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

1、创建容器

docker run -it centos

2、修改配置、安装软件

3、创建镜像

docker ps -a

docker commit 容器ID 镜像名称:标签

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"]

Dockerfile 03

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"]

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

systemctl restart docker

启动私有仓库服务

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

配置 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

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

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

创建虚拟交换机

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

docker run -it --network=docker1 myos

创建容器，使用宿主机的端口 -p 宿主机端口:容器端口

docker run -d -p 80:80 -v /var/webroot:/var/www/html 192.168.1.10:5000/myos:httpd