链滴

搭建 docker-registry

作者：happyhacker
原文链接：https://ld246.com/article/1520237631134
来源网站：链滴

准备工作

1. Ubuntu 16.04(with docker installed)

步骤

1. 安装必要的包

sudo apt install -y docker-compose apache2-utils curl

2. 创建相关目录

```
mkdir /docker-registry
mkdir  /docker-registry/data
mkdir /docker-registry/nginx
chown root:root /docker-registry
cd /docker-registry
```

3. 创建 docker-compose.yml

```
nginx:
  image: "nginx:1.9"
  ports:
    - 443:443
  links:
    - registry:registry
  volumes:
    - /docker-registry/nginx/:/etc/nginx/conf.d
registry:
  image: registry:2
  ports:
    - 127.0.0.1:5000:5000
  environment:
    REGISTRY_STORAGE_FILESYSTEM_ROOTDIRECTORY: /data
  volumes:
    - /docker-registry/data:/data
```

执行docker-compose up, 就会看到一堆信息, 没有看到错误提示的话就是安装成功了.

4. 如果是在生产环境, 还是有必要让它保持后台运行

方案可以选择 systemd 或 supervisord, 这里提供 systemd 的方案

```
[Unit]
Description=Starting docker registry

[Service]
Environment= MY_ENVIRONMENT_VAR = /docker-registry/docker-compose.yml
WorkingDirectory=/docker-registry
ExecStart=/usr/bin/docker-compose up
```

```
Restart=always

[Install]
WantedBy=multi-user.target
```

5. 修改 Nginx 配置

```
# /docker-registry/nginx/regstry.conf
upstream docker-registry {
  server registry:5000;
}

server {
  listen 443;
  server_name myregistrydomain.com;

  # SSL
  # ssl on;
  # ssl_certificate /etc/nginx/conf.d/domain.crt;
  # ssl_certificate_key /etc/nginx/conf.d/domain.key;

  # disable any limits to avoid HTTP 413 for large image uploads
  client_max_body_size 0;

  # required to avoid HTTP 411: see Issue #1486 (https://github.com/docker/docker/issues/14
6)
  chunked_transfer_encoding on;

  location /v2/ {
    # Do not allow connections from docker 1.5 and earlier
    # docker pre-1.6.0 did not properly set the user agent on ping, catch "Go *" user agents
    if ($http_user_agent ~ "^(docker\/1\.(3|4|5(?!\.[0-9]-dev))|Go ).*$" ) {
      return 404;
    }

    # To add basic authentication to v2 use auth_basic setting plus add_header
    # auth_basic "registry.localhost";
    # auth_basic_user_file /etc/nginx/conf.d/registry.password;
    # add_header 'Docker-Distribution-Api-Version' 'registry/2.0' always;

    proxy_pass                          http://docker-registry;
    proxy_set_header  Host              $http_host;   # required for docker client's sake
    proxy_set_header  X-Real-IP         $remote_addr; # pass on real client's IP
    proxy_set_header  X-Forwarded-For   $proxy_add_x_forwarded_for;
    proxy_set_header  X-Forwarded-Proto $scheme;
    proxy_read_timeout                  900;
  }
}
```

重启服务使配置生效 sudo systemctl restart docker-registry

执行curl http://localhost:5000/v2/, 如果正常, 应该输出{}

6. 配置认证信息

```
cd /docker-registry/nginx
htpasswd -c registry.password mydocker
New password:
Re-type new password:
Adding password for user mydocker
```

修改 nginx 配置(删除相应行前面的注释)

```
auth_basic "registry.localhost";
auth_basic_user_file /etc/nginx/conf.d/registry.password;
add_header 'Docker-Distribution-Api-Version' 'registry/2.0' always;
```

重启服务使配置生效 sudo systemctl restart docker-registry

执行curl http://localhost:443/v2/, 会提示401 Authorization Required

把刚刚创建的用户名和密码加上会得到正确的返回结果

curl http://mydocker:123456@localhost:443/v2/ 结果是{}

7. 配置证书

```
cd /docker-registry/nginx
# 生成一个 root key
openssl genrsa -out dockerCA.key 2048
# 生成根证书( ComonName 写你的域名, 比如域名是 registry.abc.com, 这里就写 registry.abc.com)
openssl req -x509 -new -nodes -key dockerCA.key -days 10000 -out dockerCA.crt
# 生成 server key, 在 nginx 的配置ssl_certificate_key中引用
openssl genrsa -out domain.key 2048
# 申请一个新证书( ComonName 写你的域名, 比如域名是 registry.abc.com, 这里就写 abc.com)
openssl req -new -key domain.key -out docker-registry.com.csr
# 给证书签名
openssl x509 -req -in docker-registry.com.csr -CA dockerCA.crt -CAkey dockerCA.key -CAcrea
eserial -out domain.crt -days 10000
```

因为我们生成的是一个"自签名"证书, 所以是无法被其他证书发布机构认证的, 也就是必须让客户端强信任该证书.

在宿主机上执行

```
cd /docker-registry/nginx
cp dockerCA.crt /usr/local/share/ca-certificates/
update-ca-certificates && service docker restart && service docker-registry restart
curl https://mydocker:123456@docker-server.com/v2/
# 结果应该是
{}
```

然后手工把自签名证书发布给需要的客户端,

```
scp dockerCA.crt ja@192.168.0.59:/usr/local/share/ca-certificates
ja@192.168.0.59's password:
dockerCA.crt 100% 1302 1.3KB/s 00:00
```

在客户端上, 执行

update-ca-certificates && service docker restart
#test login to fresh created repository:
docker login https://docker-server.com
Username: mydocker
Password:
Login Succeeded

8. 在客户端上测试 container 的功能

docker run -it ubuntu # 从中心 library 下载并执行 ubuntu
#re-tag images DOMAIN-NAME/NEW-TAG
docker tag ubuntu docker-server.com/test-image # 给已有container重新打 tag
#push image to repository:
docker push docker-server.com/test-image # push 到新建的 registry
docker rmi -f docker-server.com/test-image # 删除本地镜像
docker pull docker-server.com/test-image # 从新的 registry 拉取镜像