

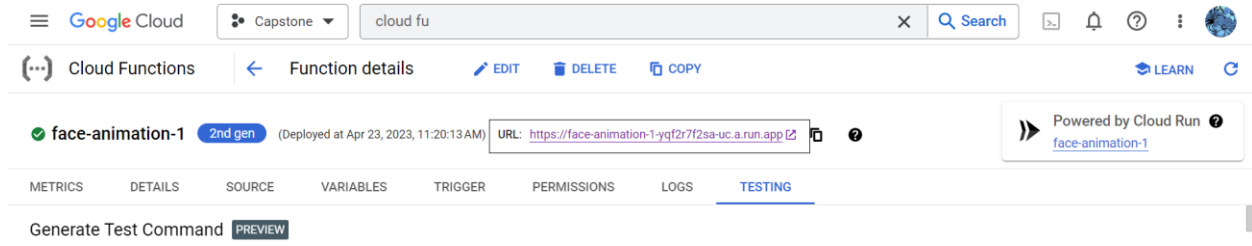
This document details the setup for the automation of the image animation in the Google Cloud Function.

1. Git branch

The branch for image animation is the “imageAnimationFinal” branch. In the ios-app/api/app.py the following lines need to be changed:

```
from joblib import Parallel, delayed
import requests
from google.cloud import storage
def sumall(audio):
    json_data = {
        'text': 'Hello, World!',
        'audio': audio,
        'image': 'photo.png',
        # replace name with the name of the pateint
        1 'name': "BananaMonster",
        # replace patientId with Id of patient
        2 'patientId': "1",
        # replace fpID with id of FP
        3 'fpId': "11"
    }
    headers = {
        'Content-Type': 'application/json',
        # Replace bearer with gcp cloud token from shell
        4 'Authorization': 'bearer ',
        5
    }
    # replace url below with url of the cloud function url
    response = requests.post('https://face-animation-1-yqf2r7f2sa-uc.a.run.app', headers=headers, json=json_data, timeout=6000)
    return response
# 6 replace audios1 with the audio files stored in gcp cloud
audios1 = ["How are you doing today", "Today is", "Where do you live", "Do you remember the time when you lost your first tooth", "It is", "It is the year twenty twenty-three"]
Parallel(n_jobs=30, verbose=10)(delayed(sumall)(i) for i in audios1)
```

1. The “name” field needs to be replaced with the name of the patient.
2. The “patientId” field needs to be replaced with the ID of the corresponding patient.
3. The “fpId” field needs to be replaced with the ID of the familiar person.
4. The “Authorization” needs to be filled in with the bearer code which is obtained by running “gcloud auth print-identity-token” in the cloud shell.
5. Replace the URL field with the URL from the cloud function as seen below:



6. In the audio array, store the name of the audio files that you want to animate the familiar person to say.

2. Setting up the Google Cloud Function

1. Create a cloud function:

<div> <div>Google Cloud</div> <div>Capstone</div> <div>cloud fu</div> <div>Search</div> <div>RELEASE NOTES</div> <div>LEARN</div> </div>										
<div> <div>Cloud Functions</div> <div>Functions</div> <div>CREATE FUNCTION</div> <div>REFRESH</div> </div>										
<div>Filter Filter functions</div>										
<input type="checkbox"/>	Environment	Name ↑	Last deployed	Region	Recommendation	Trigger	Runtime	Memory allocated	Executed function	Actions
<input type="checkbox"/>	2nd gen	face-animation	Apr 23, 2023, 11:20:13 AM	us-central1		HTTP	Python 3.8	16 GIB	iris_predict	⋮
<input type="checkbox"/>	2nd gen	face-animation-1	Apr 23, 2023, 11:20:13 AM	us-central1		HTTP	Python 3.8	32 GIB	iris_predict	⋮

2. Use the following configuration values:

Google Cloud

Capstone

cloud fu

Cloud Functions

Create function

1 Configuration

2 Code

Basics

Environment

2nd gen

Function name *

function-1

Region

us-central1

Trigger

HTTPS

Authentication

Allow unauthenticated invocations

Check this if you are creating a public API or website.

Require authentication

Manage authorized users with Cloud IAM.

ADD EVENTARC TRIGGER

Try the new Cloud Functions 2nd gen!

This next-generation of our Function-as-a-Service product comes with an advanced feature set giving you powerful infrastructure, advanced control over performance and scalability, more control around the function's runtime, more available regions, and triggers from over 90+ event sources via Cloud Audit Logs. [Learn more](#)

MORE

Runtime, build, connections and security settings

< **RUNTIME** BUILD CONNECTIONS SECURITY AND >

Memory allocated *

32 GiB preview

CPU (preview) *

8

PREVIEW

Timeout *

3600

seconds



Concurrency **PREVIEW**

Maximum concurrent requests per instance

100



Autoscaling ?

Minimum number of instances

0

Maximum number of instances *

62

Runtime service account ?

Service account

Default compute service account

By default Cloud Functions uses the automatically created Default Compute Engine Service Account. [Learn more about service accounts.](#)

Runtime environment variables ?

+ ADD VARIABLE

- Copy in the code from main.py and requirements.txt from ImageAnimation/ directory in this repo into the main.py and requirements.txt file in the cloud function:

The screenshot shows the Google Cloud Functions console. The function is named 'ins_predict' and is configured with the Python 3.8 runtime. The code editor displays the following code:

```
1 import numpy as np
2 import sys
3 import os
4 import pickle
5 from google.cloud import storage
6 def download_model_file(audio, image, name, patient_id, fp_id):
7     # Model bucket details
8     BUCKET_NAME = "familiar-person"
9     PROJECT_ID = "va-stone-374408"
10    GCS_MODEL_FILE = "Patients/{}/Familiar Person({})/Audio/{}".format(patient_id, fp_id, audio)
11
12    BUCKET_NAME2 = "familiar-person-form-data"
13    GCS_MODEL_FILE2 = "{}({})".format(name, image)
14
15
16    # Initialize a client
17    client = storage.Client(PROJECT_ID)
18
19    # Create a bucket object for our bucket
20    bucket = client.get_bucket(BUCKET_NAME)
21    bucket2 = client.get_bucket(BUCKET_NAME2)
22
23    # Create a blob object from the filepath
24    blob = bucket.blob(GCS_MODEL_FILE)
25    blob2 = bucket2.blob(GCS_MODEL_FILE2)
26
27    folder = '/tmp/'
28    if not os.path.exists(folder):
29        os.makedirs(folder)
30    # Download the file to a destination
31    blob.download_to_filename(folder + audio)
32    blob2.download_to_filename(folder + image)
33
34    return 0
```

- Deploy the cloud function and test the image animation by creating a new profile through the frontend.