采用非Google域名访问GCS的方法

Date: 20191010
Author: Wei Heng/Jun Sheng
Version: 1.0

目前在国内访问Google Cloud Storage，由于采用的是Google域名的缘故，不能直接访问。本文提供3种方法实现采用Customer Domain的方式实现对Google GCS的浏览/上传/下载/删除操作。

1. Domain-name Bucket
2. HTTP(S) Load Balancer挂载Bucket
3. Revers_Proxy方式，代理用户访问GCS请求

1 Domain-name Bucket

Google Cloud Storage在创建Bucket的时候，可以创建一个FQDN的名字，但这个domain，必须是经过Google验证的。具体的方法可以参考下面链接中的步骤：
https://www.cnblogs.com/hengwei/p/9679266.html

A 创建Bucket

通过console或gsutil创建一个domain-name的bucket：

```
gsutil mb gs://mybucket.weiheng.ink
Creating gs://mybucket.weiheng.ink/…
```

在DNS提供商，创建一条CNAME记录：

B 上传文件

GCS的认证有多种方式，本文采用的是JWT的方式，具体文档请参考：
https://cloud.google.com/endpoints/docs/openapi/service-account-authentication#python
这里采用gcloud的方法生成，并用curl发起http请求：

- 激活Service Account

```
gcloud auth activate-service-account \
    whproject@xxxxxx.iam.gserviceaccount.com \
    --key-file=/root/sa.json
```

- 生成token

```
token=`gcloud auth application-default print-access-token`
```

- 上传文件
  通过PUT方法把文件上传到刚刚创建的bucket中：

```
curl -v -X PUT -k -H "Authorization: Bearer $token" \
    --data-binary @cat1.jpg \
    https://mybucket.weiheng.ink/cat1.jpg
```

```
< HTTP/1.1 100 Continue
< HTTP/1.1 200 OK
< X-GUploader-UploadID: AEnB2UoUbsfZ1Q_OQ7usXR5Ks
< ETag: "8c2090502624d1ea4198713088612cbd"
< x-goog-generation: 1570692461153454
< x-goog-metageneration: 1
< x-goog-hash: crc32c=Cin4UA==
< x-goog-hash: md5=jCCQUCYk0epBmHEwiGEsvQ==
< x-goog-stored-content-length: 85936
< x-goog-stored-content-encoding: identity
< Vary: Origin
< Content-Length: 0
< Date: Thu, 10 Oct 2019 07:27:41 GMT
< Server: UploadServer
< Content-Type: text/html; charset=UTF-8
```

可以看到上传成功。

C 下载文件

和上传文件类似，采用GET方法：

- 生成token

```
token=`gcloud auth application-default
print-access-token`
```

- 查看Token状态

```
curl
https://www.googleapis.com/oauth2/v1/tokeninfo?access_toke
n=$token
```

```
[root@nginx-2 ~]# curl https://www.googleapis.com/oauth2/v1/tokeninfo?access_token=$token
{
  "issued_to": "101676042918541988642",
  "audience": "101676042918541988642",
  "scope": "https://www.googleapis.com/auth/cloud-platform",
  "expires_in": 1828,
  "access_type": "offline"
}
```

- 下载文件

```
curl -v -X GET -k \
    -H "Authorization: Bearer $token" -o cat1.jpg \
    https://mybucket.weiheng.ink/cat1.jpg
```

```
< HTTP/1.1 200 OK
< X-GUploader-UploadID: AEnB2Up0KHsv2yp3YL4FhToelI
< Expires: Thu, 10 Oct 2019 07:49:22 GMT
< Date: Thu, 10 Oct 2019 07:49:22 GMT
< Cache-Control: private, max-age=0
< Last-Modified: Thu, 10 Oct 2019 07:27:41 GMT
< ETag: "8c2090502624d1ea4198713088612cbd"
< x-goog-generation: 1570692461153454
< x-goog-metageneration: 1
< x-goog-stored-content-encoding: identity
< x-goog-stored-content-length: 85936
< Content-Type: application/x-www-form-urlencoded
< x-goog-hash: crc32c=Cin4UA==
< x-goog-hash: md5=jCCQUCYk0epBmHEwiGEsvQ==
< x-goog-storage-class: STANDARD
< Accept-Ranges: bytes
< Content-Length: 85936
< Server: UploadServer
```

D 浏览bucket

List Bucket采用的也是GET方法，具体实现如下：

- 生成token

```
token=`gcloud auth application-default
print-access-token`
```

- List Bucket内容，存到文件中：

```
curl -v -X GET -k \
    -H "Authorization: Bearer $token" -o list.txt \
    https://mybucket.weiheng.ink/
```

- 通过awk取出bucket名字

```
cat list.txt | awk -F '</?Name>' '{print $2}'
```

```
[root@nginx-2 ~]# cat list.txt | awk -F '</?Name>' '{print $2}'
mybucket.weiheng.ink
```

- 通过awk取出文件名

```
cat list.txt | awk -F '</?Key>' '{print $2 "\n" $4}'
```

```
[root@nginx-2 ~]# cat list.txt | awk -F '</?Key>' '{print $2 "\n" $4}'
cat1.jpg
cat2.jpg
```

E 删除文件

类似的，用DELETE方法删除：

- 生成token

```
token=`gcloud auth application-default
print-access-token`
```

- 删除文件

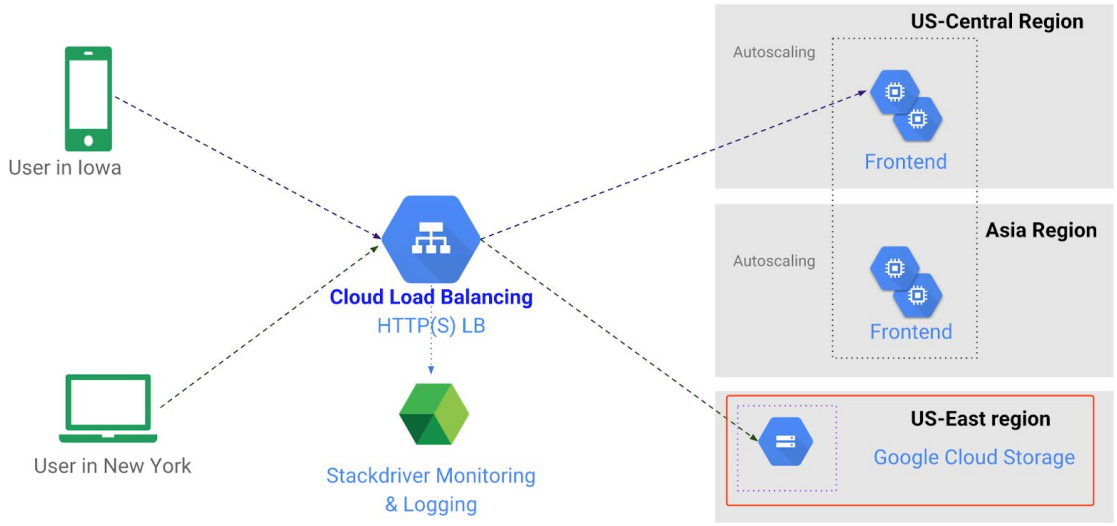
```
curl -v -k -X DELETE \
    -H "Authorization: Bearer $token" \
    https://mybucket.weiheng.ink/cat1.jpg
```

```
< HTTP/1.1 204 No Content
< X-GUploader-UploadID: AEnB2UrZY0dZR-lbCuOe55sjX5d
< Vary: Origin
< Date: Thu, 10 Oct 2019 09:00:16 GMT
< Expires: Thu, 10 Oct 2019 09:00:16 GMT
< Cache-Control: private, max-age=0
< Content-Length: 0
< Server: UploadServer
```

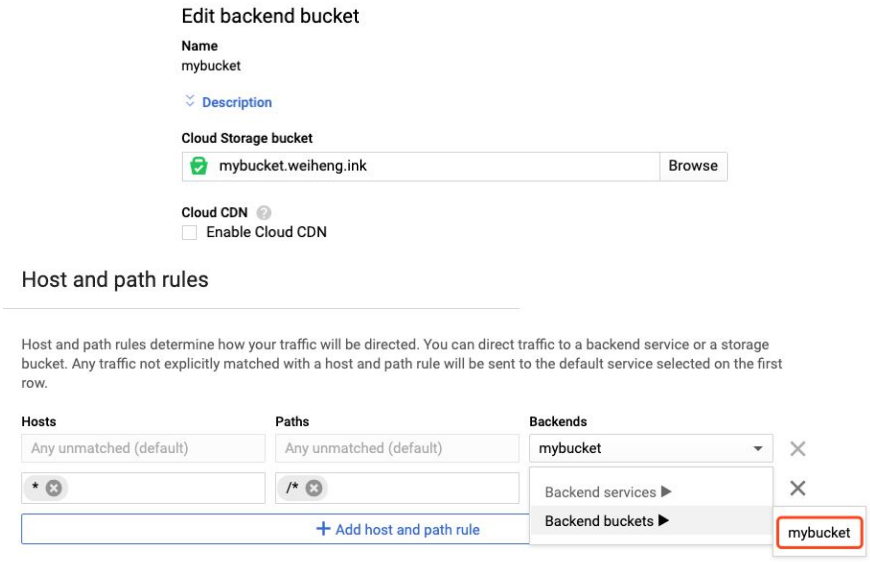
2 HTTP(S) Load Balancer挂载Bucket

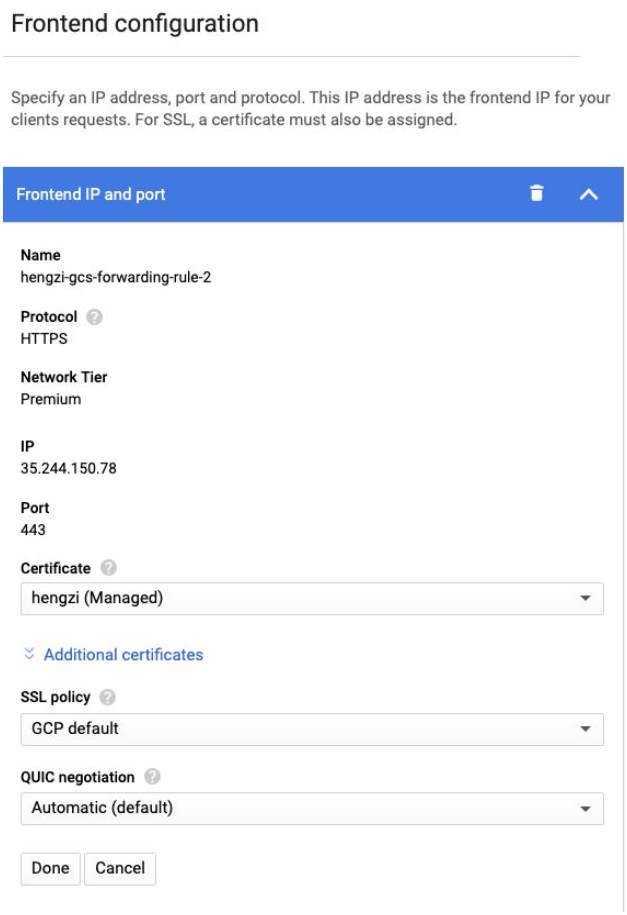
A 创建HTTP(S) LB的backend bucket

Google Cloud的HTTP(S)负载均衡可以配置backend的bucket。如下图：



具体文档请参考：
https://cloud.google.com/load-balancing/docs/backend-bucket
https://cloud.google.com/load-balancing/docs/https/adding-backend-buckets-to-load-balancers
根据文档，已经创建了一个负载均衡，配置了gs://mybucket.weiheng.ink作为buckend bucket：





另外配置了DNS的A记录，storage.hengzi.net.cn指向35.224.150.78。

B 上传文件

和Domain-name Bucket中上传文件的类似，通过PUT方法上传，但endpoint采用的是LB的域名。具体方法如下：

- 生成token

```
token=`gcloud auth application-default print-access-token`
```

- 上传

```
curl -v -X PUT \
    -H "Authorization: Bearer $token" --data @cat1.jpg \
    https://storage.hengzi.net.cn/cat1.jpg
```

C 下载文件

类似的：

- 生成token

```
token=`gcloud auth application-default
print-access-token`
```

- 下载

```
curl -v -X GET -H "Authorization: Bearer $token" \
    -o cat3.jpg https://storage.hengzi.net.cn/cat1.jpg
```

-

D 浏览Bucket

- 生成token

```
token=`gcloud auth application-default
print-access-token`
```

- 浏览Bucket

```
curl -v -X GET \
    -H "Authorization: Bearer $token" \
    -o lb_list.txt https://storage.hengzi.net.cn/
```

E 删除文件

类似的，用DELETE方法删除：

- 生成token

```
token=`gcloud auth application-default print-access-token`
```

- 删除文件

```
curl -v -X DELETE -H "Authorization: Bearer $token"
https://storage.hengzi.net.cn/cat1.jpg
```

F 断点续传

这种模式下支持断点续传功能。具体实现为：

- 生成token

```
token=`gcloud auth application-default print-access-token`
```

- 生成断点续传连接

```
curl -v -X POST -H 'content-type: text/plain' \
    -H 'x-goog-resumable:start' \
    -H "Authorization: Bearer $token" \
    -d '' 'https://storage.hengzi.net.cn/data.txt'
```

- 获取location信息
  返回值中有Location的字段，将其取出作为loc变量：

```
< Location:
https://storage.hengzi.net.cn/data.txt?upload_id=AEnB2UqelEDznBtE8n8HWRf12wDalTbqhD0-BqJvtChRCuLTo2PjiJ3WjkWc8dsYbizAFNLxN8CRv8uFTGyzYQPFpNlD2OfysFWSfZjEuuHbXHagLjc5ENI

loc='https://storage.hengzi.net.cn/data.txt?upload_id=AEnB2UqelEDznBtE8n8HWRf12wDalTbqhD0-BqJvtChRCuLTo2PjiJ3WjkWc8dsYbizAFNLxN8CRv8uFTGyzYQPFpNlD2OfysFWSfZjEuuHbXHagLjc5ENI'
```

- 上传文件

```
curl -v -X PUT --data @data.txt $loc
```

3 Revers_Proxy方式，代理用户访问GCS请求

在GCP上创建Nginx作为反向代理，访问GCS。这样用户可以不需要直接和Google的endpoint通讯。

A 安装配置Nginx

本文采用的是CentOS7的VM，只有用yum安装nginx：

```
yum install nginx -y
```

1 JSON API配置

由于Google GCS有两个REST API的endpoint，如果我们选择Json的endpoint，Nginx的配置如下：

配置Nginx作为反向代理：

```
vim /etc/nginx/nginx.conf
        location / {
          proxy_pass https://www.googleapis.com/;
        }
```

```
include /etc/nginx/conf.d/*.conf;

server {
    listen       80 default_server;
    listen       [::]:80 default_server;
    server_name  _;
    #root         /usr/share/nginx/html;

    # Load configuration files for the default
    include /etc/nginx/default.d/*.conf;

    location / {
      proxy_pass https://www.googleapis.com/;
    }

    error_page 404 /404.html;
        location = /40x.html {
    }

    error_page 500 502 503 504 /50x.html;
        location = /50x.html {
    }
}
```

2 XML API配置

XML的API的反向代理配置如下：

```
vim /etc/nginx/nginx.conf
        location / {
          proxy_pass https://storage.googleapis.com/;
        }
```

```
include /etc/nginx/conf.d/*.conf;

server {
    listen       80 default_server;
    listen       [::]:80 default_server;
    server_name  _;
    #root         /usr/share/nginx/html;

    # Load configuration files for the default server block.
    include /etc/nginx/default.d/*.conf;

    location / {
      proxy_pass https://storage.googleapis.com/;
    }

    error_page 404 /404.html;
        location = /40x.html {
    }

    error_page 500 502 503 504 /50x.html;
        location = /50x.html {
    }
}
```

3 认证包头配置

在对GCS进行操作的时候，需要在包头中配置认证信息。如果不希望在客户端不考虑认证相关信息，可以添加这个配置。
配置添加HTTP的Header，将JWT的token插入到HTTP包头中：

```
vim /etc/nginx/default.d/token.conf
     proxy_set_header Authorization 'Bearer ya29.c.Kmxxxxxxxxxx'
```

通过gcloud命令生成的token，一小时就过期，因此我们创建一个任务计划，每半小时重新生成一次:

```
crontab -e
     30 * * * * /root/update_token.sh
```

更新token脚本：

```
cat /root/update_token.sh
     token=`gcloud auth application-default print-access-token`
     echo "proxy_set_header Authorization 'Bearer $token';" >
     /etc/nginx/default.d/token.conf
     systemctl restart nginx
```

B 浏览Bucket

1 JSON API

```
curl -v -X GET   http://json.hengzi.net.cn/storage/v1/b/hengzi/o
```

2 XML API

```
curl -v -X GET http://xml.hengzi.net.cn/hengzi
```

C 上传文件

1 JSON API

```
curl -X POST --data-binary @a.txt -H "Content-Type: text/plain"
"https://json.hengzi.net.cn/upload/storage/v1/b/hengzi/o?uploadType
=media&name=a.txt"
```

2 XML API

```
curl -v -X PUT --data-binary @b.sh \
    -H "Content-Type: text/plain" \
    "http://xml.hengzi.net.cn/hengzi/b.sh"
```

D 下载文件

1 JSON API

```
curl -v -X GET -o "cat2.jpg" \
"http://json.hengzi.net.cn/storage/v1/b/hengzi/o/cat2.jpg?alt=media
"
```

2 XML API

```
curl -v -X GET -o "cat1.jpg" \
  "http://xml.hengzi.net.cn/hengzi/cat2.jpg"
```

E 删除文件

1 JSON API

```
curl -v -X DELETE \
"http://json.hengzi.net.cn/storage/v1/b/hengzi/o/a.txt"
```

2 XML API

```
curl -v -X DELETE \
"http://xml.hengzi.net.cn/hengzi/a.sh"
```

4 总结

通过以上三种方法，可以实现采用非Google域名访问Google存储。同样的，类似的方法可以用在Google的其他API上。