

PROCEDIMIENTO DE CREACIÓN DE CLOUD FUNCTIONS

Creado: 5 JUL 2023

Versión: 1.0

Autor: DFB


1 INTRODUCCIÓN

El presente documento tiene como propósito describir el proceso de creación de una Cloud Function ejecutada por un evento HTTP dentro del entorno de la plataforma Google Cloud.

2 CREACIÓN DE UN REPOSITORIO EN GOOGLE CLOUD

La creación de un repositorio en Google Cloud tiene el mismo mecanismo que el utilizado con Git Hub. De hecho, Google Cloud tiene embebido en su estructura a Git.

Los pasos son los siguientes:

- a) Crear un directorio en la computadora local que contenga los archivos y carpetas asociados a la función que se desea implementar.
- b) Posicionados en dentro del directorio, abrir una terminal Bash y ejecutar el comando '**git init**'.
- c) Crear el archivo **main.py** dentro del directorio con el código de la función deseada.
- d) Acceder a la terminal Cloud Shell desde el símbolo  ubicado en la parte superior derecha de la consola Cloud.
- e) En la terminal Cloud Shell, ejecutar el siguiente comando:
gcloud source repos create NOMBRE_REPOSITORIO
- f) Para obtener la URL del nuevo repositorio ejecutar el siguiente comando:
gcloud source repos describe NOMBRE_REPOSITORIO --format="value(url)"
- g) Gcloud devolverá una url con la siguiente estructura:
https://source.developers.google.com/p/terra-safe-391718/r/NOMBRE_REPOSITORIO
- h) Ahora vamos a vincular el repositorio Cloud creado con nuestro repositorio Git local:
git remote add origin [URL_DEL_REPOSITORIO_CLOUD]
En este caso, el comando sería:

git remote add origin https://source.developers.google.com/p/terra-safe-391718/r/NOMBRE_REPOSITORIO

- i) En la terminal Bash, ejecutar los siguientes comandos:
 - i) **Git add .**
 - ii) **Git commit -m "Mi primera Cloud Function"**
 - iii) **Git push origin master**
- j) Esta acción copiará la estructura del repositorio local al repositorio Cloud. Ingresando al menú ubicado a la izquierda de la consola Cloud, Sección CI/CD, "Source Repositories" se podrá ver el nuevo repositorio y su contenido.
- k) Ahora, vamos a clonar este repositorio en el directorio raíz de Cloud Shell, ejecutando el siguiente comando:
gcloud source repos clone NOMBRE_REPOSITORIO ~/NOMBRE_DIRECTORIO

Si se omite la opción NOMBRE_DIRECTORIO, el directorio será creado con el mismo nombre de repositorio Cloud y se lo ubicará en la misma carpeta desde la que se ejecutó el comando.

3 CREACIÓN DE LA CLOUD FUNCTION

- a) Para crear la Cloud Function, hay que estar posicionado dentro del directorio recién creado donde se encuentra el archivo main.py y ejecutar el siguiente comando:

gcloud functions deploy 'FUNCTION_NAME' --runtime python310 --trigger-http --allow-unauthenticated --entry-point hello_http --source . --gen2

El significado de las opciones es:

- i) **'FUNCTION_NAME'** es el nombre por el que la función va a ser identificada en Cloud. Admite caracteres alfanuméricos y especiales con excepción del espacio en blanco.

Ingresando al menú de la izquierda, opción "Cloud Functions" aparecerá listada luego del despliegue.
- ii) **--runtime** especifica la versión de compilador a utilizar durante la ejecución.
- iii) **--trigger-http** especifica el tipo de evento que va a disparar la ejecución de la función. En este caso, un GET HTTPS será el evento esperado.
- iv) **--allow-unauthenticated** especifica que no se solicitarán credenciales para el acceso.

- v) **--entry-point hello_http** especifica el nombre de la función que será invocada para procesar el evento. Dicha función debe estar contenida en main.py. En este caso, el nombre de la función a ser invocada es hello_http.
- vi) **--source .** especifica que main.py estará ubicado en el mismo directorio en el que se está ejecutando el comando gcloud functions deploy.
- vii) **--gen2** especifica que la función a crear será generación 2.

b) Al finalizar la ejecución del comando, el mismo mostrará la url que debe ser llamada desde un web browser para disparar el evento. El formato será similar al siguiente:

https://us-central1-terra-safe-391718.cloudfunctions.net/FUNCTION_NAME

c) El contenido de main.py es el siguiente:

```
import json
import functions_framework

@functions_framework.http

def hello_http(request):
    # Obtener los parámetros de la solicitud HTTP
    name = request.args.get('name', 'World')

    # Construir la respuesta JSON
    message = {'message': f'Hello, {name}!'}

    # Devolver la respuesta como JSON
    return json.dumps(message), 200, {'Content-Type': 'application/json'}
```

Recordar instalar las librerías json y functions_framework en el ambiente Cloud Shell.

