



Free Questions for *DVA-C02* by *dumpssheet*

Shared by *Hutchinson* on *22-07-2024*

For More Free Questions and Preparation Resources

Check the Links on Last Page

Question 1

Question Type: MultipleChoice

A developer is creating an Amazon DynamoDB table by using the AWS CLI. The DynamoDB table must use server-side encryption with an AWS owned encryption key.

How should the developer create the DynamoDB table to meet these requirements?

Options:

- A-** Create an AWS Key Management Service (AWS KMS) customer managed key. Provide the key's Amazon Resource Name (ARN) in the `KMSMasterKeyId` parameter during creation of the DynamoDB table.
- B-** Create an AWS Key Management Service (AWS KMS) AWS managed key. Provide the key's Amazon Resource Name (ARN) in the `KMSMasterKeyId` parameter during creation of the DynamoDB table.
- C-** Create an AWS owned key. Provide the key's Amazon Resource Name (ARN) in the `KMSMasterKeyId` parameter during creation of the DynamoDB table.
- D-** Create the DynamoDB table with the default encryption options.

Answer:

D

Explanation:

Default SSE in DynamoDB: DynamoDB tables are encrypted at rest by default using an AWS owned key (SSE-S3).

No Additional Action Needed: Creating a table without explicitly specifying a KMS key will use this default encryption.

DynamoDB Server-Side Encryption: <https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/Encryption>

Question 2

Question Type: MultipleChoice

An Amazon Simple Queue Service (Amazon SQS) queue serves as an event source for an AWS Lambda function. In the SQS queue, each item corresponds to a video file that the Lambda function must convert to a smaller resolution. The Lambda function is timing out on longer video files, but the Lambda function's timeout is already configured to its maximum value.

What should a developer do to avoid the timeouts without additional code changes'?

Options:

- A- Increase the memory configuration of the Lambda function
- B- Increase the visibility timeout on the SQS queue
- C- Increase the instance size of the host that runs the Lambda function.
- D- Use multi-threading for the conversion.

Answer:

B

Explanation:

Visibility Timeout: When an SQS message is processed by a consumer (here, the Lambda function), it's temporarily hidden from other consumers. Visibility timeout controls this duration.

How It Helps:

Increase the visibility timeout beyond the maximum processing time your Lambda might typically take for long videos.

This prevents the message from reappearing in the queue while Lambda is still working, avoiding premature timeouts.

[SQS Visibility Timeout:https://docs.aws.amazon.com/AWSSimpleQueueService/latest/SQSDeveloperGuide/sqs-visibility-timeout.html](https://docs.aws.amazon.com/AWSSimpleQueueService/latest/SQSDeveloperGuide/sqs-visibility-timeout.html)

Question 3

Question Type: MultipleChoice

A developer is using AWS Step Functions to automate a workflow. The workflow defines each step as an AWS Lambda function task. The developer notices that runs of the Step Functions state machine fail in the GetResource task with either an UlegalArgumentException error or a TooManyRequestsException error.

The developer wants the state machine to stop running when the state machine encounters an UlegalArgumentException error. The state machine needs to retry the GetResource task one additional time after 10 seconds if the state machine encounters a TooManyRequestsException error. If the second attempt fails, the developer wants the state machine to stop running.

How can the developer implement the Lambda retry functionality without adding unnecessary complexity to the state machine'?

Options:

- A-** Add a Delay task after the GetResource task. Add a catcher to the GetResource task. Configure the catcher with an error type of TooManyRequestsException. Configure the next step to be the Delay task. Configure the Delay task to wait for an interval of 10 seconds. Configure the next step to be the GetResource task.
- B-** Add a catcher to the GetResource task. Configure the catcher with an error type of TooManyRequestsException, an interval of 10 seconds, and a maximum attempts value of 1. Configure the next step to be the GetResource task.
- C-** Add a retrier to the GetResource task. Configure the retrier with an error type of TooManyRequestsException, an interval of 10 seconds, and a maximum attempts value of 1.

D- Duplicate the GetResource task Rename the new GetResource task to TryAgain Add a catcher to the original GetResource task Configure the catcher with an error type of TooManyRequestsException. Configure the next step to be TryAgain.

Answer:

C

Explanation:

Step Functions Retriers: Retriers provide a built-in way to gracefully handle transient errors within State Machines. Here's how to use them:

Directly attach a retrier to the problematic 'GetResource' task.

Configure the retrier:

ErrorEquals:Set this to ['TooManyRequestsException'] to target the specific error.

IntervalSeconds:Set to 10 for the desired retry delay.

MaxAttempts:Set to 1, as you want only one retry attempt.

Error Handling:

Upon 'TooManyRequestsException', the retrier triggers the task again after 10 seconds.

On a second failure, Step Functions moves to the next state or fails the workflow, as per your design.

'IllegalArgumentException' causes error propagation as intended.

Error Handling in Step Functions:<https://docs.aws.amazon.com/step-functions/latest/dg/concepts-error-handling.html>

Question 4

Question Type: MultipleChoice

A company is planning to use AWS CodeDeploy to deploy an application to Amazon Elastic Container Service (Amazon ECS) During the deployment of a new version of the application, the company initially must expose only 10% of live traffic to the new version of the deployed application. Then, after 15 minutes elapse, the company must route all the remaining live traffic to the new version of the deployed application.

Which CodeDeploy predefined configuration will meet these requirements?

Options:

- A- CodeDeployDefault ECSCanary10Percent15Minutes
- B- CodeDeployDefault LambdaCanary10Percent5Minutes
- C- CodeDeployDefault LambdaCanary10Percent15Minutes

D- CodeDeployDefault ECSELinear10PercentEvery1 Minutes

Answer:

A

Explanation:

CodeDeploy Predefined Configurations:CodeDeploy offers built-in deployment configurations for common scenarios.

Canary Deployment:Canary deployments gradually shift traffic to a new version, ideal for controlled rollouts like this requirement.

CodeDeployDefault.ECSCanary10Percent15Minutes:This configuration matches the company's requirements, shifting 10% of traffic initially and then completing the rollout after 15 minutes.

[AWS CodeDeploy Deployment Configurations:https://docs.aws.amazon.com/codedeploy/latest/userguide/deployment-configurations-create.html](https://docs.aws.amazon.com/codedeploy/latest/userguide/deployment-configurations-create.html)

Question 5

Question Type: MultipleChoice

A developer is creating a simple proof-of-concept demo by using AWS CloudFormation and AWS Lambda functions. The demo will use a CloudFormation template to deploy an existing Lambda function. The Lambda function uses deployment packages and dependencies stored in Amazon S3. The developer defined an AWS Lambda Function resource in a CloudFormation template. The developer needs to add the S3 bucket to the CloudFormation template.

What should the developer do to meet these requirements with the LEAST development effort?

Options:

- A- Add the function code in the CloudFormation template inline as the code property
- B- Add the function code in the CloudFormation template as the ZipFile property.
- C- Find the S3 key for the Lambda function. Add the S3 key as the ZipFile property in the CloudFormation template.
- D- Add the relevant key and bucket to the S3Bucket and S3Key properties in the CloudFormation template

Answer:

D

Explanation:

S3Bucket and S3Key: These properties in a CloudFormation `AWS::Lambda::Function` resource specify the location of the function's code in S3.

Least Development Effort: This solution minimizes code changes, relying on CloudFormation to reference the existing S3 deployment package.

[AWS::Lambda::Function Resource](https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/aws-resource-lambda-function.html)<https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/aws-resource-lambda-function.html>

Question 6

Question Type: MultipleChoice

A developer is working on an ecommerce platform that communicates with several third-party payment processing APIs. The third-party payment services do not provide a test environment.

The developer needs to validate the ecommerce platform's integration with the third-party payment processing APIs. The developer must test the API integration code without invoking the third-party payment processing APIs.

Which solution will meet these requirements'?

Options:

A- Set up an Amazon API Gateway REST API with a gateway response configured for status code 200. Add response templates that contain sample responses captured from the real third-party API.

- B-** Set up an AWS AppSync GraphQL API with a data source configured for each third-party API. Specify an integration type of Mock. Configure integration responses by using sample responses captured from the real third-party API.
- C-** Create an AWS Lambda function for each third-party API. Embed responses captured from the real third-party API. Configure Amazon Route 53 Resolver with an inbound endpoint for each Lambda function's Amazon Resource Name (ARN).
- D-** Set up an Amazon API Gateway REST API for each third-party API. Specify an integration request type of Mock. Configure integration responses by using sample responses captured from the real third-party API.

Answer:

D

Explanation:

Mocking API Responses: API Gateway's Mock integration type enables simulating API behavior without invoking backend services.

Testing with Sample Data: Using captured responses from the real third-party API ensures realistic testing of the integration code.

Focus on Integration Logic: This solution allows the developer to isolate and test the application's interaction with the payment APIs, even without a test environment from the third-party providers.

Amazon API Gateway Mock Integrations: <https://docs.aws.amazon.com/apigateway/latest/developerguide/how-to-mock-integration.html>

To Get Premium Files for DVA-C02 Visit

<https://www.p2pexams.com/products/dva-c02>

For More Free Questions Visit

<https://www.p2pexams.com/amazon/pdf/dva-c02>

