

GCP Lectures - Project II memo

Piotr Zalas

Overview

The application works correctly. It was written by 1 person. Key points of design are described in memo as requested. For diagram, see below.

2 points.

Source repos and CI/CD

The project code is stored in <https://source.cloud.google.com/cal-m-aegis-232817/projekt2>. I have written project in Python. I use Cloud Build to build and deploy app to App Engine and to deploy cloud functions. The whole configuration is passed to app in environment variables.

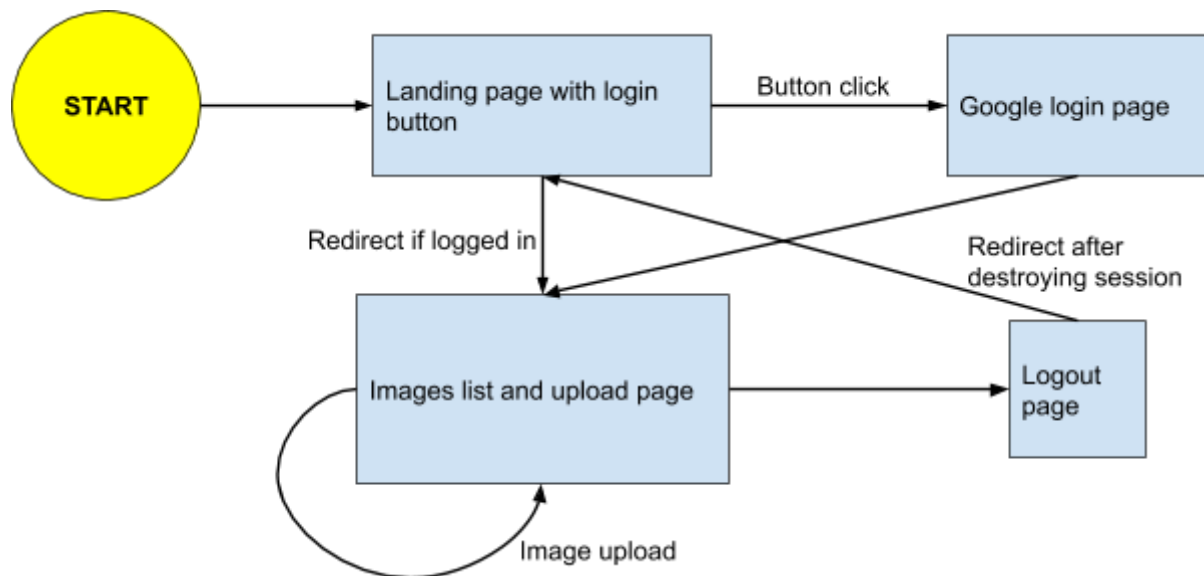
6 points.

Authentication and security

The whole web page enforces use of HTTPS. It uses SSL keys provided by App Engine. To login user I use OAuth 2 combined with Google Credentials.

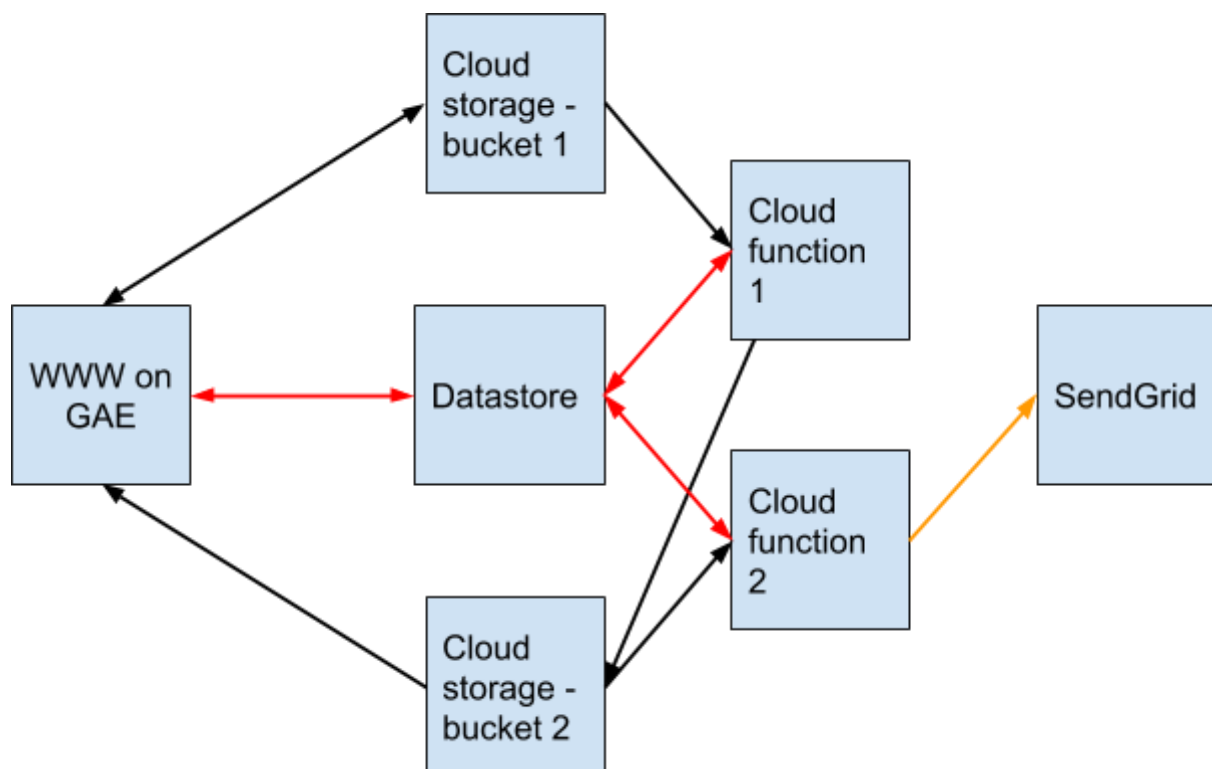
2 points.

Web application flow



Data flow

On the chart below you can see data flow. Red = image metadata, black = image blobs, orange = some metadata + blobs. WWW is deployed and running on Google App Engine.



Cloud Storage and Datastore

When user wants to upload image he provides title and file with image. All images are stored in two separate Cloud Storage buckets, one is for original images, the other one for scaled. Images are directly available only to GCF and GAE. If user wants to see image, he must use signed URL generated by webpage. Each image has a unique number generated by Datastore, the primary key of entry with image description. Datastore holds image title, original file name, URLs for original and scaled files in buckets, information about processing errors and email address of the owner. The kind of entity is "Image". When user wants to see list of images, we use his email address to query for his pictures.

4 points.

Functions

I have two functions. Both are triggered when a new file appears in corresponding buckets in cloud storage.

The first function downloads image from first bucket, downscales it, and then upload new image to bucket 2. If an error occurs, for example the image has incorrect format despite file name ending with supported extension, the original image is uploaded (because otherwise second cloud function wouldn't be triggered, and we want the second function to send email reports about errors).

The second function sends emails using SendGrid. The email contains summary on errors and two signed links to cloud storage. It seems to me that SendGrid isn't reliable service (at least with free plan). The emails almost always come with long delay of 10 - 15 minutes, and URLs in links in content are replaced to point at the sendgrid proxy. It also happens that emails are not delivered in spite of successful response from SendGrid. I wouldn't recommend this service to anyone. I've solved these problems by removing "a href" tags from email content.

7 points.