

docker仓库的用法

仓库用于存储的一个地方

docker仓库，用于存储，管理docker镜像的

docker镜像分发，传输，存储在他人机器上的问题

docker registry

docker镜像注册服务

提供一个集中式的存储，可以分发镜像

一个docker registry中可以包含多个docker 仓库，docker respoistry

每个仓库，都可以包含多个tag，每个tag标记一个版本的镜像

```
1 1.首先得有一个镜像托管平台，docker hub，可以是你自己搭建的私有registry
2 2.在该平台上，以 仓库的形式，管理某一个镜像的不同版本
3 centos Repositories
4 该仓库，以具体的某个tag来管理 具体的 某个版本的镜像文件
5
6 3. 通过命令，上传，或者下载该仓库下的镜像
7 docker pull 下载
8 docker push 上传镜像：tag # 推到镜像仓库中
9
10
11 仓库名:标签
12
13 # 如果不给与tag，默认用latest
14 centos:laste
```

```
15 centos:7.8.2003
16
17 # 具体的仓库名，编写规则
18
19 仓库名一般以两段出现，
20 仓库地址/镜像名
21
22 yuchaoge163
23
24 # 通过如下命令验证
25 [root@yc_docker01 ~]# docker images
26 REPOSITORY          TAG          IMAGE ID
27 CREATED            SIZE
28 仓库的名            镜像版本      镜像id        镜像
29 创建时间          镜像大小
30
31 # 前提就是你得在在linux客户端，登录docker hub
32 docker login命令
33
34 # 一个完整的镜像名是
35 仓库名: tag
36 # 公开仓库的填写案例
37 # yuchao163 是我docker hub的账号id
38 yuchao163/s16-centos-vim:latest
39
40 # 私有仓库的部署
41 # 通过如下地址，找到仓库在哪
42 # 本地镜像必须如下命名
43 仓库地址/软件名
44
45 10.211.55.21:8000/my_flask_web:tag
46
```

47
48
49
50
51

docker hub, 公开仓库的玩法

```
1
2 # 1.官方提示的用法
3 # 修改本地镜像的 名字
4 docker tag local-image:tagname new-repo:tagname
5 # 推送到远程仓库
6 docker push new-repo:tagname
7
8 # 实践
9
10 1. 需要在docker hub创建仓库, 然后命名, 选择公开, 或是私有
11
12 2. 修改本地镜像的名字
13 [root@yc_docker01 ~]# docker tag b4200a856253
    yuchao163/my_nginx_web
14
15 3.在客户端, 登录docker hub
16 [root@yc_docker01 ~]# docker login
17 Login with your Docker ID to push and pull images from Docker
    Hub. If you don't have a Docker ID, head over to
    https://hub.docker.com to create one.
18 Username: yuchao163
19 Password:
```

```
20 WARNING! Your password will be stored unencrypted in
    /root/.docker/config.json.
21 Configure a credential helper to remove this warning. See
22 https://docs.docker.com/engine/reference/commandline/login/#cr
    edentials-store
23
24 Login Succeeded
25
26
27 4.此时可以推送了
28 [root@yc_docker01 ~]#
29 [root@yc_docker01 ~]# docker push yuchao163/my_nginx_web
30
31 5.下载远程镜像文件的用法
32 你可以换一台linux机器，去尝试下载该公开的仓库，以及镜像文件
33
```

私有docker registry

私有镜像注册服务的搭建，创建仓库，提交，托管镜像

- 省带宽
- 传输速度很快
- 存储方便

```
1 # 运行，部署一个私有registry服务
2 # docker以及给你提供好了一个镜像，就是运行私有registry服务的
3 # docker run会自动下载该镜像
4 docker run -d \
5 --name yuchao_registry \
6 --restart=always \
7 -p 5000:5000 \
8 -v /data/registry:/var/lib/registry \
9 registry
```

```
10
11 2.上述就运行了一个私有镜像注册服务
12 该服务，并没有提供好看的web界面，而是基于api的形式，管理
13 没有界面的，但是可以和它进行http的数据交互的一个网站
14
15 # api地址如下，可以看到有哪些镜像文件
16 http://10.211.55.21:5000/v2/_catalog
17
18 3.尝试推送镜像到仓库中去
19 # 修改tag
20 [root@yc_docker01 ~]# docker tag b4200a856253
    10.211.55.21:5000/my_local_nginx
21 # 上传镜像
22
23 4.发现推不上去，是因为，客户端发出的是https请求，服务端是http响应处
    理
24 得修改docker配置文件，去掉https、基于http的推送
25
26 # 修改如下
27 {
28     "registry-mirrors": ["http://f1361db2.m.daocloud.io"],
29     "insecure-registries":["10.211.55.21:5000"]
30 }
31
32 # 重启docker生效
33 [root@yc_docker01 ~]# systemctl daemon-reload
34 [root@yc_docker01 ~]#
35 [root@yc_docker01 ~]#
36 [root@yc_docker01 ~]# systemctl restart docker
37 [root@yc_docker01 ~]#
38
39 5.再次推送镜像
40 [root@yc_docker01 ~]# docker push
    10.211.55.21:5000/my_local_nginx
```

```
41
42 6.尝试访问验证, 私有镜像
43 浏览器访问
44 http://10.211.55.21:5000/v2/_catalog
45
46 命令行测试
47 [root@yc_docker01 ~]#
48 [root@yc_docker01 ~]# curl -s 10.211.55.21:5000/v2/_catalog
49 |python -m json.tool
50 {
51     "repositories": [
52         "my_local_nginx"
53     ]
54 }
55 7.删除本地的镜像, 尝试重新下载该镜像
56
57
58
59
60
61
62
```

