

D. In the Cloud Platform Console, create a new persistent disk attached to the virtual machine, format and mount it, and configure the database service to move the files to the new disk



E. In the Cloud Platform Console, create a snapshot of the persistent disk restore the snapshot to a new larger disk, unmount the old disk, mount the new disk and restart the database service

  **Tos0** Highly Voted 5 years ago

A is the correct answer because the question says "with minimum downtime"
upvoted 28 times

  **passnow** Highly Voted 5 years ago

least amount of downtime? is the sugar word. You miss that you miss all. Everything there is correct but I believe its only A that fits that requirement
upvoted 14 times

  **raj117** 3 years, 8 months ago

but in option A, nowhere it is mentioned to shut down the VM.
upvoted 1 times

  **monkeym** 3 years, 5 months ago

No need to reboot.
upvoted 1 times

  **Ekramy_Elnaggar** Most Recent 1 month, 1 week ago



Selected Answer: A

1. Online Resizing: Google Cloud Platform allows you to increase the size of a persistent disk while it's attached to a running VM. This means don't need to shut down the database server.
2. resize2fs: This Linux command extends the file system to utilize the newly added space on the disk. It can be run while the file system is mounted, minimizing downtime.
upvoted 1 times

  **SerGCP** 1 month, 2 weeks ago

Selected Answer: C

A) can not work because you must extend partition and after than you can extend filesystem
C) ok because you can use fdisk in linux to extend partition and after than you can resize ext4 filesystem
upvoted 1 times

  **Nora9** 1 year, 1 month ago

Selected Answer: A

A is the right answer. You should resize the disk, take a snapshot, then resize the filesystem and partitions (eg.) ext4, xfs etc.
upvoted 1 times

  **AdityaGupta** 1 year, 2 months ago

Selected Answer: A

Unlike Azure, in google you can dynamically resize the persistent disk while VM is running. This narrows down the option to A or C. Since the question says "ext4-formatted persistent disk", we need to choose correct command (resize2fs or fdisk) for Linux for resizing ext4 file format disk. To resize an ext4 file system in Linux, you can use the resize2fs command. FDISK to manipulate partition tables in Linux.
upvoted 8 times

  **blackhawk86** 1 year, 3 months ago

Selected Answer: A

According to the right url, A is the right answer. <https://cloud.google.com/compute/docs/disks/resize-persistent-disk>
upvoted 4 times

  **blackhawk86** 1 year, 3 months ago

The right URL for the official document is, <https://cloud.google.com/compute/docs/disks/resize-persistent-disk>
upvoted 1 times

  **heretolearnazure** 1 year, 3 months ago

A is no brainer
upvoted 1 times

🗨️ **jalberto** 1 year, 4 months ago

Selected Answer: E

E is the correct because is true you need minimum downtime but in Production a backup is a must.
upvoted 2 times

🗨️ **jalberto** 1 year, 4 months ago

E because you are in Production, and you need a backup
upvoted 1 times

🗨️ **alekonko** 1 year, 9 months ago

Selected Answer: A

A is correct, resize disk don't required reboot or downtime
<https://cloud.google.com/compute/docs/disks/resize-persistent-disk>
upvoted 3 times

🗨️ **omermahgoub** 2 years ago

A: Increasing the size of the persistent disk in the Cloud Platform Console and using the `resize2fs` command in Linux.

Increasing the size of the persistent disk can be done without requiring the virtual machine to be shut down, and the `resize2fs` command can be used to resize the `ext4` filesystem on the disk to take advantage of the additional space. This will allow you to add more storage space to the virtual machine without disrupting the database service.

upvoted 1 times

🗨️ **megumin** 2 years, 1 month ago

Selected Answer: A

A is ok
upvoted 2 times

🗨️ **Mahmoud_E** 2 years, 1 month ago

Selected Answer: A

A https://cloud.google.com/compute/docs/disks/resize-persistent-disk?_ga=2.233866652.-3622898.1631303718
upvoted 1 times

🗨️ **minmin2020** 2 years, 2 months ago

Selected Answer: A

A. In the Cloud Platform Console, increase the size of the persistent disk and use the `resize2fs` command in Linux.
upvoted 1 times

🗨️ **AzureDP900** 2 years, 2 months ago

yes, A is right
upvoted 1 times

🗨️ **holerina** 2 years, 3 months ago

A resize the disk standard command
upvoted 1 times

Question #15

Topic 1

Your application needs to process credit card transactions. You want the smallest scope of Payment Card Industry (PCI) compliance without compromising the ability to analyze transactional data and trends relating to which payment methods are used.

How should you design your architecture?

A. Create a tokenizer service and store only tokenized data

- B. Create separate projects that only process credit card data
- C. Create separate subnetworks and isolate the components that process credit card data
- D. Streamline the audit discovery phase by labeling all of the virtual machines (VMs) that process PCI data
- E. Enable Logging export to Google BigQuery and use ACLs and views to scope the data shared with the auditor

🗨️ 👤 **AD2AD4** Highly Voted 4 years, 6 months ago

Final Decision to go with Option A. I have done PCI DSS Audit for my project and thats the best suited case. 100% sure to use tokenised data instead of actual card number

upvoted 46 times

🗨️ 👤 **Musk** 4 years, 6 months ago

But with A you cannot extract statistics. That is the second requirement.

upvoted 4 times

🗨️ 👤 **Arimaverick** 3 years, 11 months ago

Analyzing Transaction does not require Credit Card number I guess. Only amount of transaction or balance what is needed. We also perform something similar with transactional data with tokenized PII information. So CC can be tokenized. So answer should be A.

upvoted 6 times

🗨️ 👤 **Musk** 4 years, 4 months ago

Thinking about that better, I think you can because you are only tokenizing the sensitive data, not the transaction type.

upvoted 2 times

🗨️ 👤 **RitwickKumar** 2 years, 4 months ago

You can as the generated token for a given credit card would be same(generally but there are approaches which can give you different token for the same sensitive data input). Only thing that you won't know is the actual card number which is not required for the trend analysis.

When the trend analysis involves referential integrity then tokenization process becomes challenging but still once data is tokenized correctly you should be able to perform any kind of the analysis.

upvoted 2 times

🗨️ 👤 **AzureDP900** 2 years, 2 months ago

I agree. A is the best option

upvoted 3 times

🗨️ 👤 **omermahgoub** Highly Voted 2 years ago

To minimize the scope of Payment Card Industry (PCI) compliance while still allowing for the analysis of transactional data and trends related payment methods, you should consider using a tokenizer service and storing only tokenized data, as described in option A.

Tokenization is a process of replacing sensitive data, such as credit card numbers, with unique, randomly-generated tokens that cannot be used for fraudulent purposes. By using a tokenizer service and storing only tokenized data, you can reduce the scope of PCI compliance to only the tokenization service, rather than the entire application. This can help minimize the amount of sensitive data that needs to be protected and reduce the overall compliance burden.

upvoted 31 times

🗨️ 👤 **oxfordcommaa** 1 year, 11 months ago

man, this is an amazing answer. props

upvoted 5 times

🗨️ 👤 **ccpmad** 6 months, 1 week ago

thanks to chagpt, are you serious saying that?

upvoted 1 times

🗨️ 👤 **Saxena_Vibhor** 11 months, 3 weeks ago

Nicely explained, thanks.

upvoted 1 times

🗄️ 👤 **Ekramy_Elnaggar** Most Recent 1 month, 1 week ago

Selected Answer: A

1. Reduced PCI Scope: Tokenization replaces sensitive credit card data with non-sensitive tokens. This significantly reduces the scope of PCI DSS compliance, as you no longer store actual cardholder data. Only the tokenization service needs to be PCI compliant.
2. Data Analysis: You can still analyze transactional data and trends using the tokenized data. The tokens can be linked back to the original cardholder data if needed for specific analysis or reporting purposes, but this would be done in a controlled and secure environment.

upvoted 1 times

🗄️ 👤 **hzaoui** 11 months, 1 week ago

Selected Answer: A

A is correct

upvoted 1 times

🗄️ 👤 **devinss** 1 year, 3 months ago

Not sure why 100% agree on A. To limit PCI DSS scope, the data handling should be done in a separate project with very limited access. Only this project should tokenization be done and made available for analytics. The first requirement however, is isolating the payment and tokenization code in a separate project. Answer should be B.

upvoted 2 times

🗄️ 👤 **heretolearnazure** 1 year, 3 months ago

Tokenizing is the best way to protect PCI information.

upvoted 1 times

🗄️ 👤 **nocrush** 1 year, 4 months ago

Selected Answer: A

A is my best option

upvoted 1 times

🗄️ 👤 **KjChen** 2 years, 1 month ago

Selected Answer: A

<https://cloud.google.com/architecture/tokenizing-sensitive-cardholder-data-for-pci-dss>

upvoted 4 times

🗄️ 👤 **andreavale** 2 years, 1 month ago

Selected Answer: A

ok for A

upvoted 1 times

🗄️ 👤 **minmin2020** 2 years, 2 months ago

Selected Answer: A

A. Create a tokenizer service and store only tokenized data

upvoted 1 times

🗄️ 👤 **BiddlyBdoynng** 2 years, 2 months ago

B appears the most thorough but the question asks to comply with the smallest scope, network segmentation is not a must. Tokenization is simpler. C is similar to B, more than required. D & E do not address the problem.

upvoted 1 times

🗄️ 👤 **holerina** 2 years, 3 months ago

correct answer is A use tokenize

upvoted 2 times

🗄️ 👤 **abirroy** 2 years, 3 months ago

Selected Answer: A

Correct answer A



upvoted 1 times

🗄️ 👤 **Nirca** 2 years, 8 months ago

Selected Answer: A

The mandatory stage in PCI is having a encryption/ description system. Data must not be stored as is with PAN. So A IS A MUST. The rest are nice to have.


upvoted 1 times

  **ryzior** 2 years, 8 months ago

Selected Answer: A

I think it should be A and C - the paper states clearly, a proper network segmentation is still required to dispartate the vault and token servers from the rest of the flat network.

upvoted 1 times

  **sjmsummer** 2 years, 11 months ago

I chose A. But why C is not good?

upvoted 1 times

  **vincy2202** 2 years, 12 months ago

A is the correct answer

https://cloud.google.com/architecture/tokenizing-sensitive-cardholder-data-for-pci-dss#a_service_for_handling_sensitive_information

upvoted 3 times

Question #16

Topic 1

You have been asked to select the storage system for the click-data of your company's large portfolio of websites. This data is streamed in from a custom website analytics package at a typical rate of 6,000 clicks per minute. With bursts of up to 8,500 clicks per second. It must have been stored for future analysis by your data science and user experience teams.

Which storage infrastructure should you choose?

- A. Google Cloud SQL
- B. Google Cloud Bigtable
- C. Google Cloud Storage
- D. Google Cloud Datastore

  **victory108** **Highly Voted**  3 years, 5 months ago

B. Google Cloud Bigtable

upvoted 12 times

🗄️ 👤 **jeff001** Highly Voted 3 years, 8 months ago
B, Bigtable due to the IoT like requirements
upvoted 10 times

🗄️ 👤 **fff2e69** Most Recent 1 month ago
Selected Answer: B
Google Cloud Bigtable is well-suited for handling high-throughput, low-latency workloads like clickstream data. It is optimized for analytics or time-series and event data at scale, and it supports high write rates, with the capacity to handle thousands of writes per second. This makes it ideal for storing large volumes of clickstream data with bursts, ensuring data is available for analysis by data science and user experience teams.
upvoted 3 times

🗄️ 👤 **Ekramy_Elnaggar** 1 month, 1 week ago
Selected Answer: B
1. High throughput for clickstream data: Bigtable is a NoSQL database designed for high write throughput, making it ideal for handling the continuous stream of click data with bursts up to 8,500 clicks per second.
2. Scalability: Bigtable is highly scalable, allowing you to handle increasing data volumes as your website portfolio grows.
3. Low latency: Bigtable provides low latency data access, which is important for real-time analysis and reporting on clickstream data.
4. Integration with BigQuery: Bigtable integrates well with BigQuery, enabling your data science team to perform complex analysis and generate insights from the clickstream data.
upvoted 3 times

🗄️ 👤 **upliftinghut** 6 months ago
Selected Answer: C
Big table if it needs real time but here the need for analysis is not urgent => Google cloud storage.
upvoted 2 times

🗄️ 👤 **pakilodi** 1 year ago
Selected Answer: B
The right answer is BigTable for a such large volume of data.
upvoted 1 times

🗄️ 👤 **Nora9** 1 year, 1 month ago
Selected Answer: B
B. Google cloud Bigtable is most suitable as it is for analysis and streamed data.
upvoted 1 times

🗄️ 👤 **MartinFish** 1 year, 4 months ago
Selected Answer: B
Note: DataStore is now rename as FireStore.
upvoted 1 times

🗄️ 👤 **hiromi** 1 year, 8 months ago
Selected Answer: B
B is correct
upvoted 1 times

🗄️ 👤 **alekonko** 1 year, 9 months ago
Selected Answer: B
Bigtable is a high-perf NoSQL db service that handle large volumes of structured data with low latency
upvoted 3 times

🗄️ 👤 **zerg0** 1 year, 10 months ago
Selected Answer: B
The Google Cloud Bigtable goes together with the BigQuery. The question itself gives away a bit.
upvoted 1 times

🗨️ 👤 **omermahgoub** 2 years ago

For storing click-data that is streamed in at a rate of 6,000 clicks per minute, with bursts of up to 8,500 clicks per second, and that needs to be stored for future analysis by your data science and user experience teams, you should consider using a scalable, high-performance, and low-latency NoSQL database such as Google Cloud Bigtable, option B.

Google Cloud Bigtable is a fully managed, high-performance NoSQL database service that is designed to handle large volumes of structured data with low latency. It is well-suited for storing high-velocity data streams and can scale to handle millions of reads and writes per second.

Option A: Google Cloud SQL, option C: Google Cloud Storage, and option D: Google Cloud Datastore, would not be suitable for this use case as they are not designed to handle high-velocity data streams at this scale.

upvoted 4 times

🗨️ 👤 **i_am_robot** 2 years ago

B. Google Cloud Bigtable

Google Cloud Bigtable is a scalable, high-performance NoSQL database that is well-suited for storing large amounts of data with low latency. It is designed for high-throughput workloads such as streaming data, and is able to handle bursts of up to millions of reads and writes per second.

Given the high volume of click data that needs to be stored and the requirement for low latency, Google Cloud Bigtable would be a good choice for storing the data. It is able to handle the high rate of incoming data and provide fast access to the data for analysis by the data science and user experience teams.

Google Cloud SQL is a fully-managed relational database service, and may not be the best choice for storing high-volume streaming data. Google Cloud Storage is an object storage service, and may not provide the necessary performance for storing and querying large amounts of data in real-time. Google Cloud Datastore is a NoSQL document database, and while it may be suitable for storing large amounts of data, it may not provide the necessary performance for handling high volumes of streaming data.

upvoted 2 times

🗨️ 👤 **Melampos** 2 years ago

Selected Answer: B

<https://cloud.google.com/bigtable#section-9>

upvoted 1 times

🗨️ 👤 **Bry_040706** 2 years ago

B. Google Cloud Bigtable

upvoted 1 times

🗨️ 👤 **AniketD** 2 years, 1 month ago

Selected Answer: B

Cloud Bigtable has all the features to fulfill the requirements mentioned in the question

upvoted 1 times

🗨️ 👤 **Mahmoud_E** 2 years, 1 month ago

D is right answer

upvoted 1 times

Question #17

Topic 1

You are creating a solution to remove backup files older than 90 days from your backup Cloud Storage bucket. You want to optimize ongoing Cloud Storage spend.

What should you do?



- A. Write a lifecycle management rule in XML and push it to the bucket with gsutil
- B. Write a lifecycle management rule in JSON and push it to the bucket with gsutil
- C. Schedule a cron script using gsutil ls -l gs://backups/** to find and remove items older than 90 days
- D. Schedule a cron script using gsutil ls -l gs://backups/** to find and remove items older than 90 days and schedule it with cron

🗨️ 👤 **Eroc** Highly Voted 5 years, 1 month ago

All four are correct answers. Google has built in cron job scheduling with Cloud Scheduler, so that would place "D" behind "C" in Google's perspective. Google also has its own lifecycle management command line prompt gcloud lifecycle so "A" or "B" could be used. JSON is slightly better than XML.

faster than XML because of the "{" verse "<c>" distinguisher, with a Trie tree used for alphanumeric parsing. So between "A" and "B", choose "B". Between "B" and "A", "B" is slightly more efficient from the GCP operator perspective. So choose "B".

upvoted 35 times

  **ghitesh** 4 years, 11 months ago



gsutil command takes only json as input for lifecycle management. In case of API, both XML and json can be used.

<https://cloud.google.com/storage/docs/gsutil/commands/lifecycle>

<https://cloud.google.com/storage/docs/xml-api/put-bucket-lifecycle>



https://cloud.google.com/storage/docs/json_api/v1/buckets/update

upvoted 30 times

  **tartar** 4 years, 4 months ago

B is ok

upvoted 8 times

  **nitinz** 3 years, 9 months ago

B is correct. Policy = JSON format. No matter if its AWS or GCP.

upvoted 10 times

  **clouddude** Highly Voted 4 years, 7 months ago

I'll go with B.

A is not reasonable because life cycle policies are not written in XML.

B is reasonable and is cloud native.

C requires a cron script which needs something to run the script and is a non-cloud native approach.

D requires a cron script which needs something to run the script and is a non-cloud native approach.

upvoted 15 times

  **Mitthugcpguru** Most Recent 2 days, 22 hours ago

Selected Answer: B

Gsutil take only json as input so B answer

upvoted 1 times

  **Ekramy_Elnaggar** 1 month, 1 week ago

Selected Answer: B

1. Lifecycle Management: Google Cloud Storage offers built-in lifecycle management rules specifically designed for automated data retention and deletion. This is the most efficient and cost-effective way to manage your backup files.

2. JSON Format: Lifecycle rules are defined in JSON format.

3. gsutil: The gsutil command-line tool is used to interact with Cloud Storage, including setting lifecycle configuration.

upvoted 1 times

  **Hungdv** 4 months, 1 week ago

choose B

upvoted 1 times

  **squishy_fishy** 1 year, 1 month ago

The correct available answer is B. But in real life, we use Terraform tfvars file.

upvoted 1 times

  **eka_nostra** 1 year, 4 months ago

Selected Answer: B

Cloud Storage has lifecycle management rules and could be applied with gsutil and gcloud storage buckets. It is common to use JSON for transferring data.

upvoted 2 times

  **alekonko** 1 year, 9 months ago

Selected Answer: B

B, gsutil can set policy using json file

<https://cloud.google.com/storage/docs/gsutil/commands/lifecycle#examples>

upvoted 1 times

🗨️ **omer mahgoub** 2 years ago

To remove backup files older than 90 days from a Cloud Storage bucket and optimize ongoing Cloud Storage spend, you should consider writing a lifecycle management rule in JSON and pushing it to the bucket with gsutil, as described in option B.

Lifecycle management rules allow you to automatically delete objects from a Cloud Storage bucket based on age or other criteria, such as the object's storage class. By writing a rule in JSON and pushing it to the bucket with gsutil, you can specify that objects older than 90 days should be deleted, ensuring that the bucket only contains current backup files and minimizing Cloud Storage spend.

Option A, C and D would not be suitable for this use case, as they do not allow you to specify lifecycle management rules that delete objects based on age.

upvoted 3 times

🗨️ **i_am_robot** 2 years ago

B. Write a lifecycle management rule in JSON and push it to the bucket with gsutil

To remove backup files older than 90 days from a Cloud Storage bucket, you can use the lifecycle management feature in Cloud Storage. This feature allows you to specify rules to automatically delete objects based on their age.

upvoted 1 times

🗨️ **Melampos** 2 years ago

Selected Answer: B

<https://cloud.google.com/storage/docs/gsutil/commands/lifecycle>

upvoted 1 times

🗨️ **Bry_040706** 2 years ago

B. Life cycle management using JSON.

upvoted 1 times

🗨️ **AniketD** 2 years, 1 month ago

Selected Answer: B

B with JSON option is correct

upvoted 1 times

🗨️ **megumin** 2 years, 1 month ago

Selected Answer: B

B is ok

upvoted 1 times

🗨️ **Mahmoud_E** 2 years, 1 month ago

Selected Answer: B

B is the right answer

upvoted 1 times

🗨️ **zr79** 2 years, 2 months ago

life cycle management is the answer written in JSON format. JSON is easier to write and read compared to XML which you can not use in commands

upvoted 1 times

🗨️ **AzureDP900** 2 years, 2 months ago

Lifecycle management with JSON is right .. I will go with B

upvoted 1 times

Question #18

Topic 1

Your company is forecasting a sharp increase in the number and size of Apache Spark and Hadoop jobs being run on your local datacenter. You want to utilize the cloud to help you scale this upcoming demand with the least amount of operations work and code change.

Which product should you use?

- A. Google Cloud Dataflow
- B. Google Cloud Dataproc
- C. Google Compute Engine
- D. Google Kubernetes Engine

🗄️ 👤 **AWS56** Highly Voted 4 years, 11 months ago

"B. Google Cloud Dataproc" is the answer
upvoted 19 times

🗄️ 👤 **VinayakBudapanahalli** Highly Voted 3 years, 11 months ago

Dataproc is a managed Spark and Hadoop service that lets you take advantage of open source data tools for batch processing, querying, streaming, and machine learning. Dataproc automation helps you create clusters quickly, manage them easily, and save money by turning clusters off when you don't need them. With less time and money spent on administration, you can focus on your jobs and your data.

<https://cloud.google.com/dataproc/docs/concepts/overview#:~:text=Dataproc%20is%20a%20managed%20Spark,%2C%20streaming%2C%20and%20machine%20learning.&text=With%20less%20time%20and%20money,your%20jobs%20and%20your%20data>.

upvoted 13 times

🗄️ 👤 **AzureDP900** 2 years, 2 months ago

Agreed
upvoted 1 times

🗄️ 👤 **Ekramy_Elnaggar** Most Recent 1 month, 1 week ago

Selected Answer: B

1. Managed Hadoop and Spark: Dataproc is specifically designed for running and managing Apache Spark and Hadoop clusters, which directly addresses your company's needs.
2. Scalability: Dataproc allows you to easily scale your clusters to handle the increasing number and size of jobs. You can add or remove nodes as needed to accommodate the workload.
3. Minimal Operations Work: Dataproc automates cluster creation, configuration, and management, minimizing the operational overhead. This is crucial since you want to reduce operations work.
4. Code Compatibility: Dataproc is compatible with existing Spark and Hadoop code, so you can migrate your jobs with minimal or no code changes.

upvoted 1 times

🗄️ 👤 **JohnJamesB1212** 3 months, 1 week ago

Selected Answer: B

B is correct because Dataproc is used for Apache Hadoop and Spark
upvoted 1 times

🗄️ 👤 **eka_nostra** 1 year, 4 months ago

Selected Answer: B

Dataflow for data stream and batch.
Dataproc for data process with Apache Spark and Hadoop.
Compute Engine for VM.
Kubernetes Engine for Kubernetes Cluster with Compute Engine under the hood.

upvoted 3 times

🗄️ 👤 **alekonko** 1 year, 9 months ago

Selected Answer: B

B, Dataproc is Hadoop/Spark managed service in GCP
upvoted 2 times

🗄️ 👤 **examch** 1 year, 12 months ago

Selected Answer: B

Dataproc is a fully managed and highly scalable service for running Apache Hadoop, Apache Spark, Apache Flink, Presto, and 30+ open source tools and frameworks. Use Dataproc for data lake modernization, ETL, and secure data science, at scale, integrated with Google Cloud, at a fraction of the cost.

<https://cloud.google.com/dataproc>
upvoted 1 times

🗨️ **omermahgoub** 2 years ago

To scale the number and size of Apache Spark and Hadoop jobs being run on a local datacenter with the least amount of operations work and code change, you should consider using Google Cloud Dataproc, option B. Google Cloud Dataproc is a fully-managed service that makes it easy to run Apache Spark and Hadoop workloads in the cloud. It is designed to simplify the process of setting up and managing clusters for data processing, and allows you to scale quickly and easily as demand increases.

With Cloud Dataproc, you can create and delete clusters in just a few minutes, and you can use the familiar Apache Spark and Hadoop APIs and tools to process data. This means that you can utilize the cloud to scale your workloads with minimal changes to your code and operations workflow.

Option A: Google Cloud Dataflow, option C: Google Compute Engine, and option D: Google Kubernetes Engine, would not be suitable for this use case, as they do not provide the same level of support for running Apache Spark and Hadoop workloads as Cloud Dataproc.

upvoted 2 times

🗨️ **AniketD** 2 years, 1 month ago

Selected Answer: B

B. Dataproc is managed Apache Spark and Hadoop in GCP

upvoted 1 times

🗨️ **zr79** 2 years, 2 months ago

Dataproc for Hadoop and spark ecosystem

upvoted 1 times

🗨️ **minmin2020** 2 years, 2 months ago

Selected Answer: B

B. Google Cloud Dataproc

upvoted 1 times

🗨️ **holerina** 2 years, 3 months ago

B data proc for hadoop and spark

upvoted 1 times

🗨️ **Dhiraj03** 2 years, 6 months ago

Keyword - Apache Spark and Hadoop jobs - Go with Dataproc

upvoted 1 times

🗨️ **Superr** 2 years, 6 months ago

Selected Answer: B

dataproc

upvoted 1 times

🗨️ **Nirca** 2 years, 8 months ago

Selected Answer: B

Google Cloud Dataproc == managed Spark and Hadoop service

upvoted 2 times

🗨️ **pakochiu** 2 years, 8 months ago

Selected Answer: B

B - Dataproc Lift&Shift of Apache Spark and Hadoop jobs

upvoted 1 times

🗨️ **Ilanerox** 2 years, 10 months ago

B is ok.

upvoted 1 times

Question #19

Topic 1

The database administration team has asked you to help them improve the performance of their new database server running on Google Compute Engine. The database is for importing and normalizing their performance statistics and is built with MySQL running on Debian Linux. They have an n1-standard-8 virtual machine with 80 GB of SSD persistent disk.

What should they change to get better performance from this system?

- A. Increase the virtual machine's memory to 64 GB
- B. Create a new virtual machine running PostgreSQL
- C. Dynamically resize the SSD persistent disk to 500 GB
- D. Migrate their performance metrics warehouse to BigQuery
- E. Modify all of their batch jobs to use bulk inserts into the database

 **shandy** Highly Voted 5 years ago



Answer is C because persistent disk performance is based on the total persistent disk capacity attached to an instance and the number of vCPUs that the instance has. Incrementing the persistent disk capacity will increment its throughput and IOPS, which in turn improve the performance of MySQL.

upvoted 68 times

  **Eroc** Highly Voted 5 years, 1 month ago



Assuming that the database is approaching its hardware limits... both options A and C would improve performance, A would increase number CPUs and memory, but C would increase memory by more. If it a software problem, it is likly it is a hashing problem (the search and sort algorithms are not specific enough to search within the database). This problem would not be fixed just by migrating to PostgreSQL or BigQu but modifying the inserts would help the situation because it would entail specifications of data lookups. However, it wouldn't help with search performance just inserts and it doesn't help in normalization. So B, D, and E are eliminated. Since statistics is based on sets, the larger the number of sets the better the predictions. This means that the largest amount of memory would not only increase computer performance but knowledge enhancements. So C beats A.

upvoted 35 times

  **nitinz** 3 years, 9 months ago



C. universal truth - OLTP D/B performance is depended on IOPs. SSD is the best solution for higher IOPs. In GCP bigger the disk size high the IOPs.

upvoted 8 times

  **trainor** 4 years ago

Also, if you increased the memory size, it would not be a n1-standard-8 anymore. You should eventually change machine type, not simply increase memory.

upvoted 6 times

  **tartar** 4 years, 4 months ago



C is ok.

upvoted 8 times

  **haroldbenites** 3 years ago

When you increase the memory yo need to shutdown the machine, but when you increase the disk, it is not necessary. Answer is B.

upvoted 2 times

  **Ric350** 4 months ago

Repectfully, this isn't accurate. On Google Compute Engine, you can often increase the memory of a running virtual machine without needing to shut it down. This is known as live migration or memory hot-add.

upvoted 1 times

  **Mission94** 6 months, 1 week ago

Since its using SQL, so there will be a Maintenance Window, so this change can be implemented during the downtime(also there is no mention that the system should be always available)

upvoted 1 times

  **Dclaiborne41** 2 years, 7 months ago

there isn't "without downtime"



upvoted 1 times

  **Ekramy_Elnaggar** Most Recent 1 month, 1 week ago

Selected Answer: C

OLTP D/B performance is depended on IOPs. SSD is the best solution for higher IOPs. In GCP bigger the disk size higher the IOPs.

upvoted 3 times

  **Hungdv** 4 months, 1 week ago

Choose C

upvoted 1 times

  **ukivanlamlpi** 5 months, 1 week ago

Selected Answer: A

increase size will not increase performance, it either increase RAM or serverless. A or D. if no cost concern will pick D

upvoted 2 times

  **ashishdwi007** 11 months ago

Selected Answer: C

I was looking for CloudSQL in options, since it is not there, C is best

upvoted 1 times

🗨️ **hzaoui** 11 months, 1 week ago

Selected Answer: E

The fact that the database is used for importing and normalizing performance statistics suggests frequent data insertions. Optimizing this process through bulk inserts directly addresses a likely performance bottleneck.

upvoted 1 times

🗨️ **JohnDohertyDoe** 11 months, 2 weeks ago

Selected Answer: A

The answer according to Google is A. This question is part of the Google's sample questions for the certification.

upvoted 1 times

🗨️ **ccpmad** 6 months, 1 week ago

Yes, it is, and says it is C.

upvoted 1 times

🗨️ **JPA210** 1 year, 2 months ago

I see most of the people here replying C, but I do not think that the size of the disk we bring much gains in performance. D, yes, seems to me that will bring much improvements in performance, management, operations and cost. So B, Migrate their performance metrics warehouse to BigQuery

upvoted 3 times

🗨️ **Palan** 1 year, 4 months ago

I would go with option E because Bulk Insert improves performance drastically unless it is been implemented already.

upvoted 1 times

🗨️ **eka_nostra** 1 year, 4 months ago

Selected Answer: C

Increasing disk size will also increase its performance.

https://cloud.google.com/compute/docs/disks/performance#optimize_disk_performance

upvoted 4 times

🗨️ **JC0926** 1 year, 8 months ago

Selected Answer: C

C. Dynamically resize the SSD persistent disk to 500 GB

By increasing the size of the SSD persistent disk, the database server can achieve better performance. A larger SSD persistent disk provides higher IOPS (input/output operations per second) and throughput, allowing for faster read and write operations. This can help improve the performance of the MySQL database server running on the Google Compute Engine instance.

upvoted 3 times

🗨️ **mifrah** 1 year, 9 months ago

On another website I found the question with the hint "you are not allowed to reboot the VM before next maintenance window". That makes it more clear --> C.

upvoted 1 times

🗨️ **JC0926** 1 year, 9 months ago

Selected Answer: E

Question #20

Topic 1

You want to optimize the performance of an accurate, real-time, weather-charting application. The data comes from 50,000 sensors sending 10 readings a second, in the format of a timestamp and sensor reading.

Where should you store the data?

- A. Google BigQuery
- B. Google Cloud SQL
- C. Google Cloud Bigtable
- D. Google Cloud Storage

🗲️ 👤 **victory108** Highly Voted 👍 3 years, 5 months ago

C. Google Cloud Bigtable
upvoted 12 times

🗲️ 👤 **khadar** 2 years, 3 months ago

I too got this question in 10-09-22 exam with similar option and result is pass
upvoted 4 times

🗲️ 👤 **Ekramy_Elnaggar** Most Recent 🕒 1 month, 1 week ago

Selected Answer: C

1. High Write Throughput: Bigtable excels at handling high-volume write operations, which is crucial for your application receiving data from 50,000 sensors sending 10 readings per second.
2. Low Latency: Bigtable offers very low latency for read operations, essential for real-time charting and data visualization.
3. Time-Series Data: Bigtable is well-suited for storing and querying time-series data, like your weather sensor readings with timestamps.
4. Scalability: Bigtable can handle massive amounts of data and scale seamlessly as your application grows.
upvoted 2 times

🗲️ 👤 **lisabisa** 10 months ago

Bigtable - NoSQL, high-throughput, low-latency, making it suitable for storing time-series data from sensors
upvoted 2 times

🗲️ 👤 **alekonko** 1 year, 9 months ago

Selected Answer: C

BigTable is NoSQL for IoT
upvoted 2 times

🗲️ 👤 **sivaamum** 1 year, 9 months ago

C is correct
upvoted 1 times

🗲️ 👤 **omermahgoub** 2 years ago

To optimize the performance of an accurate, real-time, weather-charting application that receives data from 50,000 sensors sending 10 readings per second, it would be most appropriate to store the data in a distributed, horizontally scalable, NoSQL database such as Google Cloud Bigtable
Other options, such as Google BigQuery, Google Cloud SQL, and Google Cloud Storage, may not be as well-suited for handling high volumes of real-time data and may not provide the same level of performance and scalability as Google Cloud Bigtable.
upvoted 3 times

🗲️ 👤 **Bry_040706** 2 years ago

C. Bigtable, IoT data.
upvoted 1 times

🗲️ 👤 **AniketD** 2 years, 1 month ago

Selected Answer: C

C Bigtable
upvoted 1 times

🗲️ 👤 **Mahmoud_E** 2 years, 1 month ago

Selected Answer: C

C bigtable right answer
upvoted 1 times

🗲️ 👤 **zr79** 2 years, 2 months ago

real-time, IoT, time series and huge writes are some of the keywords to look after for Bigtable
upvoted 4 times

🗲️ 👤 **minmin2020** 2 years, 2 months ago

Selected Answer: C

C. Google Cloud Bigtable
upvoted 1 times

🗳️ 👤 **holerina** 2 years, 3 months ago

C big table for IOT data

upvoted 1 times

🗳️ 👤 **abirroy** 2 years, 3 months ago

Selected Answer: C

Google Cloud Bigtable

upvoted 1 times

🗳️ 👤 **Dhiraj03** 2 years, 6 months ago

Keyword - Timestamp - Big table

upvoted 2 times

🗳️ 👤 **szanio** 2 years, 6 months ago

Go for c

upvoted 1 times

🗳️ 👤 **Nirca** 2 years, 8 months ago

Selected Answer: C

C. Google Cloud Bigtable is the Best Practice option

upvoted 1 times

🗳️ 👤 **belly265** 2 years, 10 months ago

Ans c - when ever thier is input from IOT devices across and time series data which is huge go for big table in gcp

upvoted 2 times

🗳️ 👤 **AzureDP900** 2 years, 2 months ago

Big Table is right choice, hence C is correct

upvoted 1 times

Question #21

Topic 1

Your company's user-feedback portal comprises a standard LAMP stack replicated across two zones. It is deployed in the us-central1 region and uses autoscaled managed instance groups on all layers, except the database. Currently, only a small group of select customers have access to the portal. The portal meets a

99,99% availability SLA under these conditions. However next quarter, your company will be making the portal available to all users, including unauthenticated users. You need to develop a resiliency testing strategy to ensure the system maintains the SLA once they introduce additional user load.

What should you do?

- A. Capture existing users input, and replay captured user load until autoscale is triggered on all layers. At the same time, terminate all resources in one of the zones
- B. Create synthetic random user input, replay synthetic load until autoscale logic is triggered on at least one layer, and introduce λ chaos λ to the system by terminating random resources on both zones
- C. Expose the new system to a larger group of users, and increase group size each day until autoscale logic is triggered on all layers. At the same time, terminate random resources on both zones
- D. Capture existing users input, and replay captured user load until resource utilization crosses 80%. Also, derive estimated number of users based on existing user's usage of the app, and deploy enough resources to handle 200% of expected load

🗳️ 👤 **jcMoranp** **Highly Voted** 🍌 5 years, 1 month ago

resilience test is not about load, is about terminate resources and service not affected. Think it's B. The best for resilience in to introduce chaos the infrastructure

upvoted 90 times

  **rockstar9622** 4 years, 11 months ago

I agree with @jcmoranp, B) is correct for more info - https://cloud.google.com/solutions/scalable-and-resilient-apps#test_your_resilience
upvoted 20 times

  **AWSPro24** 3 years, 1 month ago

Isn't A superior in one way. It will demonstrate that the app is regionally redundant by demonstrating it can survive the loss of an entire zone. B only demonstrates the app is zonally redundant and can lose a random instance here and there within individual zones which is not that resilient. Thoughts?

upvoted 8 times

  **0xE8D4A51000** 2 years, 2 months ago

No. It is only terminating the service in ONE zone. B caters for terminating the service in both zones randomly. You want to be able to terminate the service when either zone has an outage.

upvoted 7 times

  **OSNG** Highly Voted 4 years ago

Will go with A. Reason:

1. SLA in question is about the Availability (The portal meets a 99.99% availability SLA under these conditions.) therefore maintaining SLA means Availability.
2. Its a user-feedback portal and type of user input is going to be similar or same (A is capturing the user input and replaying it).

Why not B:

The infrastructure is using MIG (Instances created using templates) most likely to be used with Health Check and killing random VMs cannot terminate the availability (neither affect the availability as health check will immediately kill the effected Instances and create the other one.)

Why not D:

SLA is about Availability not reliability or scaling. (As all of it does work hand to hand but still major focus should be on availability.)

--- IF AGREE PLEASE UP VOTE TO MAKE IT CLEAR FOR THE OTHERS --- Thank you.

upvoted 62 times

  **RitwickKumar** 2 years, 4 months ago

Only problem with A is that it says "replay captured user load". We are not testing for the incoming unpredictable load due to the inclusion of unauthenticated users and something that we haven't captured earlier.

Option B covers breadth and depth for the desired SLA.

upvoted 10 times

  **jay9114** 2 years, 3 months ago



What does "replay captured user load" mean?

upvoted 3 times

  **bolu** 3 years, 11 months ago


valuable input in terms of 'availability'. did you select this answer in exam too?

upvoted 1 times

  **amxexam** 3 years, 3 months ago

We are talking about resilience testing where as SLA is an argument of the system.

upvoted 1 times

  **amxexam** 3 years, 3 months ago

And resilience means the capacity to recover from failure.

upvoted 1 times

  **AWSPro24** 3 years, 1 month ago

A ensures the app can withstand the loss of a whole Zone which I think is important as well.


upvoted 1 times

  **Ekramy_Elnaggar** Most Recent 1 month, 1 week ago

Selected Answer: B

1. Synthetic Load Generation: Creating synthetic user input allows you to simulate a wide range of user behavior and load patterns, including spikes and sustained high traffic. This helps you test the system's ability to scale and handle unexpected loads.
2. Autoscaling Validation: By replaying the synthetic load, you can verify that the autoscaling logic is working correctly across all layers of the LAMP stack. This ensures that the system can dynamically adjust resources to meet demand.
3. Chaos Engineering: Introducing chaos by terminating random resources simulates real-world failures and helps you test the system's resilience to unexpected disruptions. This is crucial for maintaining the 99.99% availability SLA.
4. Controlled Environment: This approach allows you to conduct testing in a controlled environment without impacting real users. You can gradually increase the load and introduce chaos in a measured way to identify weaknesses and improve resilience.



upvoted 2 times

  **potorange** 3 months, 3 weeks ago

Selected Answer: A

A: requirements states "all layers" and "resiliency testing"

upvoted 1 times

  **Robert0** 6 months, 3 weeks ago

Selected Answer: A

I would go with A.

This solution test autoscale policy of each layer (not only one as option B refers).

Also, it propose a regional shutdown. This is a very good test commonly requested if your application is geo-redundant. In contrast, option B propose random termination of resources, not a bad practice but a little bit vague that can be implemented terrible wrong (for example you do not kill the interesting services or you kill the same service in both regions, thus generating a blackout)

upvoted 1 times

  **lisabisa** 10 months ago

Selected Answer: B

We need to do 1. load testing and 2. reliability test (failover redundancy)

B does both

A only tests one zone

C impacts real user experience

D 200% not necessary

upvoted 4 times

  **AdityaGupta** 1 year, 2 months ago

Selected Answer: A

You need to develop a resiliency testing strategy to ensure the system maintains the SLA once they introduce additional user load.

Need to maintain SLA of 99.9% means multiple zones, resilience means fault tolerance. Terminating all resources in one zone is also creating a chaos.

upvoted 1 times

  **heretolearnazure** 1 year, 3 months ago

B is correct



upvoted 1 times

  **VaraSrinvas** 1 year, 6 months ago

Selected Answer: D

Option D is the best resiliency testing strategy in this scenario as it ensures that the system is tested with actual user data, takes into account expected increase in user load, and ensures that the system is adequately scaled to handle the anticipated load.

upvoted 2 times

  **didek1986** 1 year, 4 months ago

Do not agree. B is 100% correct

upvoted 1 times

  **jrisl1991** 1 year, 2 months ago

But this would assume that the user load will not change; plus, the current application is visible only to a small group of select customers - this is the current production setup. The deployment should be prepared for all existing users plus unauthenticated users, and the load increase is unknown, so testing for 200% of "expected load" is very ambiguous.

upvoted 1 times



  **JC0926** 1 year, 8 months ago

Selected Answer: B

B. Create synthetic random user input, replay synthetic load until autoscale logic is triggered on at least one layer, and introduce λ chaos λ to system by terminating random resources on both zones.

By creating synthetic random user input and replaying the load, you can simulate the expected increased user traffic and trigger the autoscale logic on different layers of the application. Introducing chaos to the system by terminating random resources in both zones helps test the resiliency and redundancy of the system under stress. This strategy will help ensure that the system can maintain the 99.99% availability SLA when subjected to additional user load.

upvoted 5 times

  **telp** 1 year, 9 months ago

Selected Answer: B

chaos == test resilience for google

upvoted 2 times

  **Deb2293** 1 year, 9 months ago

Selected Answer: B

This is chaos engineering used by Netflix. <https://netflixtechblog.com/tagged/chaos-engineering>

upvoted 2 times

  **roaming_panda** 1 year, 11 months ago

Selected Answer: B

chaos == checking resilience

upvoted 4 times

  **holerina** 2 years ago

right thought ,

upvoted 1 times

  **AniketD** 2 years, 1 month ago

Selected Answer: B

B is correct; Using synthetic/random input is recommended. Chaos Engineering/Symian Army from Netflix is one of the proven mechanism to test the resilience of the application.

upvoted 2 times

  **megumin** 2 years, 1 month ago

Selected Answer: B

B is ok

upvoted 2 times

  **0xE8D4A51000** 2 years, 2 months ago

Selected Answer: B

B caters for terminating the service in both zones randomly. You want to be able to test resiliency when either zone has an outage.

upvoted 3 times

Question #22

Topic 1

One of the developers on your team deployed their application in Google Container Engine with the Dockerfile below. They report that their

application deployments are taking too long.

```
FROM ubuntu:16.04
```

```
COPY . /src
```

```
RUN apt-get update && apt-get install -y python python-pip
```

```
RUN pip install -r requirements.txt
```

You want to optimize this Dockerfile for faster deployment times without adversely affecting the app's functionality.

Which two actions should you take? (Choose two.)

- A. Remove Python after running pip
- B. Remove dependencies from requirements.txt
- C. Use a slimmed-down base image like Alpine Linux
- D. Use larger machine types for your Google Container Engine node pools
- E. Copy the source after the package dependencies (Python and pip) are installed

 **aviratna** Highly Voted 3 years, 5 months ago

C & E:

C: Smaller the base image with minimum dependency faster the container will start

E: Docker image build uses caching. Docker Instructions sequence matter because application's dependencies change less frequently than the Python code which will help to reuse the cached layer of dependency and only add a new layer for code change for Python Source code.

upvoted 58 times

 **vincy2202** Highly Voted 3 years ago

C & E are the correct answers.

Kindly refer - <https://www.docker.com/blog/intro-guide-to-dockerfile-best-practices/>

upvoted 12 times

 **Ekramy_Elnaggar** Most Recent 1 month, 1 week ago

Selected Answer: CE

C. Use a slimmed-down base image like Alpine Linux: The ubuntu:16.04 image is a full-fledged operating system, which means it's larger and takes longer to download and build. Alpine Linux is a minimal distribution designed for containers, resulting in significantly smaller images and faster deployments.

E. Copy the source after the package dependencies (Python and pip) are installed: Docker builds images in layers. Each RUN, COPY, and ADD instruction creates a new layer. By copying the source code after installing dependencies, you can take advantage of Docker's caching mechanism. If your source code changes, only the layers related to the source code need to be rebuilt, not the layers related to dependencies.

upvoted 2 times

 **alessandroGPC** 7 months, 4 weeks ago

Hi everyone, I have a doubt. The question talks about "deployment" not "build". CE are more correct, in my opinion, to accelerate the build process (and application management) rather than the simple deployment (with docker swarm, simple docker, kubernetes etc etc)

upvoted 1 times

 **nickcin77** 12 months ago

C&E

C - Use small images.

E - "Try to make expensive steps appear near the beginning of the Dockerfile. Steps that change often should appear near the end of the Dockerfile"

<https://docs.docker.com/build/cache/>

upvoted 3 times

 **jrisl1991** 1 year, 2 months ago

Selected Answer: CE

C is an obvious choice, as an optimized image will be much better for a container than the one in the file. For E, I found an explanation here about COPY that helped me confirm that's the other solution - https://test-dockerr.readthedocs.io/en/latest/userguide/eng-image/dockerfile_best-practices/#:~:text=required%20files%20change.-,For%20example%3A,-COPY%20requirements.txt

upvoted 1 times