Airflow Assignment: GCP Dataproc PySpark Job

Objective: Automate a workflow using Apache Airflow to process daily incoming CSV files from a GCP bucket using a Dataproc PySpark job and save the transformed data into a Hive table.

Tasks:

1. **Setup:**
   - Create a Google Cloud Platform (GCP) bucket to store the daily CSV files.
   - Set up an Apache Airflow environment and ensure GCP and Dataproc plugins/hooks are available.
2. **DAG Configuration:**
   - Create a new DAG gcp_dataproc_pyspark_dag.
   - Schedule the DAG to run once a day.
   - Ensure catchup is set to False: catchup=False.
3. **File Sensor Task:**
   - Add a GCSObjectExistenceSensor task to check for the presence of the daily CSV file in the GCP bucket.
   - Configure the task to poke for the file every 5 minutes for a maximum of 12 hours.
4. **Dataproc Cluster Creation Task:**
   - Use the DataprocClusterCreateOperator to create a new Dataproc cluster.
   - Define and configure the cluster specifications as needed.
5. **PySpark Job Execution Task:**

- Upload your PySpark script to GCP (either in a bucket or Cloud Storage).
- Use the DataProcPySparkOperator to execute the PySpark script on the created Dataproc cluster.
- The PySpark script should:
  - Read the daily CSV file from the GCP bucket.
  - Perform some logical transformations on the data.
  - Write the transformed data into a Hive table.

6. Dataproc Cluster Deletion Task:

- Use the DataprocClusterDeleteOperator to delete the Dataproc cluster once the PySpark job is successfully completed.

7. DAG Dependency Configuration:

- Set the task dependencies using the set_upstream and set_downstream methods or the bitshift operators (>> and <<).
- Ensure that the DAG tasks run in the correct sequence.

Evaluation Criteria:

- Proper configuration and structuring of the Airflow DAG.
- Successful execution and scheduling of the DAG.
- Correct sensing of the daily CSV file.
- Successful creation and deletion of the Dataproc cluster.
- Successful execution of the PySpark job with the desired transformation.
- Proper writing of the transformed data to the Hive table.

Tips:

- Remember to configure the necessary GCP connection in the Airflow web UI.
- Ensure you handle exceptions and potential issues in the workflow, such as cluster creation failures or script execution errors.
- Log important steps and outputs for easier debugging.

Submission:

- Submit the DAG python file (gcp_dataproc_pyspark_dag.py).
- Provide a brief report explaining the workflow, any challenges faced, and their solutions.
- Include screenshots of the successful DAG runs and the resulting data in the Hive table.