**AWS Certified Solutions Architect Associate Practice Questions**

**Requirement**: Create & Share 10 CSAA practice questions. These are for the API Gateway, Lambda, S3, and VPC subject areas.

**Topic**

* Questions around API Gateway, Lambda, S3, and VPC subject areas.

**Delivery Timeline**: September 3, 2021

**Question Response Types**

There are two types of questions:

* Multiple Choice Single Response – **1** correct answer **3** incorrect responses (distractors).
* Multiple Choice Multiple Response – **2** or more correct answers out of **5** or more options.

**Important Note**

* Do write Question Number for quick identification. Q# 1, Q# 2 …. & so on.
* Please mention Domain (based on ML Specialty exam blueprint), Topic & Sub-Topic (If Applicable) with every question.
* Note that we’re expecting standard scenario based questions & NOT straight-forward definition kind of questions.
* The options should not have any obviously incorrect option. We need to word the options so that all of them should appear correct for the students, but a subtle point should mark the correct answer without any ambiguity. So, one answer should be the best choice without any doubt.
* The answer / explanation section should contain explanations on why the answer is correct and others are incorrect. It should also contain the relevant resource link (for details) preferably from AWS documentation.
  + Example
    - Option A is incorrect because..
    - Option B is CORRECT because…
    - Option C is incorrect because..
    - Option D is incorrect because..
* Try to balance the domains based on weightage % defined in the exam blueprint.
* Any AWS service or feature must be approximately 6 months old to figure out in Practice Tests. Put a note in the explanation for any latest service or feature that might be on the borderline of appearing in the real exam.
* **Plagiarism** in any form - Question or in Explanation will be **rejected.** Questions & Explanations should reflect your own professional experience & AWS skills. Author’s who indulge in plagiarism will be **blacklisted** & dropped from our author’s list.
* The ownership of the questions once approved & published on Whizlabs LMS platform, lies solely with Whizlabs Software Pvt. Ltd. You can’t share or publish it elsewhere in any circumstances.

**Sample Format of** **Questions**

----------------------------------------------------------------------------------------------------------------------------

**Question​ ​:​** #

**Main​ ​Topic​ ​:​** < >

**Sub​ ​Topic​ ​:​** [optional]

**Domain:** < >

**Question text**:

<Scenario based. Should be clear in terms of requirements. No ambiguity. No duplicate options. In case of multiple answers, at the end, you should include number of expected answers. e.g. (Select TWO) or (Select THREE) etc. For single answers this is NOT required>

1. Option A...
2. Option B...
3. Option C...
4. Option D...

**Answer:** A and C

**Explanation:**

**Option A is CORRECT because...**

**Option B is incorrect because...**

**Option C is CORRECT because...**

**Option D is incorrect because...**

[Insert the explanation in clear and lucid language here.]

**Diagram:** [Optional] [Insert the architectural or conceptual diagram here.]

**Reference:** [Insert the references here - which may include links to AWS Documentation, Blog, re:Invent video, Authority YouTube video].

----------------------------------------------------------------------------------------------------------------------------

--------------------------------------Question Section Starts-----------------------------------------------------

Question: 1

**Main​ ​Topic​ ​:​** Certified Solutions Architect Associate

**Sub​ ​Topic​ ​:​ Design a multi-tier architecture solution**

**Domain:** Design Resilient Architectures

**Question text**:

You are a solutions architect working for an online retailer. Your online web site uses REST API calls via API Gateway and Lambda from your Angular SPA front-end to interact with your DynamoDB data store. Your DynamoDB tables are used for customer preferences, account, and product information. When your web traffic spikes, some requests return a 429 error response. What might be the reason your requests are returning a 429 response? (Select TWO)

1. Your Lambda function has exceeded the concurrency limit.
2. DynamoDB concurrency limit has been exceeded
3. Your Angular service failed to connect to your API Gateway REST endpoint.
4. Your Angular service cannot handle the volume spike
5. Your API Gateway has exceeded the steady-state request rate and burst limits.

**Answers:** A, E

**Explanation:**

Option A is correct. When your traffic spikes, your Lambda function can exceed the limit set on the number of concurrent instances that can be run (burst concurrency limit in US: 3,000).

Option B is incorrect. When your table exceeds its provisioned throughput DynamoDB will return a 400 error to the requesting service, in this case API Gateway. This will not result in the propagation of a 429 error response (too many requests) back to the Angular SPA service.

Option C is incorrect. If your Angular service fails to connect to your API Gateway REST endpoint your code will not generate a 429 error response (too many requests).

Option D is incorrect. Since your Angular SPA code runs in the individual user’s web browser, this option makes no sense.

Option E is correct. When your API Gateway request volume reaches the steady-state request rate and bursting limit, API Gateway throttles your requests to protect your back-end services. When these requests are throttled, API Gateway returns a 429 error response (too many requests).

**Reference:**

Please see the Amazon API Gateway developer guide titled **Throttle API requests for better throughput** (<https://docs.aws.amazon.com/apigateway/latest/developerguide/api-gateway-request-throttling.html>), the Towards Data Science article titled **Full Stack Development Tutorial: Integrate AWS Lambda Serverless Service into Angular SPA**

(<https://towardsdatascience.com/full-stack-development-tutorial-integrate-aws-lambda-serverless-service-into-angular-spa-abb70bcf417f>), the Amazon API Gateway developer guide titled **Invoking a REST API in Amazon API Gateway** (<https://docs.aws.amazon.com/apigateway/latest/developerguide/how-to-call-api.html>), the AWS Lambda developer guide titled **Lambda function scaling** (<https://docs.aws.amazon.com/lambda/latest/dg/invocation-scaling.html>), and the Amazon DynamoDB developer guide titled **Error Handling with DynamoDB** (<https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/Programming.Errors.html>)

Question: 2

**Main​ ​Topic​ ​:​** Certified Solutions Architect Associate

**Sub​ ​Topic​ ​:​ Identify elastic and scalable compute solutions for a workload**

**Domain:** Design High-Performing Architectures

**Question text**:

You are a solutions architect working for a financial services firm. Your firm requires very low latency response time for requests via API Gateway and Lambda integration to your securities master database. The securities master database, housed in Aurora, contains data about all of the securities your firm trades. The data consists of the security ticker, the trading exchange, trading partner firm for the security, etc. As this securities data is relatively static, you can improve the performance of your API Gateway REST endpoint by using API Gateway caching. Your REST API calls for equity security request types and fixed income security request types to be cached separately. Which of the following options is the most efficient way to separate your cache responses via request type using API Gateway caching?

1. Custom header
2. URL paths
3. Request method type
4. Query string

**Answer:** D

**Explanation:**

Option A is incorrect. To separate requests for the security type, the most efficient approach is to use a query string with a type parameter. A custom header is used more for setting content type, authorization, host, date, etc.

Option B is incorrect. You could use a different URL path for each type of security, but that approach would over complicate your solution.

Option C is incorrect. Your request type for security information will invariably be a GET method request type. Therefore, using the request method type would not differentiate the security types.

Option D is correct. You can use your query string parameters as part of your cache key. This allows you to separately cache responses for equity requests from fixed income request responses.

**Reference:**

Please see the Amazon API Gateway developer guide titled **Enabling API caching to enhance responsiveness**

(<https://docs.aws.amazon.com/apigateway/latest/developerguide/api-gateway-caching.html>), and the Amazon API Gateway REST API Referencepagetitled **Making HTTP Requests to Amazon API Gateway**

(<https://docs.aws.amazon.com/apigateway/api-reference/making-http-requests/>)

Question: 3

**Main​ ​Topic​ ​:​** Certified Solutions Architect Associate

**Sub​ ​Topic​ ​:​ Design secure application tiers**

**Domain:** Design Secure Applications and Architectures

**Question text**:

You are a solutions architect working for a healthcare provider. Your company uses REST APIs to expose critical patient data to internal front-end systems used by doctors and nurses when involved in patient care. The data for your patient information is stored in Aurora.

How can you ensure that your patient data REST endpoint is only accessed by your authorized internal users? (Select TWO)

1. Run your Aurora DB cluster on an EC2 instance in a private subnet
2. Use a Gateway VPC Endpoint to make your REST endpoint private and only accessible from within your VPC
3. Use IAM resource policies to restrict access to your REST APIs by adding the aws:SourceVpce condition to the API Gateway resource policy
4. Use an Interface VPC Endpoint to make your REST endpoint private and only accessible from within your VPC and through your VPCe
5. Use IAM resource policies to restrict access to your REST APIs by adding the aws:SourceArn condition to the API Gateway resource policy

**Answers:** C, D

**Explanation:**

Option A is incorrect. Controlling access to your back-end database running on Aurora will not restrict access to your API Gateway REST endpoint. Access to your API Gateway REST endpoint must be controlled at the API Gateway and VPC level.

Option B is incorrect. The Gateway VPC Endpoint is only used for the S3 and DynamoDB services.

Option C is correct. You can make your REST APIs private by using the aws:SourceVpce condition in your API Gateway resource policy to restrict access to only your VPC Endpoint.

Option D is correct. Use a VPC Interface Endpoint to restrict access to your REST APIs to traffic that arrives via the VPC Endpoint.

Option E is incorrect. The aws:SourceArn condition key is not used to restrict access to traffic that arrives via the VPC Endpoint.

**Reference:**

Please see the Amazon API Gateway developer guide titled **Creating a private API in Amazon API Gateway**

(<https://docs.aws.amazon.com/apigateway/latest/developerguide/apigateway-private-apis.html>), the Amazon API Gateway developer guidetitled **Example: Allow private API traffic based on source VPC or VPC endpoint** (<https://docs.aws.amazon.com/apigateway/latest/developerguide/apigateway-resource-policies-examples.html#apigateway-resource-policies-source-vpc-example>), the Amazon Aurora user guide titled **Amazon Aurora security** (<https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/Aurora.Overview.Security.html>), the Amazon Aurora user guide titled **Amazon Aurora DB clusters** (<https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/Aurora.Overview.html>), the Amazon Aurora user guide titled **Aurora DB instance classes** (<https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/Concepts.DBInstanceClass.html>), the Amazon API Gateway developer guidetitled **AWS condition keys that can be used in API Gateway resource policies** (<https://docs.aws.amazon.com/apigateway/latest/developerguide/apigateway-resource-policies-aws-condition-keys.html>), and the Amazon Virtual Private Cloud AWS PrivateLink page titled **VPC endpoints** (<https://docs.aws.amazon.com/vpc/latest/privatelink/vpc-endpoints.html>)

Question: 4

**Main​ ​Topic​ ​:​** Certified Solutions Architect Associate

**Sub​ ​Topic​ ​:​ Design highly available and/or fault-tolerant architectures**

**Domain:** Design Resilient Architectures

**Question text**:

You are a solutions architect working for a data analytics company that delivers analytics data to politicians that need the data to manage their campaigns. Political campaigns use your company’s analytics data to decide on where to spend their campaign money to get the best results for the efforts. Your political campaign users access your analytics data through an Angular SPA via API Gateway REST endpoints. You need to manage the access and use of your analytics platform so that you can prove that you keep your individual campaign data separate. Specifically, you need to produce logs of all user requests and responses to those requests, including request payloads, response payloads, and error traces. Which type of AWS logging service should you use to achieve your goals?

1. Use CloudWatch access logging
2. Use CloudWatch execution logging
3. Use CloudTrail logging
4. Use CloudTrail execution logging

**Answer:** B

**Explanation:**

Option A is incorrect. CloudWatch access logging captures which resource accessed an API and the method used to access the API. It is not used for execution traces, such as capturing request and response payloads.

Option B is correct. CloudWatch execution logging allows you to capture user request and response payloads as well as error traces.

Option C is incorrect. CloudTrail captures actions by users, roles, and AWS services. CloudTrail records all AWS account activity. CloudTrail does not capture error traces.

Option D is correct. CloudTrail does not have a feature called execution logging.

**Reference:**

Please see the Amazon API Gateway developer guide titled **Setting up CloudWatch logging for a REST API in API Gateway**

(<https://docs.aws.amazon.com/apigateway/latest/developerguide/set-up-logging.html>), and the AWS CloudTrail user guide titled **How CloudTrail works**

(<https://docs.aws.amazon.com/awscloudtrail/latest/userguide/how-cloudtrail-works.html>)

Question: 5

**Main​ ​Topic​ ​:​** Certified Solutions Architect Associate

**Sub​ ​Topic​ ​:​ Select appropriate data security options**

**Domain:** Design Secure Applications and Architectures

**Question text**:

You are a solutions architect working for a social media company that provides a place for civil discussion of political and news related events. Due to the ever changing regulatory requirements and restrictions placed on social media apps that provide these services, you need to build your app in a very highly flexible environment that you can change instantly without updating code. You have chosen to build the REST API endpoints used by your social media app user interface code using Lambda. How can you securely configure your Lambda functions so that you can achieve the flexibility required by your app? (Select TWO)

1. Pass environment variables to your Lambda function via the request header sent to your API Gateway methods.
2. Configure your Lambda functions to use key configuration.
3. Use encryption helpers
4. Use Lambda layers
5. Use Lambda aliases

**Answers:** B, C

**Explanation:**

Option A is incorrect. Sending environment variables to your Lambda function as request parameters would expose the environment variables as plain text. This is not a secure approach.

Option B is correct. Lambda key configuration allows you to have your Lambda functions use an encryption key. You create the key in AWS KMS. The key is used to encrypt your environment variables that you can use to change your function without deploying any code.

Option C is correct. Encryption helpers make your lambda function more secure by allowing you to encrypt your environment variables before they are sent to Lambda.

Option D is correct. Lambda layers are used to package common code such as libraries, configuration files, or custom runtime images. Layers will not give you the same flexibility as environment variables for use in managing change without deploying any code.

Option E is incorrect. Lambda aliases are used to refer to a specific version of your Lambda function. You could switch between many versions of your Lambda function, but you would have to deploy new code to create a different version of your Lambda function.

**Reference:**

Please see the AWS Lambda developer guide titled **Data protection in AWS Lambda**

(<https://docs.aws.amazon.com/lambda/latest/dg/security-dataprotection.html>), the AWS Lambda developer guide titled **Lambda concepts**

(<https://docs.aws.amazon.com/lambda/latest/dg/gettingstarted-concepts.html#gettingstarted-concepts-layer>), the AWS Lambda developer guide titled **Lambda function aliases** (<https://docs.aws.amazon.com/lambda/latest/dg/configuration-aliases.html>), and the AWS Lambda developer guide titled **Using AWS Lambda environment variables** (<https://docs.aws.amazon.com/lambda/latest/dg/configuration-envvars.html>)

Question: 6

**Main​ ​Topic​ ​:​** Certified Solutions Architect Associate

**Sub​ ​Topic​ ​:​ Select appropriate data security options**

**Domain:** Design Secure Applications and Architectures

**Question text**:

You are a solutions architect working for a media company that produces stock images and videos for sale to users via a mobile app and web site. Your app and web site allow users to gain access only to stock content they have purchased. Your content is stored in S3 buckets. You need to restrict access to multiple files that your users have purchased. Also, due to the nature of the stock content (purchasable by multiple users) you don’t want to change the URLs of each stock item.

Which access control option best fits your scenario?

1. Use CloudFront signed URLs
2. Use S3 Presigned URLs
3. UseCloudFront Signed Cookies
4. Use S3 Signed Cookies

**Answer:** C

**Explanation:**

Option A is incorrect. CloudFront signed URLs allow you to restrict access to individual files. Signed URLs require you to change your content URLs for each customer access.

Option B is incorrect. S3 Presigned URLs URLs require you to change your content URLs. The presigned URL expires after its defined expiration date.

Option C is correct. CloudFront Signed Cookies allow you to control access to multiple content files and you don’t have to change your URL for each customer access.

Option D is incorrect. There is no S3 Signed Cookies feature..

**Reference:**

Please see the Amazon CloudFront developer guide titled **Using signed cookies**

(<https://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/private-content-signed-cookies.html>), the Amazon Simple Storage Service user guide titled **Sharing an object with a presigned URL**

(<https://docs.aws.amazon.com/AmazonS3/latest/userguide/ShareObjectPreSignedURL.html>), the Amazon Simple Storage Service user guide titled **Using presigned URLs**

(<https://docs.aws.amazon.com/AmazonS3/latest/userguide/using-presigned-url.html#PresignedUrlUploadObject-LimitCapabilities>), and the Amazon CloudFront developer guide titled **Choosing between signed URLs and signed cookies**

(<https://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/private-content-choosing-signed-urls-cookies.html>)