



Exam AWS Certified Developer - Associate DVA-C02 All Questions

View all questions & answers for the AWS Certified Developer - Associate DVA-C02 exam

[Go to Exam](#)

EXAM AWS CERTIFIED DEVELOPER - ASSOCIATE DVA-C02 TOPIC 1 QUESTION 7 DISCUS...

Exam question from Amazon's AWS Certified Developer - Associate DVA-C02

Question #: 7

Topic #: 1

[\[All AWS Certified Developer - Associate DVA-C02 Questions\]](#)

A developer has an application that makes batch requests directly to Amazon DynamoDB by using the BatchGetItem low-level API operation. The responses frequently return values in the UnprocessedKeys element.

Which actions should the developer take to increase the resiliency of the application when the batch response includes values in UnprocessedKeys? (Choose two.)

- A. Retry the batch operation immediately.
- B. Retry the batch operation with exponential backoff and randomized delay.
- C. Update the application to use an AWS software development kit (AWS SDK) to make the requests.
- D. Increase the provisioned read capacity of the DynamoDB tables that the operation accesses.
- E. Increase the provisioned write capacity of the DynamoDB tables that the operation accesses.

[Show Suggested Answer](#)

by [aragon_saa](#) at March 16, 2023, 10 a.m.

Disclaimers:

- ExamTopics website is not related to, affiliated with, endorsed or authorized by Amazon.
- Trademarks, certification & product names are used for reference only and belong to Amazon.

Comments

[brandon87](#) [Highly Voted](#) 2 years, 1 month ago

[Selected Answer: BD](#)

(B) If you delay the batch operation using exponential backoff, the individual requests in the batch are much more likely to succeed.

(D) The most likely cause of a failed read or a failed write is throttling. For BatchGetItem, one or more of the tables in the batch request does not have enough provisioned read capacity to support the operation
<https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/Programming.Errors.html#Programming.Errors.RetryAndBackoff>

upvoted 24 times

[Untamables](#) [Highly Voted](#) 2 years, 1 month ago

[Selected Answer: BC](#)

B & C

<https://docs.aws.amazon.com/general/latest/gr/api-retries.html>

upvoted 19 times

[konieczny69](#) 1 year, 3 months ago

C already handles retries, why would want to do that manually?

upvoted 1 times

[test_test_raf](#) 1 month ago

Note that this is not a combination of two, these are two separate options that could work - choose SDK over low level API which, as you said, handles retries or implement retries yourself.

upvoted 1 times

[wtf3344](#) [Most Recent](#) 4 months ago

[Selected Answer: BD](#)

BatchWriteItem

- UnprocessedItems for failed write operations (exponential backoff or add WCU)

BatchGetItem

- UnprocessedKeys for failed read operations (exponential backoff or add RCU)

So answer is B & D

upvoted 2 times

[Hasitha99](#) 4 months, 1 week ago

[Selected Answer: BC](#)

D also could be an answer. But is not the most suitable since there is a possibility of getting this result intermittently even though we have necessary capacity.

upvoted 1 times

[Arad](#) 4 months, 3 weeks ago

[Selected Answer: BC](#)

BC is correct answer.

upvoted 1 times

🗳️ 👤 **sumanshu** 4 months, 4 weeks ago

Selected Answer: BD

DynamoDB BatchGetItem API operation, the request fetches multiple items in one operation. However, DynamoDB has limits on the resources it allocates for processing requests. If it cannot process some items within a batch request due to these limits, it returns the processed items (the ones it successfully retrieved), and for the remaining items, it includes them in the UnprocessedKeys element in the response.

upvoted 2 times

🗳️ 👤 **trieudo** 5 months, 1 week ago

Selected Answer: BC

Keyword: increase, resiliency, application, response, UnprocessedKeys(Unprocessed due to be not enough infra)

==> Discard A: horrible way, push so many traffic to busy system

==> Discard D, E: Error due to huge traffic, it can be unlimited. Don't come from lacking resource ==> Your resource, your money is limit >< unpredicted huge traffic

B: is good way to call x2 times after failed, call again after 1s -> 2s -> 4s -> 8s -> ... until success

C: is intergrate method of 'B' option automaically inside SDK

upvoted 1 times

🗳️ 👤 **9d8dd9c** 7 months ago

Selected Answer: BD

BD

C handles retry but using SDK is not necessary here

upvoted 1 times

🗳️ 👤 **Phongsanth** 7 months, 3 weeks ago

Selected Answer: BD

I vote for B,D

Scroll down to the bottom of this page and you will see the reason. I paste some of words here.

The most likely cause of a failed read or a failed write is throttling. For BatchGetItem, one or more of the tables in the batch request does not have enough provisioned read capacity to support the operation. For BatchWriteItem, one or more of the tables does not have enough provisioned write capacity.

If DynamoDB returns any unprocessed items, you should retry the batch operation on those items. However, we strongly recommend that you use an exponential backoff algorithm. If you retry the batch operation immediately, the underlying read or write requests can still fail due to throttling on the individual tables. If you delay the batch operation using exponential backoff, the individual requests in the batch are much more likely to succeed.

<https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/Programming.Errors.html>

upvoted 3 times

🗳️ 👤 **eberhe900** 7 months, 3 weeks ago

Selected Answer: BD

Backoff and Retry Strategy:

Implement a backoff and retry strategy to prevent overwhelming the DynamoDB service with repeated requests.

Apply an exponential backoff algorithm, where you progressively increase the delay between each retry attempt.

Consider implementing a maximum number of retries to avoid an infinite retry loop.

Fine-Tuning DynamoDB Provisioned Capacity:

If you consistently encounter unprocessed items during batch operations, it may indicate that your DynamoDB table's provisioned capacity is insufficient.

Monitor the table's consumed capacity and adjust the provisioned capacity (read capacity units) accordingly to handle the load and reduce the occurrence of unprocessed items.

By implementing these steps, you can effectively handle unprocessed items returned by the BatchGetItem operation in DynamoDB and ensure that all items are processed successfully.

upvoted 2 times

🗳️ 👤 **SherzodBek** 7 months, 3 weeks ago

Selected Answer: BD

B & D.

B is correct. Because in the question, it is mentioned that low-level API is being used. It means exponential backoff can be implemented manually.

D is correct. Because there is a frequently keyword in the question. If UnprocessedKeys error occurs frequently, DynamoDB doesn't have enough capacity to process requests. So read capacity should be increased.

upvoted 3 times

🗳️ 👤 **TheFivePips** 7 months, 3 weeks ago

Selected Answer: BD

Exponential backoff with randomized delay is a common technique used to handle transient failures and throttle errors in distributed systems like DynamoDB. This approach involves retrying the failed operation after waiting for an increasing amount of time, which helps reduce the load on the service and increases the likelihood of success during periods of high demand or throttling.

If the BatchGetItem operation frequently returns values in the UnprocessedKeys element, it indicates that the table's read capacity might be insufficient to handle the requested workload. By increasing the provisioned read capacity for the DynamoDB tables, the application can better handle the read throughput requirements and reduce the likelihood of encountering UnprocessedKeys in batch responses.

AWS SDK might provide additional features and simplifications for making requests, it does not directly address the issue of UnprocessedKeys in batch responses. This option might be beneficial for improving code maintainability and leveraging SDK features however.

upvoted 1 times

🗳️ 👤 **phongnx8** 9 months, 2 weeks ago

BC is the correct answer.

upvoted 1 times

🗳️ 👤 **Anandesh** 10 months ago

Selected Answer: BC

https://docs.aws.amazon.com/amazondynamodb/latest/APIReference/API_BatchGetItem.html

<https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/Programming.Errors.html#Programming.Errors.RetryAndBackoff>

upvoted 1 times

🗳️ 👤 **65703c1** 12 months ago

Selected Answer: BC

BC is the correct answer.


upvoted 1 times

  **drycleansing** 1 year, 1 month ago

Selected Answer: BC

among B C D, it is hard to say D copes with the problem directly I guess. Increasing RCU will affects the ratio of unprocessed items but that does not mean it handles the unprocessed items.

upvoted 1 times

  **vinfo** 1 year, 1 month ago

Selected Answer: BD

B,D. La combinación de estrategias es ideal para este comportamiento

upvoted 1 times