

用Cloud Functions同步谷歌云存储桶

本文介绍如何用谷歌云Cloud Functions来实时同步两个谷歌云存储桶，实现当源桶里有文件添加或删除时，目的桶里也自动进行对应的添加或删除。

在进行下面操作之前，先创建好用于同步的源桶和目的桶。本文示例用到了下面两个桶。

源桶：gs://youzhi-lab

目的桶：gs://youzhi-lab-bak

[一、创建用于同步文件添加的Cloud Functions函数](#)

[二、创建用于同步文件删除的Cloud Functions函数](#)

[三、查看函数状态并测试功能](#)

一、创建用于同步文件添加的Cloud Functions函数

首先在谷歌云控制台创建一个Cloud Functions函数。参照下面截图填写相应配置。其中“Bucket”选择源桶。“Event Type”选择“Finalize/Create”。

Google Cloud Platform

youzhi-lab

Cloud Functions

Create function

Name

sync_bucket

Memory allocated

256 MB

Trigger

Cloud Storage

Event Type

Finalize/Create

Bucket

youzhi-lab

Browse

Source code

☒ Inline editor

☐ ZIP upload

☐ ZIP from Cloud Storage

☐ Cloud Source repository

Runtime

Python 3.7

因为上面“Source code”选择了“Inline editor”，所以可以用内建的网页代码编辑器输入代码。将示例代码填入“main.py”标签的代码输入框。

main.py requirements.txt

```
1 from google.cloud import storage
2 import os
3
4 BUCKET_DEST = os.environ['BUCKET_DEST']
5
6 def copy_object(event, context):
7
8     bucket_name = event['bucket']
9     object_name_src = event['name']
10    object_name_dest = object_name_src
11
12    storage_client = storage.Client()
13    source_bucket = storage_client.bucket(bucket_name)
14    source_blob = source_bucket.blob(object_name_src)
15    destination_bucket = storage_client.bucket(BUCKET_DEST)
16
17    blob_copy = source_bucket.copy_blob(
18        source_blob, destination_bucket, object_name_dest
19    )
20
21    print(
22        "Blob {} in bucket {} copied to blob {} in bucket
23        source_blob.name,
24        source_bucket.name,
25
```

本例的代码可以在Github上查看。地址是

https://github.com/eugeneyu/cloud-demos/blob/master/cloud-functions/bucket_to_bucket.py

也可以从下面拷贝。

```
from google.cloud import storage
from google.cloud.exceptions import NotFound
import os

BUCKET_DEST = os.environ['BUCKET_DEST']

storage_client = storage.Client()

def copy_object(event, context):

    bucket_name = event['bucket']
    object_name_src = event['name']
    object_name_dest = object_name_src

    source_bucket = storage_client.bucket(bucket_name)
    source_blob = source_bucket.blob(object_name_src)
    destination_bucket = storage_client.bucket(BUCKET_DEST)
    destination_blob = destination_bucket.blob(object_name_dest)

    (token, bytes_rewritten, total_bytes) = destination_blob.rewrite(source_blob)
```

```

print(
    "Blob {} in bucket {} copied to blob {} in bucket {}".format(
        source_blob.name,
        source_bucket.name,
        destination_blob.name,
        destination_bucket.name,
    )
)

def delete_object(event, context):
    bucket_name = event['bucket']
    object_name_src = event['name']

    destination_bucket = storage_client.bucket(BUCKET_DEST)

    try:
        destination_bucket.delete_blob(object_name_src)
    except NotFound:
        print("Sync deletion of object {} from bucket {} to bucket {} Not Found".format(
            object_name_src,
            bucket_name,
            destination_bucket.name,
        ))

```

然后在“requirements.txt”标签输入框填写一行：

```
google-cloud-storage
```

main.py requirements.txt

```

1 # Function dependencies, for example:
2 # package>=version
3 google-cloud-storage













```

上面代码中使用blob的rewrite方法而不是bucket的blob_copy方法来复制对象，是因为如果源桶和目的桶不在一个区域，使用rewrite方法会将大文件使用多个请求分片传输，减少超时错误。如果是在同一个区域内的桶之间复制，文件也都不大，比如10MB以下，也可以使用blob_copy。具体可以参考以下文档。

https://cloud.google.com/storage/docs/json_api/v1/objects/rewrite


如果使用blob_copy跨区域传输大文件，可能会观察到下面第一行的413错误。

Errors in the last day

Resolution Status	Occurrences	Error	Seen in
 Open		3,029 GoogleAPICallError: 413 POST https://storage.googleapis.com/storage/v1/b/[redacted]/api_request (/env/local/lib/python3.7/site-packages/google/cloud/_http.py)	
 Open		40 ServiceUnavailable: 503 POST https://storage.googleapis.com/storage/v1/b:[redacted]/api_request (/env/local/lib/python3.7/site-packages/google/cloud/_http.py)	
 Open		4 ConnectionError: ('Connection aborted.', ConnectionResetError(104, 'Connection reset by peer' send (/env/local/lib/python3.7/site-packages/requests/adapters.py)	
 Open		3 NEW GatewayTimeout: 504 POST https://storage.googleapis.com/storage/v1/b/[redacted]/api_request (/env/local/lib/python3.7/site-packages/google/cloud/_http.py)	


在下面的配置项继续填写相应参数配置，包括高级配置选项。

- **Function to execute** : copy_object
- **Region** : 填写目的桶或源桶所在的区域。让函数执行与其接近，减少网络延时和开销。
- **Service account** : 可以使用默认的App Engine账号，或者改为其它账号比如Compute Engine账号等。需要确保账号在IAM配置里拥有对源桶的可读和对目的桶的可写权限。


Function to execute 

copy_object

Advanced options


Region 


asia-east2

Timeout 


60

seconds


Maximum function instances 

Service account 

App Engine default service account

☐ Retry on failure 

Networking

Ingress settings 

☒ Allow all traffic

☐ Allow internal traffic only

Only traffic from within the same project or the same VPC SC perimeter is allowed.

Egress settings 

By default, your function can send requests to the internet, but not to resources in VPC networks. To send requests to resources in your VPC network, create or select a VPC connector.

VPC connector

[+ Create a connector](#)


☒ Route only requests to private IPs through the VPC connector

☐ Route all traffic through the VPC connector

在最下面的环境变量配置中创建新的换件变量指向目的桶。

- Name : BUCKET_DEST
- Value : 目的桶名

Environment

Environment variables 

Name

Value

BUCKET_DEST

youzhi-lab-bak

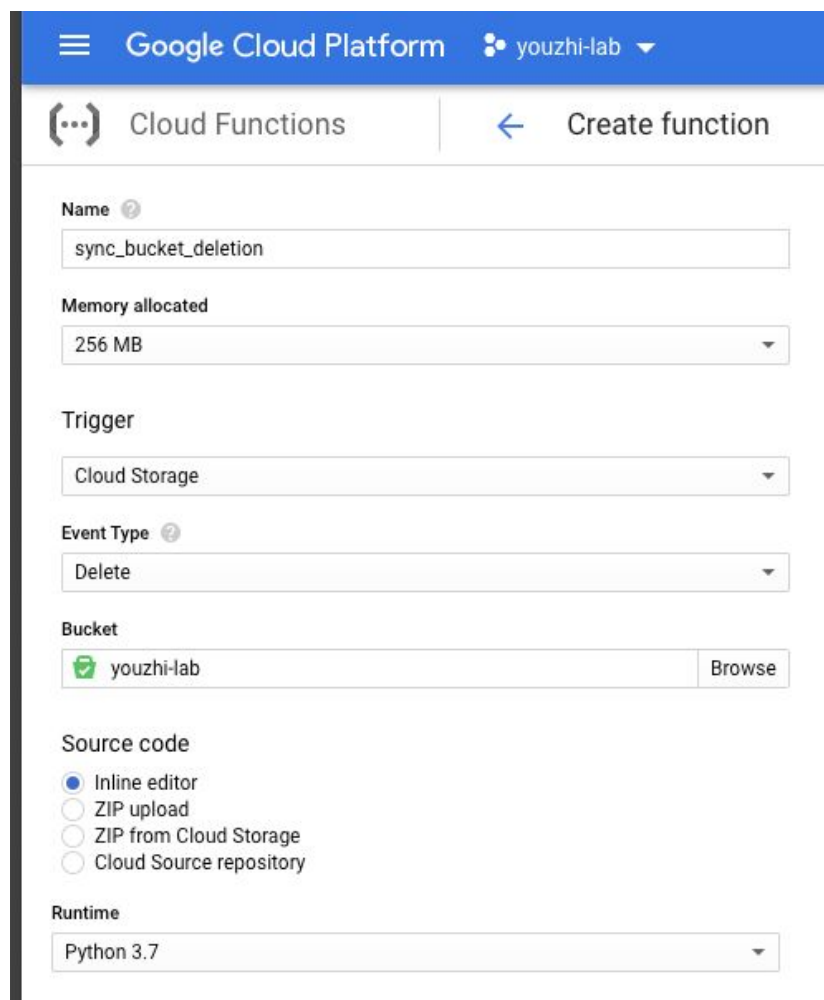
✕

[+ Add variable](#)

点击创建按钮。等待函数创建和部署完成。

二、创建用于同步文件删除的Cloud Functions函数

在谷歌云控制台创建一个Cloud Functions函数。参照下面截图填写相应配置。其中“Bucket”选择源桶。“Event Type”选择“Delete”。



The screenshot shows the Google Cloud Platform console interface for creating a new Cloud Function. The top navigation bar includes the Google Cloud Platform logo and the user account 'youzhi-lab'. The main heading is 'Cloud Functions' with a 'Create function' button. The configuration form includes the following fields:

- Name:** A text input field containing 'sync_bucket_deletion'.
- Memory allocated:** A dropdown menu set to '256 MB'.
- Trigger:** A dropdown menu set to 'Cloud Storage'.
- Event Type:** A dropdown menu set to 'Delete'.
- Bucket:** A text input field containing 'youzhi-lab' with a 'Browse' button next to it.
- Source code:** A section with four radio button options: 'Inline editor' (selected), 'ZIP upload', 'ZIP from Cloud Storage', and 'Cloud Source repository'.
- Runtime:** A dropdown menu set to 'Python 3.7'.

在代码框输入与第一节相同的代码。不过在删除时我们会调用与新建文件不同的函数来同步。

main.py requirements.txt

```
1 from google.cloud import storage
2 import os
3
4 BUCKET_DEST = os.environ['BUCKET_DEST']
5
6 def copy_object(event, context):
7
8     bucket_name = event['bucket']
9     object_name_src = event['name']
10    object_name_dest = object_name_src
11
12    storage_client = storage.Client()
13    source_bucket = storage_client.bucket(bucket_name)
14    source_blob = source_bucket.blob(object_name_src)
15    destination_bucket = storage_client.bucket(BUCKET_DEST)
16
17    blob_copy = source_bucket.copy_blob(
18        source_blob, destination_bucket, object_name_dest
19    )
20
21    print(
22        "Blob {} in bucket {} copied to blob {} in bucket
23        source_blob.name,
24        source_bucket.name,
25
```


另外也要再在requirements.txt框中填写

google-cloud-storage

main.py requirements.txt

```
1 # Function dependencies, for example:
2 # package>=version
3 google-cloud-storage
```

其他参数参照下面截图配置。注意“Function to execute”要填写“delete_object”。


Function to execute 

delete_object

Advanced options


Region 


asia-east2

Timeout 


60

seconds


Maximum function instances 

Service account 

App Engine default service account

☐ Retry on failure 


Networking

Ingress settings 

☒ Allow all traffic

☐ Allow internal traffic only

Only traffic from within the same project or the same VPC SC perimeter is allowed.

Egress settings 

By default, your function can send requests to the internet, but not to resources in VPC networks. To send requests to resources in your VPC network, create or select a VPC connector.

VPC connector


[+ Create a connector](#)

☒ Route only requests to private IPs through the VPC connector

☐ Route all traffic through the VPC connector

另外也要像之前一样创建环境变量指向目标桶。

Environment

Environment variables 

Name

Value

BUCKET_DEST

youzhi-lab-bak

✕

[+ Add variable](#)

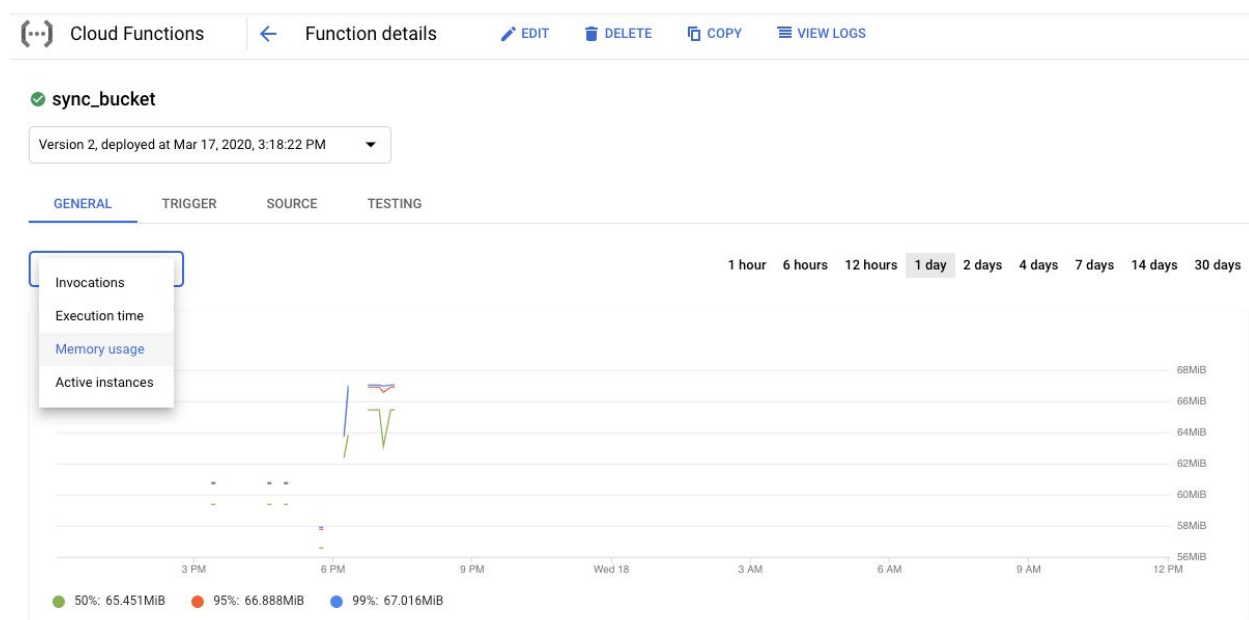
配置完后点创建并等待部署完成。

三、查看函数状态并测试功能

上面两个函数都部署完成后，可以在Cloud Functions控制台查看其状态，以及详情。

Google Cloud Platform youzhi-lab									
Cloud Functions Functions CREATE FUNCTION REFRESH DELETE COPY									
sync Filter functions									
<input type="checkbox"/>	<input checked="" type="radio"/>	Name ↑	Region	Trigger	Runtime	Memory allocated	Executed function	Last deployed	Actions
<input type="checkbox"/>	<input checked="" type="radio"/>	sync_bucket	asia-east2	Bucket: youzhi-lab	Python 3.7	256 MiB	copy_object	Mar 17, 2020, 3:18:22 PM	⋮
<input type="checkbox"/>	<input checked="" type="radio"/>	sync_bucket_deletion	asia-east2	Bucket: youzhi-lab	Python 3.7	256 MiB	delete_object	Mar 17, 2020, 7:16:17 PM	⋮

在详情中，可以看到函数的调用次数、用时、内存消耗和实例资源等统计信息。



测试函数功能可以在源桶上传一个文件，然后在目标桶查看相同文件的自动创建。之后在源桶删除一个文件，然后在目标桶查看同名文件被删除。此外，文件夹及其下文件的创建也会被同步。

要注意的是，本例的代码在同步文件时，不会复制源文件的metadata。如果需要复制metadata可以在上面代码上做相应修改再部署。