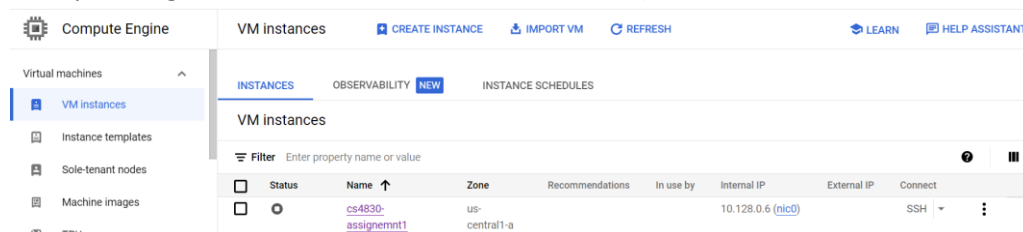


Question 1:

Use Cloud Shell to deploy a Google Cloud Function that gets triggered when you upload a file on GCS. The function should print name of the uploaded file and the number of lines in the uploaded file.

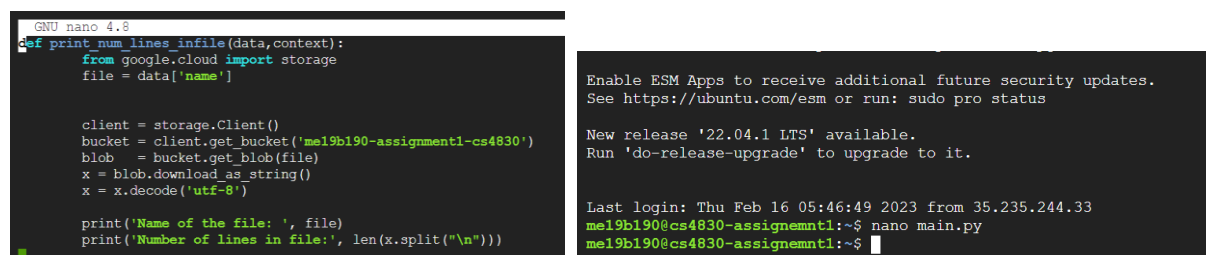
Note: In the submission for Q1, upload a Python file containing the Google Cloud Function. In the report document for Q1, note the sequence of steps used to achieve above task. Also, in the report document for Q1, post the screenshots indicating the uploaded file, function deployment, logs showing the name and number of lines of uploaded file.

Step 1: Create a bucket**Step 2: Create a VM in instance and change the operating system to Ubuntu - 20.04 LTS x86/64 (Compute Engine → VM instances → create instance)**

Step 3: The cloud function we deploy must have access to google cloud storage. Thus, the file **requirements.txt** contains the list of libraries that will be automatically be installed by Cloud function.

```
me19b190@vm-assignment1-mel19b190:~$
me19b190@vm-assignment1-mel19b190:~$
me19b190@vm-assignment1-mel19b190:~$ echo 'google-cloud-storage' >> requirements.txt
me19b190@vm-assignment1-mel19b190:~$
me19b190@vm-assignment1-mel19b190:~$
```

Step 4: Create/ Upload a python file to print the file name and number of lines present in the file, which triggers when a new file is uploaded to storage.



1. The command **nano main.py** opens an editable python script to code the required function with the name main.py
2. The function **print_num_lines_infile** is the function that will be triggered by Cloud storage in the background. The arguments data and context represent the event payload and metadata for triggering respectively. After completing the python code for the given function, the python file has been saved and closed.

Step 4: Deploy the google cloud function

For the function to get triggered each time a new event occurs in a bucket, we have to deploy the google cloud function.

```

me19b190@vm-assignment1-me19b190:~$ gcloud functions deploy print_num_lines_infile --runtime python37 --trigger-resource me19b190-assignment1-cs4830 --trigger-event google.storage.object.finalize
Deploying function (may take a while - up to 2 minutes)...!
For Cloud Build Logs, visit: https://console.cloud.google.com/cloud-build/builds;region=us-central1/1c619d07-3954-4951-92e6-ecb69301df7f?project=775468331472
Deploying function (may take a while - up to 2 minutes)...done.
availableMemoryMb: 256
buildId: 1c619d07-3954-4951-92e6-ecb69301df7f
buildName: projects/775468331472/locations/us-central1/builds/1c619d07-3954-4951-92e6-ecb69301df7f
  
```

Function deployed – *print_num_lines_infile* Respective Bucket - *me19b190-assignment1-cs4830*

Step 5: Every time we upload a new file to the bucket the function *print_num_lines_infile* will be triggered. To verify the claim, the file *config.txt* has been uploaded to the bucket *me19b190-assignment1-cs4830*, and the logs have been verified using the command “*gcloud functions logs read --limit 50*” to make sure the execution has been completed successfully.

```

me19b190@vm-assignment1-me19b190:~$
me19b190@vm-assignment1-me19b190:~$
me19b190@vm-assignment1-me19b190:~$ wget https://storage.googleapis.com/bdl-bucket-1/lab2/config.txt
--2023-02-16 11:12:58-- https://storage.googleapis.com/bdl-bucket-1/lab2/config.txt
Resolving storage.googleapis.com (storage.googleapis.com)... 172.253.119.128, 172.253.114.128, 172.217.214.128, ...
Connecting to storage.googleapis.com (storage.googleapis.com)|172.253.119.128|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 330 [text/plain]
Saving to: 'config.txt'

config.txt                                100%[=====>]          330  --.-KB/s  in 0s

2023-02-16 11:12:58 (27.2 MB/s) - 'config.txt' saved [330/330]
  
```

```

D    print_num_lines_infile f9as06k356mq 2023-02-15 18:01:43.328 Function execution took 3114 ms, finished with status: 'ok'
I    print_num_lines_infile f9as06k356mq 2023-02-15 18:01:43.326
I    print_num_lines_infile f9as06k356mq 2023-02-15 18:01:43.326 19
I    print_num_lines_infile f9as06k356mq 2023-02-15 18:01:43.326
I    print_num_lines_infile f9as06k356mq 2023-02-15 18:01:43.326 Number of lines in file:
I    print_num_lines_infile f9as06k356mq 2023-02-15 18:01:43.326 config.txt
I    print_num_lines_infile f9as06k356mq 2023-02-15 18:01:43.326
I    print_num_lines_infile f9as06k356mq 2023-02-15 18:01:43.326 Name of the file:
D    print_num_lines_infile f9as06k356mq 2023-02-15 18:01:40.214 Function execution started
me19b190@vm-assignment1-me19b190:~$
  
```

Question 2:

1. Spin up a VM and use the VM to calculate the sum of first N Fibonacci numbers. Use any value of N between 10 to 100.

2. Use Cloud Shell to deploy a Google Cloud Function that takes the value of N as argument and returns the sum of first N Fibonacci numbers

1. A VM/Google shell has been signed in
2. If for the first time, all the cloud functions and cloud API's need to be enabled, which have been enabled already in our context
3. The **"main.zip"** has been downloaded from the given [URL](#) using **"wget"** command

```
me19b190@vm-assignment1-me19b190:~$ wget https://github.com/cs22m036/bdl-2023-assignment1/archive/refs/heads/main.zip
--2023-02-17 08:06:09-- https://github.com/cs22m036/bdl-2023-assignment1/archive/refs/heads/main.zip
Resolving github.com (github.com)... 140.82.113.4
Connecting to github.com (github.com)|140.82.113.4|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://codeload.github.com/cs22m036/bdl-2023-assignment1/zip/refs/heads/main [following]
--2023-02-17 08:06:09-- https://codeload.github.com/cs22m036/bdl-2023-assignment1/zip/refs/heads/main
Resolving codeload.github.com (codeload.github.com)... 140.82.113.10
Connecting to codeload.github.com (codeload.github.com)|140.82.113.10|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1004 [application/zip]
Saving to: 'main.zip'

main.zip           100%[=====>]   1004  --.-KB/s   in 0s

2023-02-17 08:06:09 (9.74 MB/s) - 'main.zip' saved [1004/1004]

me19b190@vm-assignment1-me19b190:~$
```

To unzip the main.zip, **unzip** has been installed using **"sudo apt install unzip"** and the main.zip has been extracted using the following command

```
me19b190@vm-assignment1-me19b190:~$ unzip main.zip
Archive:  main.zip
67b9d317488b873d62702206eb52070a3f731757
creating: bdl-2023-assignment1-main/
extracting: bdl-2023-assignment1-main/README.md
inflating: bdl-2023-assignment1-main/app.py
inflating: bdl-2023-assignment1-main/main.py
me19b190@vm-assignment1-me19b190:~$
```

4. The following are the files that are extracted from main.zip

```
me19b190@vm-assignment1-me19b190:~$
me19b190@vm-assignment1-me19b190:~$
me19b190@vm-assignment1-me19b190:~$
me19b190@vm-assignment1-me19b190:~$ cd bdl-2023-assignment1-main/
me19b190@vm-assignment1-me19b190:~/bdl-2023-assignment1-main$ ls
README.md  app.py  main.py  main.py.save
me19b190@vm-assignment1-me19b190:~/bdl-2023-assignment1-main$
```

main.py.save is created in the after editing and saving the python script file main.py

```
https://ssh.cloud.google.com/v2/ssh/projects/me19b190-project1/zones/us-central1-a/instances/vm-assignment1-me19b190/authorize?authuser=0&hl=en_US&projectNumber=775468331472&useAdminProxy=true
SSH-in-browser
main.py
import os
import flask

def hello_world(request):
    return "Hello World!"

def calculate_fibonacci(n):
    # complete this function
    if n<10 or n>100:
        return "Input must be between 10 and 100"
    sum = 2
    prev_1 = 1
    prev_2 = 1
    for i in range(2,n):
        sum += prev_1+prev_2
        temp = prev_2
        prev_2 = prev_1
        prev_1 = temp
    return sum

def return_fibonacci(request):
    request.args = request.args
    if request.args and "n" in request.args:
        n = int(request.args["n"])
    else:
        n = 1
    return f'({calculate_fibonacci(n)})'
```

5. “hello_world” function has been deployed by running the following command

```
me19b190@vm-assignment1-me19b190:~/bd1-2023-assignment1-main$ gcloud functions deploy hello_world --runtime python310 --trigger-http --allow-unauthenticated
Deploying function (may take a while ~ up to 2 minutes)...WARNING: Setting IAM policy failed, try 'gcloud functions add-iam-policy-binding hello_world --region=
us-central1 --member=allUsers --role=roles/cloudfunctions.invoker'
Deploying function (may take a while ~ up to 2 minutes)...?
For Cloud Build logs, visit: https://console.cloud.google.com/cloud-build/builds;region=us-central1/0145aef5-fc78-47cd-b087-9989977ce814?project=775468331472
Deploying function (may take a while ~ up to 2 minutes)...done.
availableMemoryMb: 256
buildId: 0145aef5-fc78-47cd-b087-9989977ce814
buildName: projects/775468331472/locations/us-central1/builds/0145aef5-fc78-47cd-b087-9989977ce814
dockerRegistry: CONTAINER_REGISTRY
entryPoint: hello_world
httpsTrigger:
  securityLevel: SECURE_ALWAYS
  url: https://us-central1-me19b190-project1.cloudfunctions.net/hello_world
ingressSettings: ALLOW_ALL
labels:
  deployment-tool: cli-gcloud
name: projects/me19b190-project1/locations/us-central1/functions/hello_world
runtime: python310
serviceAccountEmail: me19b190-project1@appspot.gserviceaccount.com
sourceUploadUrl: https://storage.googleapis.com/uploads-132807434885.us-central1.cloudfunctions.appspot.com/242d46c1-13aa-4367-ab2b-ff5f400de3a5.zip
status: ACTIVE
timeout: 60s
updateTime: '2023-02-17T08:12:35.497Z'
versionId: '1'
```

The warning “The setting IAM policy failed” because of the authentication error. We have to authenticate by giving access to “***allUsers for the role Cloud function invoker***”. The following are the steps followed

- Go to cloud functions
- Click on “***hello_world***” and go to ***Permissions***
- Click on Add principals and give access to “***allUsers***” in new principals. Set the role to ***Cloud Function Invoker*** and click on save.

Grant access to "hello_world"

principals can take. Optionally, add conditions to grant access to principals only when a specific criteria is met. [Learn more about IAM conditions](#)

Resource

hello_world

Add principals

Principals are users, groups, domains, or service accounts. [Learn more about principals in IAM](#)

New principals
allUsers

Assign roles

Roles are composed of sets of permissions and determine what the principal can do with this resource. [Learn more](#)

Role *
Cloud Functions Invoker

Ability to invoke HTTP functions with restricted access.

+ ADD ANOTHER ROLE

In cloud functions, the HTTP trigger(***--trigger-http***) enables the function to run in response to HTTP(S) requests. The function is assigned a URL, when the HTTP trigger is specified at which it can receive requests.

The command ***--allow-unauthenticated*** specifies whether or not to make our service publicly available and the deploy ***hello_world*** deploys the python function “***hello_world***”

When we click on the URL returned (which has been highlighted in the above image we observe that Hello World!

us-central1-me19b190-project1.cloudfunctions.net/hello_world

Hello World!

6. As a last step, we need to delete the deployed functions as we will be charged if the function is active. The command “***gcloud functions delete hello_world --quiet***” is used to delete the triggered function.

```
me19b190@vm-assignment1-me19b190:~/bd1-2023-assignment1-main$
me19b190@vm-assignment1-me19b190:~/bd1-2023-assignment1-main$
me19b190@vm-assignment1-me19b190:~/bd1-2023-assignment1-main$ gcloud functions delete hello_world --quiet
Waiting for operation to finish...done.
Deleted [projects/me19b190-project1/locations/us-central1/functions/hello_world].
me19b190@vm-assignment1-me19b190:~/bd1-2023-assignment1-main$
```

Deploying the function “***return_fibonacci***”

1. Now, the function “***return_fibonacci***” has been deployed using the following command

“***gcloud functions deploy return_fibonacci --runtime python310 --trigger-http --allow-unauthenticated***”

```
me19b190@vm-assignment1-me19b190:~/bd1-2023-assignment1-main$ gcloud functions deploy return_fibonacci --runtime python310 --trigger-http --allow-unauthenticated
Deploying function (may take a while - up to 2 minutes)...WARNING: Setting IAM policy failed, try 'gcloud functions add-iam-policy-binding return_fibonacci --re
gion=us-central1 --member=allUsers --role=roles/cloudfunctions.invoker'
Deploying function (may take a while - up to 2 minutes)...?
For Cloud Build Logs, visit: https://console.cloud.google.com/cloud-build/builds;region=us-central1/1fbc74e3-5697-49eb-a26f-2647a35d9905?project=775468331472
Deploying function (may take a while - up to 2 minutes)...done.
availableMemoryMb: 256
buildId: 1fbc74e3-5697-49eb-a26f-2647a35d9905
buildName: projects/775468331472/locations/us-central1/builds/1fbc74e3-5697-49eb-a26f-2647a35d9905
dockerRegistry: CONTAINER_REGISTRY
entryPoint: return_fibonacci
httpsTrigger:
  securityLevel: SECURE_ALWAYS
  url: https://us-central1-me19b190-project1.cloudfunctions.net/return_fibonacci
ingressSettings: ALLOW_ALL
labels:
  deployment-tool: cli-gcloud
name: projects/me19b190-project1/locations/us-central1/functions/return_fibonacci
runtime: python310
serviceAccountEmail: me19b190-project1@appspot.gserviceaccount.com
sourceUploadUrl: https://storage.googleapis.com/uploads-132807434885.us-central1.cloudfunctions.appspot.com/c35a6c6a-7e2d-4d07-belb-elb209d44b68.zip
status: ACTIVE
timeout: 60s
updateTime: '2023-02-17T08:18:03.412Z'
versionId: '1'
me19b190@vm-assignment1-me19b190:~/bd1-2023-assignment1-main$
```

The warning “The setting IAM policy failed” because of the authentication error. We have to authenticate by giving access to “***allUsers for the role Cloud function invoker***”. The following are the steps followed

- Go to cloud functions
- Click on “***return_fibonacci***” and go to **Permissions**
- Click on Add principals and give access to “***allUsers***” in new principals. Set the role to ***Cloud Function Invoker*** and click on save.

Grant access to "return_fibonacci"

Grant principals access to this resource and add roles to specify what actions the principals can take. Optionally, add conditions to grant access to principals only when a specific criteria is met. [Learn more about IAM conditions](#)

Resource

return_fibonacci

Add principals

Principals are users, groups, domains, or service accounts. [Learn more about principals in IAM](#)

New principals
allUsers

Assign roles

Roles are composed of sets of permissions and determine what the principal can do with this resource. [Learn more](#)

Role *
Cloud Functions Invoker

Ability to invoke HTTP functions with restricted access.

ADD ANOTHER ROLE

SAVE

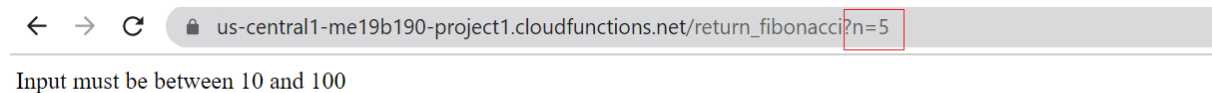
CANCEL

The function *"return_fibonacci"* is edited to return the sum of first N Fibonacci numbers given N

When we click on the returned URL or open it in the browser we see different returned values based on input provided.

The input values to the function *"return_fibonacci"* are provided by *"?n = ..."* towards the end of the URL, which have been highlighted in the below image

Input: 5 (An invalid input) Value returned: "Input must be between 10 and 100"



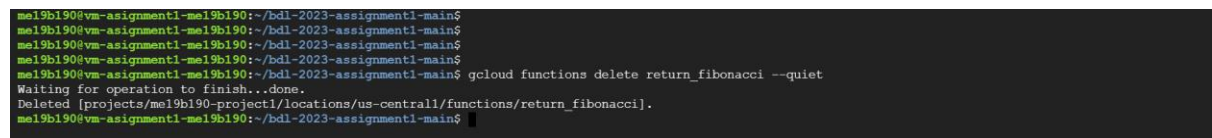
Input: 25 (A valid input (number between 10 and 25)); Value returned - 196417



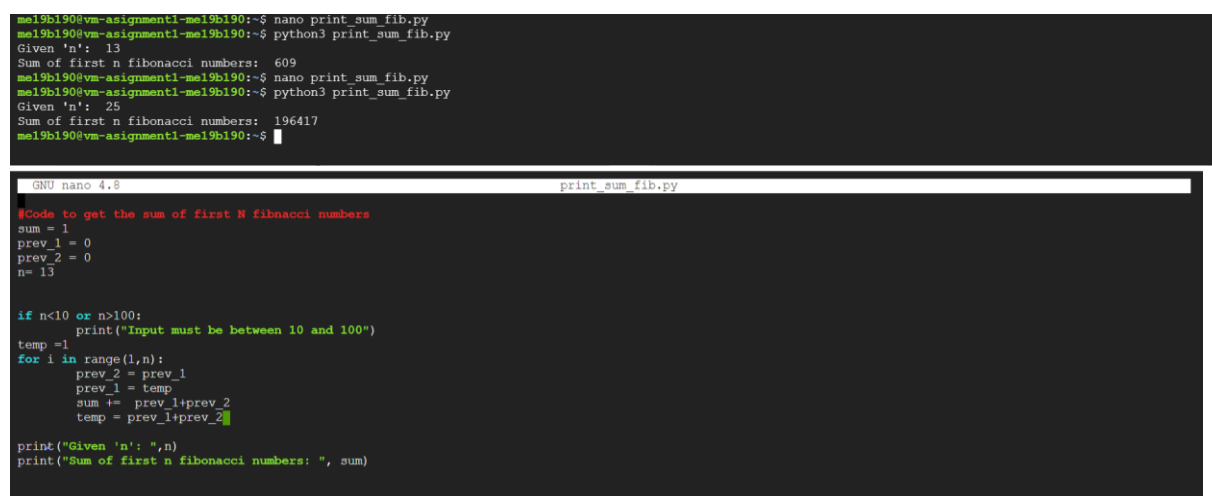
Input: 105 (An invalid input); Value returned: "Input must be between 10 and 100"



As the deployment is successful, we need to delete the function as we will be charged for cloud functions that are active. The function *"return_fibonacci"* is deleted using the following command *"gcloud functions delete return_fibonacci --quiet"*



The function to print sum of first N fibonacci numbers has been given below and has been run using a VM



Question 3:

1. Write a brief description of the following:

- a. Serverless Computing.
- b. As a student of Big Data Lab, write a comparative analysis, highlighting the pros and cons, as a student developer, b/w PaaS, IaaS, SaaS.
- c. Cloud APIs and their role in the cloud computing landscape

a. Serverless Computing:

Serverless is a cloud computing application model that enables users to build and execute computer code without the need to manage the backend servers. The term 'Serverless' does not mean no servers, but the users experience with them. Users need not manage or interact with them. Some of the leading "Serverless Cloud computing" platforms are Amazon Web Services, Microsoft Azure, IBM Cloud and Google Cloud function. One of the most salient features of cloud computing is that users do not pay for idle capacity. The cloud services are provided to the user dynamically based on demand.

Advantages of Serverless computing:

1. Pay on execution
2. Cost effective performance
3. Supports almost all programming languages
4. Better visibility and serverless models scale without intervention

- b. **PAAS:** PAAS stands for Platform As A Service(PAAS) is based on cloud, where services are managed by a third party. The provider provides the required software and hardware tools, reducing the need for complex infrastructure necessities. It provides good scaling of application being developed.

Advantages:

1. Significant cost reduction
2. Fast and flexible tools
3. Access from anywhere
4. Actively managed for developers, which enables them to focus on business needs

Disadvantages:

1. Third party services might be incompatible with current models
2. Security is based on the provider and thus data can be vulnerable to threats
3. PaaS has less scalability compared to other cloud solutions.
4. Less control stability and predictability.

IaaS: Infrastructure as a service(IaaS) is a type of cloud computing service that offers essential compute, storage and networking resources on demand.

Advantages:

1. Better control, security, increases stability and reliability
2. Optimizes costs and reduces capital expenditure
3. Good scalability and can accommodate any sudden spikes in demand

Disadvantages:

1. Security is dependent on IaaS provider
2. IT admins are needed

SaaS: Software as a service is a cloud based service and is used over an internet connection via a web browser. The provider operates, manages and maintains the infrastructure on which

Advantages:

1. Configuration-driven, for business
2. high predictability and high performance
3. Access from anywhere and cost effective

Disadvantages:

1. Least control
2. Vertical integration, highest velocity of new features.
3. Security is dependent on service provider.

c. Cloud APIs and their role in the cloud computing landscape.

(Application programming interfaces) APIs are used to integrate new applications with existing software systems as each functionality need not be written from scratch. API is a pathway between two software application systems and changes to any one of the application system does not affect the other. A cloud API allows developers/users to link to cloud computing services. There are different type of cloud APIs like PaaS API (which provide access to services like databases, SaaS APIs (which facilitate connection between cloud services) and the IaaS APIs (which enable cloud compute and storage resources). Some of the popular Cloud APIs are the Google cloud APIs, AWS APIs and Azure APIs. For any data driven business, API is an integral part. Cloud APIs enable integration of new technology and easy management of cloud applications.