



# DVA-C02<sup>Q&As</sup>

AWS Certified Developer - Associate

## Pass Amazon DVA-C02 Exam with 100% Guarantee

Free Download Real Questions & Answers **PDF** and **VCE** file from:

<https://www.pass4itsure.com/dva-c02.html>

100% Passing Guarantee  
100% Money Back Assurance

Following Questions and Answers are all new published by Amazon  
Official Exam Center

- ⚙️ **Instant Download** After Purchase
- ⚙️ **100% Money Back** Guarantee
- ⚙️ **365 Days** Free Update
- ⚙️ **800,000+** Satisfied Customers



**QUESTION 1**

A company is using Amazon RDS as the backend database for its application. After a recent marketing campaign, a surge of read requests to the database increased the latency of data retrieval from the database. The company has decided to implement a caching layer in front of the database. The cached content must be encrypted and must be highly available.

Which solution will meet these requirements?

- A. Amazon CloudFront
- B. Amazon ElastiCache for Memcached
- C. Amazon ElastiCache for Redis in cluster mode
- D. Amazon DynamoDB Accelerator (DAX)

Correct Answer: C

---

**QUESTION 2**

A developer is creating a publicly accessible enterprise website consisting of only static assets. The developer is hosting the website in Amazon S3 and serving the website to users through an Amazon CloudFront distribution. The users of this application must not be able to access the application content directly from an S3 bucket. All content must be served through the Amazon CloudFront distribution.

Which solution will meet these requirements?

- A. Create a new origin access control (OAC) in CloudFront. Configure the CloudFront distribution's origin to use the new OAC. Update the S3 bucket policy to allow CloudFront OAC with read and write access to access Amazon S3 as the origin.
- B. Update the S3 bucket settings. Enable the block all public access setting in Amazon S3. Configure the CloudFront distribution's with Amazon S3 as the origin. Update the S3 bucket policy to allow CloudFront write access.
- C. Update the S3 bucket's static website settings. Enable static website hosting and specifying index and error documents. Update the CloudFront origin to use the S3 bucket's website endpoint.
- D. Update the CloudFront distribution's origin to send a custom header. Update the S3 bucket policy with a condition by using the aws:RequestTag/tag-key key. Configure the tag-key as the custom header name, and the value being matched is the header's value.

Correct Answer: C

---

**QUESTION 3**

A developer has created a data collection application that uses Amazon API Gateway, AWS Lambda, and Amazon S3. The application's users periodically upload data files and wait for the validation status to be reflected on a processing dashboard. The validation process is complex and time-consuming for large files.

Some users are uploading dozens of large files and have to wait and refresh the processing dashboard to see if the files



have been validated. The developer must refactor the application to immediately update the validation result on the user's dashboard without reloading the full dashboard.

What is the MOST operationally efficient solution that meets these requirements?

- A. Integrate the client with an API Gateway WebSocket API. Save the user-uploaded files with the WebSocket connection ID. Push the validation status to the connection ID when the processing is complete to initiate an update of the user interface.
- B. Launch an Amazon EC2 micro instance, and set up a WebSocket server. Send the user-uploaded file and user detail to the EC2 instance after the user uploads the file. Use the WebSocket server to send updates to the user interface when the uploaded file is processed.
- C. Save the user's email address along with the user-uploaded file. When the validation process is complete, send an email notification through Amazon Simple Notification Service (Amazon SNS) to the user who uploaded the file.
- D. Save the user-uploaded file and user detail to Amazon DynamoDB. Use Amazon DynamoDB Streams with Amazon Simple Notification Service (Amazon SNS) push notifications to send updates to the browser to update the user interface.

Correct Answer: A

---

#### QUESTION 4

A developer is building a serverless application by using AWS Serverless Application Model (AWS SAM) on multiple AWS Lambda functions. When the application is deployed, the developer wants to shift 10% of the traffic to the new deployment of the application for the first 10 minutes after deployment. If there are no issues, all traffic must switch over to the new version.

Which change to the AWS SAM template will meet these requirements?

- A. Set the Deployment Preference Type to Canary10Percent10Minutes. Set the AutoPublishAlias property to the Lambda alias.
- B. Set the Deployment Preference Type to Linear10PercentEvery10Minutes. Set AutoPublishAlias property to the Lambda alias.
- C. Set the Deployment Preference Type to Canary10Percent10Minutes. Set the PreTraffic and PostTraffic properties to the Lambda alias.
- D. Set the Deployment Preference Type to Linear10PercentEvery10Minutes. Set PreTraffic and PostTraffic properties to the Lambda alias.

Correct Answer: C

---

#### QUESTION 5

A company runs a batch processing application by using AWS Lambda functions and Amazon API Gateway APIs with deployment stages for development, user acceptance testing and production. A development team needs to configure the APIs in the deployment stages to connect to third-party service endpoints.

Which solution will meet this requirement?



- A. Store the third-party service endpoints in Lambda layers that correspond to the stage
- B. Store the third-party service endpoints in API Gateway stage variables that correspond to the stage
- C. Encode the third-party service endpoints as query parameters in the API Gateway request URL.
- D. Store the third-party service endpoint for each environment in AWS AppConfig

Correct Answer: B

API Gateway stage variables are name-value pairs that can be defined as configuration attributes associated with a deployment stage of a REST API. They act like environment variables and can be used in the API setup and mapping templates. For example, the development team can define a stage variable named endpoint and assign it different values for each stage, such as dev.example.com for development, uat.example.com for user acceptance testing, and prod.example.com for production. Then, the team can use the stage variable value in the integration request URL, such as `http://$ { stageVariables.endpoint}/api`. This way, the team can use the same API setup with different endpoints at each stage by resetting the stage variable value. The other solutions are either not feasible or not cost-effective. Lambda layers are used to package and load dependencies for Lambda functions, not for storing endpoints. Encoding the endpoints as query parameters would expose them to the public and make the request URL unnecessarily long. Storing the endpoints in AWS AppConfig would incur additional costs and complexity, and would require additional logic to retrieve the values from the configuration store. References [Using Amazon API Gateway stage variables](#) [Setting up stage variables for a REST API deployment](#) [Setting stage variables using the Amazon API Gateway console](#)

[Latest DVA-C02 Dumps](#)

[DVA-C02 PDF Dumps](#)

[DVA-C02 Exam Questions](#)