

Docker部署私有仓库
Author: LiNing

Docker镜像能够以tar包的形式导出和导入，实现程序部署环境的迁移，也能够像github、bitbucket等版本仓库那样将镜像推送到Docker远程仓库上。

Docker仓库分为公有仓库和私有仓库，官方有个仓库叫Docker Hub可以供Docker开发者实现镜像的推送和拉取，也支持公有仓库和私有仓库。但是网络问题，如果从官方仓库拉取镜像速度会很慢，而且有些涉及到重要项目的镜像推送到人家服务器上，即使是私有仓库，也是感觉不安全的。

对于公司进行Docker开发的话，应该维护自己一个内网的私有仓库。这样不仅便于下载速度快，节省网络带宽，针对于每个镜像不用每个人都去中央仓库上面去下载，只需要从私有仓库中下载即可，而且提供镜像资源利用，针对于公司内部使用的镜像，推送到本地的私有仓库中，以供公司内部相关人员使用。

创建私有仓库的步骤如下：

- 首先需要有两台机器，一台作为仓库的Server，一台作为仓库的Client，用户通过Client来进行镜像的提交和拉取操作。

这里，采用两台虚拟机做例子：

Ubuntu 14.04x64 Server: ip地址 192.168.142.151
Ubuntu 14.04x64 Client: ip地址 192.168.142.152

- 在Server端启动registry容器：

启动命令：

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

```
fireling@fireling-virtual-machine:~$ docker run -d --name registry -p 5000:5000 -v /opt/data/registry:/var/lib/registry --restart=always registry
53184e30a0ceec57deec33a442cee6115330abe8814871d132706b516c71f7d
fireling@fireling-virtual-machine:~$ docker ps -l
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS               NAMES
53184e30a0ce        registry           "/entrypoint.sh /etc/"   20 seconds ago     Up 18 seconds      0.0.0.0:5000->5000/tcp   registry
fireling@fireling-virtual-machine:~$
```

这里建立了一个名为registry的守护式容器，挂载了服务器目录/opt/data/registry，映射端口为5000，这样在本地或者远程就可以通过该端口来访问仓库。

- 在Client端进行Docker配置：

修改配置文件/etc/default/docker，在DOCKER_OPTS中加入"--insecure-registry 192.168.142.151:5000"。这样把一个通过认证的安全证书加到Registry服务中。

```
# Docker Upstart and SysVinit configuration file

#
# THIS FILE DOES NOT APPLY TO SYSTEMD
#
# Please see the documentation for "systemd drop-ins":
# https://docs.docker.com/engine/articles/systemd/
#

# Customize location of Docker binary (especially for development testing).
#DOCKERD="/usr/local/bin/dockerd"

# Use DOCKER_OPTS to modify the daemon startup options.
DOCKER_OPTS="--dns 8.8.8.8 --dns 8.8.4.4 --registry-mirror=http://6d8ee622.m.daocloud.io --insecure-registry 192.168.142.151:5000"

# If you need Docker to use an HTTP proxy, it can also be specified here.
#export http_proxy="http://127.0.0.1:3128/"

# This is also a handy place to tweak where Docker's temporary files go.
#export TMPDIR="/mnt/bigdrive/docker-tmp"
~
```

修改完配置，要重启docker进程：

```
sudo service docker restart
```

- 在Client端进行镜像的推送和拉取：

标签指定服务ip和端口，即仓库名：

```
docker tag hello-world 192.168.142.151:5000/myfirstimage
```

向仓库推送镜像：

```
docker push 192.168.142.151:5000/myfirstimage
```

```
fireling@fireling-virtual-machine:~$ docker push 192.168.142.151:5000/myfirstimage
The push refers to a repository [192.168.142.151:5000/myfirstimage]
a02596fdd012: Pushed
latest: digest: sha256:a18ed77532f6d6781500db650194e0f9396ba5f05f8b50d4046b294ae5f83aa4 size: 524
fireling@fireling-virtual-machine:~$
```

删除本地镜像：

```
docker rmi 192.168.142.151:5000/myfirstimage
```

从仓库拉取镜像

```
docker pull 192.168.142.151:5000/myfirstimage
```