镜像仓库

Subtitle

2022/10/05

Table of Contents

镜像仓库

使用Harbor

安装harbor

提前创建配置文件中指定的 postgres 数据库，可以用 pgAdmin 创建

```
coreDatabase: "registry"
clairDatabase: "clair"
notaryServerDatabase: "notary_server"
notarySignerDatabase: "notary_signer"
```

使用helm安装harbor

snippet.bash

```bash
# 更新仓库,使用最新版
helm repo update
# 测试环境
helm install cr harbor/harbor --version 1.2.0 --namespace dev -f ./values-test.yaml
# 生产环境
helm install cr harbor/harbor --version 1.2.0 --namespace intra -f ./values.yaml
```

测试

测试环境生成一个自签名证书，系统种导入 ca.pem

> 由于某些需要在设置好xxnet的相关属性后，通过web浏览器访问网站的时候总是报证书错误，把相关的证书导入到浏览器里面也不行，如是有了以下的方法：
> 把CA放到`/etc/pki/ca-trust/source/anchors`，在命令行运行`/bin/update-ca-trust`，这样证书就导入到系统中去了。

样就可以顺利的使用xxnet了。

对于 docker，需要重启 docker daemon

push失败排查

报错	原因	解决方案	参考资料
denied: requested access to the resource is denied	可能是项目不存在	检查 project/image:tag 是否正确	
blob upload invalid	可能是 s3 上传失败	检查 s3 nginx代理的 client_max_body_size，设置为足够大的值，比如 5000m	

其他问题

使用外部**nginx**做代理无法**push**

此问题似乎只在 nginx 后是 docker-compose 部署的harbor 实例时出现，直接使用 Kubernetes 部署并使用 ingress nginx，无此问题。

snippet.bash

```
# docker push cr.xxx.com/test/alpine:3.10
The push refers to repository [cr.xxx.com/test/alpine]
256a7af3acb1: Pushing [==================================================>]
5.615MB
EOF
```

解决方案：https://github.com/docker/distribution/issues/970

snippet.yaml

```
# grep -B3 "relative" common/config/registry/config.yml
http:
  addr: :5000
  secret: placeholder
  relativeurls: true
```

redis cluster 报错

```
2019-09-19T15:28:15Z [DEBUG] [/common/dao/project.go:149]: sql:=select distinct p.project_id,
p.name, p.owner_id,
       p.creation_time, p.update_time  from project as p where p.deleted=false order by p.name,
param= []
2019/09/19 15:28:15 [I] [asm_amd64.s:1337] http server Running on http://:8080
2019/09/19 15:28:41 [E] [server.go:2774] MOVED 5089 192.168.64.117:6379
2019/09/19 15:28:41 [D] [server.go:2774] |  10.112.33.205| 503 |   9.958842ms| nomatch| GET
```

```
/api/ping
2019/09/19 15:28:41 [E] [server.go:2774] MOVED 13229 192.168.65.101:6379
2019/09/19 15:28:41 [D] [server.go:2774] |  10.112.33.205| 503 |   1.940507ms| nomatch| GET
/api/ping
2019/09/19 15:28:51 [E] [server.go:2774] MOVED 3558 192.168.64.117:6379
2019/09/19 15:28:51 [D] [server.go:2774] |  10.112.33.205| 503 |    6.13463ms| nomatch| GET
/api/ping
2019/09/19 15:28:51 [E] [server.go:2774] MOVED 10236 192.168.100.31:6379
2019/09/19 15:28:51 [D] [server.go:2774] |  10.112.33.205| 503 |   3.027112ms| nomatch| GET
/api/ping
2019/09/19 15:29:01 [E] [server.go:2774] MOVED 938 192.168.64.117:6379
2019/09/19 15:29:01 [D] [server.go:2774] |  10.112.33.205| 503 |   1.920783ms| nomatch| GET
/api/ping
2019/09/19 15:29:01 [E] [server.go:2774] MOVED 6511 192.168.100.31:6379
2019/09/19 15:29:01 [D] [server.go:2774] |  10.112.33.205| 503 |   2.258429ms| nomatch| GET
/api/ping
2019-09-19T15:29:01Z [INFO] [/core/main.go:146]: capture system signal terminated, to close
"closing" channel
2019-09-19T15:29:04Z [INFO] [/core/main.go:152]: Timeout waiting goroutines to exit
```

参考：

- https://github.com/goharbor/harbor/issues/6075
- https://github.com/goharbor/harbor/issues/8620

oidc模式 cli 密码失效

表现为隔一段时间未从浏览器登录harbor，则docker login 会失败，从浏览器登录一次之后 docker login 又能成功了。

怀疑是 oidc provider 未正确实现 refresh_token 功能。

验证及解决方案：调短 dex idtoken 过期时间（比如1分钟） 来验证是否是 refresh_token 的问题，然后尝试修复 refresh_token 问题。

通过实现了 refresh_token 的测试 connector, 设置 idTokens 过期时间为 20s，发现 docker login 20s 后依然可以登录。而使用线上未实现 refresh_token的 connector, 则 20 s 后登录失效。

解决方案: 直接抄 github connector 的 refresh 代码即可。

镜像存储

harbor 1.9.0 s3 不可用，镜像都是 0B□而且只显示第一个 repository

s3 垃圾回收似乎还有问题。见 https://github.com/goharbor/harbor/issues/8121

选择 ceph swift.

swift 镜像存储在 files/ 文件夹下，gc 效果

snippet.bash

```
[root@localhost harbor-deploy]# s3cmd du s3://harbor/
2803912  213 objects s3://harbor/
[root@localhost harbor-deploy]# s3cmd du s3://harbor/
2803699  210 objects s3://harbor/
[root@localhost harbor-deploy]# s3cmd du s3://harbor/
12070   205 objects s3://harbor/
[root@localhost harbor-deploy]# s3cmd ls s3://harbor/files/docker/registry/v2/repositories/library/alpine/
                       DIR   s3://harbor/files/docker/registry/v2/repositories/library/alpine/_layers/
                       DIR   s3://harbor/files/docker/registry/v2/repositories/library/alpine/_uploads/
```

可见确实清理了，但是 alpine 文件夹依然存在

先测试 ceph s3，不行在用 swift。

ceph s3 retry

snippet.bash

```
# docker push cr.play.scloud.cn/test/alpine:3.10
The push refers to repository [cr.play.scloud.cn/test/alpine]
256a7af3acb1: Pushing [==================================================>]
5.844MB
```

```
blob upload unknown
```

参考：

- https://cloud.tencent.com/developer/article/1439688

使用 swift

> 另外多个Registry需要使用共享存储，可选的有Swift、NFS、S3、azure、GCS、Ceph和OSS。我们选择使用Ceph。
> docker-registry在2.4.0版本之后移除了rados storage driver，推荐使用Swift API gateway替代，因为Ceph在rados之上提供了兼容Swift和S3的接口。Harbor是在docker-registry的基础之上做的扩展，我们用的Harbor 1.5.1所使用的registry是2.6.2版本，因此无法配置rados storage，只能使用Swift driver或者S3 driver，我们选择Swift driver。

参考：

- https://www.cnblogs.com/ltxdzh/p/9223566.html
- RadosGW详解

swift 问题

终极解决方案。ceph.conf 设置 rgw swift url = https://s3.com，不用改 Nginx 配置了

用 ingress nginx 给 radosgw 做了代理，指定 https 的 authurl，自动变成了去连 http 的 7480 端口

```
swift --os-cacert /home/deca.pem -R=1 -A https://s3.play.cn/auth -U harbor -K secretkey list
HTTPConnectionPool(host='s3.play.cn', port=7480): Max retries exceeded with url:
/swift/v1?format=json (Caused by NewConnectionError('<urllib3.connection.HTTPConnection object
at 0x7f4c0bfe1b00>: Failed to establish a new connection: [Errno 111] Connection refused',))
```

抓包 (tcpflow -cp -i ens33 port 80) 看到 X-Storage-Url 里指定了错误的 Url

```
GET /auth/v1 HTTP/1.1
Host: s3.play.cn
Accept-Encoding: identity
x-auth-user: harbor
x-auth-key: secret
user-agent: python-swiftclient-3.8.1

HTTP/1.1 204 No Content
Date: Fri, 20 Sep 2019 07:20:20 GMT
Content-Type: application/json; charset=utf-8
Connection: keep-alive
```

```
X-Storage-Url: http://s3.play.cn:7480/swift/v1
X-Storage-Token: AUTH_rgwtk
X-Auth-Token: AUTH_rgwtk
X-Trans-Id: tx000000000000000095c3-005d847db4-2a35-default
X-Openstack-Request-Id: tx000000000000000095c3-005d847db4-2a35-default
```

解决方案： 参考: https://toutiao.io/posts/xpl3rb/preview

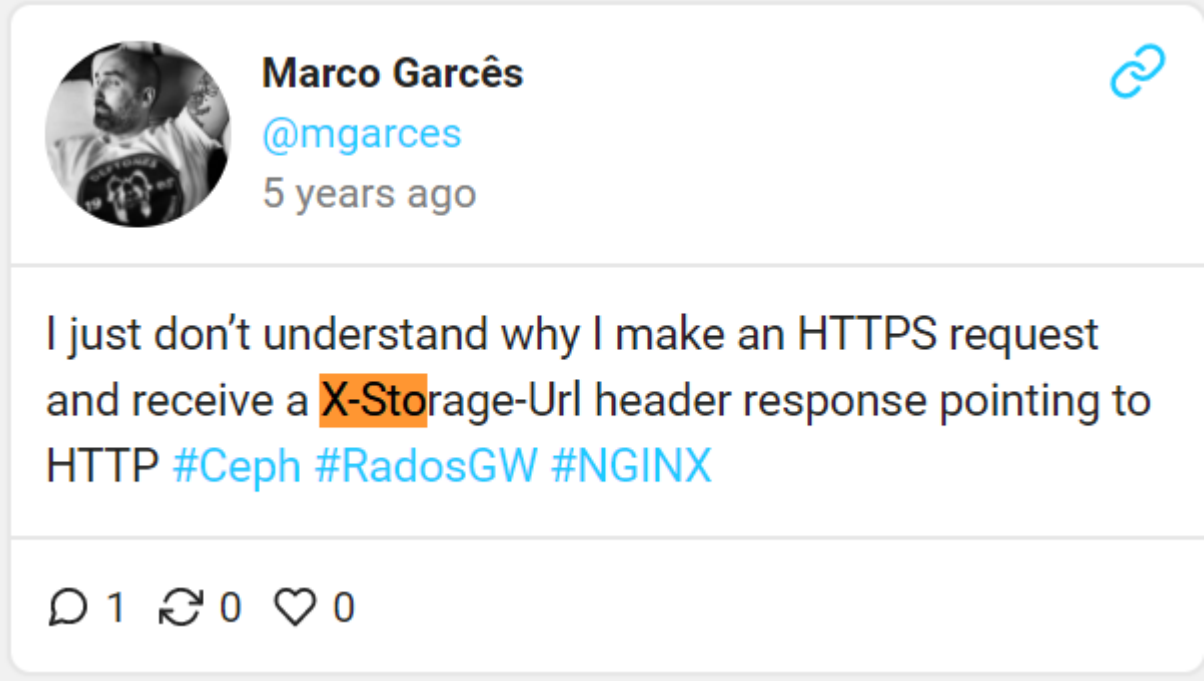
Ingress nginx 设置 annotation

```
nginx.ingress.kubernetes.io/upstream-vhost: "$host:$server_port"
```

对应于 Nginx 配置

```
proxy_set_header Host "$host:$server_port";
```

PS. 这个问题好难搜到。。



端口解决了，协议还是有问题，直接用 curl 测试, X-Storage-Url 还是错的：

```
# curl -i -H "X-Auth-User: test:swift" -H "X-Auth-Key: secret" https://s3.com/auth
HTTP/1.1 204 No Content
Date: Fri, 20 Sep 2019 08:27:15 GMT
Content-Type: application/json; charset=utf-8
Connection: keep-alive
X-Storage-Url: http://s3.com:443/swift/v1
X-Storage-Token: AUTH_rgwtk
X-Auth-Token: AUTH_rgwtk
X-Trans-Id: tx000000000000000010c-005d848d63-2a62-default
X-Openstack-Request-Id: tx000000000000000010c-005d848d63-2a62-default
```

```
Strict-Transport-Security: max-age=15724800; includeSubDomains
```

终极解决方案。ceph.conf 设置 rgw swift url = https://s3.com，不用改 Nginx 配置了

```
[global]
 ...
 rgw swift url = https://s3.com
```

参考：

- http://lists.ceph.com/pipermail/ceph-users-ceph.com/2016-April/009213.html
- http://lists.ceph.com/pipermail/ceph-users-ceph.com/2016-September/013046.html

K8S使用镜像仓库

问题	解决方案	备注
imagepullsecret是否会暴露给用户	未通过 dashboard 暴露给用户	k8s可用全局账号
是否使用harbor管理员用户	不会暴露imagepullsecret，因此可考虑用管理员账号	

为每个 namespace 添加 imagePullSecret

snippet.bash

```
for id in `kubectl get ns |awk 'NR>1{print $1}'`;do kubectl -n $id apply -f cr-key.yaml;done
```

Printed on: 2022/10/05 17:59