Free Questions for Professional-Data-Engineer

Shared by Huffman on 03-03-2025

For More Free Questions and Preparation Resources

Check the Links on Last Page

Question 1

Question Type: MultipleChoice

You want to migrate an Apache Spark 3 batch job from on-premises to Google Cloud. You need to minimally change the job so that the job reads from Cloud Storage and writes the result to BigQuery. Your job is optimized for Spark, where each executor has 8 vCPU and 16 GB memory, and you want to be able to choose similar settings. You want to minimize installation and management effort to run your job. What should you do?

Options:

A- Execute the job in a new Dataproc cluster.
B- Execute as a Dataproc Serverless job.
C- Execute the job as part of a deployment in a new Google Kubernetes Engine cluster.
D- Execute the job from a new Compute Engine VM.

Answer:

A

Question 2

Question Type: MultipleChoice

You migrated a data backend for an application that serves 10 PB of historical product data for analytics. Only the last known state for a product, which is about 10 GB of data, needs to be served through an API to the other applications. You need to choose a cost-effective persistent storage solution that can accommodate the analytics requirements and the API performance of up to 1000 queries per second (QPS) with less than 1 second latency. What should you do?

Options:

A- 1. Store the historical data in BigQuery for analytics.
2. In a Cloud SQL table, store the last state of the product after every product change.
3. Serve the last state data directly from Cloud SQL to the API.
B- 1. Store the historical data in Cloud SQL for analytics.
2. In a separate table, store the last state of the product after every product change.
3. Serve the last state data directly from Cloud SQL to the API.
C- 1. Store the products as a collection in Firestore with each product having a set of historical changes.

2. Use simple and compound queries for analytics.
3. Serve the last state data directly from Firestore to the API.

D- 1. Store the historical data in BigQuery for analytics.
2. Use a materialized view to precompute the last state of a product.
3. Serve the last state data directly from BigQuery to the API.

Answer:

D

Question 3

Question Type: MultipleChoice

You are preparing an organization-wide dataset. You need to preprocess customer data stored in a restricted bucket in Cloud Storage. The data will be used to create consumer analyses. You need to follow data privacy requirements, including protecting certain sensitive data elements, while also retaining all of the data for potential future use cases. What should you do?

Options:

A- Use Dataflow and the Cloud Data Loss Prevention API to mask sensitive data. Write the processed data in BigQuery.

B- Use the Cloud Data Loss Prevention API and Dataflow to detect and remove sensitive fields from the data in Cloud Storage. Write the filtered data in BigQuery.

C- Use Dataflow and Cloud KMS to encrypt sensitive fields and write the encrypted data in BigQuery. Share the encryption key by following the principle of least privilege.

D- Use customer-managed encryption keys (CMEK) to directly encrypt the data in Cloud Storage. Use federated queries from BigQuery. Share the encryption key by following the principle of least privilege.

Answer:

A

Question 4

Question Type: MultipleChoice

You are designing a messaging system by using Pub/Sub to process clickstream data with an event-driven consumer app that relies on a push subscription. You need to configure the messaging system that is reliable enough to handle temporary downtime of the consumer app.

You also need the messaging system to store the input messages that cannot be consumed by the subscriber. The system needs to retry failed messages gradually, avoiding overloading the consumer app, and store the failed messages after a maximum of 10 retries in a topic. How should you configure the Pub/Sub subscription?

Options:

A- Increase the acknowledgement deadline to 10 minutes.
B- Use immediate redelivery as the subscription retry policy, and configure dead lettering to a different topic with maximum delivery attempts set to 10.
C- Use exponential backoff as the subscription retry policy, and configure dead lettering to the same source topic with maximum delivery attempts set to 10.
D- Use exponential backoff as the subscription retry policy, and configure dead lettering to a different topic with maximum delivery attempts set to 10.

Answer:

D

Question 5

Question Type: MultipleChoice

The data analyst team at your company uses BigQuery for ad-hoc queries and scheduled SQL pipelines in a Google Cloud project with a slot reservation of 2000 slots. However, with the recent introduction of hundreds of new non time-sensitive SQL pipelines, the team is encountering frequent quota errors. You examine the logs and notice that approximately 1500 queries are being triggered concurrently during peak time. You need to resolve the concurrency issue. What should you do?

Options:

A- Update SQL pipelines and ad-hoc queries to run as interactive query jobs.
B- Increase the slot capacity of the project with baseline as 0 and maximum reservation size as 3000.
C- Update SOL pipelines to run as a batch query, and run ad-hoc queries as interactive query jobs.
D- Increase the slot capacity of the project with baseline as 2000 and maximum reservation size as 3000.

Answer:

C

Explanation:

To resolve the concurrency issue in BigQuery caused by the introduction of hundreds of non-time-sensitive SQL pipelines, the best approach is to differentiate the types of queries based on their urgency and resource requirements. Here's why option C is the best choice:

SQL Pipelines as Batch Queries:

Batch queries in BigQuery are designed for non-time-sensitive operations. They run in a lower priority queue and do not consume slots immediately, which helps to reduce the overall slot consumption during peak times.

By converting non-time-sensitive SQL pipelines to batch queries, you can significantly alleviate the pressure on slot reservations.

Ad-Hoc Queries as Interactive Queries:

Interactive queries are prioritized to run immediately and are suitable for ad-hoc analysis where users expect quick results.

Running ad-hoc queries as interactive jobs ensures that analysts can get their results without delay, improving productivity and user satisfaction.

Concurrency Management:

This approach helps balance the workload by leveraging BigQuery's ability to handle different types of queries efficiently, reducing the likelihood of encountering quota errors due to slot exhaustion.

Steps to Implement:

Identify Non-Time-Sensitive Pipelines:

Review and identify SQL pipelines that are not time-critical and can be executed as batch jobs.

Update Pipelines to Batch Queries:

Modify these pipelines to run as batch queries. This can be done by setting the priority of the query job to BATCH.

Ensure Ad-Hoc Queries are Interactive:

Ensure that all ad-hoc queries are submitted as interactive jobs, allowing them to run with higher priority and immediate slot allocation.

BigQuery Batch Queries

BigQuery Slot Allocation and Management

Question 6

Question Type: MultipleChoice

You are migrating your on-premises data warehouse to BigQuery. One of the upstream data sources resides on a MySQL database that runs in your on-premises data center with no public IP addresses. You want to ensure that the data ingestion into BigQuery is done securely and does not go through the public internet. What should you do?

Options:

A- Update your existing on-premises ETL tool to write to BigQuery by using the BigQuery Open Database Connectivity (ODBC) driver. Set up the proxy parameter in the Simba. googlebigqueryodbc. ini tile to point to your data center's NAT gateway.
B- Use Datastream to replicate data from your on-premises MySQL database to BigQuery. Gather Datastream public IP addresses of the Google Cloud region that will be used to set up the stream. Add those IP addresses to the firewall allowlist of your on-premises data center.
Use IP Allovlisting as the connectivity method and Server-only as the encryption type when setting up the connection profile in Datastream.
C- Use Datastream to replicate data from your on-premises MySQL database to BigQuery. Use Forward-SSH tunnel as the connectivity method to establish a secure tunnel between Datastream and your on-premises MySQL database through a tunnel server in your on-premises data center. Use None as the encryption type when setting up the connection profile in Datastream.
D- Use Datastream to replicate data from your on-premises MySQL database to BigQuery. Set up Cloud Interconnect between your on- premises data center and Google Cloud. Use Private connectivity as the connectivity method and allocate an IP address range within your VPC network to the Datastream connectivity configuration. Use Server-only as the encryption type when setting up the connection profile in Datastream.

Answer:

D

Explanation:

To securely ingest data from an on-premises MySQL database into BigQuery without routing through the public internet, using Datastream with Private connectivity over Cloud Interconnect is the best approach. Here's why:

Datastream for Data Replication:

Datastream provides a managed service for data replication from various sources, including on-premises databases, to Google Cloud services like BigQuery.

Cloud Interconnect:

Cloud Interconnect establishes a private connection between your on-premises data center and Google Cloud, ensuring that data transfer occurs over a secure, private network rather than the public internet.

Private Connectivity:

Using Private connectivity with Datastream leverages the established Cloud Interconnect to securely connect your on-premises MySQL database with Google Cloud. This method ensures that the data does not traverse the public internet.

Encryption:

Using Server-only encryption ensures that data is encrypted in transit between Datastream and BigQuery, adding an extra layer of security.

Steps to Implement:

Set Up Cloud Interconnect:

Establish a Cloud Interconnect between your on-premises data center and Google Cloud to create a private connection.

Configure Datastream:

Set up Datastream to use Private connectivity as the connection method and allocate an IP address range within your VPC network.

Use Server-only encryption to ensure secure data transfer.

Create Connection Profile:

Create a connection profile in Datastream to define the connection parameters, including the use of Cloud Interconnect and Private connectivity.

Datastream Documentation

Cloud Interconnect Documentation

Setting Up Private Connectivity in Datastream