发布镜像

简介

思想来自于集装箱。打包项目带上环境（镜像） 隔离，docker核心思想，打包装箱。每个箱子相互隔开。通过隔离机制，将服务器最好利用。 2013年开源。2014.4.9docker1.0发布。容器技术出现之前，都用虚拟机技术。Docker比起虚拟机，小巧。 Docker是基于go语言开发，开源项目。官网：https://www.docker.com/ 官方文档地址：https://docs.docker.com/ 仓库镜像地址：https://hub.docker.com/ 虚拟机技术缺点：1、资源占用多。 2、冗余步骤多， 3、启动慢。 容器化技术：不是模拟一个完整的操作系统。容器内应用直接运行在宿主机的内核，容器没有自己的内核。没有虚拟硬件，所以轻便。每个容器相互隔离，互不影响。

安装

查看Linux内核及系统版本

- uname -a 查看系统内核
- lsb_release -a 或 cat /etc/os-release 查看版本

安装docker(centos7)

参考官方文档 https://docs.docker.com/engine/install/

一、卸载旧版本docker

```
sudo yum remove docker \
                  docker-client \
                  docker-client-latest \
                  docker-common \
                  docker-latest \
                  docker-latest-logrotate \
                  docker-logrotate \
                  docker-engine
```

二、使用储存库安装，设置储存库

```
sudo yum install -y yum-utils  #安装需要的系统工具

sudo yum-config-manager \
    --add-repo \
    http://mirrors.aliyun.com/docker-ce/linux/centos/docker-ce.repo    #国内阿里云镜像源

    https://download.docker.com/linux/centos/docker-ce.repo  #设置镜像仓库，默认从国外的
```

三、安装docker

```
sudo yum install docker-ce docker-ce-cli containerd.io docker-compose-plugin  #安装最新版
# docker-ce  社区版 ee企业版   docker-ce-cli 命令行界面  containerd.io 容器  docker-compose-plugin 一个docker工具

# 如果提示接受 GPG 密钥，请验证指纹是否匹配 060A 61C5 1B55 8A7F 742B 77AA C52F EB6B 621E 9F35，如果是，则接受它。

#此命令会安装 Docker，但不会启动 Docker。它还会创建一个 docker组，但是默认情况下它不会将任何用户添加到该组中。
```

如果要安装特定版本docker，则需先列出来

```
yum list docker-ce --showduplicates | sort -r

sudo yum install docker-ce-<VERSION_STRING> docker-ce-cli-<VERSION_STRING> containerd.io docker-compose-plugin
```

四、启动docker

```
sudo systemctl start docker

docker version  #查看docker版本
```

五、测试

```
 sudo docker run hello-world
```

```
[root@e7f6c9c2d872 local]# docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
2db29710123e: Pull complete
Digest: sha256:7d246653d0511db2a6b2e0436cfd0e52ac8c066000264b3ce63331ac66dca625
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/get-started/

[root@e7f6c9c2d872 local]#
```

安装成功

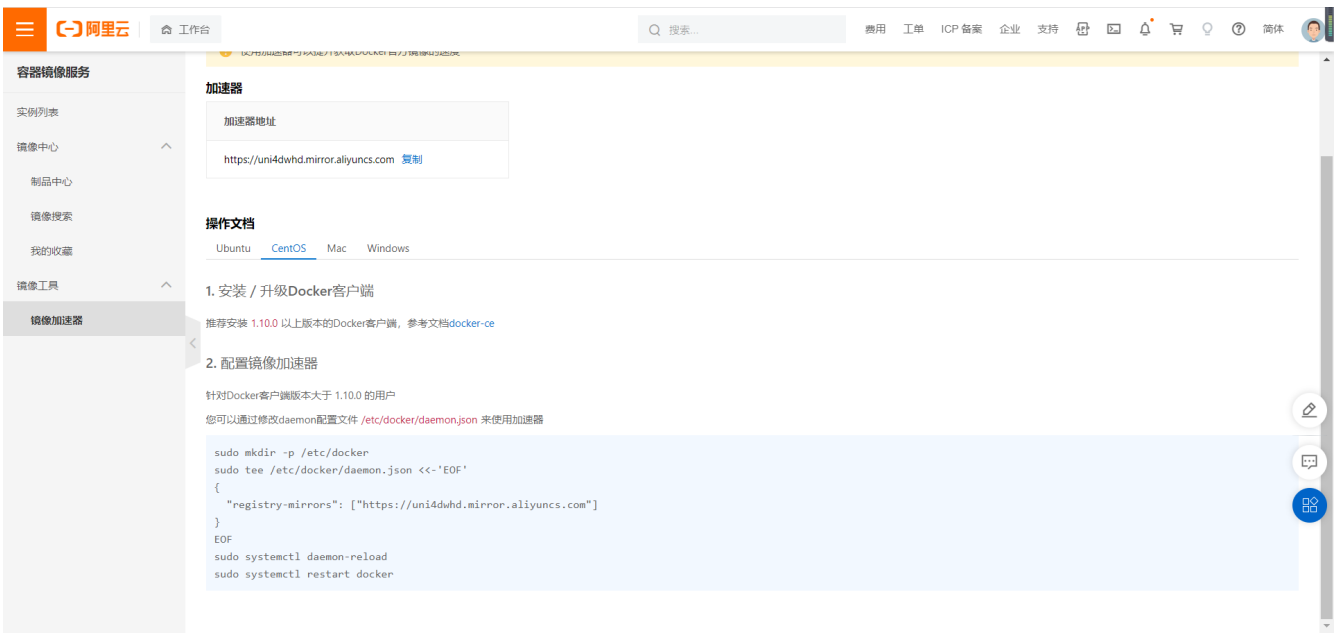
六、查看检查镜像确认

```
docker images
#结果如下，就证明ok

[root@e7f6c9c2d872 local]# docker images
REPOSITORY    TAG       IMAGE ID       CREATED          SIZE
hello-world   latest    feb5d9fea6a5   11 months ago   13.3kB
[root@e7f6c9c2d872 local]#
```

卸载docker

```
#1、卸载依赖
sudo yum remove docker-ce docker-ce-cli containerd.io docker-compose-plugin

#2、删除资源
sudo rm -rf /var/lib/docker     # /var/lib/docker docker的默认工作路径
sudo rm -rf /var/lib/containerd
```

配置阿里云镜像加速



配置使用

```
sudo mkdir -p /etc/docker
sudo tee /etc/docker/daemon.json <<-'EOF'
{
  "registry-mirrors": ["https://uni4dwhd.mirror.aliyuncs.com"]
}
EOF
sudo systemctl daemon-reload
sudo systemctl restart docker
```

修改默认储存数据路径

停掉Docker服务

```
systemctl stop docker
```

查看docker数据存储目录

```
docker info | grep "Docker Root Dir"
```

根据上面查到的路径，移动整个`/var/lib/docker`目录到数据盘的目的路径(没有rsync命令时需安装rsync)：

```
rsync -avzP /var/lib/docker  /data/docker/
```

修改配置默认路径

在/etc/docker/下更改docker配置文件daemon.json，若没有此文件则创建

```
vim daemon.json

{
  "data-root":"/data/docker_92"
}
```

重启docker

```
systemctl daemon-reload
systemctl start docker
```

查看docker数据存储目录

```
docker info | grep "Docker Root Dir"
```

确认之前的镜像是否还在

```
docker images
```

通过上述方法完成迁移之后，在确认 Docker 能正常工作之后，删除原目录数据：

```
rm -rf /var/lib/docker
```

若需要更改docker存储驱动则在配置文件中添加所要更改驱动 vim /etc/docker/daemon.json

```
{
  "storage-driver": "vfs"
}
# vfs为要更改的存储驱动
```

重启docker，查看docker存储驱动是否更改

```
systemctl daemon-reload
systemctl start docker

docker info | grep "Storage Driver"
```

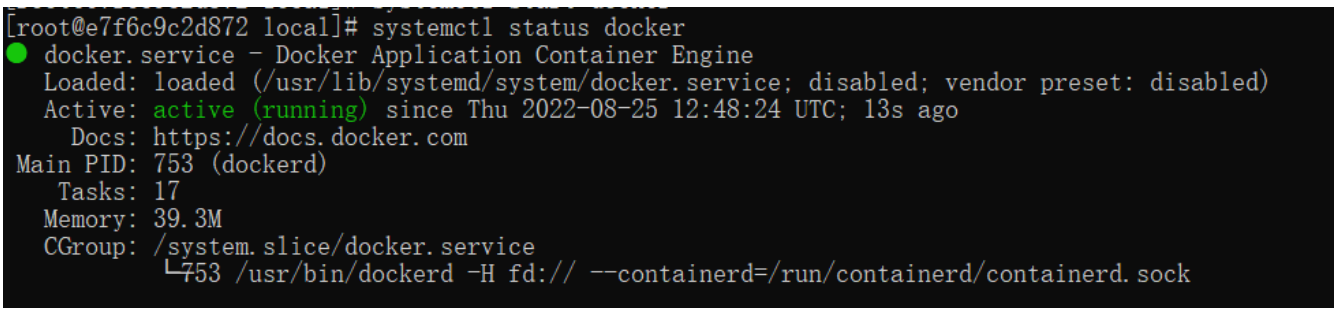
docker常用命令

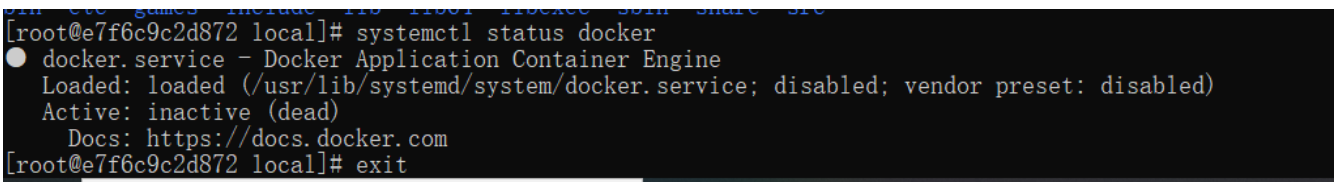
官方参考文档：https://docs.docker.com/engine/reference/commandline/docker/

一、基础命令

启动docker

```
service docker start  #Ubuntu中docker启动命令
systemctl start docker  # centos中docker启动命令,需要root权限
systemctl status docker   #查看docker服务状态，如果是在运行中 输入命令后 会看到绿色的active
```





查看docker 版本号信息

```
docker version #版本查询
docker info #查看详细信息,包括镜像和容器数量
```

docker 帮助命令

忘记命令可使用此进行查看与回顾

```
docker --help
```

或者想知道命令详细参数

```
docker [OPTIONS] --help

例: docker ps --help

root@DESKTOP-KQD226K:~# docker ps --help
Usage:  docker ps [OPTIONS]
List containers
Options:
  -a, --all             Show all containers (default shows just running)
  -f, --filter filter   Filter output based on conditions provided
      --format string   Pretty-print containers using a Go template
  -n, --last int        Show n last created containers (includes all states) (default -1)
  -l, --latest          Show the latest created container (includes all states)
      --no-trunc        Don't truncate output
  -q, --quiet           Only display container IDs
  -s, --size            Display total file sizes
```

(二) docker 镜像命令

查看镜像

```
docker  images  #查看所有本地主机上的镜像

root@DESKTOP-KQD226K:~# docker images
REPOSITORY              TAG          IMAGE ID       CREATED         SIZE
ash123fhd/mycentos      1.0          70da306aaaed   3 months ago    592MB
#解释
REPOSITORY  镜像的仓库源
TAG         镜像的标签
IMAGE ID    镜像的ID
CREATED     镜像的创建时间
SIZE        镜像的大小
# 参数
-a,--all    # 列出所有镜像
-q,--quiet  # 只显示镜像ID
```

搜索镜像

```
docker search 镜像名

[root@e7f6c9c2d872 ~]# docker search centos:7.5
NAME                                DESCRIPTION                                     STARS
    OFFICIAL   AUTOMATED
insgeek/centos                      centos:7.5.1804                                 1
jellywen/centos75-php7124-nginx114  centos:7.5.1804 php:7.1.24 nginx:1.14.1 swoo…   0
portmone/centos_php7                Centos:7.5.1804, php-7.2.14, oracle-instantc…   0

#可选项，通过搜索来过滤
docker search --filter=STARS=9000 mysql   #搜索 STARS>9000的 mysql 镜像

[root@e7f6c9c2d872 ~]# docker search --filter=STARS=9000 mysql
NAME      DESCRIPTION                                     STARS     OFFICIAL   AUTOMATED
mysql     MySQL is a widely used, open-source relation…   13108     [OK]
```

下载镜像

```
docker pull 镜像名:tag
#不加tag(版本号) 即拉取docker仓库中 该镜像的最新版本latest 加:tag 则是拉取指定版本

[root@e7f6c9c2d872 ~]# docker pull centos
Using default tag: latest
latest: Pulling from library/centos
a1d0c7532777: Pull complete
Digest: sha256:a27fd8080b517143cbbbab9dfb7c8571c40d67d534bbdee55bd6c473f432b177
Status: Downloaded newer image for centos:latest
docker.io/library/centos:latest
```

卸载镜像

镜像没有被任何容器使用才可以删除

```
docker rmi  镜像名/镜像ID
docker rmi -f 镜像名/镜像TD    #强制删除
docker rmi -f 镜像名/镜像ID 镜像名/镜像ID 镜像名/镜像ID #删除多个镜像，镜像ID或镜像名用空格隔开即可
docker rmi -f $(docker images -aq)   #删除全部镜像
```

保存镜像

将镜像保存为tar 压缩文件，可以方便镜像转移和保存，然后可以在任何一台安装了docker的服务器上加载这个镜像

```
docker save 镜像名/镜像ID -o 保存地址和文件名  #保存为tar文件

[root@e7f6c9c2d872 lian]# docker save centos -o ./centos.tar
[root@e7f6c9c2d872 lian]# ls
centos.tar
# 任何装docker的地方加载镜像保存文件,使其恢复为一个镜像

docker load -i 镜像保存文件名  #恢复

[root@e7f6c9c2d872 lian]# docker load -i ./centos.tar
74ddd0ec08fa: Loading layer [==================================================>]
238.6MB/238.6MB
Loaded image: centos:latest
```

更改镜像标签

```
docker tag 源镜像[:TAG] 想生成的镜像名[:TAG]
#如果省略TAG 则会为镜像默认打上latest TAG

[root@e7f6c9c2d872 lian]# docker images
REPOSITORY    TAG       IMAGE ID       CREATED         SIZE
hello-world   latest    feb5d9fea6a5   11 months ago   13.3kB
centos        latest    5d0da3dc9764   11 months ago   231MB
[root@e7f6c9c2d872 lian]# docker tag centos mycentos:1.0
[root@e7f6c9c2d872 lian]# docker images
REPOSITORY    TAG       IMAGE ID       CREATED         SIZE
hello-world   latest    feb5d9fea6a5   11 months ago   13.3kB
centos        latest    5d0da3dc9764   11 months ago   231MB
mycentos      1.0       5d0da3dc9764   11 months ago   231MB
```

制作/打包镜像

可以保持当前容器状态，类似VM的快照

```
docker commit -m="提交信息" -a="作者信息" 容器名/容器ID 提交后的镜像名:Tag

root@DESKTOP-KQD226K:~/file# docker images
REPOSITORY              TAG          IMAGE ID       CREATED         SIZE
mycentos                0.1          70da306aaaed   3 months ago    592MB
root@DESKTOP-KQD226K:~/file# docker commit mycentos1 mycentos:0.2
sha256:f9dc8906bc9c103099a38259fb78e285298fdf0758ab16112e509d780630d7cf
root@DESKTOP-KQD226K:~/file# docker images
REPOSITORY              TAG          IMAGE ID       CREATED          SIZE
mycentos                0.2          f9dc8906bc9c   31 seconds ago   3.94GB
mycentos                0.1          70da306aaaed   3 months ago     592MB
```

发布镜像

发布到dockerhub

1.地址https://hub.docker.com/ 注册账号

2.在服务器提交

先登录

```
root@DESKTOP-KQD226K:~# docker login -u ash123fhd
Password:
```

然后push

```
root@DESKTOP-KQD226K:~/dockerfile# docker images
REPOSITORY              TAG       IMAGE ID       CREATED         SIZE
ash123fhd/mycentos      1.0       70da306aaaed   6 minutes ago   592MB
mycentos                0.1       70da306aaaed   6 minutes ago   592MB

root@DESKTOP-KQD226K:~/dockerfile# docker push ash123fhd/mycentos:1.0
The push refers to repository [docker.io/ash123fhd/mycentos]
0dde31b64861: Pushed
deb2853b4a57: Pushed
174f56854903: Pushed
1.0: digest: sha256:ce5d0d289b152ac7950ef38a6b5d032d80df17d408409abc526b58a8ab39cc53 size:
953
```

（三）docker 容器命令

查看容器

```
docker ps       #列出当前正在运行的容器
docker ps -a    #列出所有容器（正在运行的+停止运行的）
docker ps -q    #显示正在运行容器的编号

root@DESKTOP-KQD226K:~# docker ps
CONTAINER ID   IMAGE                 COMMAND                  CREATED        STATUS
PORTS                                      NAMES
e7f6c9c2d872   mycentos:0.1          "init"                   6 days ago     Up 7 hours
80/tcp                                     mycentos1
```

查看容器挂载目录信息

```
docker inspect "Fhd_denovo_cs" | grep Source -A 1
```

创建容器

有镜像才能创建容器

```
docker run 参数 镜像名          # run的时候如果没有镜像会自动下载最新版镜像
#参数
--name="Name"    #容器名字
-d               #后台方式运行
-it              #使用交互方式运行，进入容器内查看内容（-i 创建并运行 -t 容器启动后进入命令行，通常-it一起使用）
-v 宿主机文件存储位置:容器内文件位置                    #将容器内目录挂载在宿主机
-v 宿主机文件存储位置:容器内文件位置 -v 宿主机文件存储位置:容器内文件位置 -v 宿主机文件存储位置:容器内文件位置                                    #一个容器可以挂载多个目录

-p（小写）        # 端口映射，指定容器端口
    -p  ip:主机端口：容器端口
    -p  主机端口：容器端口（常用）
    -p  容器端口
-P（大写）        # 随机指定端口
--restart=always     # 表示，该容器随docker服务启动而自动启动
--privileged=true/false      #是否给容器最高权限，意味着容器内应用将不受权限限制，一般默认false。
centos中systemctl命令需要ture
容器内使用systemctl     报错：Failed to get D-Bus connection: Operation not permitted
但是要注意，这是在创建容器时才有的选项，且要-d的方式启动，启动初始命令为init，容器启动成功后再-it方式进入：
docker run -idt --privileged=true --name mycentos1 mycentos:0.1 init
docker exec -it e7f6c9c2d872 bash
```

进入容器

进入正在运行的容器

```
docker exec -it 容器名/容器ID /bin/bash    #进入正在运行的容器
docker attach 容器名/容器ID

exec  进入容器后开启一个新的终端，可在里面操作
attach 进入容器正在执行的终端，不会启动新进程
```

退出容器

```
exit         #直接退出，容器运行未添加-d时，执行此参数 容器会被关闭
Ctrl+p+q     #无论是否添加-d参数 执行此命令容器都不会被关闭，退出容器
```

停止和启动容器

```
docker stop 容器名/容器ID     #停止当前正在运行的容器
docker kill 容器名/容器ID     #强制停止当前容器
docker start 容器名/容器ID    #启动已经存在的容器
docker restart 容器名/容器ID #重启容器
```

删除容器

```
docker rm  容器名/容器ID       #删除容器，正在运行的容器需要 -f
docker rm -f 容器名/容器ID     #强制删除容器
docker rm -f 容器名/容器ID 容器名/容器ID 容器名/容器ID #删除多个容器，容器名/容器ID用空格隔开即可
docker rm -f $(docker ps -aq)   #删除全部容器
```

容器文件拷贝

无论容器是否开启 都可以进行拷贝

```
#从容器内拷出
docker cp 容器ID/名称: 容器内路径  容器外路径

root@DESKTOP-KQD226K:~/file# ls
root@DESKTOP-KQD226K:~/file# docker cp centos3:/usr/local/hello.txt ./  #拷贝文件hello.txt到
当前目录
root@DESKTOP-KQD226K:~/file# ls
hello.txt

#将外部拷贝文件到容器内
docker  cp 容器外路径 容器ID/名称: 容器内路径

root@DESKTOP-KQD226K:~/sh# docker cp while.sh mycentos1:./
root@DESKTOP-KQD226K:~/sh# docker exec -it mycentos1 bash      #进入容器
[root@e7f6c9c2d872 local]# cd /
[root@e7f6c9c2d872 /]# ls
anaconda-post.log  bin  dev  etc  home  lib  lib64  media  mnt  opt  proc  root  run  sbin
srv  sys  tmp  usr  var  while.sh                                #while.sh文件已经在容器内了
```

更换容器名

```
docker rename 容器ID/容器名 新容器名
```

查看容器大小

```
docker ps -s

SIZE:
括号外面的，表示现在向容器的可写层写入的数据量的大小。
括号里面的（virtual表示：镜像大小+可写层数据量大小之和。
```

```
CONTAINER ID   IMAGE                         COMMAND                  CREATED        STATUS         PORTS
                    NAMES              SIZE
c8f1e62582fb   ubuntu:22.04                  "/bin/bash"              9 days ago     Up 6 days
                    liqingubuntu       10.1GB (virtual 10.1GB)
03fa26872ee8   zhangxudong2/genomedb         "init.sh"                10 days ago    Up 24 hours    5432/tcp, 0.0.0.0:7777->80/tcp, :::7777->80/tcp
                    genomedb           220MB (virtual 3.09GB)
```

（四）docker其他常用命令

查看日志

```
docker logs -tf 容器ID/容器名     #显示所有日志
    -n number        #要查看末尾多少行 默认all 容器ID
```

查看容器内进程

```
docker top 容器ID/容器名
```

查看镜像元数据

```
docker inspect 容器ID/容器名
```

查看容器环境变量

```
docker exec
```

查看docker磁盘使用情况

```
docker system df
```

DockerFile

核心用来构建镜像的文件

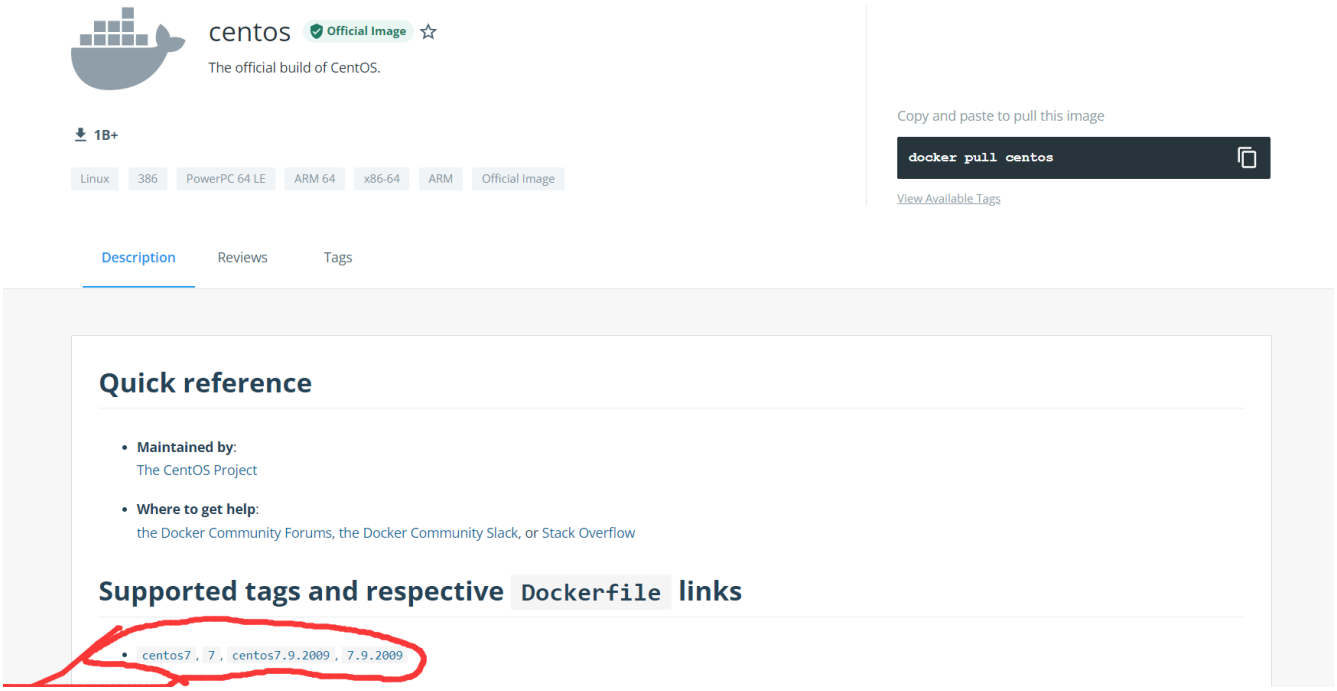
步骤：

1、编写一个dockerfile文件

2、docker build 构建成为一个镜像

3、docker run 运行镜像

4、docker push发布镜像（DockerHub,阿里云镜像库）



可以查看官网dockerfile

构建过程

1、每个保留关键字（指令）都是必须是大写字母

2、执行从上到下顺序执行

3、每个指令都会创建提交一个新的镜像层，

dockerfile面向开发的

指令说明

```
FROM            #基础镜像，一切从这里构建
MAINTAINER      #镜像谁写的（一般姓名+邮箱）
RUN             #镜像构建的时候需要运行的命令
ADD             #步骤，添加内容
WORKDIR         #镜像的工作目录
VOLUME          #挂载的目录
EXPOST          #暴露端口配置
CMD             #指定这个容器启动时需要运行的命令，只有最后一个会生效，可被替代
ENTRYPOTNT      #指定这个容器启动时需要运行的命令，可以追加命令
ONBUILD         #当构建一个被继承Dockerfile 这个时候就会运行ONBUILD ，触发指令。
COPY            #将文件拷贝到镜像中
ENV             #构建的时候设置环境变量
```

例：构建centos

Docker Hub 中99%镜像都是从这个基础镜像过来的FROM scratch,然后配置需要的软件和配置来进行的构建

```
FROM centos:7
MAINTAINER fenghuadong<1576081625@qq.com>
```

```
ENV MYPATH /usr/local
WORKDIR $MYPATH

RUN yum -y install vim
RUN yum -y install net-tools

EXPOSE 80

CMD echo $MYPATH
CMD echo "---end---"
CMD /bin/bash
```

```
docker build -f mydockerfile -t mycentos:0.1 .
```

查看镜像历史 可以查看镜像如何生成

```
a709649711c1   6 days ago     /bin/sh -c #(nop)  CMD ["/bin/sh" "-c" "/bin…   0B
4263de022a94   6 days ago     /bin/sh -c #(nop)  CMD ["/bin/sh" "-c" "echo…   0B
a3be929d57c4   6 days ago     /bin/sh -c #(nop)  CMD ["/bin/sh" "-c" "echo…   0B
ff18c28770f0   6 days ago     /bin/sh -c #(nop)  EXPOSE 80                    0B
2e47a77bc3ac   6 days ago     /bin/sh -c yum -y install net-tools             166MB
9859ce1986c3   6 days ago     /bin/sh -c yum -y install vim                   221MB
af8b8e5a0cc0   6 days ago     /bin/sh -c #(nop) WORKDIR /usr/local            0B
96e7f14fc32c   6 days ago     /bin/sh -c #(nop)  ENV MYPATH=/usr/local        0B
a8faef1d03f1   6 days ago     /bin/sh -c #(nop)  MAINTAINER fenghuadong<15…   0B
eeb6ee3f44bd   8 months ago   /bin/sh -c #(nop)  CMD ["/bin/bash"]            0B
<missing>      8 months ago   /bin/sh -c #(nop)  LABEL org.label-schema.sc…   0B
<missing>      8 months ago   /bin/sh -c #(nop) ADD file:b3ebbe8bd304723d4…   204MB
```

CMD和ENTRYPOINT区别

```
CMD          #指定这个容器启动时需要运行的命令，只有最后一个会生效，会被替代
ENTRYPOTNT   #指定这个容器启动时需要运行的命令，可以追加命令
```

命名“Dockerfile”，build会自动寻找这个文件，不需要-f 指定

发布镜像

发布到dockerhub

1.地址https://hub.docker.com/ 注册账号

2.在服务器提交

先登录

```
root@DESKTOP-KQD226K:~# docker login --help

Usage:  docker login [OPTIONS] [SERVER]
```

```
Log in to a Docker registry.
If no server is specified, the default is defined by the daemon.

Options:
  -p, --password string   Password
      --password-stdin    Take the password from stdin
  -u, --username string   Username


root@DESKTOP-KQD226K:~# docker login -u ash123fhd
```

然后push

```
root@DESKTOP-KQD226K:~/dockerfile# docker tag mycentos:0.1 ash123fhd/mycentos:1.0
root@DESKTOP-KQD226K:~/dockerfile# docker images
REPOSITORY              TAG       IMAGE ID       CREATED         SIZE
ash123fhd/mycentos      1.0       70da306aaaed   6 minutes ago   592MB
mycentos                0.1       70da306aaaed   6 minutes ago   592MB
mysql                   5.7       c20987f18b13   4 months ago    448MB
centos                  7         eeb6ee3f44bd   8 months ago    204MB
centos                  latest    5d0da3dc9764   8 months ago    231MB
zhangxudong2/genomedb   latest    8d48f7cc3071   11 months ago   2.87GB
portainer/portainer     latest    580c0e4e98b0   14 months ago   79.1MB

root@DESKTOP-KQD226K:~/dockerfile# docker push ash123fhd/mycentos:1.0
The push refers to repository [docker.io/ash123fhd/mycentos]
0dde31b64861: Pushed
deb2853b4a57: Pushed
174f568549O3: Pushed
1.0: digest: sha256:ce5d0d289b152ac795Oef38a6b5d032d80df17d408409abc526b58a8ab39cc53 size:
953
root@DESKTOP-KQD226K:~/dockerfile#
```