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 66 DISCUSSION

Actual exam question from Google's Professional Data Engineer

Question #: 66

Topic #: 1

[All Professional Data Engineer Questions]

You set up a streaming data insert into a Redis cluster via a Kafka cluster. Both clusters are running on Compute Engine instances. You need to encrypt data at rest with encryption keys that you can create, rotate, and destroy as needed. What should you do?

A. Create a dedicated service account, and use encryption at rest to reference your data stored in your Compute Engine cluster instances as part of your API service calls.

B. Create encryption keys in Cloud Key Management Service. Use those keys to encrypt your data in all of the Compute Engine cluster instances.

C. Create encryption keys locally. Upload your encryption keys to Cloud Key Management Service. Use those keys to encrypt your data in all of the Compute Engine cluster instances.

D. Create encryption keys in Cloud Key Management Service. Reference those keys in your API service calls when accessing the data in your Compute Engine cluster instances.

Show Suggested Answer

by [deleted] at *March 21, 2020, 4:16 p.m.*

Comments

Type your comment...

Submit

SonuKhan1 Highly Voted 3 years ago

Dear Admin, almost every answer is incorrect . Please check the comments and update your website.

upvoted 57 times

[Removed] Highly Voted 4 years, 7 months ago

correct: B

upvoted 22 times

[Removed] 4 years, 7 months ago

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

upvoted 4 times

tprashanth 4 years, 3 months ago

Based on the info at the link you referred, it seems C is the right answer

upvoted 4 times

baubaumiaomiao 2 years, 10 months ago

If you create it locally, you can't rotate keys. Answer should be B

upvoted 2 times

and88x Most Recent 4 months ago

Selected Answer: B

D is incorrect because referencing keys in API service calls doesn't meet the requirements for encrypting data at rest. This approach is more related to accessing data at runtime, not storing it securely.

upvoted 1 times

AmitK121981 4 months, 3 weeks ago

Selected Answer: C

CMEK is where customer managers keys, but are still created by Google (this is for KMS). CSEK is where keys are created outside GCP and used by API calls. So if customer has to create keys, it has to be outside KMS

upvoted 1 times

jatinbhatia2055 4 months, 3 weeks ago

Selected Answer: D

Best Option: This is the most accurate approach. Cloud KMS provides the ability to create, manage, and rotate encryption keys. You can use the KMS API to reference the keys when encrypting and decrypting your data. In this case, you would integrate the KMS keys into your application logic (e.g., Kafka producers/consumers, Redis clients) to encrypt and decrypt data as it is stored or processed. This approach leverages the full functionality of Cloud KMS, including the ability to rotate and destroy keys as needed.

upvoted 2 times

zevexWM 6 months, 3 weeks ago

But KMS doesnt create keys. It only stores them right?

upvoted 1 times

TVH_Data_Engineer 10 months, 3 weeks ago

Selected Answer: B

Google Cloud Key Management Service (KMS) provides a centralized cloud service for managing cryptographic keys. By creating encryption keys in Cloud KMS, you can easily manage the lifecycle of these keys, including creation, rotation, and destruction.
WYY NOT Create Keys Locally and Upload to Cloud KMS?
While it’s possible to create keys locally and then upload them to Cloud KMS, it’s generally simpler and more secure to create the keys directly in Cloud KMS. This reduces the risk associated with transferring keys and leverages the security and compliance features of Cloud KMS.

upvoted 1 times

emmylou 1 year ago

Help!
I chose "C" because of the statement, "encrypt data at rest with encryption keys that you can create, rotate, and destroy as needed" and read that as needing to generate the keys locally. Can you please explain where I went wrong?

upvoted 2 times

odiez3 1 year, 3 months ago

the answer is C Read the full statement.

" You need to encrypt data at rest with encryption keys that you can create " upvoted 1 times

theseawillclaim 1 year, 3 months ago

Selected Answer: B

B!
C is useless overhead and you cannot rotate that easily!

upvoted 1 times

Kiroo 1 year, 5 months ago

Well for what I remember from cloud arch and what I found in https://cloud.google.com/compute/docs/disks/customer-managed-encryption

There is two options or the customer manage entirely or he will use the service to generate the keys so based on that is the B

upvoted 1 times

samdhimal 1 year, 9 months ago

B. Create encryption keys in Cloud Key Management Service. Use those keys to encrypt your data in all of the Compute Engine cluster instances.

Cloud Key Management Service (KMS) is a fully managed service that allows you to create, rotate, and destroy encryption keys as needed. By creating encryption keys in Cloud KMS, you can use them to encrypt your data at rest in the Compute Engine cluster instances, which is running your Redis and Kafka clusters. This ensures that your data is protected even when it is stored on disk.

upvoted 2 times

samdhimal 1 year, 9 months ago

Option A: Create a dedicated service account, and use encryption at rest to reference your data stored in your Compute Engine cluster instances as part of your API service calls is not the best option as it does not provide encryption at rest.

Option C: Create encryption keys locally. Upload your encryption keys to Cloud Key Management Service. Use those keys to encrypt your data in all of the Compute Engine cluster instances, is not the best option as it does not provide a way to manage the encryption keys centrally.

Option D: Create encryption keys in Cloud Key Management Service. Reference those keys in your API service calls when accessing the data in your Compute Engine cluster instances, is not the best option as it does not provide encryption at rest, it only secure the data in transit.

upvoted 3 times

DGames 1 year, 10 months ago

Selected Answer: B

B is correct answer generate key using KMS, why locally again it is overhead to upload and use everywhere.

upvoted 1 times

Atnafu 1 year, 11 months ago

B
If you use Google Cloud, Cloud Key Management Service lets you create your own encryption keys that you can use to add envelope encryption to your data. Using Cloud KMS, you can create, rotate, track, and delete keys.
https://cloud.google.com/docs/security/encryption/default-encryption#:~:text=If%20you%20use%20Google%20Cloud%2C%20Cloud%20Key%20Management%20Service%20lets%20you%20create%20your%20own%20encryption%20keys%20that%20you%20can%20use%20to%20add%20envelope%20encryption%20to%20your%20data.%20Using%20Cloud%20KMS%2C%20you%20can%20create%2C%20rotate%2C%20track%2C%20and%20delete%20keys.

upvoted 1 times

medeis_jar 2 years, 10 months ago

Selected Answer: B

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

upvoted 1 times

MaxNRG 2 years, 10 months ago

Selected Answer: B

A makes no sense, you need to use your own keys.
You don’t create keys locally and upload them, you should import it to make it work..using the kms public key…not just “uploading” it. C is also out.
IT’s between B and D
Cloud KMS is a cloud-hosted key management service that lets you manage cryptographic keys for your cloud services the same way you do on-premises, You can generate, use, rotate, and destroy cryptographic keys from there.
Since you want to encrypt data at rest, is B, you don’t use them for any API calls.
https://cloud.google.com/compute/docs/disks/customer-managed-encryption

upvoted 8 times

lg1234 3 years ago

I believe you cannot upload custom keys to KMS for Compute Engine. Only via API Calls. See:
https://cloud.google.com/security/encryption/customer-supplied-encryption-keys
With that said, option B

upvoted 2 times

Load full discussion...

Platform

- Home
- Examtopics PRO
- All Exams
- Training Courses