PROFESSIONAL-DATA-ENGINEER Q&As

Professional Data Engineer on Google Cloud Platform

Pass Google PROFESSIONAL-DATA-ENGINEER Exam with 100% Guarantee

Free Download Real Questions & Answers **PDF** and **VCE** file from:

https://www.passapply.com/professional-data-engineer.html

100% Passing Guarantee
100% Money Back Assurance

Following Questions and Answers are all new published by Google Official Exam Center

QUESTION 1

You are running a streaming pipeline with Dataflow and are using hopping windows to group the data as the data arrives. You noticed that some data is arriving late but is not being marked as late data, which is resulting in inaccurate

aggregations downstream. You need to find a solution that allows you to capture the late data in the appropriate window.

What should you do?

A. Change your windowing function to session windows to define your windows based on certain activity.

B. Change your windowing function to tumbling windows to avoid overlapping window periods.

C. Expand your hopping window so that the late data has more time to arrive within the grouping.

D. Use watermarks to define the expected data arrival window Allow late data as it arrives.

Correct Answer: D

Watermarks are a way of tracking the progress of time in a streaming pipeline. They are used to determine when a window can be closed and the results emitted. Watermarks can be either event-time based or processing-time based. Event-time watermarks track the progress of time based on the timestamps of the data elements, while processing-time watermarks track the progress of time based on the system clock. Event-time watermarks are more accurate, but they require the data source to provide reliable timestamps. Processing-time watermarks are simpler, but they can be affected by system delays or backlogs. By using watermarks, you can define the expected data arrival window for each windowing function. You can also specify how to handle late data, which is data that arrives after the watermark has passed. You can either discard late data, or allow late data and update the results as new data arrives. Allowing late data requires you to use triggers to control when the results are emitted. In this case, using watermarks and allowing late data is the best solution to capture the late data in the appropriate window. Changing the windowing function to session windows or tumbling windows will not solve the problem of late data, as they still rely on watermarks to determine when to close the windows. Expanding the hopping window might reduce the amount of late data, but it will also change the semantics of the windowing function and the results. References: Streaming pipelines | Cloud Dataflow | Google Cloud Windowing | Apache Beam

QUESTION 2

You want to schedule a number of sequential load and transformation jobs Data files will be added to a Cloud Storage bucket by an upstream process There is no fixed schedule for when the new data arrives Next, a Dataproc job is triggered to perform some transformations and write the data to BigQuery. You then need to run additional transformation jobs in BigQuery The transformation jobs are different for every table These jobs might take hours to complete You need to determine the most efficient and maintainable workflow to process hundreds of tables and provide the freshest data to your end users. What should you do?

A. 1Create an Apache Airflow directed acyclic graph (DAG) in Cloud Composer with sequential tasks by using the Cloud Storage. Dataproc. and BigQuery operators 2 Use a single shared DAG for all tables that need to go through the pipeline 3 Schedule the DAG to run hourly

B. 1 Create an Apache Airflow directed acyclic graph (DAG) in Cloud Composer with sequential tasks by using the Dataproc and BigQuery operators. 2 Create a separate DAG for each table that needs to go through the pipeline 3 Use a Cloud Storage object trigger to launch a Cloud Function that triggers the DAG

C. 1 Create an Apache Airflow directed acyclic graph (DAG) in Cloud Composer with sequential tasks by using the

Cloud Storage, Dataproc. and BigQuery operators 2 Create a separate DAG for each table that needs to go through the pipeline 3 Schedule the DAGs to run hourly

D. 1 Create an Apache Airflow directed acyclic graph (DAG) in Cloud Composer with sequential tasks by using the Dataproc and BigQuery operators 2 Use a single shared DAG for all tables that need to go through the pipeline. 3 Use a Cloud Storage object trigger to launch a Cloud Function that triggers the DAG

Correct Answer: B

This option is the most efficient and maintainable workflow for your use case, as it allows you to process each table independently and trigger the DAGs only when new data arrives in the Cloud Storage bucket. By using the Dataproc and BigQuery operators, you can easily orchestrate the load and transformation jobs for each table, and leverage the scalability and performance of these services12. By creating a separate DAG for each table, you can customize the transformation logic and parameters for each table, and avoid the complexity and overhead of a single shared DAG3. By using a Cloud Storage object trigger, you can launch a Cloud Function that triggers the DAG for the corresponding table, ensuring that the data is processed as soon as possible and reducing the idle time and cost of running the DAGs on a fixed schedule4 . Option A is not efficient, as it runs the DAG hourly regardless of the data arrival, and it uses a single shared DAG for all tables, which makes it harder to maintain and debug. Option C is also not efficient, as it runs the DAGs hourly and does not leverage the Cloud Storage object trigger. Option D is not maintainable, as it uses a single shared DAG for all tables, and it does not use the Cloud Storage operator, which can simplify the data ingestion from the bucket. References:

1: Dataproc Operator | Cloud Composer | Google Cloud

2: BigQuery Operator | Cloud Composer | Google Cloud

3: Choose Workflows or Cloud Composer for service orchestration | Workflows | Google Cloud

4: Cloud Storage Object Trigger | Cloud Functions Documentation | Google Cloud [5]: Triggering DAGs | Cloud Composer | Google Cloud [6]: Cloud Storage Operator | Cloud Composer | Google Cloud

QUESTION 3

You need to compose visualizations for operations teams with the following requirements:

1.

The report must include telemetry data from all 50,000 installations for the most resent 6 weeks (sampling once every minute).

2.

The report must not be more than 3 hours delayed from live data.

3.

The actionable report should only show suboptimal links.

4.

Most suboptimal links should be sorted to the top.

5.

Suboptimal links can be grouped and filtered by regional geography.

6.

User response time to load the report must be