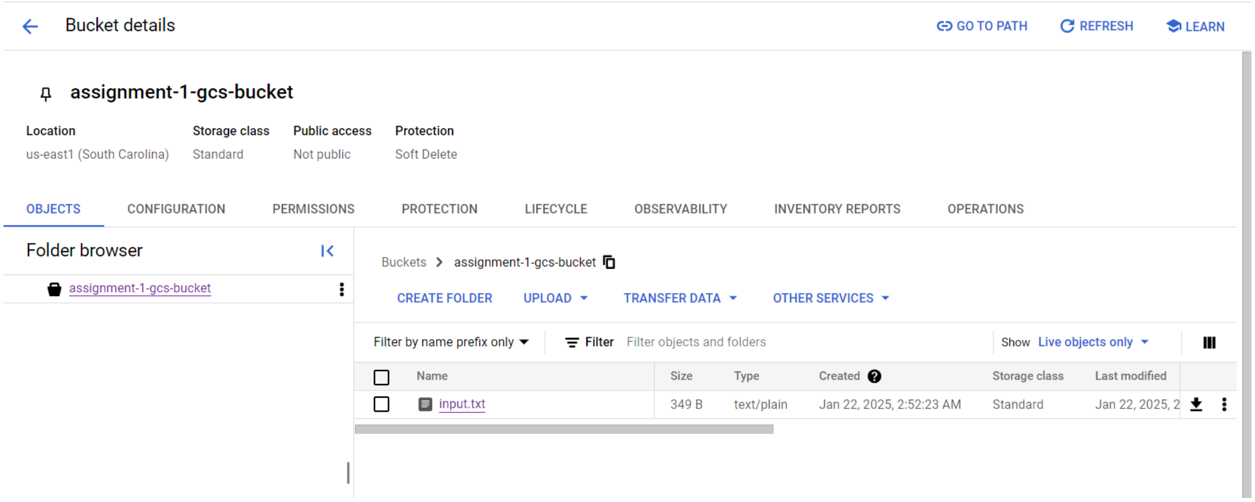
# Question : Spin Up a VM and write a python program to count lines of a file placed in GCS.

**Submitted By: Shramana Sinha, 23F1002703**

## 1. Setup and Prerequisites

### 1.1 Creating a GCS Bucket

1. A GCS bucket named "assignment-1-gcs-bucket" was created with default settings, except for data storage location set to "Lowest latency within a single region".
2. A file named "input.txt" was uploaded to the created bucket.



assignment-1-gcs-bucket

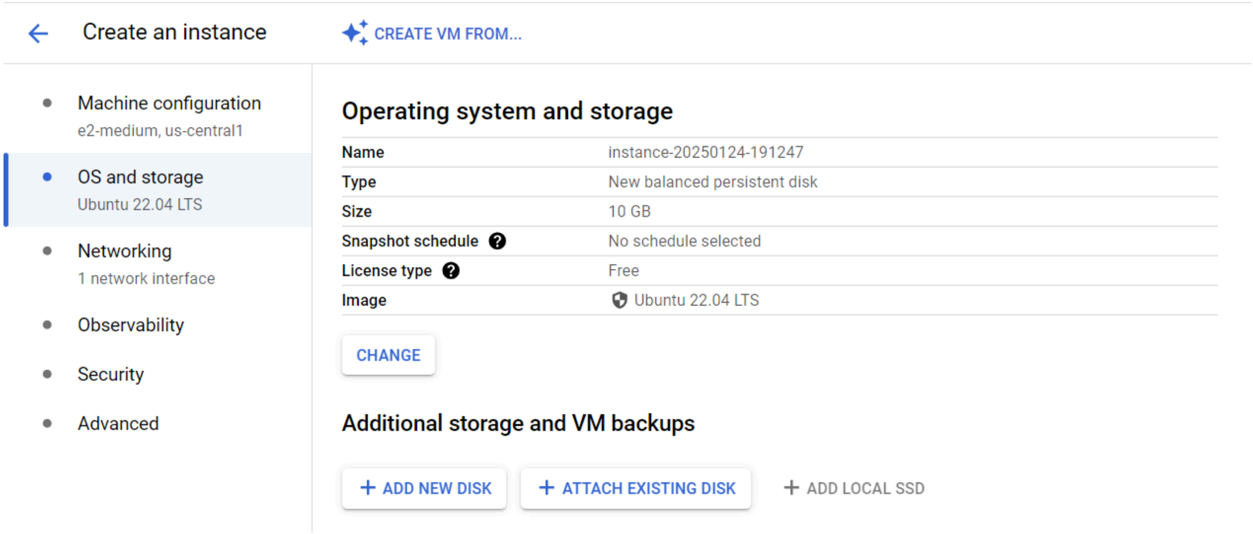
| Location | Storage class | Public access | Protection |
|---|---|---|---|
| us-east1 (South Carolina) | Standard | Not public | Soft Delete |

| □ | Name | Size | Type | Created | Storage class | Last modified |
|---|---|---|---|---|---|---|
| □ | input.txt | 349 B | text/plain | Jan 22, 2025, 2:52:23 AM | Standard | Jan 22, 2025, 2 |

*Screenshot 1: GCS bucket and file*

### 1.2 Spinning up a Virtual Machine

A Virtual Machine (VM) was created with default settings, except for using Ubuntu 22.04 LTS as the Operating System.



- Machine configuration
  e2-medium, us-central1
- OS and storage
  Ubuntu 22.04 LTS
- Networking
  1 network interface
- Observability
- Security
- Advanced

Operating system and storage

| | |
|---|---|
| Name | instance-20250124-191247 |
| Type | New balanced persistent disk |
| Size | 10 GB |
| Snapshot schedule | No schedule selected |
| License type | Free |
| Image | Ubuntu 22.04 LTS |

Additional storage and VM backups

*Screenshot 2: VM creation settings*

### 1.3 Installing Required Software

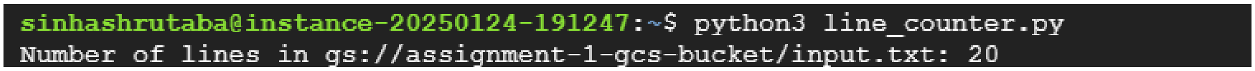
The necessary softwares were installed on the VM:

- Package lists were updated: `sudo apt update`
- Python 3 and pip were installed: `sudo apt install python3 python3-pip`
- Google Cloud Storage library was installed: `pip3 install google-cloud-storage`

## 2. Code Execution and Results

The script was executed using the command:

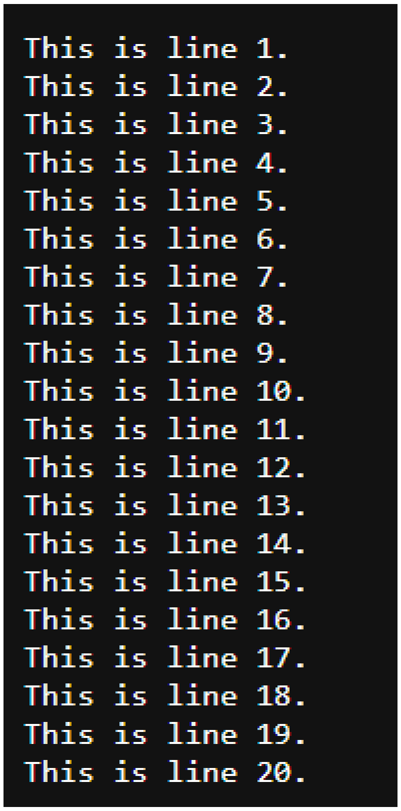
```
python3 line_counter.py
```



```
sinhashrutaba@instance-20250124-191247:~$ python3 line_counter.py
Number of lines in gs://assignment-1-gcs-bucket/input.txt: 20
```

*Screenshot 3: Terminal showing the execution of the script and its output*

The output matches the number of lines in the input file.



```
This is line 1.
This is line 2.
This is line 3.
This is line 4.
This is line 5.
This is line 6.
This is line 7.
This is line 8.
This is line 9.
This is line 10.
This is line 11.
This is line 12.
This is line 13.
This is line 14.
This is line 15.
This is line 16.
This is line 17.
This is line 18.
This is line 19.
This is line 20.
```

*Screenshot 4: File showing the number of lines*

## 3. Code Explanation

The Python script consists of two main functions:

### 3.1 count_lines_in_gcs(bucket_name: str, blob_name: str) -> int

This function is responsible for connecting to GCS, downloading the specified file, and counting its lines. Here's a breakdown of its operations:

1. Initialize the GCS client using `storage.Client()`. It uses the default authentication method, which assumes that the VM has the necessary permissions to access the GCS bucket.
2. Get the specified bucket and blob (file) using the provided names.
3. Download the file content as text using `blob.download_as_text()`.
4. Split the text content into a list of lines using `splitlines()`. This method handles different line ending characters (`\\n`, `\\r`, or `\\r\\n`) automatically.
5. Count the number of lines using `len()`.
6. Return the line count.

### 3.2 main()

This function serves as the entry point of the script:

1. Define the bucket name and blob name (file path).
2. Call `count_lines_in_gcs()` with the specified bucket and blob names.
3. Print the result or any error messages.