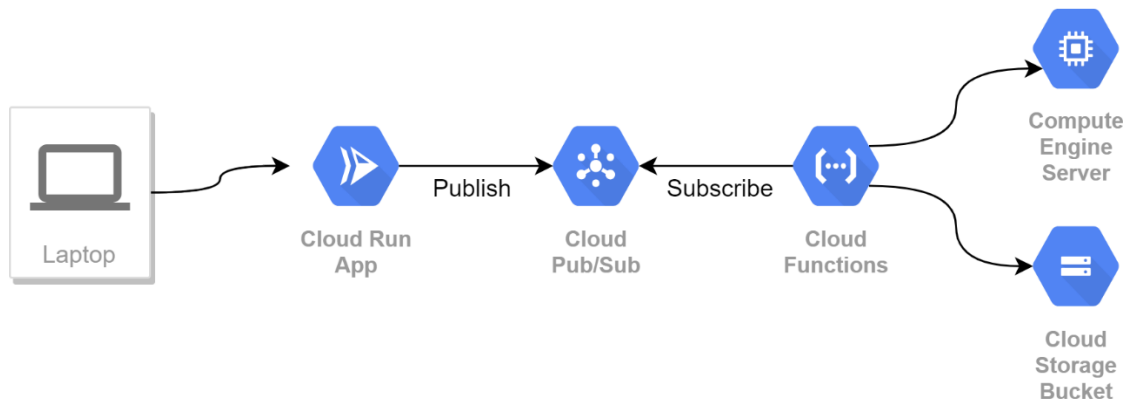


Project 1: Cloud Computing Lab

Assignment

Follow each step in this document to create the system shown below. Submit a report to include the following:

- A screenshot of your Cloud Run app which includes the buttons and address bar of the browser
- A screenshot of your server created by the cloud function (found here: <https://console.cloud.google.com/compute/instance>)
- A screenshot of your storage bucket created by the cloud function (found here: <https://console.cloud.google.com/storage/browser>)
- Your source code for your Cloud Run app and your cloud function
- A short write up (2-3 paragraphs) explaining the serverless components of this project
- A short write up (2-3 paragraphs) describing the purpose of pub/sub components



Step by Step Instructions

Step 1: Setting up Your Cloud Environment

In this step, you will prepare your cloud project to deploy this project. To do so, you will need to set up a development environment. You can use your existing laptop (recommended), or you can create a Windows server in your Google Cloud Project to use as your development environment. In either case, I highly recommend using PyCharm as your IDE and requesting the academic license. You should also use Python 3.7 or above, but you are welcome to also use other supported languages if you are more comfortable doing so (i.e., C#, Go, Java, Node.js, PHP, Ruby).

Follow these steps to set up your IDE and cloud project:

- 1) Install the Google Cloud SDK on your development server by following the tutorial here: <https://cloud.google.com/sdk/docs/quickstarts>
 - a) When running `gcloud init`, select the project for which you have applied the Google Cloud credits, and use the region `us-central1`.
- 2) Clone the git repository <https://github.com/pdhuff/cpsc4387.git>

- 3) Copy the `gcloud_setup.ps1` script to a location where you want to clone a new repository (e.g. `C:\Users\pdhuff\Documents\cpssc4387`)
- 4) Run the script `gcloud_setup.ps1` (i.e., `powershell -file gcloud_setup.ps1`)

Step 2: Deploying a Cloud Run App

Follow the steps to deploy a cloud run app: <https://cloud.google.com/run/docs/quickstarts/build-and-deploy#python>. You will later extend this application to interact with cloud components.

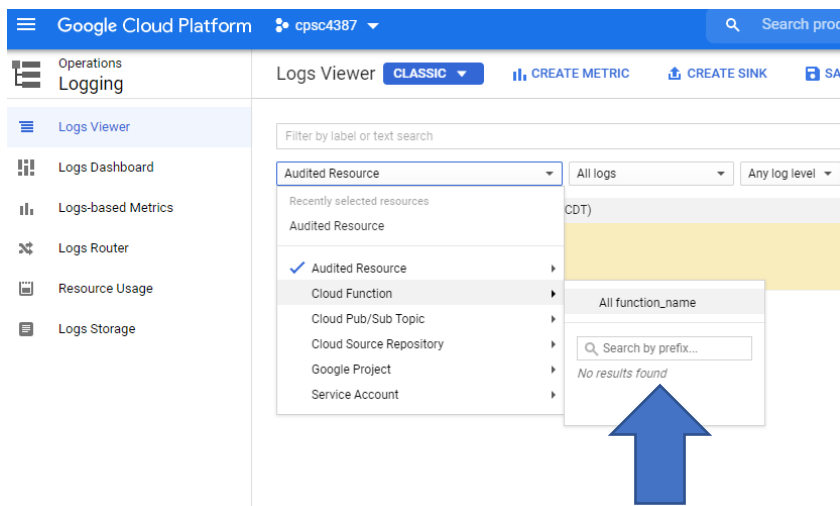
Step 3: Pub/Sub Topic

Create a pub/sub topic for interacting with compute and storage in the cloud. In the Flask app you created in the previous step, add a button to trigger the pub/sub topic. Follow the tutorial here for sending messages to a pub/sub topic: <https://cloud.google.com/pubsub/docs/publisher>.

Step 4: Deploy Cloud Function and Tie-In Cloud Run App through Pub/Sub

Cloud functions can be deployed to trigger on a message from a pub/sub topic. The stop-server function in the `cpssc4387` is an example. Write a new function that will subscribe to the pub/sub topic you just created.

Test the script by clicking on the button in your cloud run app created in step 2. Then, go to <https://console.cloud.google.com/logs/> and select the function you just created. You should see any print statements you made in the function show up here.



Step 5: Deploy Compute and Storage Functionality in your Cloud Run App

Now, for the fun part. Tie in the automation of other cloud resources. Use the app you created in step 2 and create two buttons for the following functions:

- Create a new server image:
<https://cloud.google.com/compute/docs/reference/rest/v1/instances/insert>¹
- Create a new storage bucket: https://cloud.google.com/storage/docs/creating-buckets#storage-create-bucket-code_samples

Have the cloud function you created in Step 4 parse the pub/sub message to determine whether to create a server or create a storage bucket.

¹ Scroll to the bottom for sample code. Make sure to turn the server off after you create it! You can add this to your cloud function, but you do not have to.