Week 2 Assignment

Virtual Machine in GCP

Submitted By: Anhat Singh (21F2000381)

Contents

List of Figures

1 Objective

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

2 Steps followed

2.1 Create a Google Cloud Account

- Go to google cloud platform, and create an account with an email ID.
- Add the payment details to start the free trial.
- Created a new project named `ibd-sept2024`.

2.2 Install and setup gcloud-cli on local system

To facilitate using the cloud platform, I chose to use the gcloud cli considering its simplicity and speed.

1. To install the CLI on Ubuntu/WSL, I ran the following commands in a linux shell:

```
$ sudo apt-get update
$ sudo apt-get install apt-transport-https ca-certificates gnupg curl
$ curl https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo gpg
↪  --dearmor -o /usr/share/keyrings/cloud.google.gpg
$ echo "deb [signed-by=/usr/share/keyrings/cloud.google.gpg]
↪  https://packages.cloud.google.com/apt cloud-sdk main" | sudo tee -a
↪  /etc/apt/sources.list.d/google-cloud-sdk.list
$ sudo apt-get update && sudo apt-get install google-cloud-cli
```

2. After successfully installing the gcloud cli, I ran the following command to configure it for local usage:

   ```
   $ gcloud init
   ```

3. This gives a link to open in a browser to login into the gcloud account. I copied the link, pasted in a browser and logged-in into my Google Cloud account.

4. After logging-in to my account, on the terminal window, I selected the project `ibd-sept2024` I wanted to use in my gcloud-cli

5. Then, I set the region and the zone to `asia-south1` and `asia-south1-a` respectively, using the following commands:

   ```
   $ gcloud compute project-info add-metadata \
   --metadata google-compute-default-region=asia-south1,\
   google-compute-default-zone=asia-south1-a
   ```

6. Reran the `gcloud init` command and refollow the steps 2-4 again.

7. Ran the following command to allow access to python scripts to access my cloud resources:

   ```
   $ gcloud auth application-default login
   ```

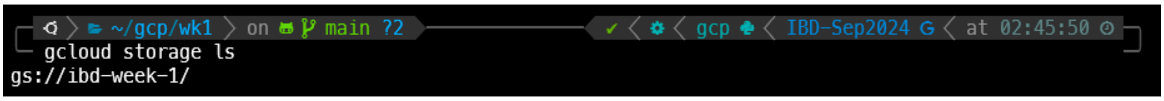
   This again gives a link to open in a browser to login into the gcloud account. I copied the link, pasted in a browser and logged-in into my Google Cloud account.

2.3 Create a Google Storage Bucket

1. I ran the following command to create a storage bucket in gcloud (running it for the first time ever will ask to enable Storage API first. Typed "Y" and pressed enter to enable them):

   ```
   $ gcloud storage buckets create gs://ibd-week-1 --location=asia-south1
   ```

2. After successful creation, the following command generated the list of buckets.



```
~/gcp/wk1 on main ?2 ──────── ✓ gcp IBD-Sep2024 at 02:45:50
gcloud storage ls
gs://ibd-week-1/
```

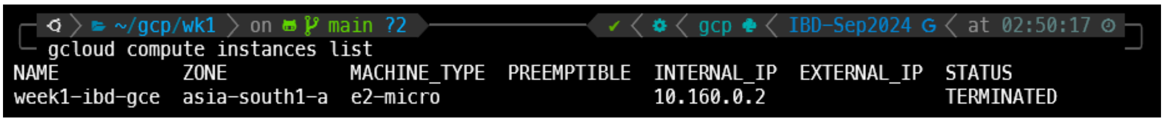
Figure 1: List of buckets in the Google Cloud

2.4 Create a Compute Engine VM

1. I ran the following command to create a virtual machine in gcloud (running it for the first time ever will ask to enable GCE API first. Typed "Y" and pressed enter to enable them):

```
$ gcloud compute instances create week1-ibd-gce --machine-type e2-micro
```

2. After successful creation, the following command generated the list of instances.

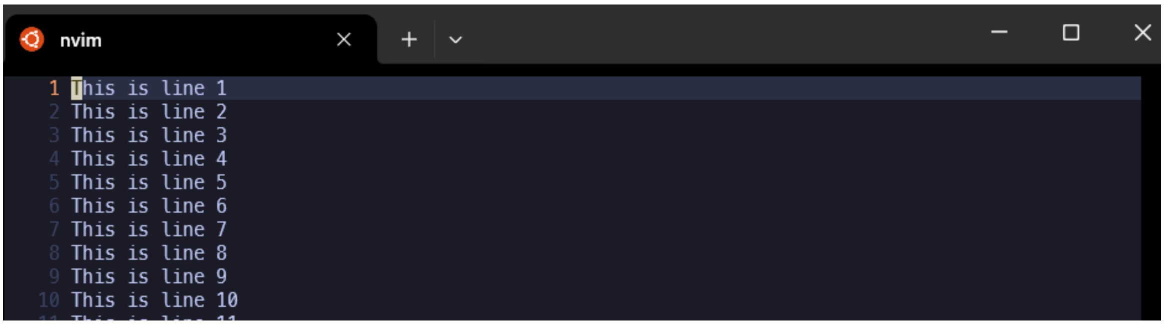


```
~/gcp/wk1 on main ?2                                ✓ gcp IBD-Sep2024 at 02:50:17
gcloud compute instances list
NAME           ZONE           MACHINE_TYPE  PREEMPTIBLE  INTERNAL_IP  EXTERNAL_IP  STATUS
week1-ibd-gce  asia-south1-a  e2-micro                   10.160.0.2                TERMINATED
```

Figure 2: List of GCE instance in the Google Cloud

2.5 Create a dummy input file and upload to the bucket

1. To use as input to my Python program later on, I used a dummy txt file having the following content in it:



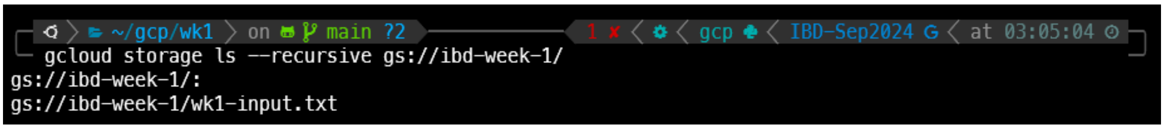
```
This is line 1
This is line 2
This is line 3
This is line 4
This is line 5
This is line 6
This is line 7
This is line 8
This is line 9
This is line 10
```

Figure 3: Dummy text file used for input to final code

2. To upload to it to my bucket, I used the following command:

```
$ gcloud storage cp wk1-input.txt gs://ibd-week-1
```

3. This successfully added the file into my GCP Storage bucket:



```
~/gcp/wk1 on main ?2                              1 ✗ gcp IBD-Sep2024 at 03:05:04
gcloud storage ls --recursive gs://ibd-week-1/
gs://ibd-week-1/:
gs://ibd-week-1/wk1-input.txt
```

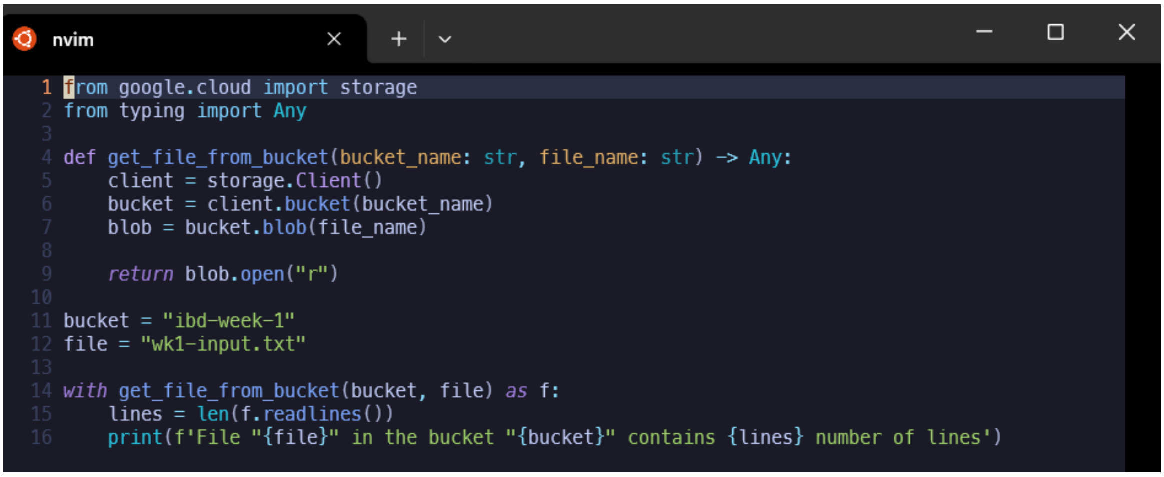
Figure 4: List of objects in my bucket

2.6 Write the Python code locally and upload it to the VM

1. Installed the `google-cloud-storage` python dependency using pip

```
$ pip install google-cloud-storage
```

2. Created the following python script to run the code:



```python
from google.cloud import storage
from typing import Any

def get_file_from_bucket(bucket_name: str, file_name: str) -> Any:
    client = storage.Client()
    bucket = client.bucket(bucket_name)
    blob = bucket.blob(file_name)

    return blob.open("r")

bucket = "ibd-week-1"
file = "wk1-input.txt"

with get_file_from_bucket(bucket, file) as f:
    lines = len(f.readlines())
    print(f'File "{file}" in the bucket "{bucket}" contains {lines} number of lines')
```

Figure 5: Python code to count the number of lines in my bucket

3. Saved the file as `code.py` and ran the following command to copy it into my GCE VM:

```
$ gcloud compute scp ./code.py week1-ibd-gce:~/wk1/code.py
```

2.7 Run the Python code on the VM

1. SSH into the GCE VM using the following command:

```
$ gcloud compute ssh week1-ibd-gce
```

If running for the first time, this will ask for a SSH paraphrase. I left it blank, as I don't care about vm security for now.

2. Installed Python, pip and the dependencies to run my python code:

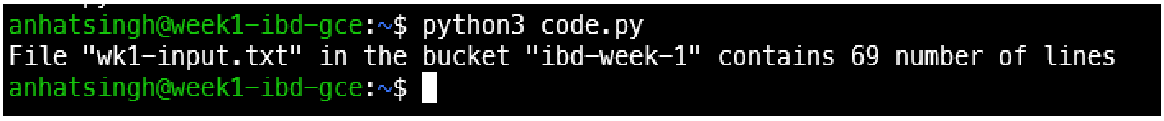
```
$ sudo apt update
$ sudo apt install python pip
$ pip install google-cloud-storage
```

3. Run the code:

```
$ python3 ~/wk1/code.py
```

3 Output

If the code successfully runs, the following output is generated on the console:



```
anhatsingh@week1-ibd-gce:~$ python3 code.py
File "wk1-input.txt" in the bucket "ibd-week-1" contains 69 number of lines
anhatsingh@week1-ibd-gce:~$
```

Figure 6: Output of the Python code generated