<https://opsx.alibaba.com/mirror>  **#阿里巴巴镜像站**

**修改nginx默认首页**

[root@docker01 ~]# docker run -d nginx **# -d 启动的是服务**

43d2116b92d26b2612f7980fb8ebdd534b33ba195e9f244a7a156acfc441c316

[root@docker01 ~]# docker exec -it 43 /bin/bash **#进入容器**

root@43d2116b92d2:/# nginx -h **#查看帮助**

root@43d2116b92d2:/# nginx -T  **#查看配置文件**

root@43d2116b92d2:/# cd /usr/share/nginx/html

root@43d2116b92d2:/usr/share/nginx/html# ls

50x.html index.html

root@43d2116b92d2:/usr/share/nginx/html# echo hello world > index.html

[root@docker01 ~]# curl <http://172.17.0.2> #退出访问

root@43d2116b92d2:/# cat /etc/debian\_version 查看

8.6

root@43d2116b92d2:/# dpkg -l #查看ngixn装的包

root@43d2116b92d2:/# dpkg -l |grep nginx

root@43d2116b92d2:/# dpkg -S nginx

**1 docker 安装**

#----------------------物理机--------------------------#

创建一台 4G 内存的虚拟机

创建虚拟机硬盘

qemu-img create -b node.qcow2 -f qcow2 node1.img 20G

创建虚拟机配置文件

sed 's/demo/node1/' demo.xml >/etc/libvirt/qemu/node1.xml

定义虚拟机

virsh define /etc/libvirt/qemu/node1.xml

编辑虚拟机修改内存

virsh edit node1

启动虚拟机

virsh start node1

创建自定义 yum 源

拷贝 docker-engine\* /var/ftp/public

cd /var/ftp/public

createrepo .

#--------------------虚拟机--------------------#

修改主机名和静态 ip 地址

echo docker01 >/etc/hostname

配置静态 ip 地址 /etc/sysconfig/network-scripts/ifcfg-eth0

BOOTPROTO="static"

IPADDR="192.168.1.10"

NETMASK="255.255.255.0"

GATEWAY="192.168.1.254"

配置 /etc/yum.repos.d/docker.repo

[local\_docker]

name=CentOS docker

baseurl="ftp://192.168.1.254/public"

enabled=1

gpgcheck=0

安装 docker

yum install docker-engine

设置开机自启动

systemctl enable docker

重启虚拟机

reboot

docker 常用命令

查看版本

docker version

查看镜像

docker images

搜索镜像

docker search 关键字

下载镜像

docker pull 镜像的名字

查看命令帮助

docker help 命令

导出镜像

docker save busybox:latest >busybox.tar

导入镜像

docker load <busybox.tar

启动容器

docker run -it 镜像的名字:标签 镜像里面的命令

查看容器

docker ps -a [查看所有容器]

删除镜像

docker rmi 镜像名称:标签

创建镜像别名

docker tag 源镜像名称:标签 新名称:新标签

容器命令介绍

docker run

-i 交互式

-t 终端

-d 后台进程

启动一个交互式的终端 /bin/bash /usr/bin/python

docker run -it centos /bin/bash

docker run -it centos /usr/bin/python

启动一个服务(非交互) nginx

docker run -d nginx

启动一个交互式的终端，放在后台运行 /bin/bash &

docker run -itd centos

查看容器 docker ps

-a 所有容器

-q 只显示id

停止所有正在运行中的容器

docker stop $(docker ps -q)

删除所有容器

docker rm $(docker ps -aq)

查看容器内进程列表

docker top 容器id

查看容器详细信息，查看容器 ip

docker inspect 容器id

docker inspect -f "{{.NetworkSettings.IPAddress}}" 容器id

进入一个已经运行的容器

docker exec -it 容器id /bin/bash

连接一个已经运行容器的 pid为1 的进程

退出后容器会结束，如果不想结束需要把容器放入后台(ctrl + pq)

docker attach 容器id

#------------------------------------------#

redhat debina

#------------------------------------------#

RHEL,CentOS debina

fedora ubuntu

安装 yum apt-get

管理 rpm dpkg

#------------------------------------------#

编排镜像 （**自定义镜像**）

1、创建容器

docker run -it centos **#启动容器**

[root@eccc1cc70777 yum.repos.d]# rm -f \*  **# 删除系统自动生成的yum源**

[root@eccc1cc70777 yum.repos.d]# vi local.repo  **#创建自定义yum源**

[root@eccc1cc70777 yum.repos.d]# yum install net-tools psmisc lftp iproute vim **#安装常用软件包**

[root@docker01 ~]# docker commit eccc1cc70777 myos #封装一个叫myos的镜像 docker commit 容器ID 镜像名称:标签

[root@docker01 ~]#docker images #查看镜像

[root@docker01 ~]#docker ps -a

[root@docker01 ~]#docker rm c0 #删除刚刚创建的c0容器

[root@docker01 ~]#docker run -it myos #创建容器

1. 修改配置、安装软件（自动化封装，自动创建镜像）

[root@docker01 ~]#mkdir xx #创建文件夹

[root@docker01 ~]#cd xx

[root@docker01 ~]#cp ../oo/local.repo ./ #把yum文件cp到当前目录

[root@docker01 xx]#touch Dockerfile #创建Dockerfile 文件

[root@docker01 xx]#vim Dockerfile #编写文件

[root@docker01 xx]#docker build -t testos . # . 表示当前路径

Dockerfile 01:

FROM centos:latest

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

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

RUN yum install -y net-tools psmisc lftp iproute vim

Dockerfile 02:

FROM myos:latest

CMD ["/usr/bin/pythono"]

**手动创建（手动封装一个apache服务）**

[root@docker01 ~]# docker run -it myos #启动容器

[root@078f1b8f5609 /]# yum -y install httpd #安装httpd

[root@078f1b8f5609 /]# httpd -t #报错

[root@078f1b8f5609 /]# vim /etc/httpd/conf/httpd.conf

ServerName localhost #在ServerName后面添加

[root@078f1b8f5609 /]# httpd -t

Syntax OK

[root@078f1b8f5609 /]# systemctl start httpd #这样启动会报错

[root@078f1b8f5609 /]#cat /usr/lib/systemd/system/httpd.service

EnvironmentFile=/etc/sysconfig/httpd #$OPTIONS在里面为空

ExecStart=/usr/sbin/httpd $OPTIONS -DFOREGROUND

[root@078f1b8f5609 /]# /usr/sbin/httpd -DFOREGROUND #启动httpd

**#按ctrl+p+q退出**

[root@docker01 ~]# docker inspect 07 #查看ip

[root@docker01 ~]# curl -I <http://172.17.0.3> #访问

设置http默认首页

[root@docker01 ~]# docker exec -it 07 /bin/bash

[root@078f1b8f5609 /]# rpm -ql httpd

[root@078f1b8f5609 /]# cd /var/www/html

[root@078f1b8f5609 html]# vim index.html

Hello world

[root@078f1b8f5609 html]# exit

[root@docker01 ~]# curl -i <http://172.17.0.3>

**自动创建（封装一个apache服务）**

[root@docker01 ~]# mkdir bb

[root@docker01 ~]# cd bb

[root@docker01 bb]# vim Dockerfile

FROM myos

RUN yum install -y httpd

ADD httpd.conf /etc/httpd/conf/httpd.conf

WORKDIR /var/www/html

RUN echo "hello nsd1803" >index.html

EXPOSE 80

MAINTAINER lixin

ENV EnvironmentFile=/etc/sysconfig/httpd

CMD ["/usr/sbin/httpd", "-DFOREGROUND"]

[root@docker01 bb]# docker exec -it 07 /bin/bash

[root@078f1b8f5609 /]# scp /etc/httpd/conf/httpd.conf 172.17.0.1:./

[root@docker01 bb]# mv /root/httpd.conf ./

[root@docker01 bb]# ls

Dockerfile httpd.conf

[root@docker01 bb]# docker build -t myos:httpd . #开始创建

[root@docker01 bb]# docker images

[root@docker01 bb]# docker ps -a

[root@docker01 bb]# docker stop 07 #停掉

[root@docker01 bb]# docker rm 07 #删除

[root@docker01 bb]# docker run -d myos:httpd #启动httpd服务

[root@docker01 bb]# docker inspect f2 #查看信息

[root@docker01 bb]# curl -i <http://172.17.0.3> #验证

创建私有仓库

配置文件 /etc/docker/daemon.json

{

"insecure-registries" : ["192.168.1.10:5000"]

}

完成配置以后重启 docker 服务

[root@docker01 ~]# docker ps

[root@docker01 ~]# docker stop f2 #停掉httpd服务

[root@docker01 ~]#systemctl restart docker #重起docker

启动私有仓库服务

[root@docker01 ~]#docker run -d -p 5000:5000 registry

打标记

docker tag busybox:latest 192.168.1.10:5000/busybox:latest

docker tag myos:latest 192.168.1.10:5000/myos:latest

docker tag myos:python 192.168.1.10:5000/myos:python

docker tag myos:httpd 192.168.1.10:5000/myos:httpd

上传镜像

docker push 192.168.1.10:5000/busybox:latest

docker push 192.168.1.10:5000/myos:latest

docker push 192.168.1.10:5000/myos:python

docker push 192.168.1.10:5000/myos:httpd

客户机使用私有镜像源

[root@docker03 ~]# cd /etc/docker/

配置 daemon.json

[root@docker03 docker]# scp 192.168.1.10:/etc/docker/daemon.json ./

重启服务 systemctl restart docker

启动容器

docker run -it 192.168.1.10:5000/busybox

docker run -it 192.168.1.10:5000/myos:latest

docker run -d 192.168.1.10:5000/myos:httpd

查看私有仓库有什么样的镜像

curl http://192.168.1.10:5000/v2/\_catalog

查看私有仓库的镜像有什么样的标签

curl http://192.168.1.10:5000/v2/myos/tags/list

持久化存储

[root@docker02 ~]#cd /var/

[root@docker02 var]# ls

[root@docker02 var]#mkdir webroot

[root@docker02 var]#cd webroot/

[root@docker02 webroot]# docker images

[root@docker02 webroot]# docker run -d -v /var/webroot:/var/www/html 192.168.1.10:5000/myos:httpd #启动服务

[root@docker02 webroot]# cd /var/webroot/

[root@docker02 webroot]# echo 12345 > index.html #修改默认首页

**使用共享存储**

[root@room9pc01 ~]# virsh start --console node6 #启动并连接到虚拟机

[root@docker03 docker]# mkdir /var/webroot

[root@docker03 docker]# yum -y install nfs-utils #安装nfs

[root@docker03 docker]# vim /etc/exports

/var/webroot \*(rw)

[root@docker03 var]# chmod 777 webroot

[root@docker03 var]# exportfs -vr

[root@docker03 var]# systemctl start nfs

[root@docker01 ~]# yum -y install nfs-utils

[root@docker01 ~]# mount -t nfs 192.168.1.16:/var/webroot /mnt

[root@docker01 ~]# cd /mnt/

[root@docker01 mnt]# echo 123456 > index.html

[root@docker01 mnt]# docker run -it -v /mnt:/var/www/html myos

#启动容器

[root@484a99089d58 /]# cd /var/www/html/

[root@484a99089d58 html]# ls

index.html

[root@484a99089d58 html]# cat index.html

存储卷映射

docker run -itd -v 物理机文件夹:容器内文件夹 镜像:标签

**Docker 网络架构**

**Linux网桥**

**创建虚拟网卡**

[root@docker03 ~]# cd /etc/sysconfig/network-scripts/

[root@docker03 network-scripts]# cp ifcfg-eth0 ifcfg-br0

[root@docker03 network-scripts]# vim ifcfg-br0

# Generated by dracut initrd

DEVICE="br0"

ONBOOT="yes"

TYPE="Bridge"

BOOTPROTO="static"

IPADDR="192.168.2.16"

NETMASK="255.255.255.0"

[root@docker03 network-scripts]# systemctl restart network

#如果报ERROR：…… brctl not found 这个错，需要安装bridge-utils驱动

创建虚拟交换机

[root@docker01 ~]#docker network create --driver=bridge --subnet 192.168.100.0/24 docker1

创建一个新的容器，使用新的交换机

[root@docker03 network-scripts]# vim ifcfg-br0

[root@docker01 ~]# docker run -it --network=docker1 myos

[root@docker01 ~]# iptables -t filter -vnL DOCKER-ISOLATION

[root@docker01 ~]# iptables -t filter -F DOCKER-ISOLATION

#清除规则才能ping通192.168.100.3

### ifconfig 能看到不同的ip

创建容器，使用宿主机的端口 -p 宿主机端口:容器端口

[root@docker02 ~]# docker stop $(docker ps -q) #关掉所有容器

[root@docker02~]# docker run -d -p 80:80 -v /var/webroot:/var/www/html 192.168.1.10:5000/myos:httpd #使用容器

[root@docker02 ~]# cd /var/webroot/

[root@docker02 webroot]# ls

index.html

[root@docker02 webroot]# curl http://www.baidu.com -o index.html

#下载百度页面，访问百度

**使用nginx容器**

[root@docker01 ~]# docker run -it nginx /bin/bash #启动容器

root@1f374d3b7183:/# nginx -T #查看root在哪个目录

root@1f374d3b7183:/# exit #ctrl+p+q退出不结束，exit是退出并结束

[root@docker01 ~]# mkdir /var/webroot

[root@docker01 ~]# docker run -d -p 80:80 -v /var/webroot:/usr/share/nginx/html nginx

892663544d21eb20dbfe1b4e8579ba6b500512cec2d6d22d82c97d694777a810

[root@docker01 ~]# cd /var/webroot/

[root@docker01 webroot]# curl http://www.sohu.com -o index.html

#使用访问页面