## 1、创建docker镜像Hub

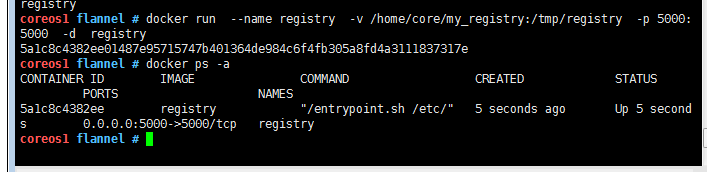
https://github.com/docker/docker.github.io/blob/master/registry/deploying.md

docker run -d -p 5000:5000 --restart=always --name registryV2 -v my\_regisitry :/var/lib/registry registry:2

docker run -d -p 5000:5000 --restart=always --name registryV2 -v `pwd`/data:/var/lib/registry registry:2

docker run --name registry -v /home/core/my\_registry:/tmp/registry -p 5000:5000 -d registry

docker run --name registry --restart=always -v /my\_registry:/tmp/registry -p 5000:5000 -d registry

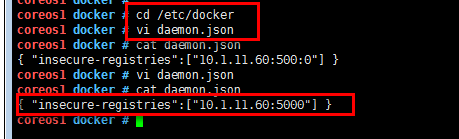


## 2、把调用Hub的主机docker直接方式从https修改为http

所有需要从该hub中访问，其os中都需要进行如下操作

在”*/etc/docker/*“目录下，创建”daemon.json“文件。在文件中写入：

{ "insecure-registries":["10.1.11.60:5000"] }[[1]](#footnote-1)



然后调用

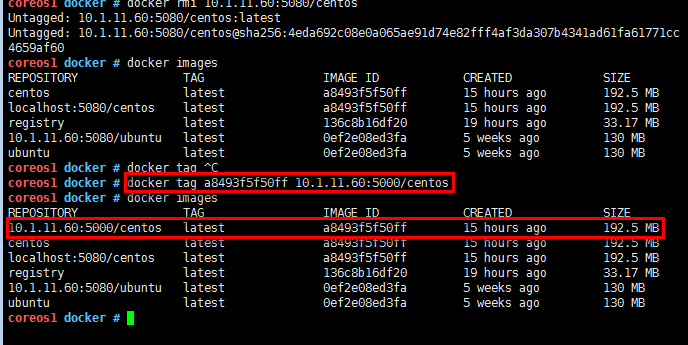
sudo systemctl daemon-reload

sudo systemctl restart docker

|  |
| --- |
| centos中修改docker配置文件，指定私有仓库url  [root@localhost ~]# vim /etc/sysconfig/docker 修改此行  OPTIONS='--insecure-registry C:\Users\prk\AppData\Local\Temp\%W@GJ$ACOF(TYDYECOKVDYB.png10.1.11.60：5000' [root@localhost ~]# service docker restart  重启docker |

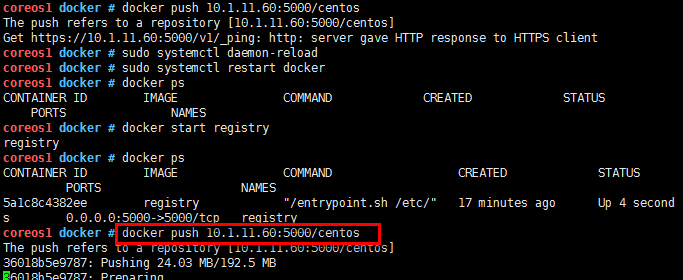
## 3、 tag 镜像

docker tag a8493f5f50ff 10.1.11.60:5000/centos

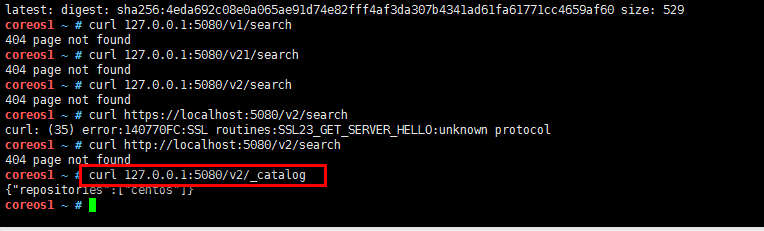


## 4、上传：

docker push 10.1.11.60:5000/centos



## 5、查询



访问私有仓库

# curl 127.0.0.1:5000/v1/search ##网上用的这个命令报404

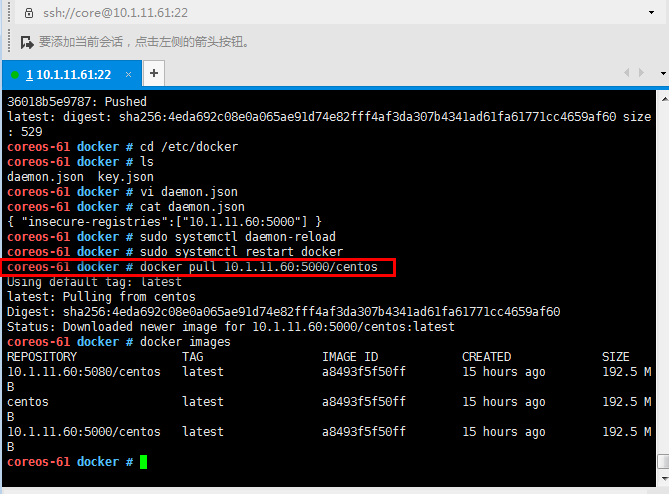
404 page not found

后面查了资料发现是v1版本的api查看方式，我们现在的版本是v2，用下面命令即可

# curl 127.0.0.1:5000/v2/\_catalog

{"repositories":["registry"]} ##有一个registry镜像

## 5、下载



## 参考：

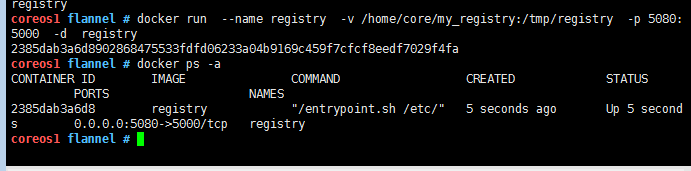
Docker支持采用仓库（本处指的是registry）来支持镜像的分发和更新管理。这极大的便利了用户。官方提供了dockerhub网站来作为一个公开的集中仓库。然而，本地访问dockerhub速度往往很慢，并且很多时候我们需要一个本地的私有仓库只供网内使用。

1 使用registry启动私有仓库的容器  
docker run -d -p 5000:5000 -v /root/my\_registry:/tmp/registry registry

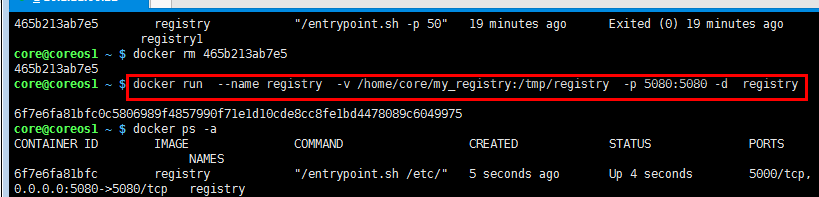
docker run -d -p 5000:5000 -v /home/core/my\_registry:/tmp/registry registry --name registry --restart=always

docker run --name registry -v /home/core/my\_registry:/tmp/registry -p 5080:5000-d registry --restart=always

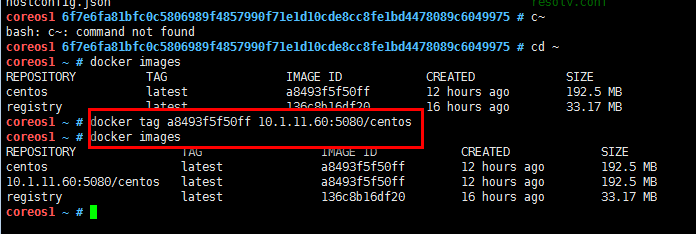
docker run --name registry -v /home/core/my\_registry:/tmp/registry -p 5080:5000 -d registry



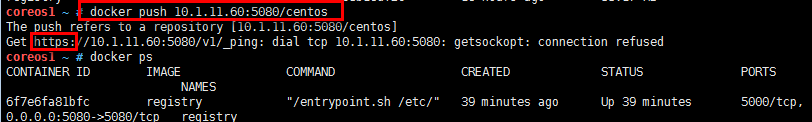
说明：若之前没有安装registry容器则会自动下载并启动一个registry容器，创建本地的私有仓库服务。默认情况下，会将仓库创建在容器的/tmp/registry目录下，可以通过 -v 参数来将镜像文件存放在本地的指定路径上（例如，放在本地目录/root/my\_registry下）。

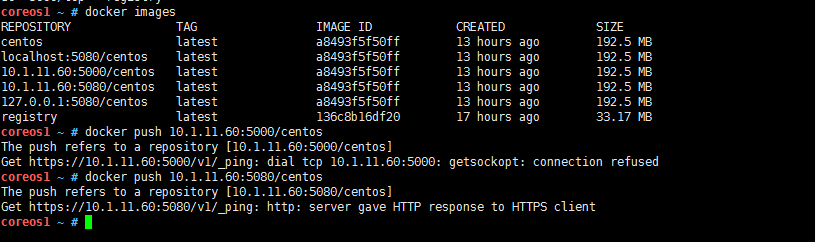


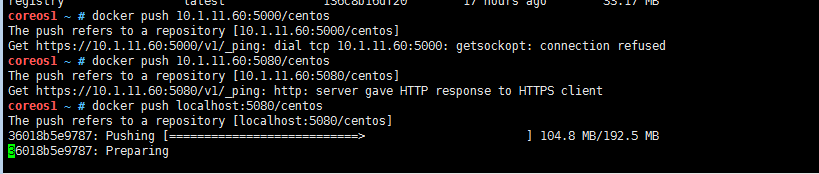
docker tag a8493f5f50ff 10.1.11.60:5080/centos



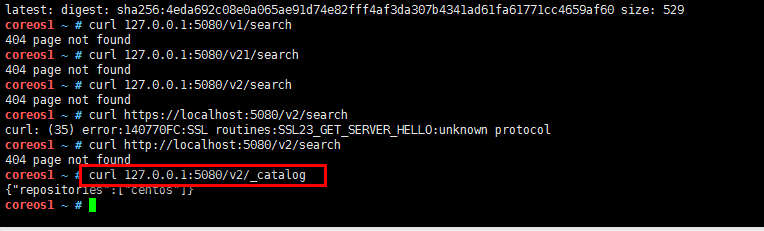
docker push 10.1.11.60:5080/centos







因为Docker从1.3.x之后，与docker registry交互默认使用的是https，但是此处搭建私有仓库却只提供http服务，所以当和私有仓库交互时报上述错误。因此需要在启动docker server时增加启动参数为默认使用http访问。



访问私有仓库

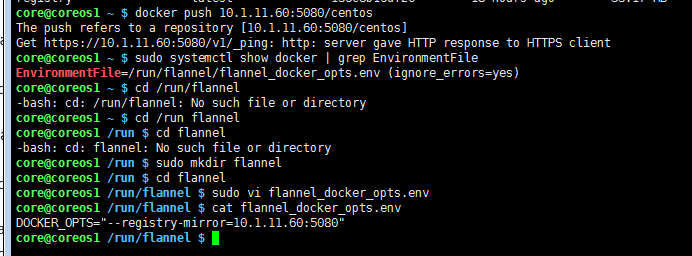
# curl 127.0.0.1:5000/v1/search ##网上用的这个命令报404

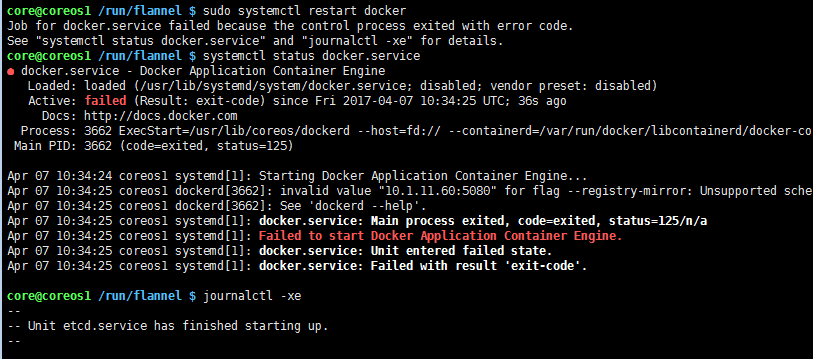
404 page not found

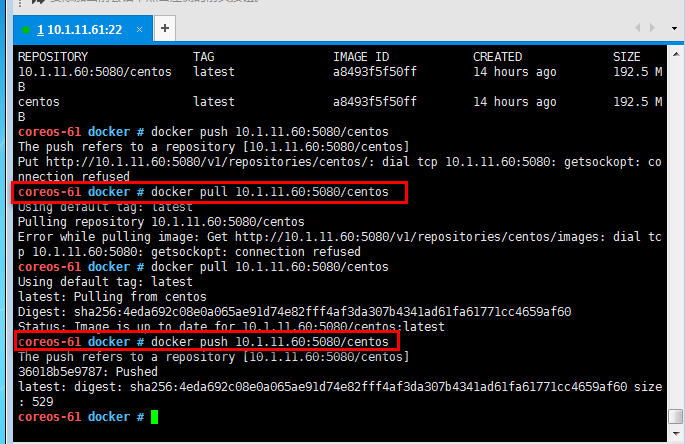
后面查了资料发现是v1版本的api查看方式，我们现在的版本是v2，用下面命令即可

# curl 127.0.0.1:5000/v2/\_catalog

{"repositories":["registry"]} ##有一个registry镜像







2 向私有仓库push镜像  
启动完register的镜像后，将register镜像上传到本地仓库上作为测

    localhost my\_registry # ls -R  
    .:

http://www.cnblogs.com/hobinly/p/6110624.html

<http://www.cnblogs.com/zeppelin/p/5906410.html>

docker版本： 1.12.3（注意版本，可能存在不同版本设置不同的情况）

docker registry版本：2.4.1

问题：

成功安装docker registry，在浏览器中输入http://192.168.1.100:5000/v2,成功返回json数据。在push 到docker registry时，报：

[root@master sysconfig]# docker push 192.168.1.100:5000/registry:2.4.1

The push refers to a repository [192.168.1.100:5000/registry]

Get https://192.168.1.100:5000/v1/\_ping: http: server gave HTTP response to HTTPS client

这个问题可能是由于客户端采用https，docker registry未采用https服务所致。一种处理方式是把客户对地址“192.168.1.100:5000”请求改为http。

目前很多文章都是通过修改docker的配置文件“etc/systemconfig/docker"，重启docker来解决这个问题。但发现docker1.12.3版本并无此文件，根据网上创建此文件，并填入相应内容，重启docker无效果，仍然报此错误。

解决方法：

在”*/etc/docker/*“目录下，创建”daemon.json“文件。在文件中写入：

{ "insecure-registries":["192.168.1.100:5000"] }

保存退出后，重启docker。问题解决：

[复制代码](javascript:void(0);)

[root@master docker]# docker push 192.168.1.100:5000/registry:2.4.1

The push refers to a repository [192.168.1.100:5000/registry]

ee8e809cfde5: Pushed

ba20d499f984: Pushed

705e35f12f24: Pushed

42755cf4ee95: Pushed

2.4.1: digest: sha256:b66c4af9577744ae6d32e975808230e2ff558a5d50a7968d5102a900e147f3d5 size: 1158

[复制代码](javascript:void(0);)

window server 2016,默认安装docker的配置文件在“C:\ProgramData\docker\config\”。可以在该目录下创建”daemon.json“文件，解决此问题。

1. [↑](#footnote-ref-1)