Cloud functions

Big Data and Cloud Computing (CC4093)

Eduardo R. B. Marques, DCC/FCUP

Introduction

Cloud Functions

We'll go through an overview of Google Cloud Functions (GCF) for event-driven computation.

[Two sample function examples available here ...]

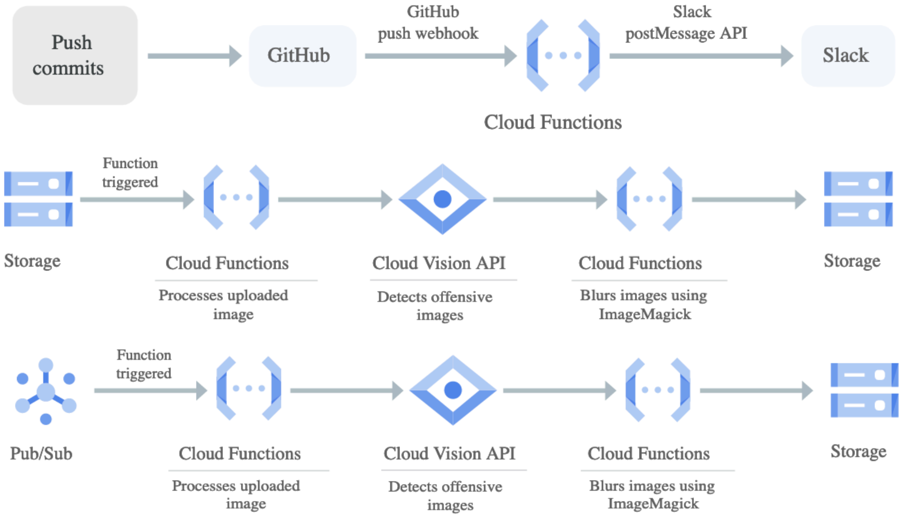
Cloud functions

Cloud functions designate cloud computation that:

- runs for a short time in event-driven manner;
- maintains no internal state, typically it reacts to the event by deriving outputs;
- is billed on per-invocation basis and function running time;
- computational resources are provisioned and scaled on-the-fly - serverless computation

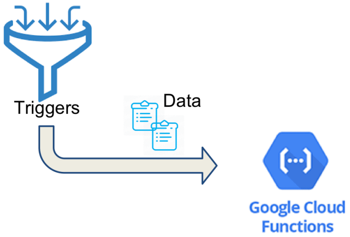
There are several **FaaS (Function-as-a-Service)** offerings like AWS Lambda, Google Cloud Functions (GCF), and Azure Functions.

Example use-cases



Examples from GCF Use Cases.

GCF triggers



GCF triggers include:

- HTTP requests - an HTTP trigger enables a function to run in response to HTTP(S) requests (as in the previous example)
- Google Cloud storage events: a Cloud Storage trigger enables a function to be called in response to changes in Cloud Storage.
- Pub/Sub events: a Pub/Sub trigger enables a function to be called in response to a Pub/Sub messages.

2nd generation GCF supports events generated by various cloud services $\rightarrow$ see here

GCF - a simple example

```python
# The code of a GCF triggered by HTTP
import functions_framework
@functions_framework.http
def hello_via_http(request):
    if request.args and 'name' in request.args:
        name = request.args['name']
    else:
        name = 'World'
    return 'Hello {}!'.format(name)
```

Code: a variety of languages can be used (Javascript, Python, Go, Java, ...).

Stateless computation: the memory state set by a previous invocation cannot be reused. To maintain state across invocations storage services (e.g. GCS, databases) must be used.

Runtime environment: container image built automatically include base runtime, source code, package dependencies, and auxiliary files.

GCF example (cont.)

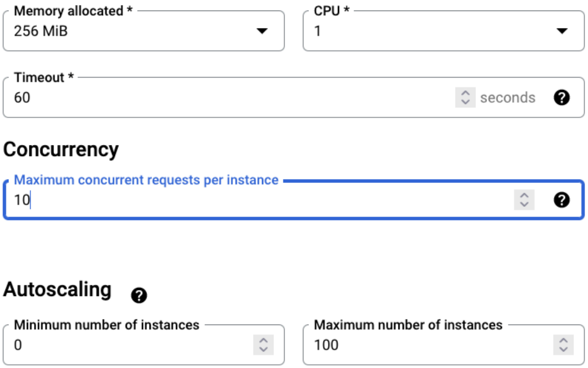
Deployment using `gcloud`

```
$ gcloud functions deploy hello-function \
  --gen2 --runtime=python312 --region=us-central1 \
  --source=. --entry-point=hello_via_http --trigger-http \
  --allow-unauthenticated
Preparing function...done.
Updating function (may take a while)...
...
url: https://us-central1-bdcc202324.cloudfunctions.net/hello-f
```

Example invocation

```
$ curl \
https://us-central1-bdcc202324.cloudfunctions.net/hello-functi
Hello Eduardo!
```

GCF - serverless operation



Memory allocated *	256 MiB
CPU *	1
Timeout *	60 seconds

Concurrency

Maximum concurrent requests per instance	10

Autoscaling

Minimum number of instances	0
Maximum number of instances	100

Containers called **function instances** are provisioned and scaled on-the-fly through Cloud Run. Each instance has an associated CPU/RAM configuration and may serve up to a maximum number of concurrent requests. New instances are created / destroyed according to the volume of requests. You may define a minimum and a maximum number of instances.

GCF billing

GCF invocations are billed according to:

- how many times they are invoked;
- how long functions run (time);
- resources (CPU + RAM) that provisioned for function instances: and
- outbound data transfer (price per GB).

GCF and PubSub

```
import functions_framework
import google.cloud.storage as gcs
...
@functions_framework.cloud_event
def subscribe(event):
    url_to_fetch = base64.b64decode(
      event.data["message"]["data"]).decode("utf-8")
    data = download(url_to_fetch)
    object_name = ...
    gcs_client = gcs.Client()
    gcs_bucket = gcs.Bucket(client=gcs_client,
                            name=BUCKET_NAME)
    gcs_blob = gcs.Blob(object_name, bucket=gcs_bucket)
    gcs_blob.upload_from_string(data)
```

Function consumes a PubSub event indicating a file stored online. It fetches the corresponding file and stores it into a bucket.

GCF and PubSub (cont.)

Topic creation (only done once)

(topic name in this case: `PROCESS_URL`)

```
$ gcloud pubsub topics create PROCESS_URL
Created topic [projects/bdcc202324/topics/handle_image].
```

Function deployment

```
$ gcloud functions deploy process_url_function \
 --entry-point=subscribe --trigger-topic=PROCESS_URL \
   ...
```

GCF and PubSub (cont.)

Event triggering

```
gcloud pubsub topics publish PROCESS_URL \
  --message="https://SomeSite.com/SomePhoto.jpg"
```

Function output / logging

The output of functions triggered by PubSub events is written to a cloud log.

```
$ gcloud functions logs read
LEVEL:
NAME: process-url-function
EXECUTION_ID:
TIME_UTC: 2024-03-05 10:03:20.091
LOG: ...
...
```