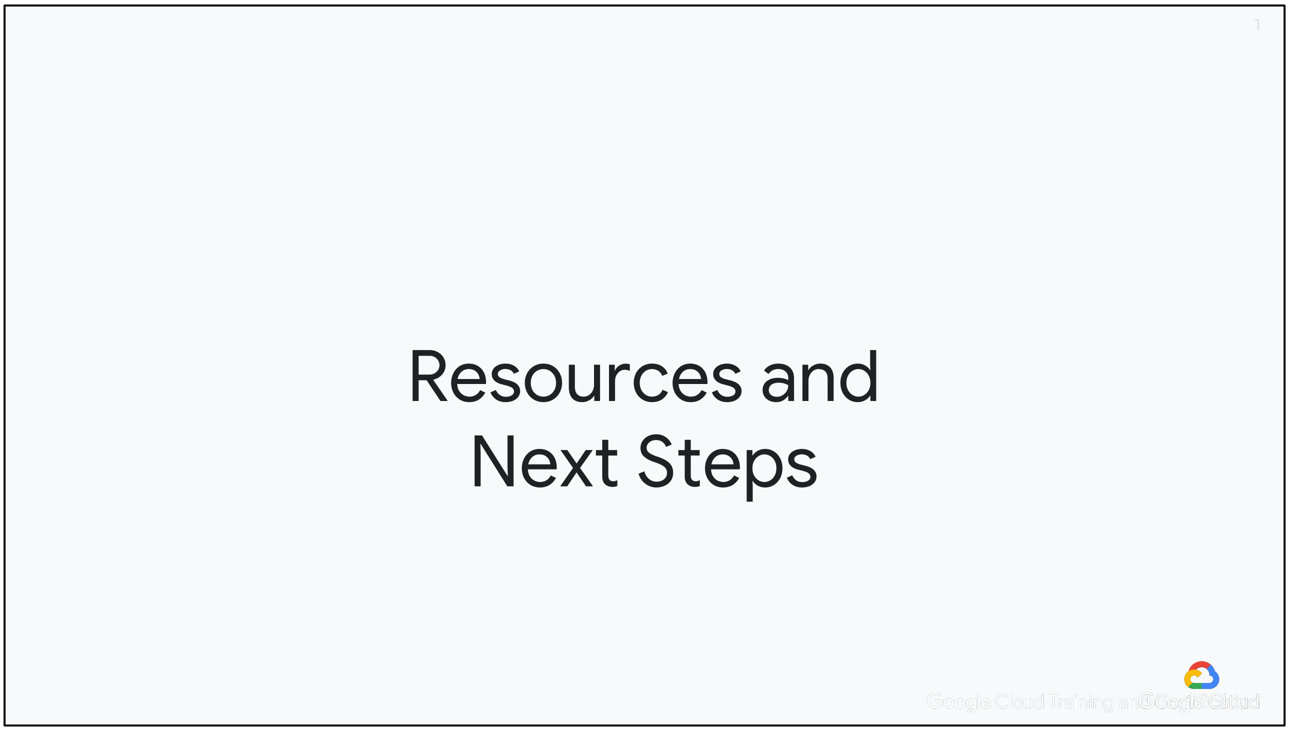


Resources and Next Steps

DEPE

Storage of JSON files with occasionally changing schema, for ANSI SQL queries.

A. Store in BigQuery. Provide format files for data load and update them as needed.
B. Store in BigQuery. Select "Automatically detect" in the Schema section.
C. Store in Cloud Storage. Link data as temporary tables in BigQuery and turn on the "Automatically detect" option in the Schema section of BigQuery.
D. Store in Cloud Storage. Link data as permanent tables in BigQuery and turn on the "Automatically detect" option in the Schema section of BigQuery.

DEPE

Storage of JSON files with occasionally changing schema, for ANSI SQL queries.

A. Store in BigQuery. Provide format files for data load and update them as needed.
B. Store in BigQuery. Select "Automatically detect" in the Schema section. ✓
C. Store in Cloud Storage. Link data as temporary tables in BigQuery and turn on the "Automatically detect" option in the Schema section of BigQuery.
D. Store in Cloud Storage. Link data as permanent tables in BigQuery and turn on the "Automatically detect" option in the Schema section of BigQuery.

Solution

B is correct because of the requirement to support occasionally (schema) changing JSON files and aggregate ANSI SQL queries: you need to use BigQuery, and it is quickest to use 'Automatically detect' for schema changes.

A is not correct because you should not provide format files: you can simply turn on the 'Automatically detect' schema changes flag.

C and D are not correct because you should not use Cloud Storage for this scenario: it is cumbersome and doesn't add value.

https://cloud.google.com/bigquery/docs/schemas
https://cloud.google.com/bigquery/docs/schema-detect
https://cloud.google.com/bigquery/docs/loading-data

DEPE

Low-cost one-way one-time migration of two 100-TB file servers to GCP;
data will only be accessed from Germany.

A. Use Transfer Appliance. Transfer to a Cloud Storage Regional storage bucket.
B. Use Transfer Appliance. Transfer to a Cloud Storage Multi-Regional bucket.
C. Use Storage Transfer Service. Transfer to a Cloud Storage Regional bucket.
D. Use Storage Transfer Service. Transfer to a Cloud Storage Multi-Regional bucket.

DEPE

Low-cost one-way one-time migration of two 100-TB file servers to GCP; data will only be accessed from Germany.

A. Use Transfer Appliance. Transfer to a Cloud Storage Regional storage bucket. ✓
B. Use Transfer Appliance. Transfer to a Cloud Storage Multi-Regional bucket.
C. Use Storage Transfer Service. Transfer to a Cloud Storage Regional bucket.
D. Use Storage Transfer Service. Transfer to a Cloud Storage Multi-Regional bucket.

Solution

A is correct because you are performing a one-time (rather than an ongoing series) data transfer from on-premises to Google Cloud Platform for users in a single region (Germany). Using a Regional storage bucket will reduce cost and also conform to regulatory requirements.

B, C, and D are not correct because you should only use Transfer Service for a one-time one-way transfer (C,D) and because you should not use a Multi-Regional storage bucket for users in a single region (B,D). Also, Storage Transfer Service does not work for data stored on-premises..

https://cloud.google.com/storage-transfer/docs/overview
https://cloud.google.com/storage-transfer/docs/
https://cloud.google.com/transfer-appliance/
https://cloud.google.com/transfer-appliance/docs/2.0/concepts

DEPE

Cost-effective backup to GCP of multi-TB databases from another cloud including monthly DR drills.

A. Use Transfer Appliance. Transfer to Cloud Storage Nearline bucket.
B. Use Transfer Appliance. Transfer to Cloud Storage Coldline bucket.
C. Use Storage Transfer Service. Transfer to Cloud Storage Nearline bucket.
D. Use Storage Transfer Service. Transfer to Cloud Storage Coldline bucket.

DEPE

Cost-effective backup to GCP of multi-TB databases from another cloud including monthly DR drills.

A. Use Transfer Appliance. Transfer to Cloud Storage Nearline bucket.
B. Use Transfer Appliance. Transfer to Cloud Storage Coldline bucket.
[✓ C. Use Storage Transfer Service. Transfer to Cloud Storage Nearline bucket.] ✓
D. Use Storage Transfer Service. Transfer to Cloud Storage Coldline bucket.

Solution

C is correct because you will need to access your backup data monthly to test your disaster recovery process, so you should use a Nearline bucket; also because you will be performing ongoing, regular data transfers, so you should use Storage Transfer Service.

A, B, and D are not correct because you should not use Coldline if you want to access the files monthly (B,D) and you should not use Transfer Appliance for repeated data transfers (A,B).

https://cloud.google.com/storage/docs/concepts
https://cloud.google.com/storage/docs/storage-classes
https://cloud.google.com/storage-transfer/docs/overview
https://cloud.google.com/transfer-appliance/

DEPE

250,000 devices produce a JSON device status every 10 seconds. How do you capture event data for outlier time series analysis?

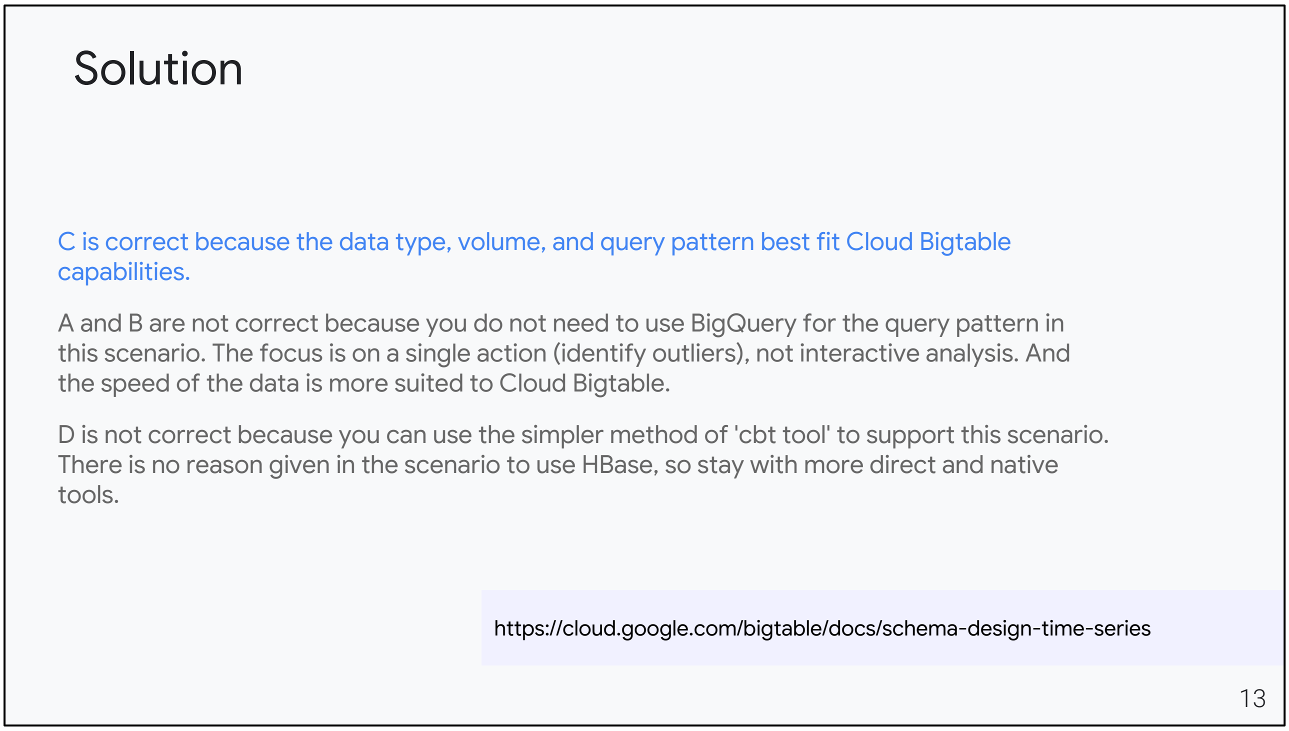
A. Capture data in BigQuery. Develop a BigQuery API custom application to query the dataset and display device outlier data.
B. Capture data in BigQuery. Use the BigQuery console to query the dataset and display device outlier data.
C. Capture data in Cloud Bigtable. Use the Cloud Bigtable cbt tool to display device outlier data.
D. Capture data in Cloud Bigtable. Install and use the HBase shell for Cloud Bigtable to query the table for device outlier data.

How do you identify outliers from a stream of data in order to remove them? The answer is to use forecasting to establish a sense of what upcoming data is out of range and to adjust the range/forecast as the stream progresses.

DEPE

250,000 devices produce a JSON device status every 10 seconds. How do you capture event data for outlier time series analysis?

A. Capture data in BigQuery. Develop a BigQuery API custom application to query the dataset and display device outlier data.
B. Capture data in BigQuery. Use the BigQuery console to query the dataset and display device outlier data.
C. Capture data in Cloud Bigtable. Use the Cloud Bigtable cbt tool to display device outlier data. ✓
D. Capture data in Cloud Bigtable. Install and use the HBase shell for Cloud Bigtable to query the table for device outlier data.



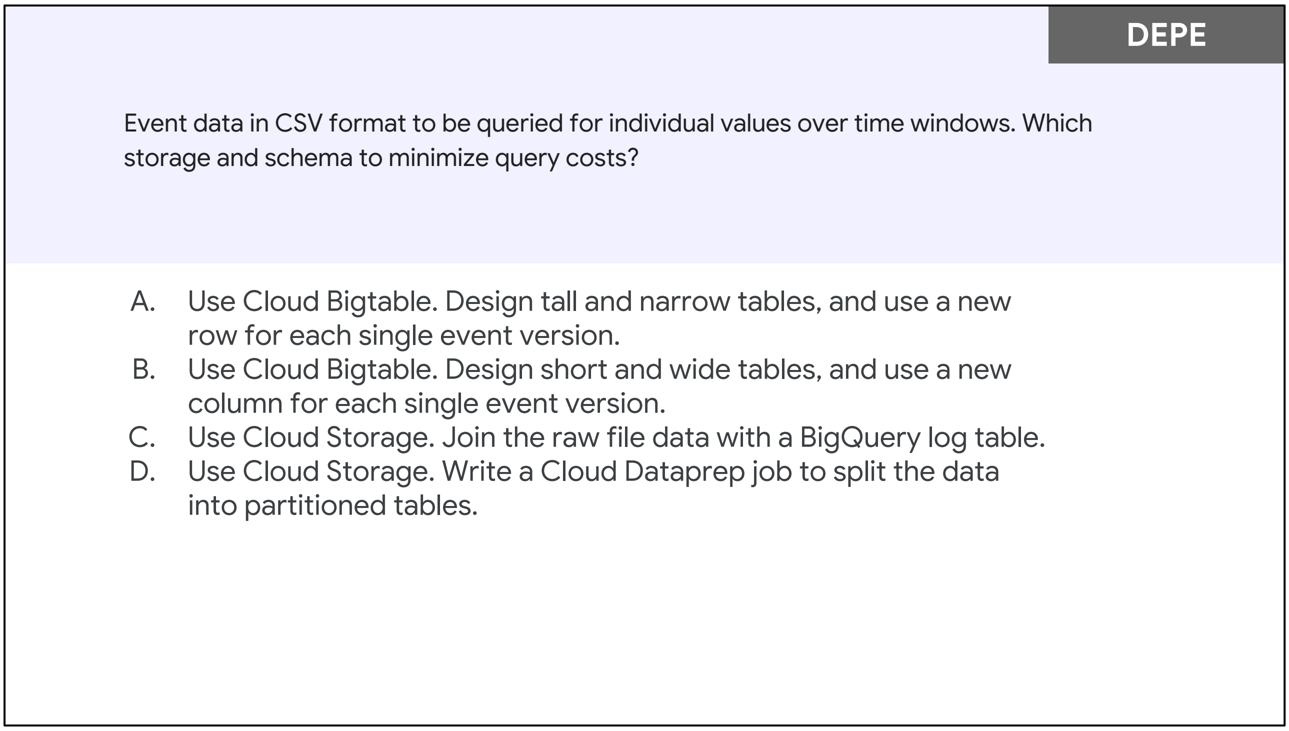
Solution

C is correct because the data type, volume, and query pattern best fit Cloud Bigtable capabilities.

A and B are not correct because you do not need to use BigQuery for the query pattern in this scenario. The focus is on a single action (identify outliers), not interactive analysis. And the speed of the data is more suited to Cloud Bigtable.

D is not correct because you can use the simpler method of 'cbt tool' to support this scenario. There is no reason given in the scenario to use HBase, so stay with more direct and native tools.

https://cloud.google.com/bigtable/docs/schema-design-time-series



DEPE

Event data in CSV format to be queried for individual values over time windows. Which storage and schema to minimize query costs?

A. Use Cloud Bigtable. Design tall and narrow tables, and use a new row for each single event version.
B. Use Cloud Bigtable. Design short and wide tables, and use a new column for each single event version.
C. Use Cloud Storage. Join the raw file data with a BigQuery log table.
D. Use Cloud Storage. Write a Cloud Dataprep job to split the data into partitioned tables.

Cloud Storage is going to have cheaper STORAGE costs than Cloud Bigtable, but we want to minimize QUERY costs.
Then process with BigQuery log table or Cloud Dataprep partitioned tables.
Partitions sound like a natural fit for windows.
Using a new row for each event sounds better than using a new column. The rows are suited to process on a primary key.

DEPE

Event data in CSV format to be queried for individual values over time windows. Which storage and schema to minimize query costs?

A. Use Cloud Bigtable. Design tall and narrow tables, and use a new row for each single event version. ✓
B. Use Cloud Bigtable. Design short and wide tables, and use a new column for each single event version.
C. Use Cloud Storage. Join the raw file data with a BigQuery log table.
D. Use Cloud Storage. Write a Cloud Dataprep job to split the data into partitioned tables.

Solution

A is correct because it is a recommended best practice. Use Cloud Bigtable and this schema for this scenario. Cloud Storage would have cheaper STORAGE costs than Cloud Bigtable, but we want to *minimize QUERY costs*.

B is not correct because the query should be based on a new row for each event, not a new column. You should design tall and narrow tables, not short and wide tables.

C and D are not correct because you do not need to use Google Cloud Storage for this scenario. It might be cheaper for storage, but not for processing.

https://cloud.google.com/bigtable/docs/schema-design-time-series
https://cloud.google.com/bigquery/docs/reference/standard-sql/analytic-function-concepts

Customer wants to maintain investment in existing Apache Spark code data pipeline.

A. BigQuery
B. Cloud Dataflow
C. Cloud Dataproc
D. Cloud Dataprep

Customer wants to maintain investment in existing Apache Spark code data pipeline.

A. BigQuery
B. Cloud Dataflow
[✓ C. Cloud Dataproc ✓]
D. Cloud Dataprep

Solution

C is correct because Cloud Dataproc is a managed Hadoop service and runs Apache Spark applications.

B would require recreating or adapting the code (an additional investment). A and D: BigQuery and Dataprep would not be appropriate tools and cannot run Spark applications.

h
https://cloud.google.com/bigquery/docs/loading-data

DEPE

Host a deep neural network machine learning model on GCP. Run and monitor jobs that could occasionally fail.

A. Use Cloud Machine Learning Engine to host your model. Monitor the status of the Operation object for 'error' results.
B. Use Cloud Machine Learning Engine to host your model. Monitor the status of the Jobs object for 'failed' job states.
C. Use a Kubernetes Engine cluster to host your model. Monitor the status of the Jobs object for 'failed' job states.
D. Use a Kubernetes Engine cluster to host your model. Monitor the status of the Operation object for 'error' results.

DEPE

Host a deep neural network machine-learning model on GCP. Run and monitor jobs that could occasionally fail.

A. Use Cloud Machine Learning to host your model. Monitor the status of the Operation object for 'error' results.
B. Use Cloud Machine Learning to host your model. Monitor the status of the Jobs object for 'failed' job states. ✓
C. Use a Kubernetes Engine cluster to host your model. Monitor the status of the Jobs object for 'failed' job states.
D. Use a Kubernetes Engine cluster to host your model. Monitor the status of the Operation object for 'error' results.

Solution

B is correct because of the requirement to host an ML DNN. Cloud ML Engine for Tensorflow can handle DNNs. Google recommends monitoring Jobs, not Operations.

A is not correct because you should not use the Operation object to monitor failures.

C and D are not correct because you should not use a Kubernetes Engine cluster for this scenario.

https://cloud.google.com/ml-engine/docs/tensorflow/managing-models-jobs
https://cloud.google.com/kubernetes-engine/docs/how-to/monitoring
https://cloud.google.com/ml-engine/docs/tensorflow/getting-started-training-prediction
https://cloud.google.com/ml-engine/docs/tensorflow/troubleshooting

DEPE

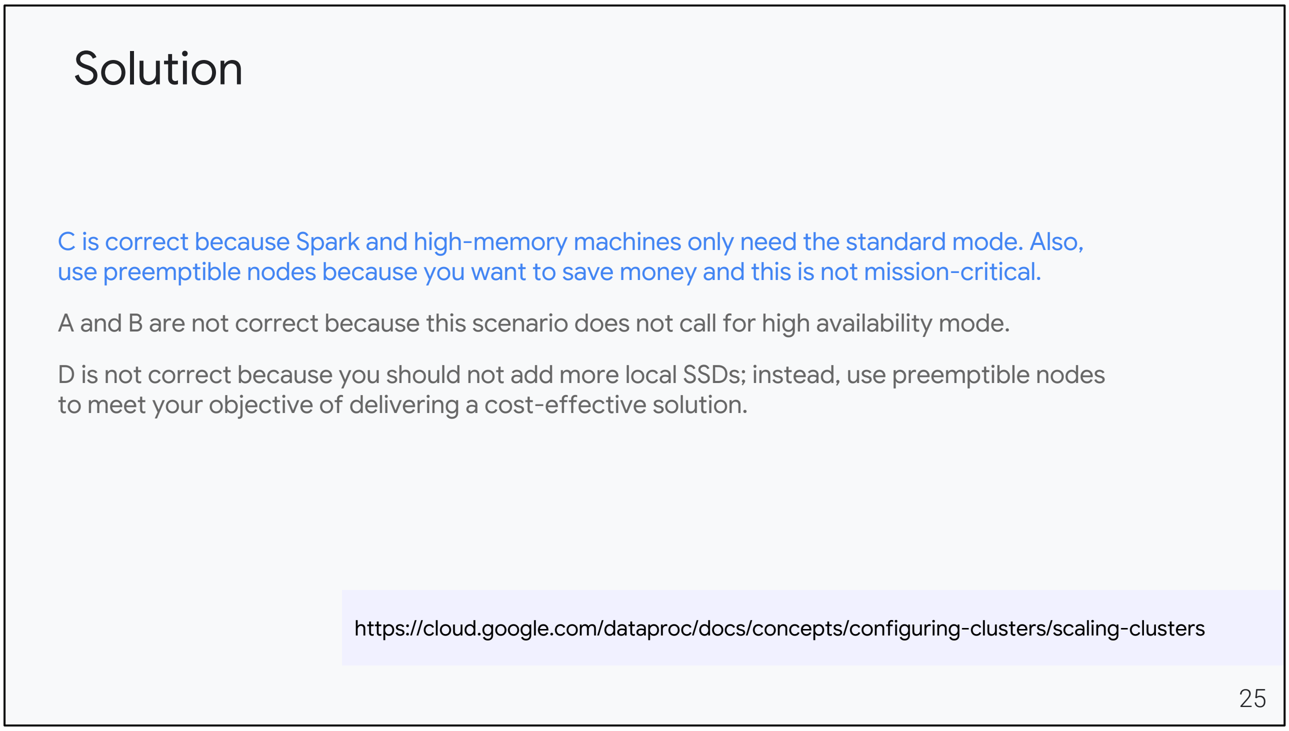
Cost-effective way to run non-critical Apache Spark jobs on Cloud Dataproc?

A. Set up a cluster in high availability mode with high-memory machine types. Add 10 additional local SSDs.
B. Set up a cluster in high availability mode with default machine types. Add 10 additional preemptible worker nodes.
C. Set up a cluster in standard mode with high-memory machine types. Add 10 additional preemptible worker nodes.
D. Set up a cluster in standard mode with the default machine types. Add 10 additional local SSDs.

DEPE

Cost-effective way to run non-critical Apache Spark jobs on Cloud Dataproc?

A. Set up a cluster in high availability mode with high-memory machine types. Add 10 additional local SSDs.
B. Set up a cluster in high availability mode with default machine types. Add 10 additional preemptible worker nodes.
[✓ C. Set up a cluster in standard mode with high-memory machine types. Add 10 additional preemptible worker nodes. ✓]
D. Set up a cluster in standard mode with the default machine types. Add 10 additional local SSDs.



Solution

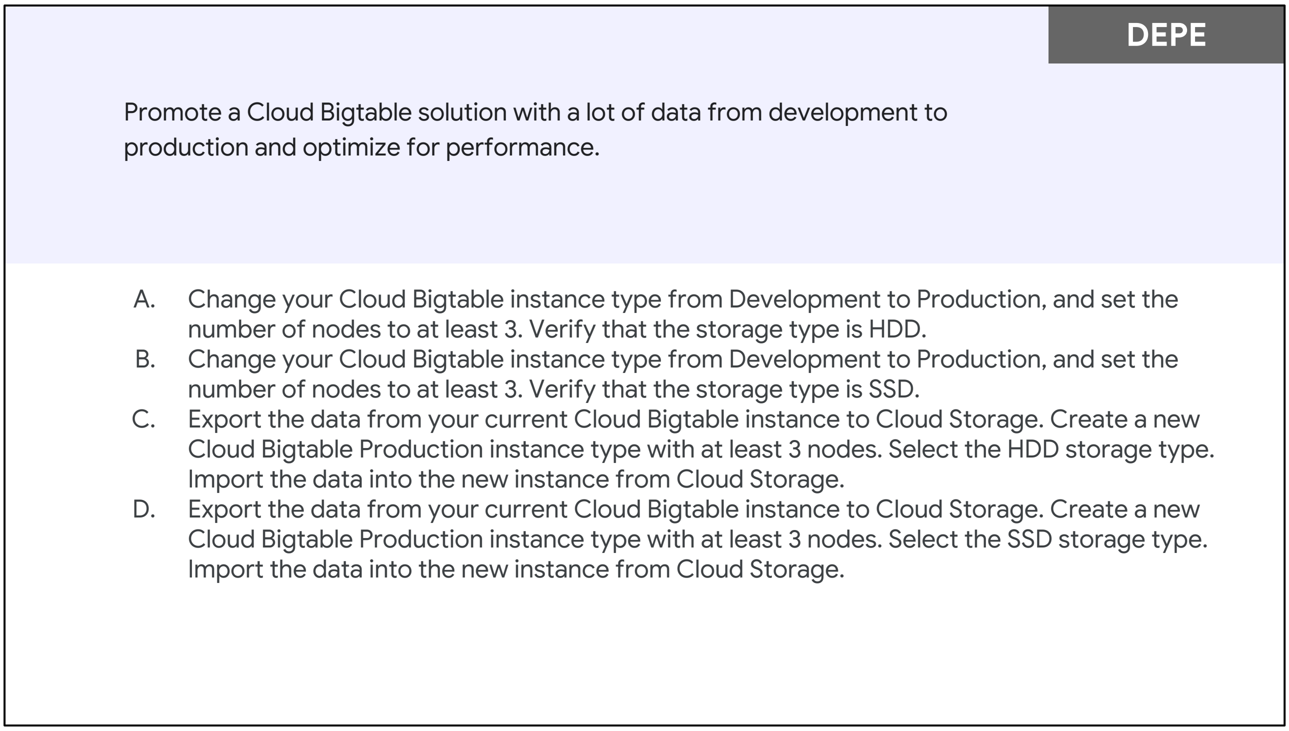
C is correct because Spark and high-memory machines only need the standard mode. Also, use preemptible nodes because you want to save money and this is not mission-critical.

A and B are not correct because this scenario does not call for high availability mode.

D is not correct because you should not add more local SSDs; instead, use preemptible nodes to meet your objective of delivering a cost-effective solution.

https://cloud.google.com/dataproc/docs/concepts/configuring-clusters/scaling-clusters

HA will cost more than Standard mode. This is not production. SSDs will make it FASTER, not CHEAPER.

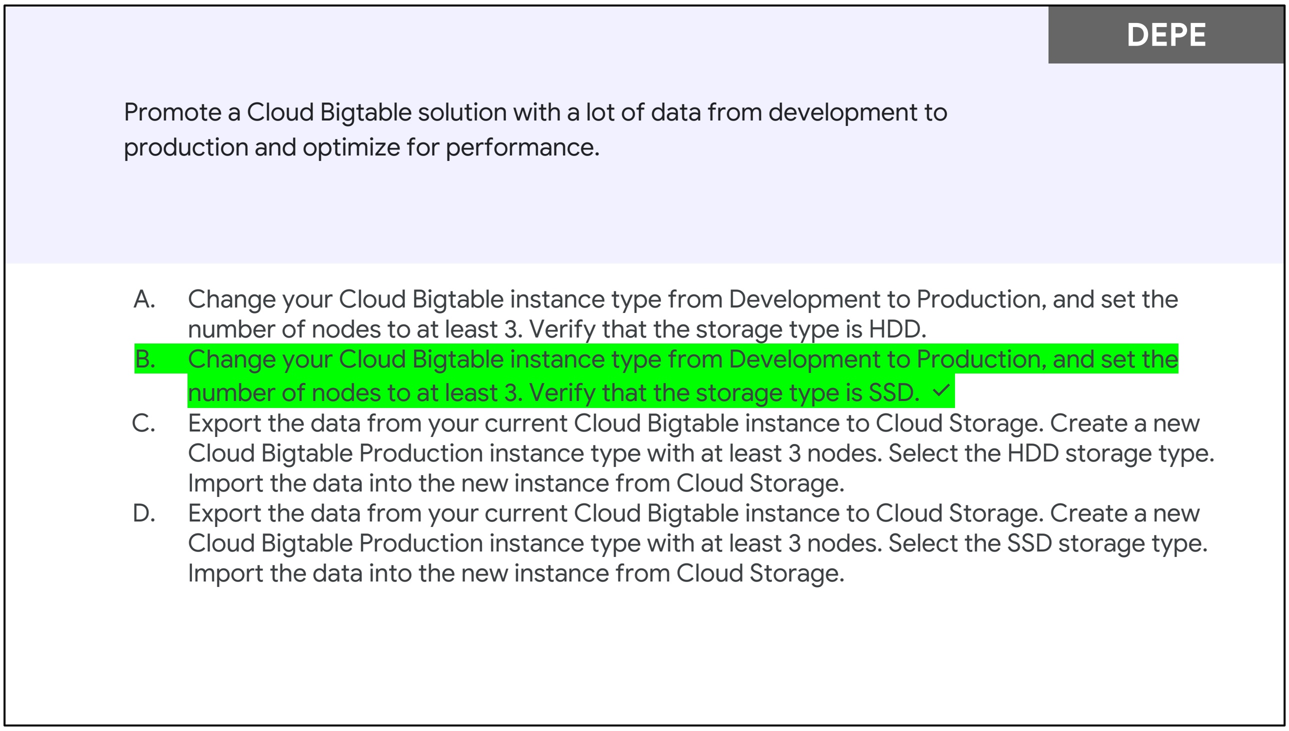


DEPE

Promote a Cloud Bigtable solution with a lot of data from development to production and optimize for performance.

A. Change your Cloud Bigtable instance type from Development to Production, and set the number of nodes to at least 3. Verify that the storage type is HDD.
B. Change your Cloud Bigtable instance type from Development to Production, and set the number of nodes to at least 3. Verify that the storage type is SSD.
C. Export the data from your current Cloud Bigtable instance to Cloud Storage. Create a new Cloud Bigtable Production instance type with at least 3 nodes. Select the HDD storage type. Import the data into the new instance from Cloud Storage.
D. Export the data from your current Cloud Bigtable instance to Cloud Storage. Create a new Cloud Bigtable Production instance type with at least 3 nodes. Select the SSD storage type. Import the data into the new instance from Cloud Storage.

SSD will improve performance. 3 nodes is required. Both modification and export methods are available. You cannot change the disk type or modify the cluster ID on a running cluster.
https://cloud.google.com/bigtable/docs/modifying-instance
https://cloud.google.com/bigtable/docs/modifying-instance#upgrade-development-instance


DEPE

Promote a Cloud Bigtable solution with a lot of data from development to production and optimize for performance.

A. Change your Cloud Bigtable instance type from Development to Production, and set the number of nodes to at least 3. Verify that the storage type is HDD.

B. Change your Cloud Bigtable instance type from Development to Production, and set the number of nodes to at least 3. Verify that the storage type is SSD. ✓

C. Export the data from your current Cloud Bigtable instance to Cloud Storage. Create a new Cloud Bigtable Production instance type with at least 3 nodes. Select the HDD storage type. Import the data into the new instance from Cloud Storage.

D. Export the data from your current Cloud Bigtable instance to Cloud Storage. Create a new Cloud Bigtable Production instance type with at least 3 nodes. Select the SSD storage type. Import the data into the new instance from Cloud Storage.

SSD will improve performance. 3 nodes is required. Both modification and export methods are available. You cannot change the disk type or modify the cluster ID on a running cluster.
https://cloud.google.com/bigtable/docs/modifying-instance
https://cloud.google.com/bigtable/docs/modifying-instance#upgrade-development-instance

Solution

B is correct because Cloud Bigtable allows you to 'scale in place,' which meets your requirements for this scenario.

A is not correct because you should be using SSD storage for this scenario.

C and D are not correct because creating a new Cloud Bigtable instance is extraneous and not needed to export; you can upgrade in place for nodes, but the storage type cannot be changed.

https://cloud.google.com/bigtable/docs/modifying-instance
https://cloud.google.com/bigtable/docs/modifying-instance#upgrade-development-instance

DEPE

As part of your backup plan, you want to be able to restore snapshots of Compute Engine instances using the fewest steps.

A. Export the snapshots to Cloud Storage. Create disks from the exported snapshot files. Create images from the new disks.
B. Export the snapshots to Cloud Storage. Create images from the exported snapshot files.
C. Use the snapshots to create replacement disks. Use the disks to create instances as needed.
D. Use the snapshots to create replacement instances as needed.

DEPE

As part of your backup plan, you want to be able to restore snapshots of Compute Engine instances using the fewest steps.

A. Export the snapshots to Cloud Storage. Create disks from the exported snapshot files. Create images from the new disks.
B. Export the snapshots to Cloud Storage. Create images from the exported snapshot files.
C. Use the snapshots to create replacement disks. Use the disks to create instances as needed.
D. Use the snapshots to create replacement instances as needed. ✓

Solution

D is correct because the scenario asks how to recreate instances. You can create an instance directly from a snapshot without restoring to disk first.

A and B are not correct because you don't need to export the snapshot to use it.

C is not correct because it includes the step of creating a persistent disk before creating an instance, which is not necessary.

https://cloud.google.com/compute/docs/instances/create-start-instance

DEPE

You want to minimize costs to run Google Data Studio reports on BigQuery queries by using prefetch caching.

A. Set up the report to use the Owner's credentials to access the underlying data in BigQuery, and direct the users to view the report only once per business day (24-hour period).
B. Set up the report to use the Owner's credentials to access the underlying data in BigQuery, and verify that the 'Enable cache' checkbox is selected for the report.
C. Set up the report to use the Viewer's credentials to access the underlying data in BigQuery, and also set it up to be a 'view-only' report.
D. Set up the report to use the Viewer's credentials to access the underlying data in BigQuery, and verify that the 'Enable cache' checkbox is not selected for the report.

You want to minimize costs to run Data Studio reports on BigQuery queries by using prefetch caching.

A. Set up the report to use the Owner's credentials to access the underlying data in BigQuery, and direct the users to view the report only once per business day (24-hour period).

B. Set up the report to use the Owner's credentials to access the underlying data in BigQuery, and verify that the 'Enable cache' checkbox is selected for the report. ✓

C. Set up the report to use the Viewer's credentials to access the underlying data in BigQuery, and also set it up to be a 'view-only' report.

D. Set up the report to use the Viewer's credentials to access the underlying data in BigQuery, and verify that the 'Enable cache' checkbox is not selected for the report.

Solution

B is correct because you must set Owner credentials to use the 'enable cache' option in BigQuery. It is also a Google best practice to use the ‘enable cache’ option when the business scenario calls for using prefetch caching. 1) Report must use Owner's Credentials. 2) You don't need to tell the users not to use the report, you need to tell the system to use Query and Pre-fetch caching to cut down on BigQuery jobs.

A, C, and D are not correct because a cache auto-expires every 12 hours; a prefetch cache is only for data sources that use the Owner's credentials (not the Viewer's credentials).

Article: "How Data Studio caches data" in Data Studio dashboard Help

A Data Analyst is concerned that a BigQuery query could be too expensive.

A. Use the LIMIT clause to limit the number of values in the results.
B. Use the SELECT clause to limit the amount of data in the query. Partition data by date so the query can be more focused.
C. Set the Maximum Bytes Billed, which will limit the number of bytes processed but still run the query if the number of bytes requested goes over the limit.
D. Use GROUP BY so the results will be grouped into fewer output values.

A Data Analyst is concerned that a BigQuery query could be too expensive.

A. Use the LIMIT clause to limit the number of values in the results.

B. Use the SELECT clause to limit the amount of data in the query. Partition data by date so the query can be more focused. ✓

C. Set the Maximum Bytes Billed, which will limit the number of bytes processed but still run the query if the number of bytes requested goes over the limit.

D. Use GROUP BY at the end so the results will be grouped into fewer output values.

Solution

B is correct.

A is not correct because the LIMIT clause limits the output, but does not limit data processes.

C is not correct because if the query contains too many bytes, the job will fail and not be run.

D is not correct because ordering the output will have no effect on the data processed up to that point.

https://cloud.google.com/bigquery/docs/best-practices-costs

DEPE

BigQuery data is stored in external CSV files in Cloud Storage; as the data has increased, the query performance has dropped.

A. Import the data into BigQuery for better performance.
B. Request more slots for greater capacity to improve performance.
C. Divide the data into partitions based on date.
D. Time to move to Cloud Bigtable; it is faster in all cases.

DEPE

BigQuery data is stored in external CSV files in Cloud Storage; as the data has increased, the query performance has dropped.

A. Import the data into BigQuery for better performance. ✓
B. Request more slots for greater capacity to improve performance.
C. Divide the data into partitions based on date.
D. Time to move to Cloud Bigtable; it is faster in all cases.

Solution

A is correct. The performance issue is because the data is stored in a non-optimal format in an external storage medium.

B is incorrect because a slot is a measure of processing power, and the bottleneck is in the data access, not the data processing.

C *might* improve performance by focusing the query to a date-range if the data was already imported into a dataset.

D is incorrect because a solution is available using BigQuery, and redesigning the data pipeline would be more complicated than adjusting the existing solution.

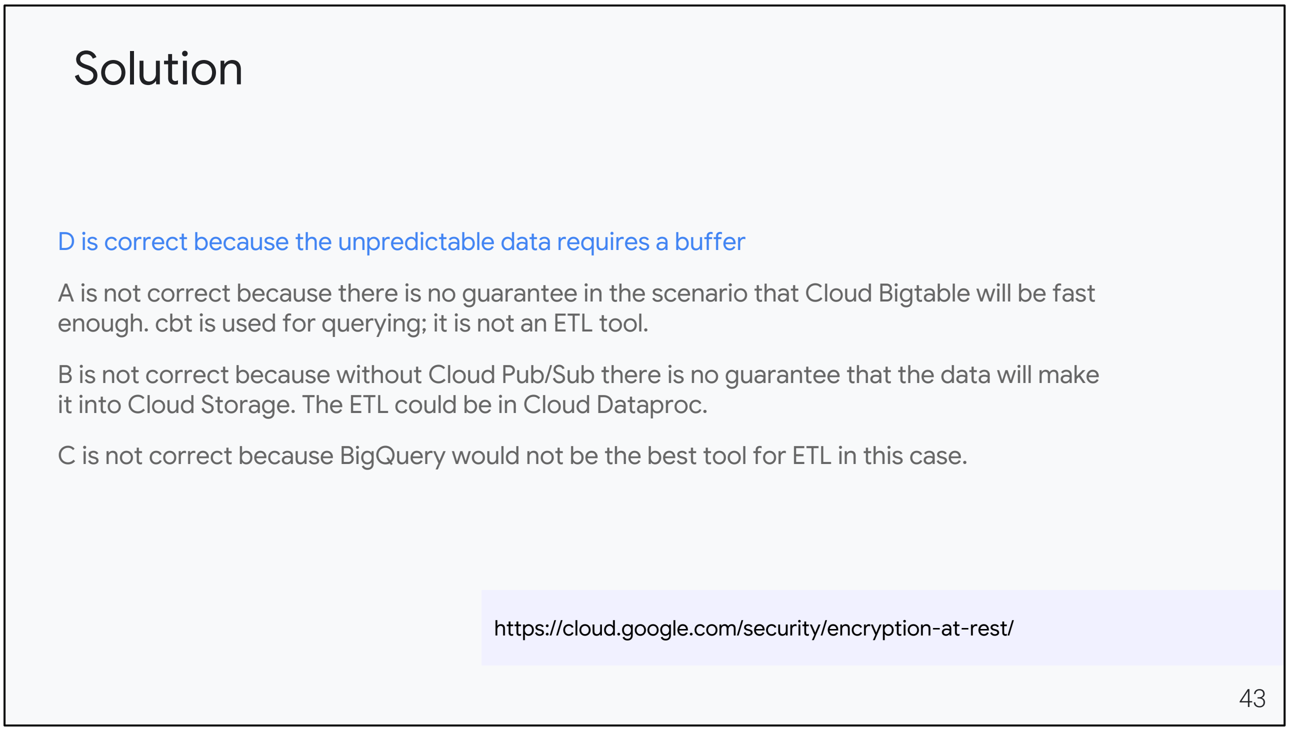
h
https://cloud.google.com/bigquery/docs/loading-data

Source data is streamed in bursts and must be transformed before use.

A. Use Cloud Bigtable for fast input and cbt for ETL.
B. Ingest data to Cloud Storage. Use Cloud Dataproc for ETL.
C. Use Cloud Pub/Sub to buffer the data, and then use BigQuery for ETL.
D. Use Cloud Pub/Sub to buffer the data, and then use Cloud Dataflow for ETL.

Source data is streamed in bursts and must be transformed before use.

A. Use Cloud Bigtable for fast input and cbt for ETL.
B. Ingest data to Cloud Storage. Use Cloud Dataproc for ETL.
C. Use Cloud Pub/Sub to buffer the data, and then use BigQuery for ETL.
D. Use Cloud Pub/Sub to buffer the data, and then use Cloud Dataflow for ETL. ✓



Solution

D is correct because the unpredictable data requires a buffer

A is not correct because there is no guarantee in the scenario that Cloud Bigtable will be fast enough. cbt is used for querying; it is not an ETL tool.

B is not correct because without Cloud Pub/Sub there is no guarantee that the data will make it into Cloud Storage. The ETL could be in Cloud Dataproc.

C is not correct because BigQuery would not be the best tool for ETL in this case.

https://cloud.google.com/security/encryption-at-rest/

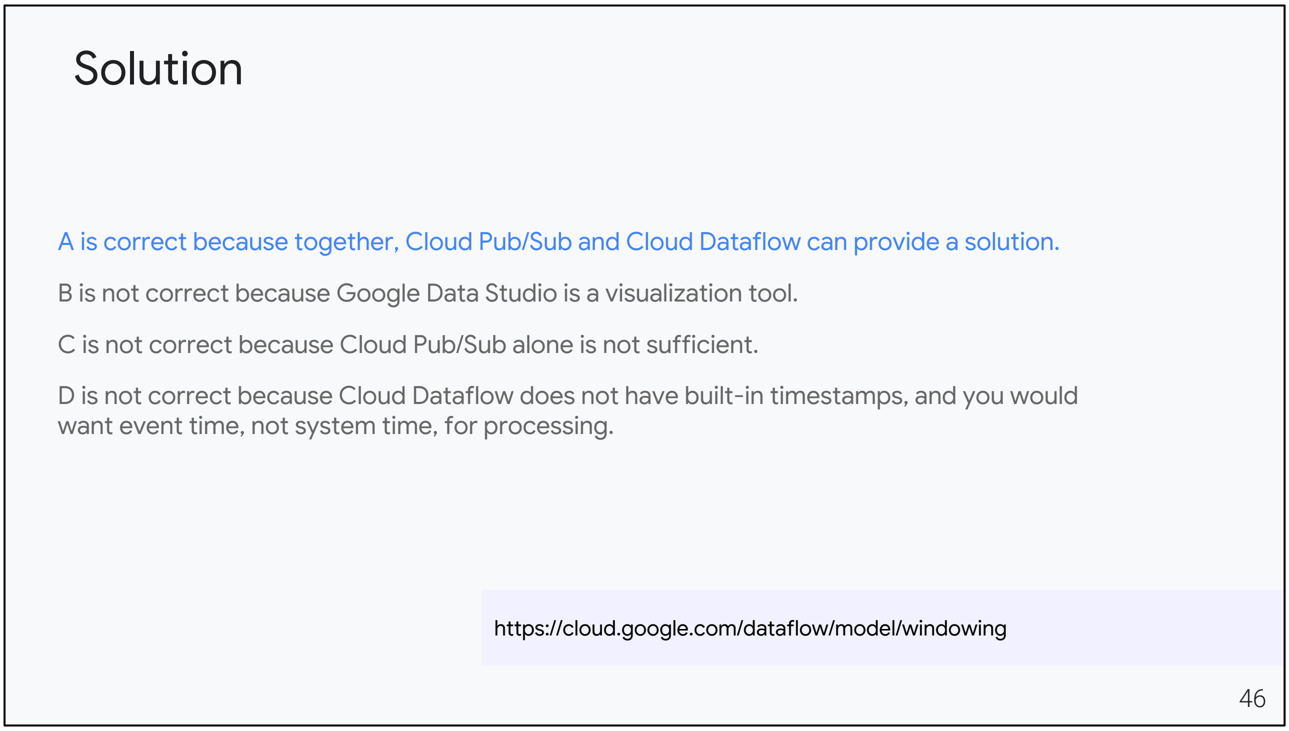
https://cloud.google.com/pubsub/docs/overview

Calculate a running average on streaming data that can arrive late and out of order.

A. Use Cloud Pub/Sub and Cloud Dataflow with Sliding Time Windows.
B. Use Cloud Pub/Sub and Google Data Studio.
C. Cloud Pub/Sub can guarantee timely arrival and order.
D. Use Cloud Dataflow's built-in timestamps for ordering and filtering.

Calculate a running average on streaming data that can arrive late and out of order.

A. Use Cloud Pub/Sub and Cloud Dataflow with Sliding Time Windows. ✓
B. Use Cloud Pub/Sub and Google Data Studio.
C. Cloud Pub/Sub can guarantee timely arrival and order.
D. Use Cloud Dataflow's built-in timestamps for ordering and filtering.



Solution

A is correct because together, Cloud Pub/Sub and Cloud Dataflow can provide a solution.

B is not correct because Google Data Studio is a visualization tool.

C is not correct because Cloud Pub/Sub alone is not sufficient.

D is not correct because Cloud Dataflow does not have built-in timestamps, and you would want event time, not system time, for processing.

https://cloud.google.com/dataflow/model/windowing

https://cloud.google.com/dataflow/model/windowing