Introduction to Big Data

Jan 2025 Term – Graded Assignment 1

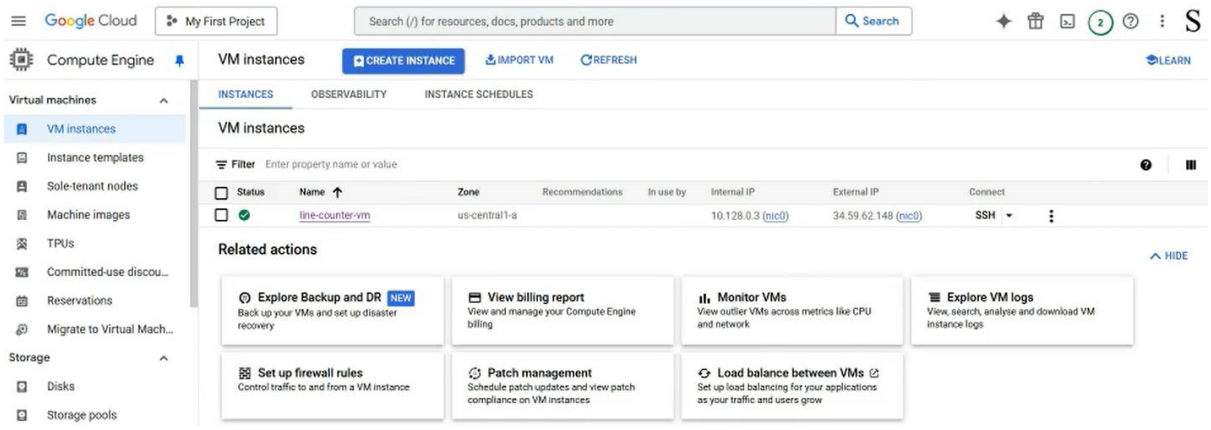
Name :- Satvik Chandrakar

Roll no :- 21f1000344

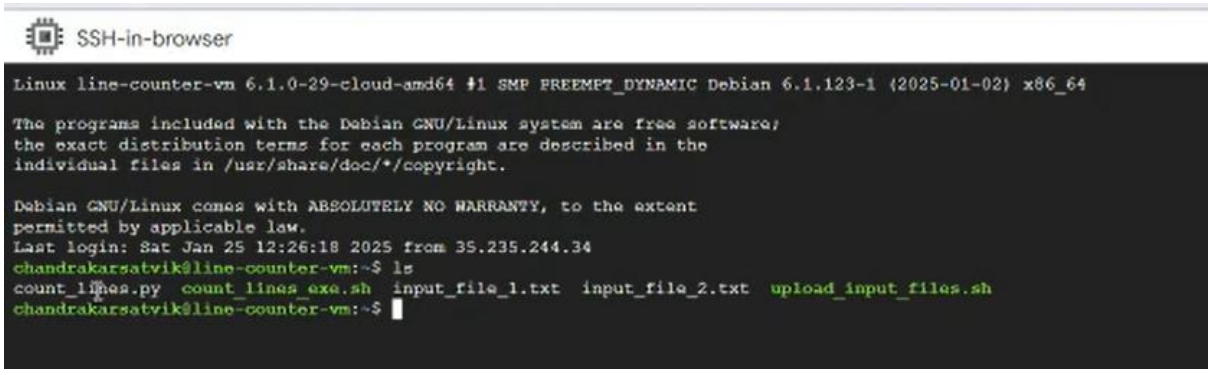
Task :- Spin up a VM and write a python program to count lines of a file placed in GCS.

My Approach :-

- Step 0 :- I created an account in the Google Cloud Platform(GCP) and read about how to set up a virtual machine in the GCP. And also designed the code to count the number of lines in a given file.
- Step 1 :- Logged into the GCP console and created a new VM instance.
    - Name :- line-counter-vm
    - Region :- us-central1-a
    - Machine :- e2-micro (for cost efficiency)
    - Boot disk :- Debian, Debian GNU/Linux

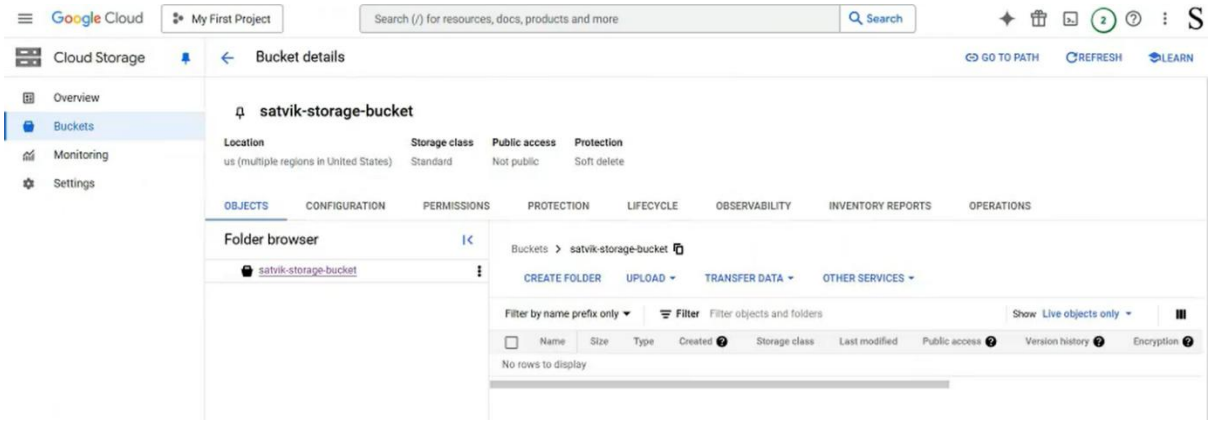


(Fig 1 :- VM Instance)



(Fig 2 :- SSH into the VM)

- Step 2 :- Initialized a storage bucket in the Google Cloud Storage (GCS) with the name satvik-storage-bucket and rest of the configurations in the default setting.



(Fig 3 :- GCS Bucket)

- Step 3 :- Wrote the python program to count the number of lines in the given file
  - SSH into the VM and created a python script.

    $ *nano count_lines.py*
  - Added the python code to count_lines.py
  - Saved it and exited the text editor

    CTRL + X -> Y -> ENTER

SSH-in-browser

```
 GNU nano 7.2                                                    count_lines.py
import sys

def count_lines(file_path):
    output_file = "output_file.txt"

    try:
        with open(file_path, 'r') as file:
            line_count = sum(1 for line in file)

        result = f"Total lines in '{file_path}': {line_count}\n"
        print(result.strip())  # Print to console

        # Create the output file if it doesn't exist and write the output
        with open(output_file, 'w') as out_file:
            out_file.write(result)

        print(f"Output written to '{output_file}'")

    except FileNotFoundError:
        print(f"Error: The file '{file_path}' was not found.")
    except Exception as e:
        print(f"An error occurred: {e}")

if __name__ == "__main__":
    if len(sys.argv) != 2:
        print("Please entre the input file. Usage: python count_lines.py <file_path>")
    else:
        count_lines(sys.argv[1])
```

(Fig 4 :- count_lines.py)

- Step 4 :- Wrote the upload_input_files.sh script to upload both the input files into the GCS from the VM and then remove them from the VM
  - SSH into the VM and created a shell script

    $ *nano upload_input_files.sh*

- Added the bash commands to the script
- Saved it and exited the text editor

CTRL + X -> Y -> ENTER

SSH-in-browser

```
  GNU nano 7.2                                                upload_input_files.sh
#!/bin/bash
gsutil cp /home/chandrakarsatvik/input_file_1.txt gs://satvik-storage-bucket/
gsutil cp /home/chandrakarsatvik/input_file_2.txt gs://satvik-storage-bucket/
rm input_file_1.txt input_file_2.txt
```

(Fig 5 :- upload_input_files.sh)

- Step 5 :- Wrote the count_lines_exe.sh script to download the input file based on the input file number parameter, count the number of lines in the input file and generate the output_file.txt, remove any preexisting output_file.txt from GCS, upload the newly generated output_file.txt to the GCS and remove the input file and output_file from the VM.
  - SSH into the VM and created a shell script

    $ *nano count_lines_exe.sh*
  - Added the bash commands to the script
  - Saved it and exited the text editor

    CTRL + X -> Y -> ENTER

SSH-in-browser

```
  GNU nano 7.2                                                count_lines_exe.sh
#!/bin/bash
if [[ "$1" == "1" || "$1" == "2" ]]
then
  gsutil cp gs://satvik-storage-bucket/input_file_$1.txt /home/chandrakarsatvik/ #Download the input_file.txt from the Google Cloud Storage
  python3 count_lines.py input_file_$1.txt #Run the python script to count the number of lines in the input_file
  gsutil rm gs://satvik-storage-bucket/output_file.txt #Remove the any preexisting output_file.txt
  gsutil cp /home/chandrakarsatvik/output_file.txt gs://satvik-storage-bucket/ #Upload the output_file.txt to the GCS
  rm output_file.txt input_file_$1.txt #Remove the files after processing them
  echo "Successfully completed the process"
else
  echo "Please entre the valid  input file number (1 or 2). Usage ./count_lines_exe.sh 1"
fi
```

(Fig 6 :- count_lines_exe.sh)

- Step 6 :- Downloaded the required dependencies
  - SSH into the VM
  - Installed the Google cloud SDK

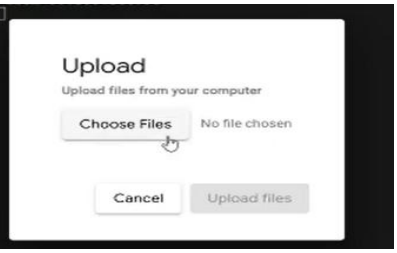
    $ sudo apt update && sudo apt install -y google-cloud-sdk
  - Authenticated GCP CLI in the VM

    $ gcloud auth login

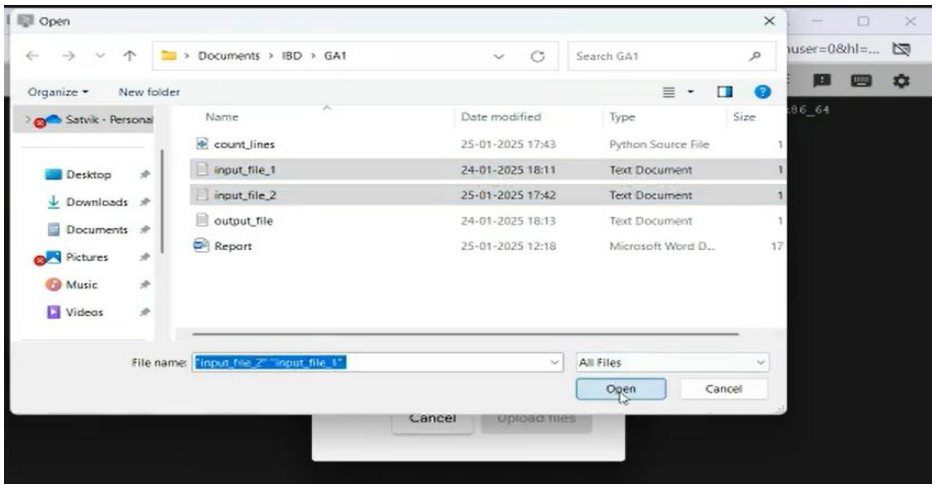
    $ gcloud config set project eminent-crane-448810-s3
  - Installed the python

    $ sudo apt update && sudo apt install -y python3
- Step 7 :- Uploaded the input_file_1.txt and input_file_2.txt to the VM from my local machine

(Fig 7.1 :- Click on UPLOAD FILE)

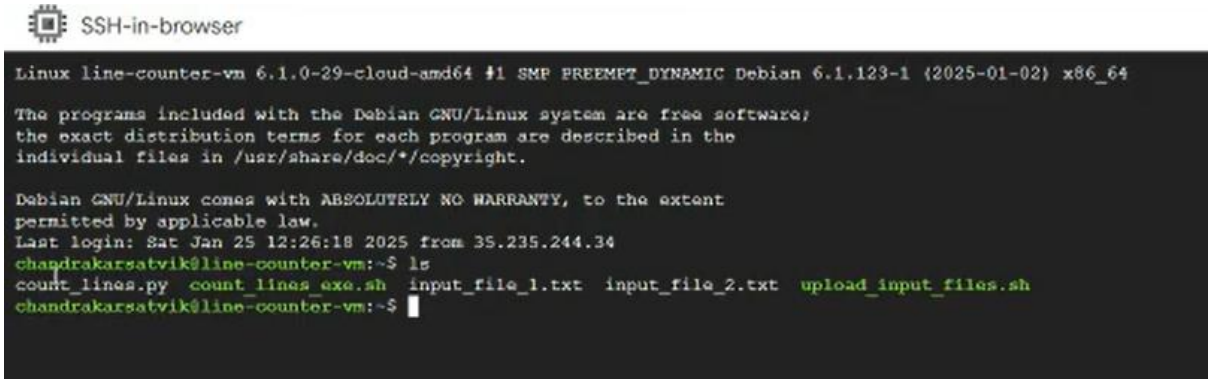


(Fig 7.2 :- Click on Choose Files)



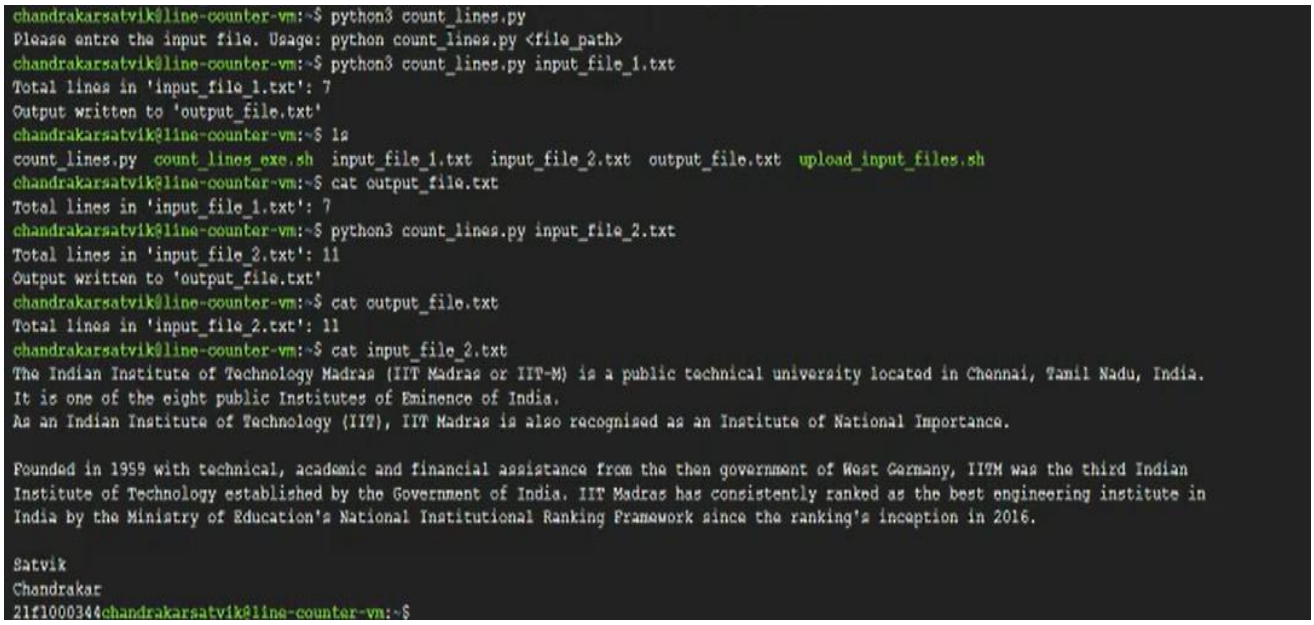
(Fig 7.3 :- Selected both the input files)

(Fig 7.4 :- Click Upload files)



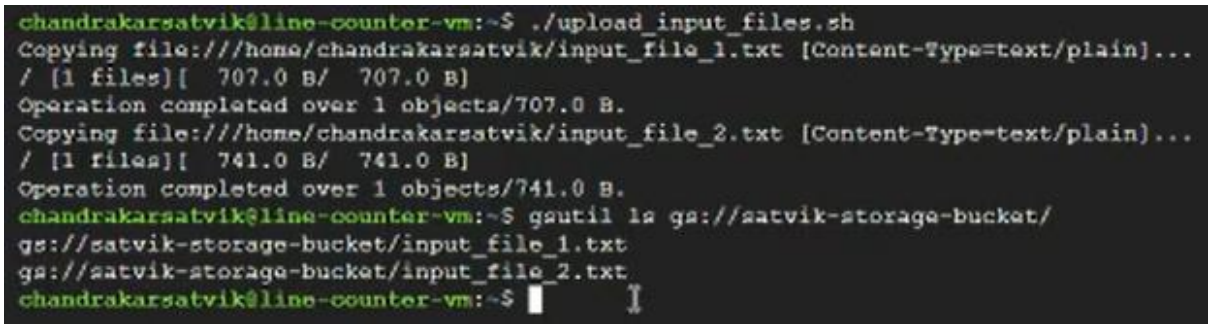
(Fig 7.5 :- list of all the files in currently in the VM)

- Step 8 :- Test Run. Executing the count_lines.py using the input files in the VM.



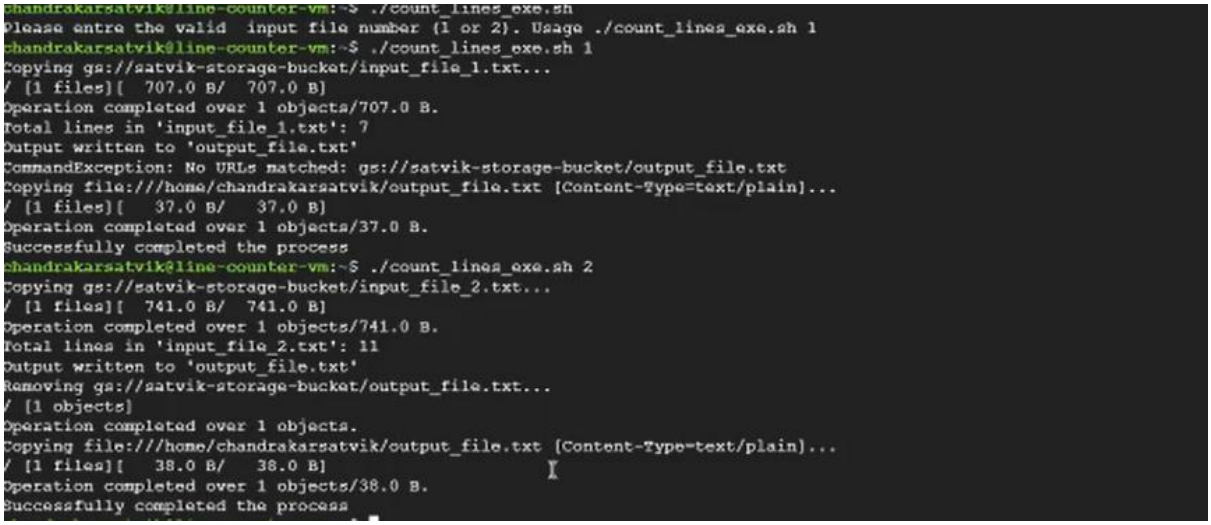
(Fig 8 :- Running the count_lines.py by passing the input files in the VM to test its working)

- Step 9 :- Uploaded both the input files to the GCS bucket and removed them from the VM by running *./upload_input_files.sh* and then executed the command *gsutil ls gs://satvik-storage-bucket/* to check whether they were successfully uploaded to the GCS bucket or not.



(Fig 9 :- Uploaded the input files to the GCS bucket)

- Step 10 :- Final Run. Executed the command *./count_lines_exe.sh 1* to fetch the input_file_1.txt, count the number of lines it in, wrote the output to output_file.txt and uploaded the output_file.txt to GCS bucket. Then executed the command *./count_lines_exe.sh 2* for input_file_2.txt file.



(Fig 10.1 :- Execution of the python program to count lines of the file placed in the GCS)

(Fig 10.2 :- output_file.txt)

Note :- First ./count_lines_exe.sh 1 was implemented. It removed any preexisting output_file.txt from the GCS bucket, which there was none hence the message “CommandException: No URLs matched…...” is displayed. Then it uploaded the newly generated output_file.txt in the GCS bucket. ./count_lines_exe.sh commands can be seen in the Fig 6.

Then ./count_lines_exe.sh 2 was implemented and it removed the output_file.txt from the GCS bucket which was uploaded during the ./count_lines_exe.sh 2 implementation. It uploaded the output_file.txt containing the line “Total lines in ‘input_file_2.txt’: 11” to the google cloud storage.

----------X----------