CS4830 – Big Data Laboratory

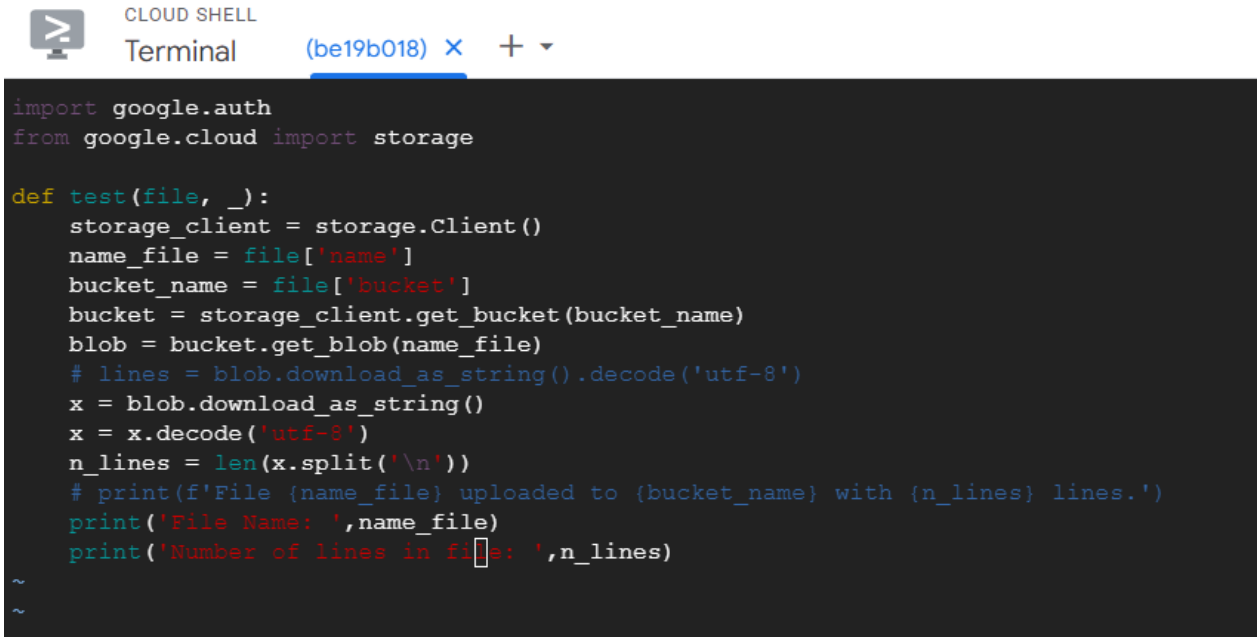
Assignment-1

Ishan Chokshi – BE19B018

January-May 2023

Indian Institute of Technology Madras

1. Step 1: Open cloud shell from the console.
   Step 2: Create a new directory to store the python file. Here I have named the directory as q1.
   Step 3: Create the python file which takes the file uploaded as an input and displays the name of the file as well as the number of lines in the file. Name this file as main.py. This python file is created by using the editor available in the console. Note the name of the function used in this file as the same function name needs to be used while creating the google cloud function. So our function is named as 'test'.



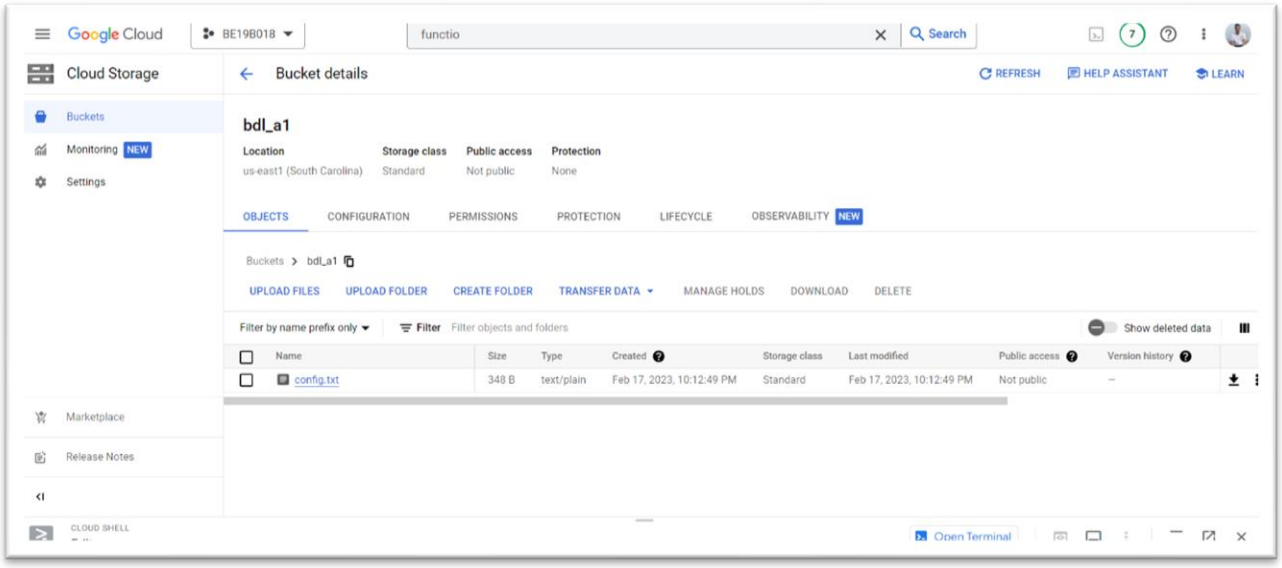

```
import google.auth
from google.cloud import storage

def test(file, _):
    storage_client = storage.Client()
    name_file = file['name']
    bucket_name = file['bucket']
    bucket = storage_client.get_bucket(bucket_name)
    blob = bucket.get_blob(name_file)
    # lines = blob.download_as_string().decode('utf-8')
    x = blob.download_as_string()
    x = x.decode('utf-8')
    n_lines = len(x.split('\n'))
    # print(f'File {name_file} uploaded to {bucket_name} with {n_lines} lines.')
    print('File Name: ',name_file)
    print('Number of lines in file: ',n_lines)
~
~
```

   Step 4: Create a requirements.txt file which has the content 'google-cloud-storage' written in it.
   Step 5: We create a bucket where we will be uploading the file. This bucket will be called be19b018_a1.
   Step 6: Now, we deploy the function by running the following command on the cloud shell as shown in the screenshot given below:

```
be19b018@cloudshell:~/q1 (be19b018)$ gcloud functions deploy test --runtime python39 --trigger-resource be19b018_a1 --trigger-event google.storage.object.finalize --entry-point test
```

```
be19b018@cloudshell:~/a1 (be19b018)$ gcloud functions deploy test --runtime python39 --trigger-resource bdl_a1 --trigger-event google.storage.object.finalize --entry-point test
Deploying function (may take a while - up to 2 minutes)...working..
For Cloud Build Logs, visit: https://console.cloud.google.com/cloud-build/builds;region=us-central1/69905e14-b048-4188-b4f2-68629289dccd?project=820790181286
Deploying function (may take a while - up to 2 minutes)...done.
availableMemoryMb: 256
buildId: 69905e14-b048-4188-b4f2-68629289dccd
buildName: projects/820790181286/locations/us-central1/builds/69905e14-b048-4188-b4f2-68629289dccd
dockerRegistry: CONTAINER_REGISTRY
entryPoint: test
eventTrigger:
  eventType: google.storage.object.finalize
  failurePolicy: {}
  resource: projects/_/buckets/bdl_a1
  service: storage.googleapis.com
ingressSettings: ALLOW_ALL
labels:
  deployment-tool: cli-gcloud
maxInstances: 3000
name: projects/be19b018/locations/us-central1/functions/test
runtime: python39
serviceAccountEmail: be19b018@appspot.gserviceaccount.com
sourceUploadUrl: https://storage.googleapis.com/uploads-897758075667.us-central1.cloudfunctions.appspot.com/a5d7e0b9-ff98-4b6d-a083-32f0a0b51518.zip
status: ACTIVE
timeout: 60s
```

Step 7: Now we upload our file to be read into the bucket.



Step 8: To check if our code is working, we can view the logs of our function created by using the command shown in the screenshot below. The output is shown below as well:

```
be19b018@cloudshell:~/a1 (be19b018)$ gcloud functions logs read test
LEVEL: D
NAME: test
EXECUTION_ID: bregkbuu3p5g
TIME_UTC: 2023-02-17 16:42:52.084
LOG: Function execution took 889 ms, finished with status: 'ok'

LEVEL:
NAME: test
EXECUTION_ID: bregkbuu3p5g
TIME_UTC: 2023-02-17 16:42:52.082
LOG: Number of lines in file:  19

LEVEL:
NAME: test
EXECUTION_ID: bregkbuu3p5g
TIME_UTC: 2023-02-17 16:42:52.082
LOG: File Name:  config.txt
```

2. **Part 1:**

   Step 1: Open the VM. Check if pip and python are installed in the VM.
   Step 2: Open a new python file in an editor using the vi command
   Step 3: Make the python function returning the sum of the first N Fibonacci numbers. I have considered Fibonacci series to start with 0. The value N I have taken is 50.
   Step 4: Run the python file using the python3 command.

```
be19b018@instance-1:~$ vi fibo2.py
be19b018@instance-1:~$ python3 fibo2.py
20365011073
```

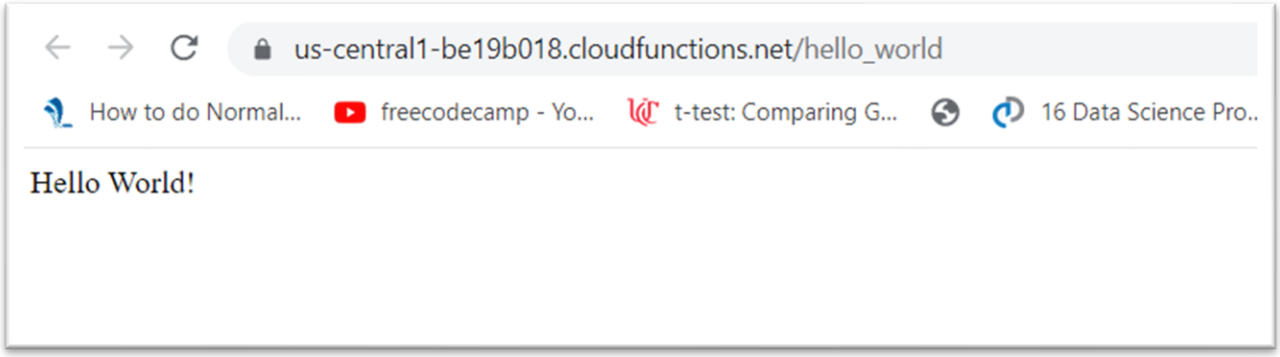
   Part 2:

   Step 1: Open cloud shell and make a new directory.
   Step 2: Upload the zipped folder to this new directory.
   Step 3: Unzip the folder using the unzip command in the cloud shell.

Step 4: Deploy the hello_world function.

```
be19b018@cloudshell:~/q2 (be19b018)$ ls
bdl-2023-assignment1-main  bdl-2023-assignment1-main.zip  requirements.txt
be19b018@cloudshell:~/q2 (be19b018)$ cd bdl-2023-assignment1-main/
be19b018@cloudshell:~/q2/bdl-2023-assignment1-main (be19b018)$ ls
app.py  main.py  README.md
be19b018@cloudshell:~/q2/bdl-2023-assignment1-main (be19b018)$ gcloud functions deploy hello_world --runtime python310 --trigger-http
--allow-unauthenticated
Allow unauthenticated invocations of new function [hello_world]? (y/N)?  y

Deploying function (may take a while - up to 2 minutes)...working.
For Cloud Build Logs, visit: https://console.cloud.google.com/cloud-build/builds;region=us-central1/80f36638-3e72-4d24-bec0-23bf8e9b2438?project=820790181286
Deploying function (may take a while - up to 2 minutes)...done.
availableMemoryMb: 256
buildId: 80f36638-3e72-4d24-bec0-23bf8e9b2438
buildName: projects/820790181286/locations/us-central1/builds/80f36638-3e72-4d24-bec0-23bf8e9b2438
dockerRegistry: CONTAINER_REGISTRY
entryPoint: hello_world
httpsTrigger:
  securityLevel: SECURE_ALWAYS
  url: https://us-central1-be19b018.cloudfunctions.net/hello_world
```

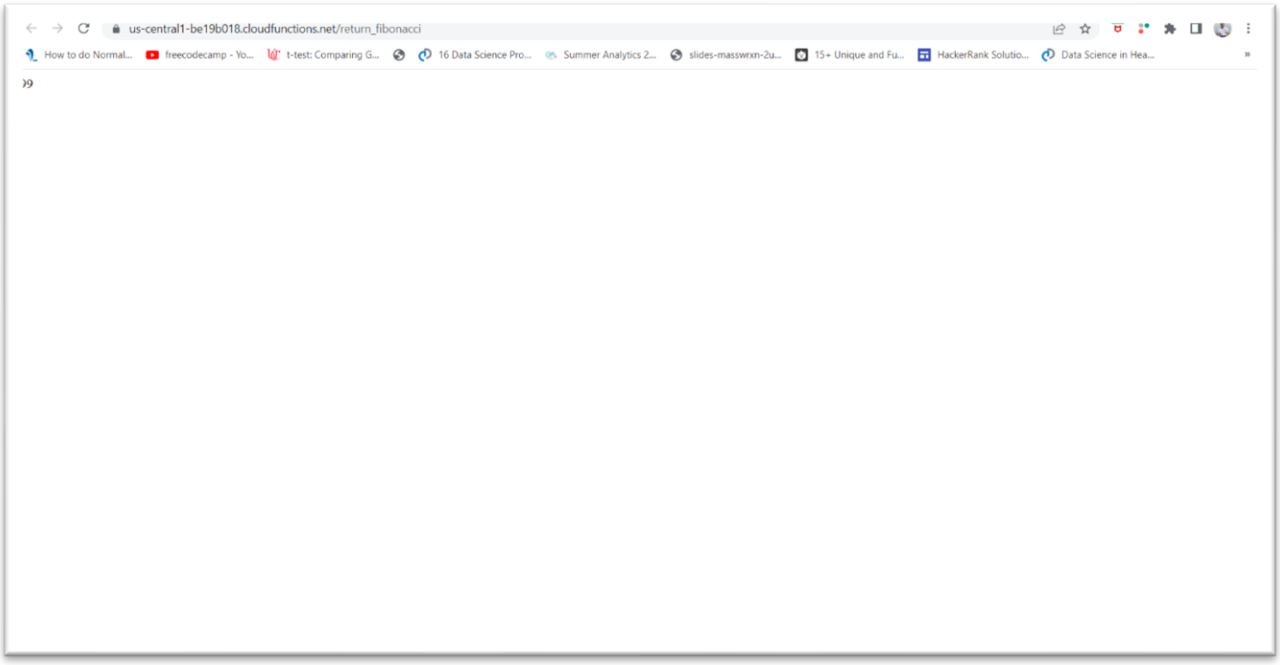
Step 5: Open the webpage using the url:
https://us-central1-be19b018.cloudfunctions.net/hello_world

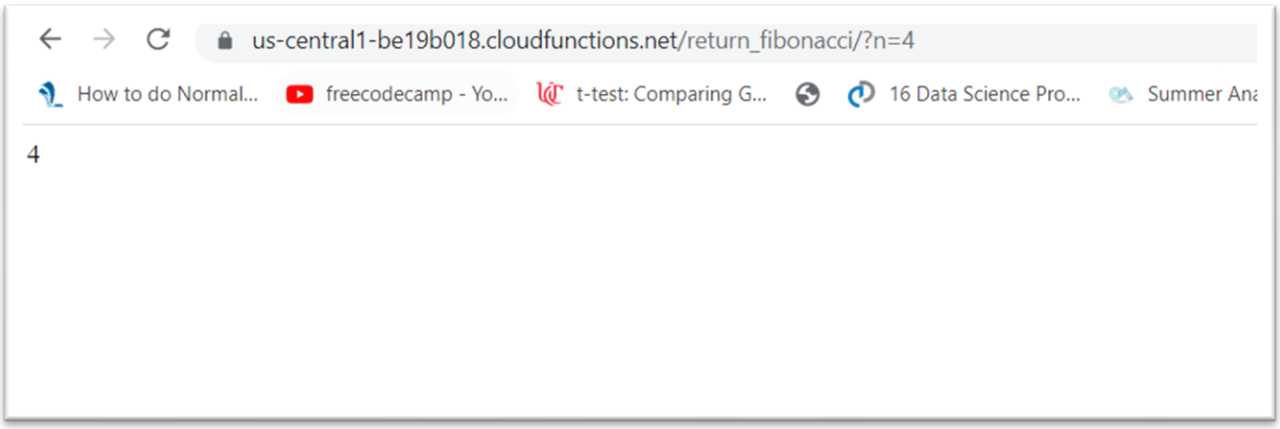


Step 6: Shut down the hello_world function
Step 7: Deploy the return_fibonacci function
Step 8: The output is 99. This means we need to edit the fibonacci function. So first delete the google clod function return_fibonacci
Step 9: The required changes are made by opening the online text editor from the cloud shell.
Step 10: Deploy the return_fibonacci function again.
Step 11: Open the webpage using the url:
https://us-central1-be19b018.cloudfunctions.net/return_fibonacci
Step 12: In order to pass ‘n’ as a parameter for the input, add the following suffix to the URL: /?n=<any number>. Shown below is the output for n = 4. Note that I have considered Fibonacci series to being with 0.

```python
def calculate_fibonacci(n):
    a = 0
    b = 1
    if n == 1:
        return 0
    total = 0
    for i in range(0,n):
        total += a
        a,b = b,a+b
    return total
    # return 99
    # complete this function
```

```
be19b018@cloudshell:~/q2/bdl-2023-assignment1-main (be19b018)$ gcloud functions deploy return_fibonacci --runtime python310 --trigger-http --allow-unauthenticated
Deploying function (may take a while - up to 2 minutes)...working.
For Cloud Build Logs, visit: https://console.cloud.google.com/cloud-build/builds;region=us-central1/244d416a-5b63-4403-bbd7-c231baf28a55?project=820790181286
Deploying function (may take a while - up to 2 minutes)...done.
availableMemoryMb: 256
buildId: 244d416a-5b63-4403-bbd7-c231baf28a55
buildName: projects/820790181286/locations/us-central1/builds/244d416a-5b63-4403-bbd7-c231baf28a55
dockerRegistry: CONTAINER_REGISTRY
entryPoint: return_fibonacci
httpsTrigger:
  securityLevel: SECURE_OPTIONAL
  url: https://us-central1-be19b018.cloudfunctions.net/return_fibonacci
ingressSettings: ALLOW_ALL
labels:
  deployment-tool: cli-gcloud
name: projects/be19b018/locations/us-central1/functions/return_fibonacci
runtime: python310
serviceAccountEmail: be19b018@appspot.gserviceaccount.com
sourceUploadUrl: https://storage.googleapis.com/uploads-897758075667.us-central1.cloudfunctions.appspot.com/ce2cc9b0-2a18-4fa3-af10-05de30596b52.zip
status: ACTIVE
timeout: 60s
updateTime: '2023-02-17T16:19:01.343Z'
```





3. **Serverless Computing:** Serverless computing is a cloud computing model which allows users to write and run code for any application without the need for managing any kind of backend infrastructure, server or any other backend resource. Any kind of backend resource or some other provision which is required for the application is managed and allocated by the cloud provider itself. This enables the developer to focus on their code without worrying about other things such as the underlying infrastructure, deployment and scalability of the application.
   The biggest advantage of using serverless computing is its cost-effectiveness when it comes to scalability of the applications to be run by the developer. This is one the reasons why serverless computing can be quite useful in the field of big data. It can easily handle processing, analysing, transformation and integration of big data for different uses in applications. It can also be used or large-scale machine learning purposes.

   PaaS (Platform as a Software): A computing service which provides support for the development, maintenance, management, and deployment of various applications to the developers, without the need of having knowledge regarding the underlying infrastructure necessary for running the application. As a student developer, using a PaaS would be useful when we want to use it for scaling of applications without worrying too much about how to streamline the development process. There might be a few drawbacks of using PaaS. One of them is that the PaaS we are using may not support code in all languages. There is also less control over the underlying infrastructure such as virtual machines, virtual storage, etc. So if any issues arise during the development stage from the platform side, it might be difficult to debug the issue.
   IaaS (Infrastructure as a Service): A computing service which provides servers, virtual machines, storage, which is used for end-to-end development of an application. For a student developer, IaaS provides more control over the underlying infrastructure which can be used for better customization and scalability of the application. The disadvantage of IaaS is its higher maintenance cost since the developer has to handle all the underlying infrastructure. This also

requires extensive knowledge about the same which might be difficult for a student developer to understand.

SaaS (Software as a Service): A computing service which gives users access to online software applications. These services can be used without the requirement of any maintenance or installation by the user. The user only requires an internet connection to access this service. For student developers, SaaS can be useful if they want to access some software online in relation to their application. But due to limited or no access to the underlying infrastructure, it is difficult to customize the software for specific uses.

Cloud API: It acts as a gateway to access for software made by developers to access some particular cloud services such as databases, storage, etc. Cloud APIs enable the developers to access the cloud services without having the need to have any knowledge about the underlying infrastructure. This is a major plus point since it allows the developers to focus on the code which leads to efficient, faster, and cost-effective development of applications by availing the resources provided by the cloud service for better scalability and streamlining of applications.