

# 用Cloud Functions压缩GCS中的日志文件

在Cloud Functions控制台界面，创建新的Function。

The screenshot shows the 'Create function' interface in the Google Cloud Functions console. At the top, there is a breadcrumb 'Cloud Functions' and a 'Create function' button. Below this, the form is divided into several sections: 'Name' with a text input containing 'compress\_file'; 'Memory allocated' with a dropdown menu set to '256 MB'; 'Trigger' with a dropdown menu set to 'Cloud Storage'; 'Event Type' with a dropdown menu set to 'Finalize/Create'; and 'Bucket' with a text input containing 'oppo-cdn-log-uncompressed' and a 'Browse' button.

- Trigger选择Cloud Storage
- Event Type选择Finalize/Create
- Bucket选择未压缩日志文件所存放的存储桶

按如下内容填写Function的代码。

## Source code

- ☒ Inline editor
- ☐ ZIP upload
- ☐ ZIP from Cloud Storage
- ☐ Cloud Source repository

## Runtime

Python 3.7

main.py requirements.txt

```
1 from google.cloud import storage
2 import os
3 import tarfile
4 from shutil import copyfile
5
6 storage_client = storage.Client()
7 BUCKET_PREFIX = "oppo-cdn-log-compressed"
8 TMP_FOLDER = "/tmp"
9
10 def compress_file(event, context):
11     """Triggered by a change to a Cloud Storage bucket.
12     Args:
13         event (dict): Event payload.
14         context (google.cloud.functions.Context): Metadata
15     """
16     file = event
17     print(f"Processing file: {file['name']}.")
18     print(f"Bucket: {file['bucket']}.")
19
20     tmp_folder = "/tmp"
21
22     bucket_name = file['bucket']
23     bucket_src_filename = file['name']
24
25
```

## Function to execute ?

compress\_file

其中，main.py的内容为如下代码。注意将BUCKET\_PREFIX的值改为你创建的压缩日志存放桶的名称。

```
from google.cloud import storage
import os
import tarfile
from shutil import copyfile

storage_client = storage.Client()
BUCKET_PREFIX = "cdn-log-compressed"
TMP_FOLDER = "/tmp"
```

```

def compress_file(event, context):
    """Triggered by a change to a Cloud Storage bucket.
    Args:
        event (dict): Event payload.
        context (google.cloud.functions.Context): Metadata for the event.
    """
    file = event
    print(f"Processing file: {file['name']}.")
    print(f"Bucket: {file['bucket']}.")

    tmp_folder = "/tmp"

    bucket_name = file['bucket']
    bucket_src_filename = file['name']

    if (bucket_src_filename.startswith(BUCKET_PREFIX)):
        return

    filename_w_ext = os.path.basename(bucket_src_filename)
    filename, file_extension = os.path.splitext(filename_w_ext)
    local_src_file_name = TMP_FOLDER + "/" + filename_w_ext

    local_dest_file_name = TMP_FOLDER + "/" + filename + ".tar.gz"
    bucket_dest_filename = BUCKET_PREFIX + "/" + filename + ".tar.gz"

    bucket = storage_client.get_bucket(bucket_name)
    blob = bucket.blob(bucket_src_filename)
    blob.download_to_filename(local_src_file_name)

    #copyfile(local_src_file_name, local_dest_file_name)
    make_tarfile(local_dest_file_name, local_src_file_name)

    blob = bucket.blob(bucket_dest_filename)
    blob.upload_from_filename(local_dest_file_name)

    os.remove(local_src_file_name)
    os.remove(local_dest_file_name)

def make_tarfile(output_filename, source_dir):
    with tarfile.open(output_filename, "w:gz") as tar:

```

```
tar.add(source_dir, arcname=os.path.basename(source_dir))
```

在requirements.txt中加上google-cloud-storage。

main.py requirements.txt

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

点击代码框下方的“more”，在Advanced options中选择合适的区域。

#### Advanced options

Region ?

asia-east2

点击创建。