



3、Docker镜像相关命令

1、镜像相关命令

(0) 可以先配置下国内的镜像加速器，为了解决去远程dockerhub仓库拉取镜像时超时的错误，用国内阿里云的镜像加速器。

- 拉取elasticsearch镜像时的错误信息



```
Trying to pull repository docker.io/library/elasticsearch ...
```

```
Get https://registry-1.docker.io/v2/: read tcp 10.0.24.2:47824->34.238.187.50:443: read: connection reset by peer
```



- 登录阿里云，找到容器镜像服务-镜像工具-镜像加速器，在centos中配置即可



容器镜像服务

实例列表

镜像中心 ^

镜像搜索

我的收藏

镜像工具 ^

镜像加速器

! 使用加速器可以提升获取Docker官方镜像的速度

加速器

加速器地址

<https://ppbksqjv.mirror.aliyuncs.com> 复制

操作文档

Ubuntu

CentOS

Mac

Windows

1. 安装 / 升级Docker客户端

推荐安装 1.10.0 以上版本的Docker客户端，参考文档[docker-ce](#)

2. 配置镜像加速器

针对Docker客户端版本大于 1.10.0 的用户

您可以通过修改daemon配置文件 `/etc/docker/daemon.json` 来使用加速器

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



LINUX中执行如下命令：

```
sudo mkdir -p /etc/docker
```

#tee方式处理：

```
$ sudo tee /etc/docker/daemon.json <<-'EOF'
```

//上一步sudo tee /etc/docker/daemon.json <<-'EOF'输入后回车，再将下面的阿里云的镜像仓库配置复制粘贴到控制台，再回车即可

//推荐阿里云的镜像仓库

```
{  
  "registry-mirrors": ["https://ppbksqjv.mirror.aliyuncs.com"]  
}  
EOF
```

或

```
{  
"registry-mirrors":["https://docker.mirrors.ustc.edu.cn"]  
}  
EOF
```

或

```
{  
"registry-mirrors":["https://docker.mirrors.ustc.edu.cn"]  
}  
EOF
```

#vim方式处理：

```
$ vim /etc/docker/daemon.json
```

```
{  
  "registry-mirrors":["https://docker.mirrors.ustc.edu.cn"]  
}
```

wq保存退出。

```
//执行完毕后，daemon.json文件会增加一行配置
//重启docker服务
sudo systemctl daemon-reload
sudo systemctl restart docker
```

```
//重新拉取需要的镜像即可
```



(1) 使用**search**从DockerHub远程仓库 (<https://hub.docker.com/>) 上**检索镜像**

```
docker search 关键字
docker search mysql
```

(2) 使用**pull**从docker hub远程仓库上**拉取镜像**

- docker pull 镜像名:tag (tag表示标签，多为软件版本，默认是latest可省略)
- 可去dockerhub上搜索框直接搜索指定版本

```
docker pull mysql
docker pull mysql:8.0.20
docker pull tomcat:8-jre8
docker pull registry.docker-cn.com/library/redis
```

(3) 使用**push**往dockerhub中**上传镜像**

- 先创建dockerhub的仓库hgm/mysql （ hgm是用户名，mysql是仓库名称）

```
docker login
docker images
docker tag 旧镜像 新标签名//docker tag 旧镜像名:版本号 hgm/mysql:版本号
docker push test/mysql:5.0
```

（ 4 ）使用**images**查看linux系统本地**所有镜像**

```
docker images
```

（ 5 ）使用**rmi**将linux系统本地中**指定镜像删除**(IMAGE_ID代表镜像id)

```
docker rmi IMAGE_ID //仅删除没有容器引用的镜像
docker rmi -f //强制删除镜像，即使有容器引用该镜像（一般不推荐，正确是先删除容器再删除镜像）
docker rmi -no-prune //不删除未带标签的父镜像（即删除镜像的新标签时，如果此镜像有多个标签，删除众多标签的指定某一个标签时，不会删除父镜像；如果只有一个标签，删除此标签时父镜像也会删除）。
```

（ 6 ）使用**tag**命令为现有拉取到本地的镜像**添加一个新的标签（镜像副本）**

```
docker tag 镜像名 新的标签名
```

（ 7 ）使用**inspect**命令**获取镜像或容器详细信息**（-f参数来获取指定的内容）

```
docker inspect 镜像名或镜像id或容器名或容器id
docker inspect -f {{.Size}} mysql
```



(8) 使用**history**命令**查看镜像历史**(--no-trunc 查看具体信息)

```
docker history mysql
```

(9) 使用**image prune**来进行**清理残存的临时文件及未被使用的镜像文件**

```
docker image prune
```

(10) 使用**container commit**命令**基于已有镜像，创建（克隆）新的镜像。**

```
docker container commit -m "创建提交信息" -a "作者信息" 旧镜像ID 新镜像名称:版本号  
docker container commit -m "这是创建新镜像" -a "1534834526@qq.com" 906859c32a88 hgm-mysql:1.0
```

(11) 先编写一个Dockerfile文件指令文件(文件中是基于旧镜像创建名为python:3新镜像的指令)，然后通过编译命令：

```
docker image build -t python:3 .
```

(12) 使用**save**从本地系统**导出一个镜像为tar包**

```
docker save -o 归档文件tar的路径 IMAGEID或镜像名:tag  
docker save -o mysql.tar mysql:版本号
```

12 使用**load**往本地系统**加载tar包成为一个镜像**

```
docker load -i mysql.tar或 docker load < mysql.tar
```



