« 7.2 Beyond the cloud (/css/2022/serverless/exercise-edge-fo...

CS-E4190 (/css/2022/) / 7. Serverless, edge and fog (/css/2022/serverless/) / 7.3 Assignment: Deploy and run serverless functions

Assignment: Deploy and run serverless functions

This assignment introduces the concepts of serverless functions through hands-on tasks involving different types of triggers. The goal is to deploy functions on Google Cloud Platform (GCP) that respond to HTTP, Cloud Storage and Pub/Sub events.

Warning

The activities in this course are individual work. **Do not read or copy solutions from other students**. **Do not share solutions**. Remember that episodes of plagiarism and collusion are fraudulent means in studying according to the Aalto University Code of Academic Integrity (https://into.aalto.fi/display/ensaannot/Aalto+University+Code+of+Academic+Integrity+and+Handling+Violations+Thereof) which may result in caution or suspension. See also the code of conduct (https://mycourses.aalto.fi/mod/page/view.php?id=916404) of the course.

Tip

Carefully read the related instructions (https://mycourses.aalto.fi/mod/page/view.php?id=916406) before submitting the assignment(s).

Overview

Your task is to create three separate Google Cloud Functions that achieve simple tasks. For each task, you must submit the code for the function as well as the Google Cloud Project ID.

This course has already ended.

Getting started

First, create a Google Cloud Platform project with billing enabled. You can use the billing account to which the course credits were redeemed in module 1.

Note

There is no separate coupon for this module. If you haven't already redeemed the credits for this course, please follow the instructions on My Courses (https://mycourses.aalto.fi/mod/page/view.php?id=916414).

Next, enable the Cloud Functions, Cloud Pub/Sub, Cloud Build, Artifact Registry, Cloud Run Admin, Eventarc and Google Cloud Storage JSON APIs in your project. Additionally, the second task also requires to enable the Vision API. You can enable all the required service APIs through this link (https://console.cloud.google.com/flows/enableapi?apiid=cloudfunctions.googleapis.com,pubsub,storage_api,vision.googleapis.com,cloudbuild.googleapis.com,artifactregistry.googleapis.com,run.goo After enabling the APIs, it may prompt you to set up credentials. For this assignment, you don't need to create separate credentials for each API.

You can work on the assignment locally from your machine by installing the Google Cloud SDK (https://cloud.google.com/sdk/docs/install) (to deploy and manage the functions).

Getting familiar with Cloud Functions

Complete the basic tutorials on triggering Cloud Functions:

- HTTP (https://cloud.google.com/functions/docs/tutorials/http) (Note that you cannot use the `--allow-unauthenticated` flag, as this is against Aalto policy. Check the tutorial (https://mycourses.aalto.fi/mod/page/view.php?id=916473&forceview=1) on MyCourses on how to test your function.),
- Cloud Storage (https://cloud.google.com/functions/docs/tutorials/storage) (only object finalize, i.e., when a file is created),
- PubSub (https://cloud.google.com/functions/docs/tutorials/pubsub) events.

Once you have completed the basic tutorials, you might find the following resources useful when completing the assignments:

- Write event-driven functions (2nd gen) (https://cloud.google.com/functions/docs/writing/write-event-driven-functions)
- Specifying dependencies with pip (https://cloud.google.com/functions/docs/writing/specifying-dependencies-python#specifying_dependencies_with_pip)
- Setting environment variables for Cloud Functions with gcloud (https://cloud.google.com/functions/docs/env-var)

Grading

Submissions are evaluated by first trying to create a cloud function and then evaluating the function through `pytest`.

Attention

Make sure you have added the A Plus grader account as described here (https://mycourses.aalto.fi/mod/page/view.php?id=916414)

The submission should include one file and a text field.

GCP project ID

This should be the project ID of your GCP project.

Cloud Functions code

This ZIP file must include all source files necessary for your function deployment. Make sure your source files are at the root of the ZIP file, rather than a folder containing the files. See Structuring source code (https://cloud.google.com/functions/docs/writing#structuring_source_code) for more information. Please name your file *code.zip* .

Note

Each task first deploys your Cloud Function. If this step is successful, the grader tests the basic functionality of the Cloud Function. The points are assigned according to the table below.

Test	Points
Output of cloud function is correct	48

Task 1: function with an HTTP trigger

Create a cloud function (gen 2) with the entry-point `create_text_file_http` (see below for scaffolding code). The function is triggered when an HTTP POST request with a JSON body containing `fileName` and `fileContent` is sent to the trigger endpoint.

Note

Although the entry-point has underscores in its name, generation 2 cloud functions don't allow underscores in cloud function names. You can name your cloud function freely when deploying/testing, for example to create-text-file-http. The grader will assign a random name for your cloud function.

The function must:

- Get the bucket name from the environment variable `BUCKET_ENV_VAR`
- Create a file named with its name set to the value of `fileName` inside the bucket
- Set the contents of the file to the value of `fileContent`
- Return the `fileName` in the body of the response (JSON).
- Return a HTTP status code of 200.

For example, when this HTTP request is sent:

```
curl -X POST <YOUR_CLOUD_FUNCTION_URI> -H "Content-Type:application/json" -H "Authorization: bearer $(gcloud auth p
```

Your function should:

- Create a file called `c9880557-cb3d-49dd-8ab2-1a13b4f2b575` inside a bucket (the name of the bucket is the environment variable `BUCKET_ENV_VAR`)
- Set the contents of the created file to `86df48ab-2d5e-41f0-8cb1-263c49360992`
- Return HTTP status code 200 and the below JSON:

```
{"fileName":"c9880557-cb3d-49dd-8ab2-1a13b4f2b575"}
```

Note

To get started you can download a scaffolding application **here** (https://gitmanager.cs.aalto.fi/static/CS-E4190_2022Autumn/_downloads/template_http.zip).

You'll find these resources useful:

- Google Cloud Functions Overview (https://cloud.google.com/functions/docs/writing)

Your code must be successful deployed by our grader for points. Make sure you have added our grader to your project and tested deploying and running the function on your own account.

⚠ The deadline for the assignment has passed (Wednesday, 21 December 2022, 14:30).

Serverless assignment: create an HTTP-triggered function

Upload your solution files.

Project ID

code.zip

Choose File No file chosen

Submit

Task 2: function with a storage trigger

Create a cloud function with the entry_point `image_to_text_storage` (see below for scaffolding code). The function is triggered when a file is uploaded to a storage bucket. The storage bucket will be created by the grader.

Note

Although the entry-point has underscores in its name, generation 2 cloud functions don’t allow underscores in cloud function names. You can name your cloud function freely when deploying/testing, for example to image-to-text-storage. The grader will assign a random name for your cloud function.

The function must:

- Get the storage bucket from the Cloud Storage event data
- Download the image from the storage bucket to the `/tmp` directory
- Use the Google Cloud Vision api to detect text in the image (this should be the full text detected by Google Cloud Vision, no text manipulation is required)
- Save the detected text to a `.txt` file with the same name as the image to the same storage bucket

For example:

A Storage Bucket called `EXAMPLE_BUCKET` will be created automatically by the grader. When an image called `testimage.jpg` is uploaded to `EXAMPLE_BUCKET`, the Cloud Function will be triggered. The function downloads the image to a temporary directory, detects text in the image and uploads a text file called `testimage.txt` to the same bucket with the detected text.

Attention

Make sure to tell your function to ignore files that end with `.txt`

Note

To get started you can download a scaffolding application **`here`** (https://gitmanager.cs.aalto.fi/static/CS-E4190_2022Autumn/_downloads/template_storage.zip).

You’ll find these resources useful:

- Google Cloud Vision Docs (https://cloud.google.com/vision/docs/ocr)
- Google Storage Docs (https://cloud.google.com/storage/docs/uploads-downloads)

Your code must be successful deployed by our grader for points. Make sure you have added our grader to your project and tested deploying and running the function on your own account.

⚠ The deadline for the assignment has passed (Wednesday, 21 December 2022, 14:30).

You cannot submit this assignment

The deadline for the assignment has passed (Wednesday, 21 December 2022, 14:30).

Show anyway

Serverless assignment: create a Cloud Storage-triggered function

Upload your solution files.

Project ID

code.zip

Choose File No file chosen

Submit

Task 3: function with a pub/sub trigger

Create a Pub/Sub-triggered function with the entry-point `restaurant_orders_pubsub` (see below for scaffolding code) that is triggered when a message is published to a Pub/Sub topic called `restaurant_orders_queue` .

> **Note**
>
> Although the entry-point has underscores in its name, generation 2 cloud functions don't allow underscores in cloud function names. You can name your cloud function freely when deploying/testing, for example to restaurant-orders-pubsub. The grader will assign a random name for your cloud function.

The function must:

- Read the JSON Base64 Encoded Pub/Sub message
- If the order is a **takeout** order, publish the JSON order to a Pub/Sub topic called `restaurant_takeout_orders`
- If the order is an **eat-in** order, publish the JSON order to a Pub/Sub topic called `restaurant_eat-in_orders`
- Ignore all other messages

For example when your functions receives this JSON string (which will be base64 encoded):

```
{
        "orderID": "c9880557-cb3d-49dd-8ab2-1a13b4f2b575",
        "type": "takeout",
        "items": [
                "fish",
                "chips"
        ]
}
```

Your function should send it to the Pub/Sub topic called `restaurant_takeout_orders` . Similarly if the `type` was **eat-in** you should send it to the Pub/Sub topic called `restaurant_eat-in_orders`

> **Note**
>
> To get started you can download a scaffolding application **`here`** (https://gitmanager.cs.aalto.fi/static/CS-E4190_2022Autumn/_downloads/template_pubsub.zip).
>
> You'll find these resources useful:
>
> - Google Cloud Functions Pub/Sub (https://cloud.google.com/functions/docs/calling/pubsub)

Your code must be successful deployed by our grader for points. Make sure you have added our grader to your project and tested deploying and running the function on your own account.

⚠ The deadline for the assignment has passed (Wednesday, 21 December 2022, 14:30).

Serverless assignment: create a Pub/Sub-triggered function

Upload your solution files.

Project ID

code.zip

Choose File No file chosen

Submit

« 7.2 Beyond the cloud (/css/2022/serverless/exercise-edge-fo...