //查找tab补全软件包

bash-completion-2.1-6.el7.noarch

[root@7c0e7c2139d2 /]# echo $$ //查看当前进程

19

[root@7c0e7c2139d2 /]# pstree -p 0

?()-+-bash(1) //进程号为1是上帝进程

`-bash(19)---pstree(118)

**进入容器 docker attach attach**

[root@docker1 ~]# docker attach 7c //退出会关闭进程

如果误操作使用attach,使用Ctrl+p,在按Q,相当于Ctl+z

**自定义镜像**

起一个容器,配置yum源,安装常用软件包

[root@docker1 ~]# docker run -it docker.io/centos

[root@9ecf2790f6d6 /]# rm -rf /etc/yum.repos.d/\*

[root@9ecf2790f6d6 /]# vi /etc/yum.repos.d/local.repo

[local\_repo]

name=CentOS-$releasever - Base

baseurl="ftp://192.168.1.254/centos-1804"

enabled=1

gpgcheck=0

[root@9ecf2790f6d6 /]# yum -y install vim net-toold iproute lftp psmisc bash-completion

[root@9ecf2790f6d6 /]# exit

把安装修改过的荣器新打一个包

[root@docker1 ~]# docker ps -a

9ecf2790f6d6

[root@docker1 ~]# docker commit 9ecf2790f6d6 myos

[root@docker1 ~]# docker rm 9ecf2790f6d6

[root@docker1 ~]# docker images

[root@docker1 ~]# docker history myos //查看myos怎么打的包

启动自定义的镜像

[root@docker1 ~]# docker run -it myos:latest

**Dockerfile**

把镜像写成一个脚本

Dockerfile语法格式

**FROM**:基础镜像

**MAINTAINER**:镜像创建者信息

**EXPOSE**:开放端口

**ENV:**设置变量

**ADD**:复制文件到镜像

**RUN:**制作镜像时执行的命令,可以有多个

**WORKDIR**:定义容器默认工作目录 相当于CD命令,

**CMD**:容器启动时执行命令,尽可以有一条CMD

[root@docker1 ~]# vim Dockerfile

FROM docker.io/centos:latest

RUN rm -rf /etc/yum.repos.d/\*

ADD local.repo /etc/yum.repos.d/local.repo

RUN yum -y install vim net-toold iproute lftp psmisc bash-completion

[root@docker1 ~]# docker build -t test:latest **. <---有点**

[root@docker1 ~]# docker images //查看是否有test镜像

[root@docker1 ~]# docker run -it test:latest //查看是否有之前的配置

**ENV设置变量的使用**

[root@docker1 ~]# vim Dockerfile

FROM myos:latest

ENV PS1="[###] "

[root@docker1 ~]# docker build -t myos:test .

[root@docker1 ~]# docker run -it myos:test

[###] //改变了提示符

**156**

mkdir dirtest

echo this is test of dir!!! > dirtest/haha.txt

zip -re dirtest.zip dirtest

 rm -rf dirtest

unzip dirtest.zip