



CKA-docker基础



1.白话介绍容器docker

- 容器的介绍
- 了解镜像
- 配置docker代理
- 配置docker加速器
- 使用国内仓库
 - 网易云
 - 阿里云



2.docker镜像管理

- 镜像的命名方式
- docker pull 镜像
- docker tag 镜像
- docker rmi 镜像
- docker save 镜像名 > filename.tar
- docker load -i filename.tar
- 把容器导出为镜像 docker export 容器名 > filename.tar
- 导入 cat filename.tar | docker import - 镜像名



3.docker管理容器

- docker run 镜像 ---最简单的一个容器
- docker run -it **--rm** hub.c.163.com/library/centos/bin/bash
- docker run -dit -h node --name=c1 镜像名 命令
- docker run -dit --restart=always 镜像名 命令
- docker run -dit --restart=always -e 变量1=值1 -e 变量2=值2 镜像



练习:

- 安装nginx
- 安装mysql
- 安装tomcat
- 安装redis
- 安装apache



4.管理容器的常见命令

- docker exec xxxx 命令
- docker start xxxx
- docker stop xxxxx
- docker restart xxxxx
- docker top xxxxx
- docker logs -f node
- docker inspect 容器

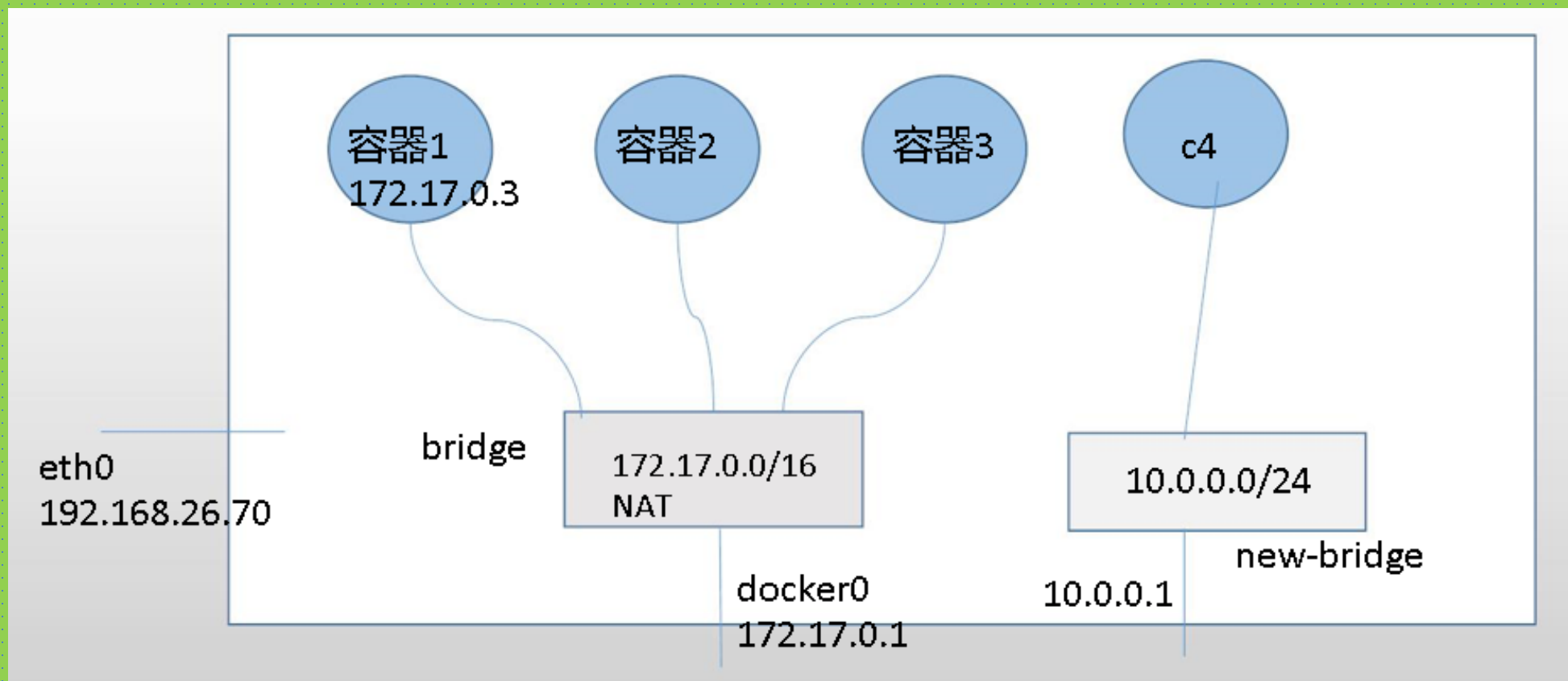


5.数据卷的使用

- `docker run -dit --restart=always -v p_path1:c_path2 镜像名 命令`
- `docker run -dit --restart=always -v c_path2 镜像名 命令`

6.docker网络管理

- docker network list
- docker network inspect 6f70229c85f0





docker网络管理

- `docker network list`
- `man -k docker`
- `man docker-network-create`
- `docker network create -d bridge --subnet=10.0.0.0/24 mynet`
- `docker run --net=mynet --rm -it centos /bin/bash`
- `docker run -dit -p 物理机端口:容器端口 镜像`



容器互联

- 先创建一个容器
- `docker run -it --rm --name=h1 hub.c.163.com/library/centos /bin/bash`
- 再创建一个容器h2，和h1通信有两种方式
- `docker inspect h1 | grep -i ipaddr`
- `docker run -it --rm --name=h2 hub.c.163.com/library/centos ping 172.17.0.2`
- `docker run -it --rm --name=h2 --link h1:h1 hub.c.163.com/library/centos ping h1`



lab-搭建一个wordpress博客

- `docker run -dit --name db --restart=always -e MYSQL_ROOT_PASSWORD=redhat -e MYSQL_DATABASE=wordpress mysql`
- `docker run -dit --name blog --restart=always -e WORDPRESS_DB_HOST=172.17.0.2 -e WORDPRESS_DB_USER=root -e WORDPRESS_DB_PASSWORD=redhat -e WORDPRESS_DB_NAME=wordpress -p 80:80 wordpress`
- 或者
- `docker run -dit --name blog --link=db:mysql -p 80:80 wordpress`



7.自定义镜像

```
FROM hub.c.163.com/library/centos
MAINTAINER duan
RUN rpm --import /etc/pki/rpm-gpg/RPM-GPG-KEY-
CentOS-*
ADD epel.repo /etc/yum.repos.d/
RUN yum install -y nginx

EXPOSE 80 端口2 端口3

CMD ["nginx", "-g","daemon off;"]
```

`docker build -t v4 . -f filename`

`docker build -t name .`

COPY和ADD的意思是一样

ADD带有自动解压功能
COPY没有自动解压功能



自定义镜像-常用到的一些命令

ENV指定变量

```
[root@docker1 xx]# cat Dockerfile
#test
FROM centos
MAINTAINER hello1
ENV MYDIR /mydir
RUN mkdir $MYDIR
RUN echo hello world > $MYDIR/test.txt
[root@docker1 xx]#

[root@bc92722e52be /]# echo $MYDIR
/mydir
[root@bc92722e52be /]#
```

在命令里指定变量

```
[root@docker1 ~]# docker run --rm -it -e "XX=123" local/c3 bash
WARNING: IPv4 forwarding is disabled. Networking will not work.
[root@eafcd01ffac /]# echo $MYDIR
/mydir
[root@eafcd01ffac /]# echo $XX
123
```

USER指定用户

```
#TEST
FROM centos
MAINTAINER hello1
ENV MYDIR /mydir
RUN mkdir $MYDIR
RUN echo hello world > $MYDIR/test.txt
RUN groupadd group1 && useradd -g group1 user1
USER user1

#docker run --rm -it -u user1 local/c4 bash
```



练习:配置可以ssh的centos镜像

```
FROM centos:v1
MAINTAINER duan
RUN rm -rf /etc/yum.repos.d/*
ADD epel.repo /etc/yum.repos.d/
ADD CentOS-Base.repo /etc/yum.repos.d/
RUN yum install openssh-clients openssh-server -y
RUN ssh-keygen -t rsa -f /etc/ssh/ssh_host_rsa_key
RUN ssh-keygen -t ecdsa -f /etc/ssh/ssh_host_ecdsa_key
RUN ssh-keygen -t ed25519 -f /etc/ssh/ssh_host_ed25519_key
RUN sed -i '/UseDNS/cUseDNS no' /etc/ssh/sshd_config

RUN echo "root:redhat" | chpasswd
EXPOSE 22
CMD ["/usr/sbin/sshd", "-D"]
```



8.配置docker本地仓库

- `docker pull registry`
- `docker run -d --name registry -p 5000:5000 --restart=always -v /myreg:/var/lib/registry registry`

- 修改默认地址

`ADD_REGISTRY='--add-registry 192.168.26.70:5000'`



docker使用本地仓库

```
cat /etc/docker/daemon.json
```

```
{  
  "registry-mirrors": ["https://i1pfdcu7.mirror.aliyuncs.com"],  
  "insecure-registries": ["192.168.26.70:5000"]  
}
```

```
curl http://192.168.26.70:5000/v2/_catalog
```

```
docker push 192.168.26.70:5000/rhce/busybox
```

```
docker pull 192.168.26.70:5000/rhce/busybox 或者
```

```
curl -XGET http://192.168.26.70:5000/v2/_catalog
```

```
curl -XGET http://192.168.26.70:5000/v2/rhce/nginx/tags/list
```




删除本地仓库里的镜像

- `curl https://raw.githubusercontent.com/burnettk/delete-docker-registry-image/master/delete_docker_registry_image.py | sudo tee /usr/local/bin/delete_docker_registry_image >/dev/null`
- `chmod a+x /usr/local/bin/delete_docker_registry_image`
- 查看到本地挂载点为/myreg
- `export REGISTRY_DATA_DIR=/myreg/docker/registry/v2`
- `/usr/local/bin/delete_docker_registry_image -i rhce/nginx:latest`



9.限制容器资源

- 了解Cgroup的使用

对内存的限制

`/etc/systemd/system/memload.service.d`

```
cat oo-aa.conf
```

```
[Service]
```

```
MemoryLimit=512M
```

对CPU亲和性限制

```
ps mo pid,comm,psr $(pgrep httpd)
```

`/etc/systemd/system/httpd.service.d`

```
cat oo-aa.conf
```

```
[Service]
```

```
CPUAffinity=0
```



对容器内存的限制

```
[root@docker2 ~]# docker run -dit --name t1 -m 200M --restart=always -v /data:/data centos /bin/bash
8567a8bef8b5ed2e0197209e5aa1552d4bef693faa4a6a09b9eb88abca67883c
```

```
[root@8567a8bef8b5 data]# bigmem 500M
Attempting to allocate 500 Mebibytes of resident memory...
Killed
```

#docker stats

CONTAINER	CPU %	MEM USAGE / LIMIT	MEM %
PIDS			
t1	0.00%	3.465 MiB / 200 MiB	1.73%
MB 0			



对容器CPU的限制

```
[root@docker2 ~]# docker run -dit --name t1 --cpuset-cpus=1 --restart=always -v /data:/data centos /bin/bash  
71fb1ab099385240e6927eb1d22d572bc967b49ad97c410c1f6b69048d2c8708
```

```
[root@docker2 data]# ps mo pid,comm,psr 778
```

PID	COMMAND	PSR
778	burnBX	-
-	-	0



10. 监控容器

- `docker pull hub.c.163.com/xbingo/cadvisor:latest`

```
docker run \  
> -v /var/run:/var/run \  
> -v /sys:/sys:ro \  
> -v /var/lib/docker:/var/lib/docker:ro \  
> -d -p 8080:8080 --name=mon \  
> hub.c.163.com/xbingo/cadvisor:latest
```



cAdvisor - h1 (/sys) x

← → ↻ ⓘ 192.168.26.62:8080/docker/3aacc0ffd7a4a736fe1a2089ed3fd709bb28c06824aa5d35d8b4129a7bd9814e

Usage

Overview

CPU

0 100

Memory

0 100

Processes

No processes found

CPU

Total Usage



11.compose

- ~~yum install python2-pip -y~~
- ~~pip install docker-compose~~
- ~~pip uninstall docker-compose~~
- ~~docker-compose -v~~
- yum install docker-compose -y



cat docker-compose.yml

blog:

image: hub.c.163.com/public/wordpress:4.5.2

restart: always

links:

- db:mysql

ports:

- "80:80"

db:

image: hub.c.163.com/library/mysql

restart: always

environment:

- MYSQL_ROOT_PASSWORD=redhat

- MYSQL_DATABASE=wordpress

关于docker-compose的一些命令

docker-compose up -d

docker-compose ps

docker-compose start/stop/restart

docker-compose rm

db:

image: hub.c.163.com/library/mysql

restart: always

environment:

- MYSQL_ROOT_PASSWORD=redhat

- MYSQL_DATABASE=wordpress

volumes:

- /xx:/var/lib/mysql



12.harbor的使用

安装并启动docker

上传harbor的离线包

导入harbor的镜像

编辑harbor.cfg

修改hostname 为自己的主机名

harbor_admin_password 登录密码

安装compose

运行脚本 ./install.sh

在浏览器里输入IP

docker login IP --家目录下会有一个.docker文件夹