



## 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 12 DISCU...

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

Question #: 12

Topic #: 1

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

A developer is creating an application that will give users the ability to store photos from their cellphones in the cloud. The application needs to support tens of thousands of users. The application uses an Amazon API Gateway REST API that is integrated with AWS Lambda functions to process the photos. The application stores details about the photos in Amazon DynamoDB.

Users need to create an account to access the application. In the application, users must be able to upload photos and retrieve previously uploaded photos. The photos will range in size from 300 KB to 5 MB.

Which solution will meet these requirements with the LEAST operational overhead?

- A. Use Amazon Cognito user pools to manage user accounts. Create an Amazon Cognito user pool authorizer in API Gateway to control access to the API. Use the Lambda function to store the photos and details in the DynamoDB table. Retrieve previously uploaded photos directly from the DynamoDB table.
- B. Use Amazon Cognito user pools to manage user accounts. Create an Amazon Cognito user pool authorizer in API Gateway to control access to the API. Use the Lambda function to store the photos in Amazon S3. Store the object's S3 key as part of the photo details in the DynamoDB table. Retrieve previously uploaded photos by querying DynamoDB for the S3 key.
- C. Create an IAM user for each user of the application during the sign-up process. Use IAM authentication to access the API Gateway API. Use the Lambda function to store the photos in Amazon S3. Store the object's S3 key as part of the photo details in the DynamoDB table. Retrieve previously uploaded photos by querying DynamoDB for the S3 key.
- D. Create a users table in DynamoDB. Use the table to manage user accounts. Create a Lambda authorizer that validates user credentials against the users table. Integrate the Lambda authorizer with API Gateway to control access to the API. Use the Lambda function to store the photos in Amazon S3. Store the object's S3 key as part of the photo details in the DynamoDB table. Retrieve previously uploaded photos by querying DynamoDB for the S3 key.

Show Suggested Answer

by [Untamables](#) at March 21, 2023, 6:35 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

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

**Selected Answer: B**

B

<https://docs.aws.amazon.com/apigateway/latest/developerguide/apigateway-integrate-with-cognito.html>  
<https://aws.amazon.com/blogs/big-data/building-and-maintaining-an-amazon-s3-metadata-index-without-servers/>  
 upvoted 12 times

[jayvarma](#) Highly Voted 1 year, 9 months ago

As it is not a good practice to create a new IAM user for each user that signs up for the application, Option C is ruled out. Amazon Cognito user pools primary purpose is to authenticate and authorize web and mobile applications.

As the solution requires the application to store images that are between 300KB and 5MB in size, The idea of storing the images in the DynamoDB is ruled out because the object size in a dynamoDb table cannot exceed 400kb. The ideal solution for this problem would be to store the photos in S3 and store the object's key in the DynamoDB table.

So, Option B is the right answer

upvoted 6 times

[sumanshu](#) Most Recent 4 months, 4 weeks ago

**Selected Answer: B**

- A) Eliminated - DynamoDB is not optimized for large binary objects
  - C) Eliminated - Creating IAM users for each application user is not scalable
  - D) Eliminated - Using a custom table and Lambda authorizer for user authentication increases operational complexity.
- upvoted 1 times

[trieudo](#) 5 months, 1 week ago

**Selected Answer: B**

keyword: LEAST operational overhead, size from 300 KB to 5 MB.

==> discard A: item limit upto 400KB only

==> discard C: each user per IAM user --> really bad practice

==> discard D: violate "LEAST operational overhead", you must build a lot (write lambda function), result in manage it alot

B is best choice to use services: s3, dynamoDB, Amazon Cognito with maximum automatic build-in feature and least effort for operation

upvoted 1 times

🗨️ 👤 **Saudis** 6 months, 3 weeks ago

The difference between B and C the key word create user > cognito

upvoted 1 times

🗨️ 👤 **65703c1** 12 months ago

**Selected Answer: B**

B is the correct answer.

upvoted 1 times

🗨️ 👤 **badsati** 1 year, 1 month ago

**Selected Answer: B**

No question, Answer is B

upvoted 1 times

🗨️ 👤 **TheFivePips** 1 year, 2 months ago

**Selected Answer: B**

Definitely do not add all users manually, so that rules out C and D.

You wouldn't use DynamoDB to store the photos because DynamoDB limits the size of each item that you store in a table to 400 KB. So that rules out A

upvoted 2 times

🗨️ 👤 **leonardoliveros** 1 year, 6 months ago

**Selected Answer: B**

It's easier if you leverage all pros of Amazon Cognito you don't need creating a IAM user by employee

upvoted 1 times

🗨️ 👤 **ihta\_2031** 2 years, 1 month ago

**Selected Answer: B**

Cognito,

Item size in dynamodb is less than this scenario

upvoted 4 times

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

**Selected Answer: B**

B is the most valid solution.

A nearest, but invalid, because you cannot store object in Dynamo.

upvoted 3 times