Docker Registry

Docker Register作为Docker的核心组件之一负责镜像内容的存储与分发，客户端的docker pull以及push命令都将直接与register进行交互，最初版本的registry由python是心啊，由于涉及初期在安全性、性能以及API的设计上有着诸多的缺陷，该版本在0.9之后停止了开发，由新的项目distribution（新的docker register被称为Distribution）来重新设计并开发了下一代的registry，新的项目由go语言开发，所有的api底层存储方式，系统架构都进行了全面的重新设计已解决上一代registry钟村镇的问题。

官方文档地址：https://docs.docker.com/registry

官方github地址: https://github.com/docker/distribution

搭建镜像

- 下载docker registry镜像

```
[root@docker-server ~]# docker pull registry
```

- 创建授权使用目录

```
[root@docker-server ~]# mkdir /docker/auth -p
```

- 创建用户

```
[root@docker-server docker]# yum install httpd-tools.x86_64 -y
[root@docker-server docker]# htpasswd -Bbn jack 123456 > auth/htpasswd
[root@docker-server docker]# cat auth/htpasswd
jack:$2y$05$a2wtUYyoC8p/eXzoseT9Q.dhMDgQgwkUiKVfs1z6zijk6M4UIiUsq
[root@docker-server docker]# docker run -d -p 5000:5000 -v /docker/auth/:/auth -e \
"REGISTRY_AUTH=htpasswd" \
-e "REGISTRY_AUTH_HTPASSWD_REALM=Registry Realm" \
-e "REGISTRY_AUTH_HTPASSWD_PATH=/auth/htpasswd" registry
[root@docker-server docker]# docker ps -l
CONTAINER ID   IMAGE      COMMAND                  CREATED         STATUS         PORTS                                       NAMES
7be5539c6cbe   registry   "/entrypoint.sh /etc…"   7 seconds ago   Up 6 seconds   0.0.0.0:5000->5000/tcp, :::5000->5000/tcp   nostalgic_stonebraker
[root@docker-server docker]# curl 127.0.0.1:5000
[root@docker-server docker]# docker login  127.0.0.1:5000
Username: jack
Password:
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded

```

Docker 仓库之分布式harbor

参考网址：https://goharbor.io/docs/2.3.0/install-config/

Harbor是一个用于存储和分发Docker镜像的企业级Registry服务器，由vmware开源，其通过添加一些企业必须的功能特性，例如安全、标识和管理等，扩展了开源的Docker Distribution。Harbor也提供了高级的安全特性，诸如用户管理，访问控制和活动审计等。

安装

- 下载镜像

```
[root@docker-server1 ~]# wget https://github.com/goharbor/harbor/releases/download/v2.3.1/harbor-offline-installer-v2.3.1.tgz
```

- 解压缩

```
[root@docker-server1 ~]# tar xzvf harbor-offline-installer-v2.3.1.tgz
[root@docker-server1 ~]# ln -sv /root/harbor /usr/local/
"/usr/local/harbor" -> "/root/harbor"
```

- 安装harbor

```
[root@docker-server1 harbor]# cp harbor.yml.tmpl harbor.yml
[root@docker-server1 harbor]#  grep -Ev '#|^$' harbor.yml.tmpl > harbor.yml
[root@docker-server1 harbor]# cat harbor.yml
hostname: 192.168.175.10
http:
  port: 80
harbor_admin_password: Harbor12345
database:
  password: root123
  max_idle_conns: 100
  max_open_conns: 900
data_volume: /data
trivy:
  ignore_unfixed: false
  skip_update: false
  insecure: false
jobservice:
  max_job_workers: 10
notification:
  webhook_job_max_retry: 10
chart:
  absolute_url: disabled
log:
  level: info
  local:
    rotate_count: 50
    rotate_size: 200M
    location: /var/log/harbor
_version: 2.3.0
proxy:
  http_proxy:
```

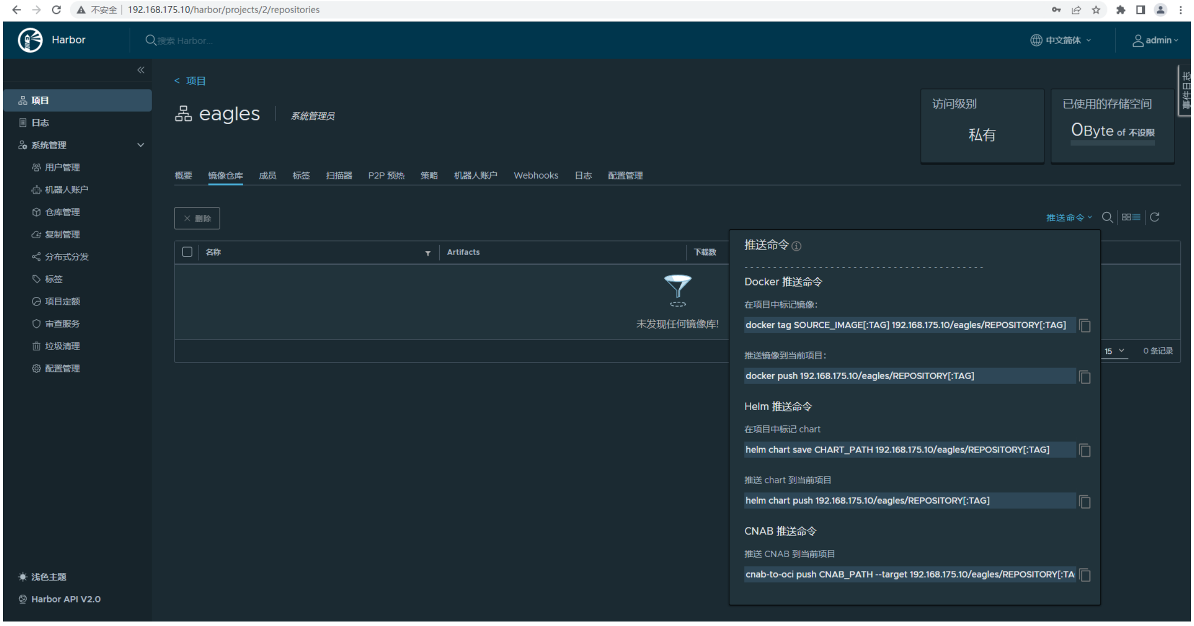
```
  https_proxy:
  no_proxy:
  components:
    - core
    - jobservice
    - trivy
[root@docker-server1 harbor]# ./prepare
[root@docker-server1 harbor]# ./install.sh

[root@docker-server1 harbor]# ss -tnl
State      Recv-Q Send-Q Local Address:Port               Peer Address:Port
LISTEN     0      128                *:80                            *:*
LISTEN     0      128                *:22                            *:*
LISTEN     0      100        127.0.0.1:25                            *:*
LISTEN     0      128        127.0.0.1:1514                          *:*
LISTEN     0      128               :::80                           :::*
LISTEN     0      128               :::22                           :::*
LISTEN     0      100              ::1:25                           :::*

---
# 之后的启动关闭可以通过docker-compose管理，自动生成docker-compose.yml文件
[root@docker-server1 harbor]# ls
common     docker-compose.yml    harbor.yml       install.sh  prepare
common.sh  harbor.v2.3.1.tar.gz  harbor.yml.tmpl  LICENSE
```

web访问

默认用户名和密码在harbor.yml中设置为：harbor_admin_password: Harbor12345

推送镜像

- 网站上有推送教学



推送命令

Docker 推送命令

在项目中标记镜像：

docker tag SOURCE_IMAGE[:TAG] 192.168.175.10/eagles/REPOSITORY[:TAG]

推送镜像到当前项目：

docker push 192.168.175.10/eagles/REPOSITORY[:TAG]

Helm 推送命令

在项目中标记 chart

helm chart save CHART_PATH 192.168.175.10/eagles/REPOSITORY[:TAG]

推送 chart 到当前项目

helm chart push 192.168.175.10/eagles/REPOSITORY[:TAG]

CNAB 推送命令

推送 CNAB 到当前项目

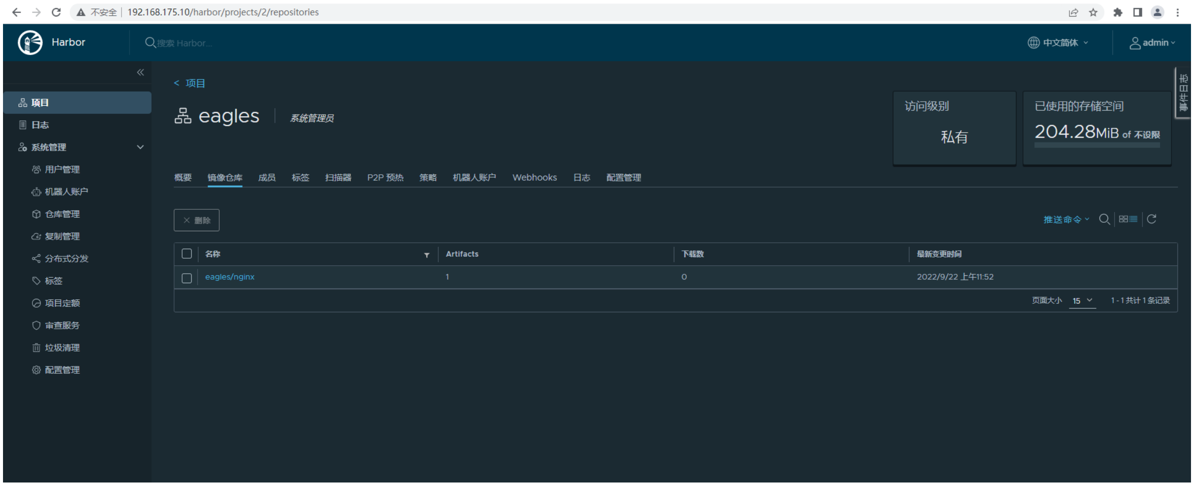
cnab-to-oci push CNAB_PATH --target 192.168.175.10/eagles/REPOSITORY[:TA

- 登陆成功

```
[root@docker-server2 ~]# vim /etc/docker/daemon.json
{
 "registry-mirrors":["https://almtd3fa.mirror.aliyuncs.com"],
 "insecure-registries":["192.168.175.10"]
}
[root@docker-server2 ~]# systemctl daemon-reload
[root@docker-server2 ~]# systemctl restart docker
[root@docker-server2 ~]# docker login 192.168.175.10
Username: admin
Password:
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
```

- 打标签推送镜像

```
[root@docker-server2 nginx]# docker images
REPOSITORY    TAG       IMAGE ID       CREATED          SIZE
nginx         v1        9502baadabe2   14 seconds ago   525MB
nginx         latest    4cdc5dd7eaad   2 weeks ago      133MB
centos        7         8652b9f0cb4c   8 months ago     204MB
[root@docker-server2 nginx]# docker tag nginx:v1 192.168.175.10/eagles/nginx:v1
[root@docker-server2 nginx]# docker push 192.168.175.10/eagles/nginx:v1
67fdcbb9bb94: Pushed
907624200ee8: Pushed
b3438bfb2fad: Pushed
810a6b84c4e8: Pushed
174f56854903: Pushed
v1: digest: sha256:d60c3659de05a6c1acfd3d0f7746ca189b42fca9dfece0667b543c44e6cf49b8 size: 1372
```



不安全 | 192.168.175.10/harbor/projects/2/repositories

< 项目

eagles 系统管理员

访问级别: 私有

已使用的存储空间: 204.28MiB of 不设限

名称	Artifacts	下载数	最新变更时间
eagles/nginx	1	0	2022/9/22 上午11:52

页面大小 15 1 - 1 共计 1 条记录