**Morning：**

1. 自定义镜像

**-自定义镜像**

1）另存为新镜像

[root@bc59f63 ~]# vi /etc/yum.repos.d/local.repo

[root@bc59f63 ~]# yum -y net-tools install psmisc vim lftp iproute

[root@docker01 ~]# docker commit bc59 myos:latest

#把原镜像内容与自定义内容合并成新的镜像

1. Dockerfile语法格式

FROM 镜像名:标签 #基础镜像

ADD 本地文件 镜像目录 #复制文件到镜像(只支持本地当前目录)

RUN 命令 #制作镜像时执行的命令(可以有多个)

MAINTAINER #镜像创建者信息

EXPOSE 端口号 #开放的端口

ENV 环境变量 #设置变量(通过/usr/lib/xx.service查看)

WORKDIR #定义容器默认工作目录(相当于cd)

CMD [“xx”,“-x”] #容器启动时默认执行命令(不写继承父模板)

1. Dockerfile工作流程

[root@docker01 ~]# mkdir build ; cd build

[root@docker01 build]# cat Dockerfile

FROM centos:latest

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

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

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

[root@docker01 ~]# docker build -t myos:v01 /root/build

**-案例：Dockerfile脚本**

脚本一：自定义apache镜像，启动容器默认开启httpd服务

[root@docker01 build]# cat Dockerfile

FROM myos:v01

RUN yum -y install httpd

MAINTAINER gjq jerry@qq.com

EXPOSE 80

EXPOSE 443

ENV EnvironmentFile=/etc/sysconfig/httpd

WORKDIR /var/www/html

RUN echo hello hema! > index.html

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

[root@docker01 ~]# docker build -t myos:http /root/build

[root@docker01 ~]# docker run -itd myos:http

[root@docker01 ~]# curl 172.17.0.x

hello hema!

脚本二：自定义sshd镜像，启动容器后默认开启sshd服务

[root@docker01 build]# cat Dockerfile

FROM myos:v01

RUN yum -y install openssh-server initscripts

RUN sshd-keygen

RUN echo 123 | passwd --stdin root

MAINTAINER gjq jerry@qq.com

EXPOSE 22

ENV EnvironmentFile=/etc/sysconfig/sshd

CMD ["/usr/sbin/sshd","-D"]

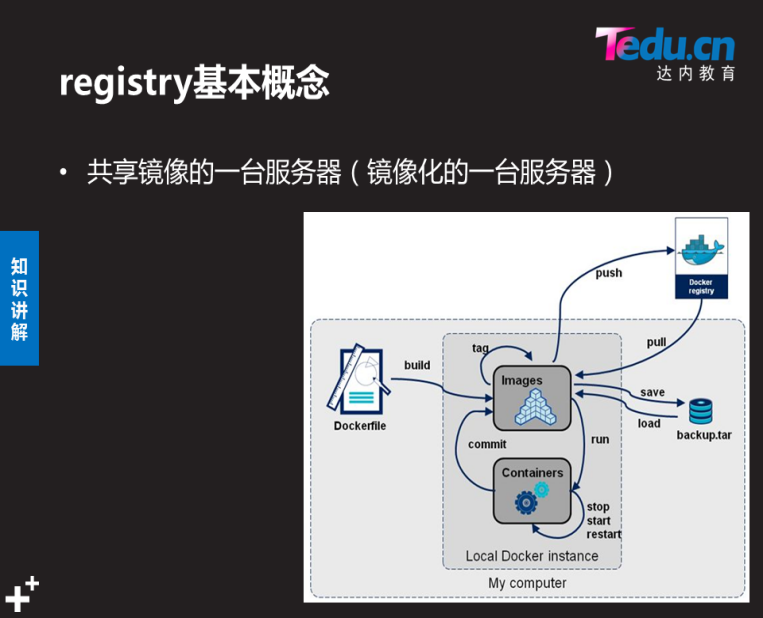
[root@docker01 ~]# docker build -t myos:sshd /root/build

[root@docker01 ~]# docker run -itd myos:sshd

[root@docker01 ~]# ssh 172.17.0.x

2、自定义私有仓库

**-registry基本概念**



**-创建仓库、上传镜像**

json： javascript对象表示法，基于文本，独立于语言，轻量级数据交换格式。

数组[“x”,”y”]、键值对key:value、支持嵌套

1. 定义一个私有仓库

[root@docker01 ~]# vim /etc/docker/daemon.json

{

"insecure-registries" : ["192.168.1.10:5000"] #宿主机IP

}

[root@docker01 ~]# systemctl restart docker

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

#创建仓库

[root@docker02 ~]# curl 192.168.1.10:5000/v2/

{} #出现括号

[root@docker1 ~]# docker tag myos:v01 \

> 192.168.1.10:5000/myos:v01

[root@docker1 ]# docker push 192.168.1.10:5000/myos:v01

#打标签、上传

1. 在docker2上面启动

[root@docker2 ~]# scp 192.168.1.10:/etc/docker/daemon.json \

> /etc/docker/

[root@docker2 ~]# systemctl restart docker

[root@docker2 ~]# docker run -it 192.168.1.10:5000/myos:v01 \

> /bin/bash

**-查看私有仓库**

1. 查看仓库的镜像名

[root@docker1 ~]# curl http://192.168.1.10:5000/v2/\_catalog

{"repositories":["busybox","myos"]}

1. 查看镜像里面的标签

[root@docker1 ~]# curl http://192.168.1.10:5000/v2/myos/tags/list

{"name":"myos","tags":["http"]}

**Afternoon：**

1、持久化存储

**-主机卷的映射**

1. 卷的概念

-容器不保持任何数据

-可以挂载真实机目录或共享存储为卷，实现数据持久化

1. 主机卷的映射

[root@docker01 ~]# docker run -itd -v /opt:/tmp myos:http

#将真机的/opt映射到容器的/tmp

**-案例：使用共享存储映射到容器**

1）真机作为NFS服务器

[root@room12pc2 ~]# mkdir /docker

[root@room12pc2 ~]# cat /etc/exports

/docker/ 192.168.1.0/24(rw,no\_root\_squash)

[root@room12pc2 ~]# systemctl restart nfs

1. 虚拟机docker01~02挂载

[root@docker0x ~]# yum -y install nfs-utils

[root@docker0x ~]# mkdir /docker

[root@docker0x ~]# mount -t nfs -o rw \

> 192.168.1.254:/docker /docker

1. 映射主机目录到容器

[root@docker01 ~]# docker run -itd -v /docker:/var/www/html \

> myos:http

[root@docker02 ~]# docker run -itd -v /docker:/var/www/html \

> 192.168.1.10:5000/myos:http

[root@docker01 ~]# echo “hello hema!” > /docker/index.html

[root@docker01 ~]# curl 172.17.0.x

#实现网页文件的共享

1. Docker网络架构

**-Linux虚拟网卡/网桥**

1）虚拟网卡

[root@localhost network-scripts]# cat /etc/sysconfig//ifcfg-eth0:0

DEVICE="eth0:0"

NAME=”eth0:0”

ONBOOT=yes

BOOTPROTO=static

TYPE=Ethernet

IPADDR=192.168.1.10

NETMASK=255.255.255.0

[root@localhost ~]# systemctl restart network

[root@localhost ~]# ifconfig

1）虚拟网桥

[root@localhost network-scripts]# cat /etc/sysconfig//ifcfg-br0

DEVICE="br0"

NAME=”br0”

ONBOOT=yes

BOOTPROTO=static

TYPE=Bridge

IPADDR=192.168.1.10

NETMASK=255.255.255.0

[root@localhost ~]# systemctl restart network

[root@localhost ~]# brctl show

**-Docker自定义网桥**

1）查看Docker默认创建的网络模型

[root@docker01 ~]# docker network ls

NETWORK ID NAME DRIVER SCOPE

2ae0a5aac05f bridge bridge local #桥接模型

ddf52a098ca55 host host local #主机模型

27ce48a1635b none null local #无网络

2）使用Docker创建虚拟网桥

[root@docker01 ~]# docker network create --driver bridge \

> --subnet=10.10.10.0/24 docker1

#不写--driver默认为bridge，不写--subnet默认顺序分配网段

[root@docker01 ~]# brctl show

1. 使用自定义网桥启动容器

[root@docker01 ~]# docker run --network=docker1 -it myos:v01

**-Docker端口映射**

[root@docker01 ~]# docker run -id -p 80:80 \

> -v /docker:/var/www/html myos:http

[root@docker01 ~]# curl 192.168.1.10

hello world!

【拓展】k8s集群管理工具