🔍

⊖ **Google Discussions**

Exam Professional Data Engineer All Questions
View all questions & answers for the Professional Data Engineer exam

Go to Exam

📄 **EXAM PROFESSIONAL DATA ENGINEER TOPIC 1 QUESTION 208 DISCUSSION**

Actual exam question from Google's Professional Data Engineer

Question #: 208

Topic #: 1

[All Professional Data Engineer Questions]

A live TV show asks viewers to cast votes using their mobile phones. The event generates a large volume of data during a 3-minute period. You are in charge of the "Voting infrastructure" and must ensure that the platform can handle the load and that all votes are processed. You must display partial results while voting is open. After voting closes, you need to count the votes exactly once while optimizing cost. What should you do?
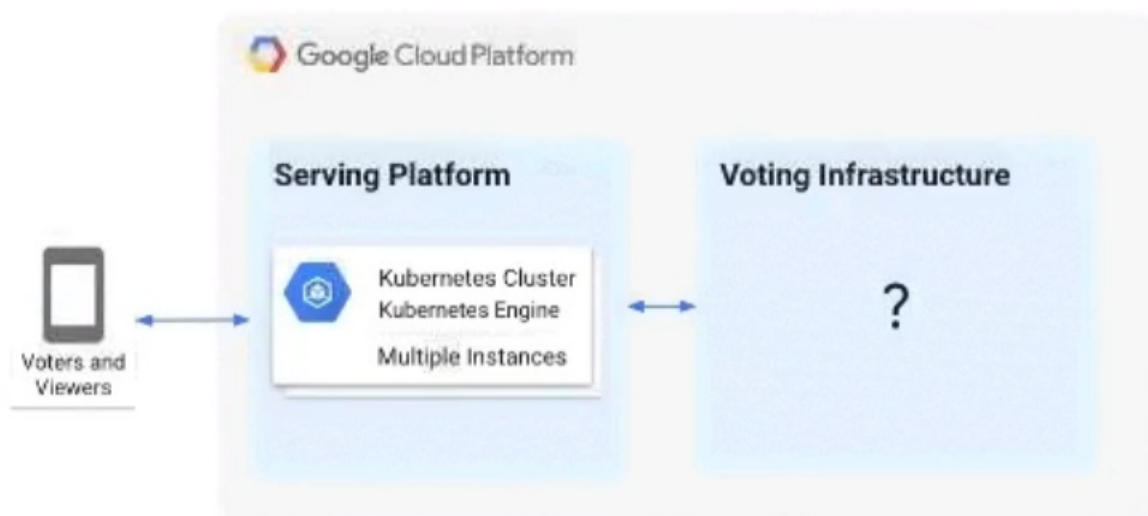
A. Create a Memorystore instance with a high availability (HA) configuration.

B. Create a Cloud SQL for PostgreSQL database with high availability (HA) configuration and multiple read replicas.

C. Write votes to a Pub/Sub topic and have Cloud Functions subscribe to it and write votes to BigQuery.

D. Write votes to a Pub/Sub topic and load into both Bigtable and BigQuery via a Dataflow pipeline. Query Bigtable for real-time results and BigQuery for later analysis. Shut down the Bigtable instance when voting concludes.

Show Suggested Answer

by 👤 **e70ea9e** at *Dec. 30, 2023, 9:30 a.m.*

Comments

    Type your comment...

Submit

⊟ 👤 **MaxNRG** `Highly Voted 👍` 10 months ago

`Selected Answer: D`

Since cost optimization and minimal latency are key requirements, option D is likely the best choice to meet all the needs:

The key reasons option D works well:

Using Pub/Sub to ingest votes provides scalable, reliable transport.

Loading into Bigtable and BigQuery provides both:

Low latency reads from Bigtable for real-time results.
Cost effective storage in BigQuery for longer term analysis.
Shutting down Bigtable after voting concludes reduces costs.

BigQuery remains available for cost-optimized storage and analysis.

So you are correct that option D combines the best of real-time performance for queries using Bigtable, with cost-optimized storage in BigQuery.

The only additional consideration may be if 3 minutes of Bigtable usage still incurs higher charges than ingesting directly into BigQuery. But for minimizing latency while optimizing cost, option D is likely the right architectural choice given the requirements.

👍 ↩ 🏳 upvoted 7 times

⊟ 👤 **f74ca0c** `Most Recent ⊘` 4 months, 1 week ago

`Selected Answer: C`

Modern Capabilities: BigQuery's advancements make it suitable for both real-time and historical querying.
Cost Efficiency: No need to spin up and shut down a Bigtable instance.
Simplified Workflow: Real-time and post-event data are stored in the same system, reducing the need to synchronize or transfer data between systems.

👍 ↩ 🏳 upvoted 2 times

⊟ 👤 **JyoGCP** 8 months, 3 weeks ago

`Selected Answer: D`

D. Write votes to a Pub/Sub topic and load into both Bigtable and BigQuery via a Dataflow pipeline. Query Bigtable for real-time results and BigQuery for later analysis. Shut down the Bigtable instance when voting concludes.

👍 ↩ 🏳 upvoted 1 times

⊟ 👤 **Matt_108** 9 months, 3 weeks ago

`Selected Answer: D`

D, i do agree with everything MaxNRG said.

👍 ↩ 🏳 upvoted 1 times

⊟ 👤 **Smakyel79** 10 months ago

`Selected Answer: C`

Pub/Sub for sure, and Cloud Functions + BigQuery Streaming seems a good solution. Won't use BigTable as need at least 100GB of data (don't thing a voting system could arrive to that amount of data) and needs to "heat" to work right for >10 minutes... and would be $$$ over C solution

👍 ↩ 🏳 upvoted 1 times

⊟ 👤 **raaad** 10 months, 1 week ago

1aaaa 10 months, 1 week ago

Answer is D:
- Google Cloud Pub/Sub can manage the high-volume data ingestion.
- Google Cloud Dataflow can efficiently process and route data to both Bigtable and BigQuery.
- Bigtable is excellent for handling high-throughput writes and reads, making it suitable for real-time vote tallying.
- BigQuery is ideal for exact vote counting and deeper analysis once voting concludes.

👍 ↩ ⚑ upvoted 2 times

⊟ 👤 **e70ea9e** 10 months, 1 week ago

Handling High-Volume Data Ingestion:

Pub/Sub: Decouples vote collection from processing, ensuring scalability and resilience under high load.
Dataflow: Efficiently ingests and processes large data streams, scaling as needed.
Real-Time Results with Exactly-Once Processing:

Bigtable: Optimized for low-latency, high-throughput reads and writes, ideal for real-time partial results.
Exactly-Once Semantics: Dataflow guarantees each vote is processed only once, ensuring accurate counts.
Cost Optimization:

Temporary Bigtable Instance: Running Bigtable only during voting minimizes costs.
BigQuery Storage: Cost-effective for long-term storage and analysis.

👍 ↩ ⚑ upvoted 3 times