GA-6

Report on Counting Lines in a File Using Google Cloud Functions and Pub/Sub

Due Date: 2024-08-07, 23:59 IST

Objective: To create a system that counts the number of lines in a file in real-time using Google Cloud Functions (GCF) and Google Cloud Pub/Sub. The solution involves writing a Google Cloud Function that triggers on file upload to a Google Cloud Storage bucket, publishes the file name to a Pub/Sub topic, and a Python program that subscribes to that Pub/Sub topic, reads the file, counts the lines, and prints the result.

---

1. Setup and Configuration

Google Cloud Storage (GCS) Bucket:

- Created a Google Cloud Storage bucket (`ibd-ga6-bucket`) to store the input files.

Google Cloud Pub/Sub:

- Created a Pub/Sub topic (`ibd-ga6-topic`).
- Created a Pub/Sub subscription (`ibd-ga6-subscription`) for the topic.

Google Cloud Function (GCF):

- Developed a Google Cloud Function that is triggered by a file upload to the GCS bucket.
- The function publishes the name of the uploaded file to the Pub/Sub topic.

Python Subscriber Script:

- Implemented a Python program that subscribes to the Pub/Sub topic, receives the file name, and processes the file to count its lines.

---

2. Google Cloud Function Code

Function Description: The function is triggered by an event in Google Cloud Storage. Upon receiving the event, it extracts the file name and publishes a message containing the file name to the Pub/Sub topic.

```
from google.cloud import pubsub_v1
import os

def publish_file_name(event, context):
    file_name = event['name']
    topic_name = 'projects/your-project-id/topics/ibd-ga6-topic'

    publisher = pubsub_v1.PublisherClient()
    publisher.publish(topic_name, file_name.encode('utf-8'))
    print(f'Published message for file: {file_name}')
```

Deployment:

- The Google Cloud Function is deployed with the trigger set to the specified GCS bucket.

---

3. Python Subscriber Script

Script Description: The Python script subscribes to the Pub/Sub topic, receives the file name, retrieves the file from GCS, counts the number of lines, and prints the count.

```
from google.cloud import pubsub_v1, storage
import io

def message_handler(message):
    file_name = message.data.decode('utf-8')
    process_file(file_name)
    message.ack()

def process_file(file_name):
    storage_client = storage.Client()
    bucket = storage_client.bucket('ibd-ga6-bucket')
    blob = bucket.blob(file_name)

    with io.BytesIO(blob.download_as_bytes()) as file:
        line_count = sum(1 for _ in file)
        print(f'The file {file_name} has {line_count} lines.')

def main():
```

```
    subscriber = pubsub_v1.SubscriberClient()
    subscription_path =
'projects/your-project-id/subscriptions/ibd-ga6-subscription'

    streaming_pull_future = subscriber.subscribe(subscription_path,
callback=message_handler)
    print('Listening for messages on {}'.format(subscription_path))

    try:
        streaming_pull_future.result()
    except KeyboardInterrupt:
        streaming_pull_future.cancel()

if __name__ == '__main__':
    main()
```

Execution:

- The Python script is executed in an environment with the appropriate permissions to access Pub/Sub and GCS.

---

4. Testing

Testing Steps:

1. Upload a test file (`tiny-shakespeare.txt`) to the GCS bucket.
2. Verify that the Google Cloud Function is triggered and publishes the file name to the Pub/Sub topic.
3. Ensure that the Python script receives the message, retrieves the file, counts the lines, and prints the result.

Results:

- The system successfully counts the number of lines in the uploaded file and prints the count.

---

5. Issues and Resolutions

Issue 1: File Not Found (404 Error)

- Error: `No such object: ibd-ga6-bucket/gs://ibd-ga6-bucket/tiny-shakespeare.txt`
- Resolution: Ensure that the file path is correct and that the file exists in the bucket. The path should not include the `gs://` prefix when accessing the file within the Python script.

Issue 2: Project Not Found (404 Error)

- Error: `Requested project not found or user does not have access to it (project=ibd-ga6-pubsub)`
- Resolution: Verify that the correct project ID is used and that the necessary permissions are granted to access the Pub/Sub topic and GCS bucket.

---

6. Conclusion

The solution effectively uses Google Cloud Functions and Pub/Sub to create a real-time file processing system. The Google Cloud Function publishes the file name to Pub/Sub upon upload, and the Python script processes the file by counting the lines and printing the result. The implementation meets the requirements and handles the specified use case.