Exam

DVA-C02

AWS Certified Developer - Associate (DVA-C02)

Free demo	80 sample questions
Full exam	595 questions with answers and explanations
Contents	Questions with answers and explanations

Exam domains

Domain	Exam weight	Questions
D1 Development with AWS Services		54
D2 Security		15
D3 Deployment		7
D4 Troubleshooting and Optimization		4

This is a free demo from Dumpsgate: 80 of the exam's 595 questions. **Get the full DVA-C02 exam** with every question, answer and explanation.

Question 1

Single choice · Domain: D1 Development with AWS Services › Publish custom metrics to Amazon CloudWatch and configure threshold alarms

A company maintains an application responsible for processing several thousand external callbacks each day. The company's system administrators want to know how many callbacks are being received on a rolling basis, and they want this data available for 10 days.

The company also wants the ability to issue automated alerts if the number of callbacks exceeds the defined thresholds.

What is the MOST cost-effective way to address the need to track and alert on these statistics?

A. Push callback data to an Amazon RDS database that can be queried to show historical data and to alert on exceeded thresholds.

B. Push callback data to AWS X-Ray and use AWS Lambda to query, display, and alert on exceeded thresholds.

C. Push callback data to Amazon Kinesis Data Streams and invoke an AWS Lambda function that stores data in Amazon DynamoDB and sends the required alerts.

D. Push callback data to Amazon CloudWatch as a custom metric and use the CloudWatch alerting mechanisms to alert system administrators.

Answer: D. Push callback data to Amazon CloudWatch as a custom metric and use the CloudWatch alerting mechanisms to alert system administrators.

Explanation

This question tests the ability to select a serverless AWS monitoring solution that tracks callback counts over time and automatically alerts when a custom metric exceeds a threshold.

✗ **A** Push callback data to an Amazon RDS database that can be queried to show historical data and to alert on exceeded thresholds.
Incorrect. Amazon RDS is unnecessarily expensive and operationally complex for storing a simple callback count and evaluating thresholds. CloudWatch custom metrics and alarms provide the required monitoring and alerting capabilities without maintaining a database.

https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/publishingMetrics.html

✗ **B** Push callback data to AWS X-Ray and use AWS Lambda to query, display, and alert on exceeded thresholds.
Incorrect. AWS X-Ray is designed for tracing requests and analyzing application performance, not for maintaining rolling business statistics or threshold alarms. CloudWatch custom metrics are the appropriate service for this requirement.

https://docs.aws.amazon.com/xray/latest/devguide/aws-xray.html

✗ **C** Push callback data to Amazon Kinesis Data Streams and invoke an AWS Lambda function that stores data in Amazon DynamoDB and sends the required alerts.
Incorrect. This solution introduces Kinesis Data Streams, Lambda, and DynamoDB, creating unnecessary services and costs for a straightforward metric-counting and alerting requirement. CloudWatch can receive custom metrics and trigger alarms directly.

https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/publishingMetrics.html

✓ **D** Push callback data to Amazon CloudWatch as a custom metric and use the CloudWatch alerting mechanisms to alert system administrators.
Correct. CloudWatch custom metrics can record the number of callbacks, provide historical metric data, and support CloudWatch alarms that notify administrators when thresholds are exceeded. This is the

Free demo · full exam: 595 questions

most cost-effective and simplest solution because it avoids additional storage, streaming, and processing services. and.

https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/publishingMetrics.html
https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/AlarmThatSendsEmail.html

---

Question 2

Single choice · Domain: D1 Development with AWS Services › Select an appropriate caching strategy to improve read-heavy application performance

A company has a three-tier application that is deployed in Amazon ECS. The application is using an Amazon RDS for MySQL DB instance. The application performs more database reads than writes.

During times of peak usage, the application's performance degrades. When this performance degradation occurs, the DB instance's ReadLatency metric in Amazon CloudWatch increases suddenly.

How should a developer modify the application to improve performance?

A. Use Amazon ElastiCache to cache query results.
B. Scale the ECS cluster to contain more ECS instances.
C. Add read capacity units (RCUs) to the DB instance.
D. Modify the ECS task definition to increase the task memory.

Answer: A. Use Amazon ElastiCache to cache query results.

Explanation

This question tests the ability to identify a database read bottleneck and use Amazon ElastiCache to cache frequently requested query results, reducing read pressure on an Amazon RDS for MySQL DB instance.

✓ **A** Use Amazon ElastiCache to cache query results.
Correct. Because the workload performs more reads than writes and RDS ReadLatency increases during peak usage, caching frequently accessed query results with Amazon ElastiCache can reduce database reads and improve application performance.

https://docs.aws.amazon.com/AmazonElastiCache/latest/mem-ug/WhatIs.html

✗ **B** Scale the ECS cluster to contain more ECS instances.
Incorrect. Adding ECS instances increases application compute capacity but does not directly reduce the read workload or latency of the RDS database, which is the indicated bottleneck.

https://docs.aws.amazon.com/AmazonECS/latest/developerguide/cluster-auto-scaling.html

✗ **C** Add read capacity units (RCUs) to the DB instance.
Incorrect. Read capacity units are a DynamoDB capacity concept, not a scaling option for an Amazon RDS for MySQL DB instance. RDS performance is managed through options such as instance sizing, read replicas, and caching.

https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/CHAP_ReadReplicas.html

✗ **D** Modify the ECS task definition to increase the task memory.
Incorrect. Increasing ECS task memory may help if the application is constrained by container memory, but it does not address the sudden increase in the RDS ReadLatency metric caused by database read pressure.

https://docs.aws.amazon.com/AmazonECS/latest/developerguide/

Free demo · full exam: 595 questions

task_definition_parameters.html#task_definition_memory

Question 3

Single choice · Domain: D1 Development with AWS Services › Optimize AWS Lambda function performance by configuring memory allocation

A company has a monolithic desktop-based application that processes images. A developer is converting the application into an AWS Lambda function by using Python. Currently, the desktop application runs every 5 minutes to process the latest image from an Amazon S3 bucket. The desktop application completes the image processing task within 1 minute.

During testing on AWS, the developer notices that the Lambda function runs at the specified 5-minute interval. However, the Lambda function takes more than 2 minutes to complete the image processing task. The developer needs a solution that will improve the Lambda function's performance.

Which solution will meet this requirement?

A. Update the instance type of the Lambda function to a compute optimized instance with at least eight virtual CPUs (vCPUs).

B. Update the configuration of the Lambda function to use the latest Python runtime.

C. Increase the memory that is allocated to the Lambda function.

D. Configure reserved concurrency on the Lambda function.

Answer: C. Increase the memory that is allocated to the Lambda function.

Explanation

This question tests the ability to identify the Lambda configuration that improves execution performance for a CPU-intensive image-processing workload. Increasing Lambda memory also proportionally increases the allocated CPU capacity, which can reduce execution time.

✗ **A** Update the instance type of the Lambda function to a compute optimized instance with at least eight virtual CPUs (vCPUs).
Incorrect. AWS Lambda does not use selectable EC2 instance types or allow developers to configure a specific number of vCPUs directly. Lambda allocates CPU power in proportion to the configured memory.

https://docs.aws.amazon.com/lambda/latest/dg/configuration-memory.html

✗ **B** Update the configuration of the Lambda function to use the latest Python runtime.
Incorrect. Changing to the latest supported Python runtime does not directly provide the proportional CPU increase needed to improve the processing time of this function. Lambda performance is primarily improved here by increasing the function's memory allocation.

https://docs.aws.amazon.com/lambda/latest/dg/lambda-runtimes.html

✓ **C** Increase the memory that is allocated to the Lambda function.
Correct. Lambda allocates CPU power in proportion to the configured memory. Increasing memory therefore provides more CPU capacity and can improve the performance of CPU-intensive image processing.

https://docs.aws.amazon.com/lambda/latest/dg/configuration-memory.html

✗ **D** Configure reserved concurrency on the Lambda function.
Incorrect. Reserved concurrency limits or guarantees the number of concurrent executions; it does not

Free demo · full exam: 595 questions

increase the CPU or memory available to an individual invocation and therefore does not reduce the execution time of the image-processing task.

https://docs.aws.amazon.com/lambda/latest/dg/configuration-concurrency.html

Question 4

Single choice · Domain: D1 Development with AWS Services › Design an event-driven fanout workflow that invokes Lambda functions across multiple AWS accounts using Amazon SNS

A developer works in an environment with multiple AWS accounts that have AWS Lambda functions processing the same 100 KB payloads. The developer wants to centralize the point of origin of the payloads to one account and have all the Lambda functions be invoked whenever the initiating event occurs in the parent account.

How can the developer design the workflow in the MOST efficient way, so all the multi-account Lambda functions get invoked when the event occurs?

A. Create a Lambda function in the parent account and use cross-account IAM roles with the AWS STS AssumeRole API call to make AWS Lambda invoke the API call to invoke all the cross-account Lambda functions.

B. Subscribe all the multi-account Lambda functions to an Amazon SNS topic and make an SNS Publish API call with the payload to the SNS topic.

C. Set up an Amazon SQS queue with the queue policy permitting the ReceiveMessage action for multi-account Lambda functions. Then send the payload to the SQS queue using the sqs:SendMessage permission and poll the queue using multi-account Lambda functions.

D. Use a worker on an Amazon EC2 instance to poll for the payload event. Invoke all Lambda functions using the Lambda Invoke API after using cross-account IAM roles with the AWS STS AssumeRole API call.

Answer: B. Subscribe all the multi-account Lambda functions to an Amazon SNS topic and make an SNS Publish API call with the payload to the SNS topic.

Explanation

This question tests the ability to select an efficient AWS service integration for distributing the same event payload to multiple Lambda functions in different accounts. Amazon SNS provides publish-subscribe fanout, allowing one publish operation to invoke all subscribed Lambda functions.

✗ **A** Create a Lambda function in the parent account and use cross-account IAM roles with the AWS STS AssumeRole API call to make AWS Lambda invoke the API call to invoke all the cross-account Lambda functions.
Incorrect. This requires custom orchestration, repeated cross-account role assumption, and separate Lambda Invoke API calls, which is less efficient than using a managed fanout service.

https://docs.aws.amazon.com/STS/latest/APIReference/API_AssumeRole.html

✓ **B** Subscribe all the multi-account Lambda functions to an Amazon SNS topic and make an SNS Publish API call with the payload to the SNS topic.
Correct. Amazon SNS supports publish-subscribe fanout, so the initiating account can publish the 100 KB payload once and SNS can invoke all subscribed Lambda functions. Cross-account access requires the appropriate SNS topic and Lambda resource-based permissions.

Free demo · full exam: 595 questions

https://docs.aws.amazon.com/sns/latest/dq/sns-lambda.html

✗ **C** Set up an Amazon SQS queue with the queue policy permitting the ReceiveMessage action for multi-account Lambda functions. Then send the payload to the SQS queue using the sqs:SendMessage permission and poll the queue using multi-account Lambda functions.
Incorrect. An SQS queue is primarily a point-to-point message buffer. Multiple Lambda consumers would not each reliably receive the same message from one queue because messages are distributed among consumers. Separate queues and additional fanout logic would be needed.

https://docs.aws.amazon.com/lambda/latest/dq/with-sqs.html

✗ **D** Use a worker on an Amazon EC2 instance to poll for the payload event. Invoke all Lambda functions using the Lambda Invoke API after using cross-account IAM roles with the AWS STS AssumeRole API call.
Incorrect. An EC2 worker introduces unnecessary infrastructure, polling, operational overhead, and repeated cross-account invocation logic. Amazon SNS provides a managed and more efficient fanout mechanism.

https://docs.aws.amazon.com/sns/latest/dq/sns-lambda.html

Question 5

Single choice · Domain: D1 Development with AWS Services › Test AWS Lambda functions locally with AWS SAM CLI using simulated Amazon S3 events

A developer is implementing an AWS Lambda function that will be invoked when an object is uploaded to Amazon S3. The developer wants to test the Lambda function in a local development machine before publishing the function to a production AWS account.

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

A. Upload an object to Amazon S3 by using the aws s3api put-object CLI command. Wait for the local Lambda invocation from the S3 event.

B. Create a sample JSON text file for a put object S3 event. Invoke the Lambda function locally. Use the aws lambda invoke CLI command with the JSON file and Lambda function name as arguments.

C. Use the sam local start-lambda CLI command to start Lambda. Use the sam local generate-event s3 put CLI command to create the Lambda test JSON file. Use the sam local invoke CLI command with the JSON file as the argument to invoke the Lambda function.

D. Create a JSON string for the put object S3 event. In the AWS Management Console, use the JSON string to create a test event for the local Lambda function. Perform the test.

Answer: C. Use the sam local start-lambda CLI command to start Lambda. Use the sam local generate-event s3 put CLI command to create the Lambda test JSON file. Use the sam local invoke CLI command with the JSON file as the argument to invoke the Lambda function.

Explanation

This question tests the ability to generate an Amazon S3 event and invoke a Lambda function locally before deployment. AWS SAM CLI provides local emulation and event-generation commands for this purpose.

✗ **A** Upload an object to Amazon S3 by using the aws s3api put-object CLI command. Wait for the local Lambda invocation from the S3 event.
Incorrect. An Amazon S3 event invokes a Lambda function configured in AWS; uploading to S3 does not automatically invoke a Lambda function running on a developer's local machine.

Free demo · full exam: 595 questions

https://docs.aws.amazon.com/lambda/latest/dg/with-s3.html

✗ **B** Create a sample JSON text file for a put object S3 event. Invoke the Lambda function locally. Use the aws lambda invoke CLI command with the JSON file and Lambda function name as arguments.
Incorrect. The aws lambda invoke command invokes a Lambda function in AWS and does not provide local Lambda emulation. AWS SAM CLI is intended for invoking Lambda functions locally with test events.

https://docs.aws.amazon.com/serverless-application-model/latest/developerguide/using-sam-cli-local-invoke.html

✓ **C** Use the sam local start-lambda CLI command to start Lambda. Use the sam local generate-event s3 put CLI command to create the Lambda test JSON file. Use the sam local invoke CLI command with the JSON file as the argument to invoke the Lambda function.
Correct. AWS SAM CLI can emulate Lambda locally, generate a sample Amazon S3 put event, and invoke the function with that event without deploying it to AWS. This provides the required local testing with the least operational overhead. and.

https://docs.aws.amazon.com/serverless-application-model/latest/developerguide/using-sam-cli-local-generate-event.html
https://docs.aws.amazon.com/serverless-application-model/latest/developerguide/using-sam-cli-local-invoke.html

✗ **D** Create a JSON string for the put object S3 event. In the AWS Management Console, use the JSON string to create a test event for the local Lambda function. Perform the test.
Incorrect. The AWS Lambda console creates test events for Lambda functions deployed in AWS; it cannot create or invoke a function running on a local development machine.

https://docs.aws.amazon.com/lambda/latest/dg/testing-functions.html

Dumpsgate

Question 6

Single choice · Domain: D1 Development with AWS Services › Configure Amazon S3 lifecycle rules for noncurrent object versions and select an appropriate archival storage class

A company runs an analytics application that creates daily reports. The company stores the reports in an Amazon S3 bucket that has versioning enabled. The application stores several drafts of each report every day as S3 object versions before submitting a final report for the day.

The company needs a solution to delete draft reports that are older than 1 year. The company accesses draft reports frequently for the first month. After 1 month, the company accesses draft reports rarely during the next 11 months. Draft reports must be accessible within minutes.

Which solution will meet these requirements MOST cost-effectively?

A. Add an expiration action for object versions that have been noncurrent for over 30 days. Add a noncurrent version transition action that uses a storage class of GLACIER_IR for object versions that have been noncurrent for over a year.

B. Add an expiration action for versions that have been noncurrent for over 30 days. Add a noncurrent version transition action that uses a storage class of DEEP_ARCHIVE for objects that have been noncurrent for over a year.

C. Add a noncurrent version transition action that uses a storage class of GLACIER_IR for objects that have been noncurrent for over 30 days. Add an expiration action for objects that have been noncurrent for more than 1 year.

D. Add a noncurrent version transition action that uses a storage class of DEEP_ARCHIVE for objects that have been noncurrent for over 30 days. Add an expiration action for objects that have been noncurrent for more than 1 year.

Answer: C. Add a noncurrent version transition action that uses a storage class of GLACIER_IR for objects that have been noncurrent for over 30 days. Add an expiration action for objects that have been noncurrent for more than 1 year.

Explanation

This question tests the ability to configure lifecycle transitions and expiration actions for versioned S3 objects based on noncurrent age. It also evaluates selection of S3 Glacier Instant Retrieval to provide access within minutes while reducing storage costs.

✗ **A** Add an expiration action for object versions that have been noncurrent for over 30 days. Add a noncurrent version transition action that uses a storage class of GLACIER_IR for object versions that have been noncurrent for over a year.
Incorrect. This expires the draft versions after 30 days, but the requirements are to retain them for 1 year. The transition to S3 Glacier Instant Retrieval after 1 year also occurs too late for the 11-month rarely accessed period.

https://docs.aws.amazon.com/AmazonS3/latest/userguide/intro-lifecycle-rules.html

✗ **B** Add an expiration action for versions that have been noncurrent for over 30 days. Add a noncurrent version transition action that uses a storage class of DEEP_ARCHIVE for objects that have been noncurrent for over a year.
Incorrect. Expiring the versions after 30 days does not satisfy the requirement to delete drafts older than 1 year. In addition, S3 Glacier Deep Archive retrieval can take hours, so it does not provide access within minutes.

https://docs.aws.amazon.com/AmazonS3/latest/storage-class-intro.html

✓ **C** Add a noncurrent version transition action that uses a storage class of GLACIER_IR for objects that have been noncurrent for over 30 days. Add an expiration action for objects that have been noncurrent for more than 1 year.
Correct. The rule keeps drafts in their original storage class during the first month, transitions noncurrent versions to S3 Glacier Instant Retrieval after 30 days for lower-cost storage with millisecond access, and expires them after 1 year.

https://docs.aws.amazon.com/AmazonS3/latest/userguide/lifecycle-configuration-examples.html

✗ **D** Add a noncurrent version transition action that uses a storage class of DEEP_ARCHIVE for objects that have been noncurrent for over 30 days. Add an expiration action for objects that have been noncurrent for more than 1 year.
Incorrect. Although the one-year expiration is appropriate, S3 Glacier Deep Archive is intended for long-term archival and retrieval can take hours. It does not meet the requirement that draft reports be accessible within minutes.

https://docs.aws.amazon.com/AmazonS3/latest/storage-class-intro.html

Question 7

Single choice · Domain: D1 Development with AWS Services › Configure Amazon SNS subscription filter policies to route selected messages to AWS Lambda

A development team is using an Amazon SNS topic as part of an application. The development team is using an AWS SDK to send messages to the SNS topic. The SNS topic sends an email message to a specific person when each message arrives in the topic.

The development team wants a solution to invoke a specific AWS Lambda function if a message that arrives in the topic contains the word “Important” in the title attribute of the JSON message. The solution must not require any changes to the application code.

Which solution will meet these requirements?

A. Add a new subscription to the topic that will invoke the specified Lambda function. Add a subscription filter policy that filters on a title that contains “Important.”

B. Add a new subscription to the topic that will invoke the specified Lambda function. Add a redrive policy that filters on a title that contains “Important.”

C. Create a new SNS topic that will invoke the specified Lambda function. Add a subscription filter policy that filters on a title that contains “Important.”

D. Create a new SNS topic that will invoke the specified Lambda function. Add a redrive policy that filters on a title that contains “Important.”

Answer: A. Add a new subscription to the topic that will invoke the specified Lambda function. Add a subscription filter policy that filters on a title that contains “Important.”

Explanation

This question tests the ability to add a Lambda subscription to an existing SNS topic and use a subscription filter policy to deliver only messages whose title contains a specified value. This solution does not require changes to the publishing application.

✓ **A** Add a new subscription to the topic that will invoke the specified Lambda function. Add a subscription filter policy that filters on a title that contains “Important.”
Correct. An SNS subscription filter policy can selectively deliver messages to the Lambda subscription based on message attributes or message-body properties. Adding a subscription to the existing topic avoids changes to the application that publishes messages.

https://docs.aws.amazon.com/sns/latest/dq/sns-message-filtering.html

✗ **B** Add a new subscription to the topic that will invoke the specified Lambda function. Add a redrive policy that filters on a title that contains “Important.”
Incorrect. An SNS redrive policy sends undeliverable messages to a dead-letter queue; it does not filter messages based on their contents. Content-based routing requires a subscription filter policy.

https://docs.aws.amazon.com/sns/latest/dq/sns-dead-letter-queues.html

✗ **C** Create a new SNS topic that will invoke the specified Lambda function. Add a subscription filter policy that filters on a title that contains “Important.”
Incorrect. Creating a new topic does not cause messages published to the existing topic to be delivered to it, and it would require changing the application or adding message forwarding. A new filtered subscription should be added to the existing topic.

https://docs.aws.amazon.com/sns/latest/dq/sns-message-filtering.html

✗ **D** Create a new SNS topic that will invoke the specified Lambda function. Add a redrive policy that filters on a title that contains “Important.”
Incorrect. A redrive policy is for handling undeliverable messages and does not provide content-based

Free demo · full exam: 595 questions

filtering. Additionally, a new topic would not automatically receive messages from the existing topic.
https://docs.aws.amazon.com/sns/latest/dq/sns-dead-letter-queues.html

---

Question 8

Multiple choice · Domain: D1 Development with AWS Services › Use Amazon CloudWatch custom metrics and alarms with AWS Lambda and Amazon SNS

A company has an application that receives batches of orders from partners every day. The application uses an AWS Lambda function to process the batches.

If a batch contains no orders, the Lambda function must publish to an Amazon SNS topic as soon as possible.

Which combination of steps will meet this requirement with the LEAST implementation effort?

(Select TWO.)

A. Update the existing Lambda function's code to send an Amazon CloudWatch custom metric for the number of orders in a batch for each partner.

B. Create a new Lambda function as an Amazon Kinesis data stream consumer. Configure the new Lambda function to track orders and to publish to the SNS topic when a batch contains no orders.

C. Set up an Amazon CloudWatch alarm that will send a notification to the SNS topic when the value of the custom metric is 0.

D. Schedule a new Lambda function to analyze Amazon CloudWatch metrics every 24 hours to identify batches that contain no orders. Configure the Lambda function to publish to the SNS topic.

E. Modify the existing Lambda function to log orders to an Amazon Kinesis data stream.

Answer

- A. **Update the existing Lambda function's code to send an Amazon CloudWatch custom metric for the number of orders in a batch for each partner.**
- C. **Set up an Amazon CloudWatch alarm that will send a notification to the SNS topic when the value of the custom metric is 0.**

Explanation

This question tests the ability to instrument a Lambda function with a custom metric and configure a CloudWatch alarm to notify an SNS topic when a batch contains zero orders. This approach detects the condition without adding a separate stream-processing or scheduled-analysis component.

✓ **A** Update the existing Lambda function's code to send an Amazon CloudWatch custom metric for the number of orders in a batch for each partner.
Correct. The existing Lambda function already processes each batch, so it can publish a custom metric with the batch order count. This requires less implementation effort than introducing Kinesis and another consumer function.

https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/publishingMetrics.html

✗ **B** Create a new Lambda function as an Amazon Kinesis data stream consumer. Configure the new Lambda function to track orders and to publish to the SNS topic when a batch contains no orders.
Incorrect. This adds a new Lambda function and Kinesis processing infrastructure. Also, a batch with no orders would not produce order records in the stream, so the consumer cannot reliably determine that an empty batch was received without additional batch-marker logic.

Free demo · full exam: 595 questions

https://docs.aws.amazon.com/lambda/latest/dg/with-kinesis.html

✓ **C** Set up an Amazon CloudWatch alarm that will send a notification to the SNS topic when the value of the custom metric is 0.
Correct. A CloudWatch alarm can monitor the custom order-count metric and publish to an SNS topic when the metric reaches zero, providing prompt notification with minimal additional implementation.

https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/AlarmThatSendsEmail.html

✗ **D** Schedule a new Lambda function to analyze Amazon CloudWatch metrics every 24 hours to identify batches that contain no orders. Configure the Lambda function to publish to the SNS topic.
Incorrect. This introduces an additional scheduled Lambda function and delays detection for up to 24 hours, which does not meet the requirement to notify as soon as possible.

https://docs.aws.amazon.com/lambda/latest/dg/with-schedule.html

✗ **E** Modify the existing Lambda function to log orders to an Amazon Kinesis data stream.
Incorrect. Sending order records to Kinesis alone does not publish a notification to SNS and does not provide a direct mechanism to detect an empty batch. It would also require additional stream-consumer logic.

https://docs.aws.amazon.com/streams/latest/dev/introduction.html

Question 9

Single choice · Domain: D1 Development with AWS Services › Design an ordered, coordinated serverless workflow using AWS Lambda to manage inventory and payment processing

A developer is designing a serverless application that customers use to select seats for a concert venue. Customers send the ticket requests to an Amazon API Gateway API with an AWS Lambda function that acknowledges the order and generates an order ID. The application includes two additional Lambda functions: one for inventory management and one for payment processing. These two Lambda functions run in parallel and write the order to an Amazon DynamoDB table.

The application must provide seats to customers according to the following requirements. If a seat is accidentally sold more than once, the first order that the application received must get the seat. In these cases, the application must process the payment for only the first order. However, if the first order is rejected during payment processing, the second order must get the seat. In these cases, the application must process the payment for the second order.

Which solution will meet these requirements?

A. Send the order ID to an Amazon SNS FIFO topic that fans out to one Amazon SQS FIFO queue for inventory management and another SQS FIFO queue for payment processing.

B. Change the Lambda function that generates the order ID to initiate the Lambda function for inventory management. Then initiate the Lambda function for payment processing.

C. Send the order ID to an Amazon SNS topic. Subscribe the Lambda functions for inventory management and payment processing to the topic.

D. Deliver the order ID to an Amazon SQS queue. Configure the Lambda functions for inventory management and payment processing to poll the queue.

Answer: B. Change the Lambda function that generates the order ID to initiate the Lambda function for inventory management. Then initiate the Lambda function for payment processing.

Free demo · full exam: 595 questions

Dumpsgate

Question 10

Single choice · Domain: D1 Development with AWS Services › Package AWS SAM application source code for deployment with AWS CloudFormation

Given the source code for an AWS Lambda function in the local file store.py containing a handler function called getstore and the following AWS CloudFormation template:

Transform: AWS::Serverless-2016-10-31

Resources:

StoreFunc:

Type: AWS::Serverless::Function

Properties:

Handler: store.getstore

Runtime: python3.13

What should be done to prepare the template so that it can be deployed using the AWS CLI command aws cloudformation deploy?

A. Use aws cloudformation compile to base64 encode and embed the source file into a modified CloudFormation template.

B. Use aws cloudformation package to upload the source code to an Amazon S3 bucket and produce a modified CloudFormation template.

C. Use aws lambda zip to package the source file together with the CloudFormation template and deploy the resulting .zip archive.

D. Use aws serverless create-package to embed the source file directly into the existing CloudFormation template.

Answer: B. Use aws cloudformation package to upload the source code to an Amazon S3 bucket and produce a modified CloudFormation template.

Explanation

This question tests the ability to prepare local Lambda source code referenced by an AWS SAM template for deployment using the AWS CLI. The aws cloudformation package command uploads local artifacts to Amazon S3 and generates a transformed template that aws cloudformation deploy can use.

✗ **A** Use aws cloudformation compile to base64 encode and embed the source file into a modified CloudFormation template.
Incorrect. aws cloudformation compile is not an AWS CloudFormation CLI command, and Lambda source code is not prepared for deployment by embedding it as base64 in the template. AWS documents the package command for uploading local artifacts.

https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/using-cfn-cli-package.html

✓ **B** Use aws cloudformation package to upload the source code to an Amazon S3 bucket and produce a modified CloudFormation template.
Correct. The aws cloudformation package command uploads local Lambda artifacts to Amazon S3 and outputs a modified template with the S3 location. The resulting template can then be deployed with aws cloudformation deploy.

https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/using-cfn-cli-package.html

✗ **C** Use aws lambda zip to package the source file together with the CloudFormation template and deploy the resulting .zip archive.

Free demo · full exam: 595 questions

Incorrect. aws lambda zip is not a valid AWS CLI command, and the CloudFormation template is not packaged into the Lambda deployment archive. CloudFormation packages the local Lambda artifact and uploads it to Amazon S3 separately.

https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/using-cfn-cli-package.html

✗ **D** Use aws serverless create-package to embed the source file directly into the existing CloudFormation template.
Incorrect. aws serverless create-package is not the required AWS CLI command, and Lambda source code is not embedded directly into a CloudFormation template. The supported workflow uses aws cloudformation package to upload artifacts to Amazon S3.

https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/using-cfn-cli-package.html

---

Question 11

Single choice · Domain: D1 Development with AWS Services › Use AWS CloudFormation change sets to preview and review stack changes before updating resources

A developer creates an AWS CloudFormation template for a development environment. The developer configures a CloudFormation stack for the development environment.

The developer must occasionally update the CloudFormation template. The developer must review changes that are made to the stack when the developer updates the CloudFormation template.

Which solution will meet these requirements?

A. Create a stack set for CloudFormation stack updates.
B. Create a change set for CloudFormation stack updates.
C. Configure AWS X-Ray to inspect the CloudFormation stack.
D. Add a CloudFormation::HookTypeConfig resource to the template.

Answer: B. Create a change set for CloudFormation stack updates.

Explanation

This question tests the ability to select the CloudFormation feature that previews the proposed changes resulting from an updated template. Change sets allow developers to review changes before executing a stack update.

✗ **A** Create a stack set for CloudFormation stack updates.
Incorrect. CloudFormation StackSets are used to create and manage stacks across multiple AWS accounts and Regions; they do not provide the required change review mechanism for a single stack update.

https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/what-is-cfnstacksets.html

✓ **B** Create a change set for CloudFormation stack updates.
Correct. A change set provides a summary of the proposed changes to a stack, allowing the developer to review the impact of an updated template before executing the update.

https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/using-cfn-updating-stacks-changesets.html

✗ **C** Configure AWS X-Ray to inspect the CloudFormation stack.
Incorrect. AWS X-Ray analyzes and traces requests through applications and services. It does not preview or review CloudFormation template changes.

https://docs.aws.amazon.com/xray/latest/devguide/aws-xray.html

✗ **D** Add a CloudFormation::HookTypeConfig resource to the template.

Free demo · full exam: 595 questions

Incorrect. AWS::CloudFormation::HookTypeConfig configures a CloudFormation Hook, which can validate or control resource operations. It is not the feature used to preview changes from a template update.

https://docs.aws.amazon.com/AWSCloudFormation/latest/TemplateReference/aws-resource-cloudformation-hooktypeconfig.html

---

Question 12

Single choice · Domain: D1 Development with AWS Services › Configure AWS SDK retry behavior to handle throttling errors

A developer is testing a Python application that is deployed on AWS Elastic Beanstalk. The application is using the Boto3 library to send events to an Amazon SQS queue. An external tool consumes the events.

During the tests, a ThrottlingException error occurs at irregular intervals.

What should the developer do to resolve this error?

A. Increase the number of instances that are assigned to the Elastic Beanstalk deployment.

B. Scale up the instances in the Elastic Beanstalk deployment to provide more RAM and CPU to the environment.

C. Configure Boto3 to use the adaptive retry mode to retry the failing step with progressively longer waits between calls.

D. Redeploy the Elastic Beanstalk environment. Choose a less memory-intensive programming language.

Answer: C. Configure Boto3 to use the adaptive retry mode to retry the failing step with progressively longer waits between calls.

Explanation

This question tests the ability to configure Boto3 retry behavior when an AWS service throttles API requests. Adaptive retry mode uses client-side rate limiting and backoff to reduce request pressure and retry throttled calls.

✗ **A** Increase the number of instances that are assigned to the Elastic Beanstalk deployment.
Incorrect. Increasing the number of Elastic Beanstalk instances does not directly resolve SQS API throttling and could increase the number of requests made to the queue. The retry behavior of the AWS SDK should be configured instead.

https://docs.aws.amazon.com/sdkref/latest/guide/feature-retry-behavior.html

✗ **B** Scale up the instances in the Elastic Beanstalk deployment to provide more RAM and CPU to the environment.
Incorrect. Adding CPU or memory does not address throttling imposed on requests to Amazon SQS. Throttled SDK requests should be handled with an appropriate retry mode and backoff strategy.

https://docs.aws.amazon.com/sdkref/latest/guide/feature-retry-behavior.html

✓ **C** Configure Boto3 to use the adaptive retry mode to retry the failing step with progressively longer waits between calls.
Correct. Boto3 supports adaptive retry mode, which uses client-side rate limiting and backoff to respond to throttling and retry failed requests. This helps reduce request pressure on the throttled service.

✗ **D** Redeploy the Elastic Beanstalk environment. Choose a less memory-intensive programming

language.
Incorrect. Redeploying the environment or changing the programming language does not resolve Amazon SQS API throttling. The application should use an AWS SDK retry mode that handles throttled requests.
https://docs.aws.amazon.com/sdkref/latest/guide/feature-retry-behavior.html

Question 13

Single choice · Domain: D1 Development with AWS Services › Restrict API Gateway REST API access by source IP address using a resource policy

A developer is designing an internal single-page website for a company. The website has static content that is hosted in an Amazon S3 bucket that is configured as a static web host. The design requires the web application to invoke a REST API in Amazon API Gateway to run an AWS Lambda function.

The developer must ensure that the REST API can be invoked only by computers that have IP addresses within the company's corporate CIDR range.

Which solution will meet this requirement?

A. Define an S3 bucket policy to restrict website access to the corporate CIDR range.

B. Update the IAM service-linked role that is associated with API Gateway to restrict access to the corporate CIDR range.

C. Define a network ACL to restrict API access to the corporate CIDR range.

D. Define an API Gateway resource policy to restrict API access to the corporate CIDR range.

Answer: D. Define an API Gateway resource policy to restrict API access to the corporate CIDR range.

Explanation

This question tests the ability to configure an Amazon API Gateway resource policy that allows or denies invocation based on the caller's source IP address. The policy can use the aws:SourceIp condition key to restrict access to the company's corporate CIDR range.

✗ **A** Define an S3 bucket policy to restrict website access to the corporate CIDR range.
Incorrect. An S3 bucket policy controls access to the static website content in Amazon S3, not invocation of the separate API Gateway REST API.

https://docs.aws.amazon.com/AmazonS3/latest/userguide/WebsiteAccessPermissionsReqd.html

✗ **B** Update the IAM service-linked role that is associated with API Gateway to restrict access to the corporate CIDR range.
Incorrect. The API Gateway service-linked role grants API Gateway permissions to interact with other AWS services; it does not restrict which client source IP addresses can invoke an API.

https://docs.aws.amazon.com/apigateway/latest/developerguide/using-service-linked-roles.html

✗ **C** Define a network ACL to restrict API access to the corporate CIDR range.
Incorrect. Network ACLs control traffic at the subnet level for resources in a VPC, whereas the API Gateway REST API is accessed through an API Gateway endpoint. An API Gateway resource policy is the appropriate control for source IP restrictions.

https://docs.aws.amazon.com/vpc/latest/userguide/vpc-network-acls.html

Free demo · full exam: 595 questions

✓ **D** Define an API Gateway resource policy to restrict API access to the corporate CIDR range.
Correct. An API Gateway resource policy can allow or deny access to a REST API based on the caller's source IP address by using the aws:SourceIp condition key and the corporate CIDR range.
https://docs.aws.amazon.com/apigateway/latest/developerguide/apigateway-resource-policies.html

---

Question 14

Single choice · Domain: D1 Development with AWS Services › Configure AWS CodeBuild batch builds with dependency graphs and appropriate compute capacity

A development team is working on a project in AWS CodeBuild that requires multiple interdependent builds to run simultaneously. The development team wants to optimize resource usage by combining Amazon EC2 On-Demand Instances and reserved capacity fleets for the builds.

The builds include tasks for multiple platforms, such as web, mobile, and API components.

Which solution will meet these requirements in the MOST cost-effective way?

A. Create separate CodeBuild projects for each platform. Use On-Demand Instances for all builds.
B. Set up batch builds in CodeBuild. Define dependencies between platforms. Use a combination of Reserved Instances and On-Demand Instances.
C. Create separate CodeBuild projects for each platform. Assign Reserved Instances or On-Demand Instances to each project based on resource availability.
D. Configure a single CodeBuild project. Run all builds sequentially by using Reserved Instances.

Answer: B. Set up batch builds in CodeBuild. Define dependencies between platforms. Use a combination of Reserved Instances and On-Demand Instances.

Explanation

This question tests the ability to use CodeBuild batch builds to execute interdependent builds concurrently while selecting a cost-effective combination of reserved capacity and On-Demand compute resources.

✗ **A** Create separate CodeBuild projects for each platform. Use On-Demand Instances for all builds.
Incorrect. Separate projects do not provide the dependency-aware orchestration of a CodeBuild batch build, and using On-Demand capacity for every build does not optimize costs when reserved capacity is available.

https://docs.aws.amazon.com/codebuild/latest/userguide/batch-build.html

✓ **B** Set up batch builds in CodeBuild. Define dependencies between platforms. Use a combination of Reserved Instances and On-Demand Instances.
Correct. CodeBuild batch builds support multiple builds in a batch and build graphs that define dependencies, allowing independent builds to run concurrently while dependent builds wait. Combining reserved capacity with On-Demand capacity provides flexibility and can reduce costs compared with using On-Demand capacity for all builds. and.

https://docs.aws.amazon.com/codebuild/latest/userguide/batch-build.html
https://aws.amazon.com/codebuild/pricing/

✗ **C** Create separate CodeBuild projects for each platform. Assign Reserved Instances or On-Demand Instances to each project based on resource availability.
Incorrect. Although using reserved capacity for suitable projects can reduce cost, separate projects do not model the inter-build dependencies or provide the coordinated parallel execution available through

CodeBuild batch builds.

https://docs.aws.amazon.com/codebuild/latest/userguide/batch-build.html

✗ **D** Configure a single CodeBuild project. Run all builds sequentially by using Reserved Instances.
Incorrect. Running builds sequentially fails the requirement to run multiple interdependent builds simultaneously and can increase total build time. A CodeBuild batch build can coordinate dependencies while allowing independent builds to run in parallel.

https://docs.aws.amazon.com/codebuild/latest/userguide/batch-build.html

Question 15

Single choice · Domain: D1 Development with AWS Services › Publish and use a custom Amazon CloudWatch metric to scale resources based on application-level demand

A company wants to migrate its web application to AWS and leverage auto scaling to handle peak workloads. The solutions architect determined that the best metric for an auto scaling event is the number of concurrent users.

Based on this information, what should the developer use to auto scale based on concurrent users?

A. An Amazon SNS topic to be invoked when a concurrent user threshold is met
B. An Amazon CloudWatch NetworkIn metric
C. Amazon CloudFront to leverage AWS edge locations
D. A custom Amazon CloudWatch metric for concurrent users

Answer: D. A custom Amazon CloudWatch metric for concurrent users

Explanation

This question tests the ability to select a custom CloudWatch metric when the required scaling signal, such as concurrent users, is not provided as a standard AWS service metric. Application-level metrics can be published to CloudWatch and used with scaling policies.

✗ **A** An Amazon SNS topic to be invoked when a concurrent user threshold is met
Incorrect. Amazon SNS is a notification and messaging service, not a metric that measures concurrent users or directly provides the scaling signal. CloudWatch alarms can send notifications to SNS, but the underlying metric must still be published or collected separately.

https://docs.aws.amazon.com/sns/latest/dq/welcome.html

✗ **B** An Amazon CloudWatch NetworkIn metric
Incorrect. NetworkIn measures the amount of network traffic received by an EC2 instance, not the number of concurrent users. It therefore does not directly represent the application demand specified in the question.

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/viewing_metrics_with_cloudwatch.html

✗ **C** Amazon CloudFront to leverage AWS edge locations
Incorrect. CloudFront improves content delivery by using edge locations, but it does not provide an application-specific concurrent-user metric for scaling the web application.

https://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/Introduction.html

✓ **D** A custom Amazon CloudWatch metric for concurrent users
Correct. Concurrent users is an application-level metric that is not represented by the standard infrastructure metrics listed. The developer should publish a custom CloudWatch metric for concurrent

Free demo · full exam: 595 questions

users and use it in an alarm or target-tracking scaling policy.
https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/publishingMetrics.html

Question 16

Single choice · Domain: D1 Development with AWS Services › Construct an Amazon EventBridge event pattern that filters Amazon S3 object deletion events by event type and bucket ARN

A developer at a company needs to send email notifications to the company's application security team each time an object is deleted from an Amazon S3 bucket named amzn-s3-demo-bucket. The developer uses Amazon EventBridge and Amazon SNS to send the notifications. The developer creates the following EventBridge rule:

{ "source": ["aws.s3"], "detail-type": [{"suffix": "Deleted"}] } The security team reports that it receives email notifications about objects being deleted from S3 buckets other than amzn-s3-demo-bucket. The security team also receives notifications when other S3 events happen across all buckets in the company's AWS account.

The developer needs to modify the EventBridge rule to ensure that the security team receives email notifications only when an object is deleted from the amzn-s3-demo-bucket S3 bucket.

Which solution will meet this requirement?

A. Specify the source as aws.s3.amzn-s3-demo-bucket. Specify the bucket Amazon Resource Name (ARN) in a resources property.

B. Change the detail-type property to Object Deleted. Specify the source as amzn-s3-demo-bucket.

C. Specify the bucket Amazon Resource Name (ARN) in a resources property. Change the detail-type property to Object Deleted.

D. Specify the bucket Amazon Resource Name (ARN) as the source. Specify aws.sns in a target property.

Answer: C. Specify the bucket Amazon Resource Name (ARN) in a resources property. Change the detail-type property to Object Deleted.

Explanation

This question tests the ability to configure EventBridge filtering for Amazon S3 events and route only matching events to an Amazon SNS notification target. The event pattern must retain the aws.s3 source, use the exact Object Deleted detail type, and identify the target bucket in the resources field.

✗ **A** Specify the source as aws.s3.amzn-s3-demo-bucket. Specify the bucket Amazon Resource Name (ARN) in a resources property.
Incorrect. Amazon S3 EventBridge events use aws.s3 as the source; the bucket name is not appended to the source value. Although filtering by the bucket ARN in resources is appropriate, the suffix Deleted filter can still match multiple deletion-related event types.

https://docs.aws.amazon.com/AmazonS3/latest/userguide/ev-events.html

✗ **B** Change the detail-type property to Object Deleted. Specify the source as amzn-s3-demo-bucket.
Incorrect. The source must be aws.s3, not the S3 bucket name. Also, changing the detail type alone does not restrict events to the required bucket; the bucket ARN must be specified in the resources property.

https://docs.aws.amazon.com/AmazonS3/latest/userguide/ev-events.html

✓ **C** Specify the bucket Amazon Resource Name (ARN) in a resources property. Change the detail-type

property to Object Deleted.
Correct. With the existing source of aws.s3 retained, setting detail-type to Object Deleted and specifying the amzn-s3-demo-bucket ARN in resources restricts the rule to object deletion events from that bucket.
https://docs.aws.amazon.com/AmazonS3/latest/userguide/ev-events.html

✗ **D** Specify the bucket Amazon Resource Name (ARN) as the source. Specify aws.sns in a target property.
Incorrect. The bucket ARN cannot be used as the event source; S3 events use aws.s3. An SNS topic is configured as a target, not as a way to filter events, and this solution does not restrict the event type or bucket correctly.
https://docs.aws.amazon.com/eventbridge/latest/userguide/eb-targets.html

---

Question 17

Single choice · Domain: D1 Development with AWS Services › Configure AWS Step Functions logging to exclude execution data

A developer is reviewing an AWS Step Functions state machine to process files that are uploaded to an Amazon S3 bucket. The state machine is created by an AWS CloudFormation template. The state machine includes three sequential AWS Lambda functions. The state machine runs when new objects are uploaded to the S3 bucket. Full logging to Amazon CloudWatch and AWS X-Ray is enabled in the template.

When the developer reviews the state machine output, the developer sees that the logs include sensitive information. The developer must prevent the logs from storing sensitive information.

Which solution will meet this requirement?

A. Update the StateMachine properties of the logging configuration in the CloudFormation template to Level: ERROR. Redeploy the template.

B. Update the StateMachine properties of the logging configuration in the CloudFormation template to Level: FATAL. Redeploy the template.

C. Update the StateMachine properties of the logging configuration in the CloudFormation template to IncludeExecutionData: false. Redeploy the template.

D. Update the StateMachine properties of the tracing configuration in the CloudFormation template to Enabled: false. Redeploy the template.

Answer: C. Update the StateMachine properties of the logging configuration in the CloudFormation template to IncludeExecutionData: false. Redeploy the template.

Explanation

This question tests the ability to configure Step Functions CloudWatch logging so that execution input and output data, which may contain sensitive information, are not included in logs.

✗ **A** Update the StateMachine properties of the logging configuration in the CloudFormation template to Level: ERROR. Redeploy the template.
Incorrect. Changing the logging level to ERROR only limits which events are logged; it does not prevent execution data from being included in the logged events.
https://docs.aws.amazon.com/step-functions/latest/dg/cw-logs.html

✗ **B** Update the StateMachine properties of the logging configuration in the CloudFormation template to Level: FATAL. Redeploy the template.

Free demo · full exam: 595 questions

Incorrect. A more restrictive logging level does not specifically disable logging of execution input and output data, so sensitive execution data can still be included in logged events.

https://docs.aws.amazon.com/step-functions/latest/dg/cw-logs.html

✓ **C** Update the StateMachine properties of the logging configuration in the CloudFormation template to IncludeExecutionData: false. Redeploy the template.
Correct. Setting IncludeExecutionData to false prevents Step Functions from logging execution input and output data to CloudWatch Logs, while still allowing other execution-history events to be logged.

https://docs.aws.amazon.com/step-functions/latest/dg/cw-logs.html

✗ **D** Update the StateMachine properties of the tracing configuration in the CloudFormation template to Enabled: false. Redeploy the template.
Incorrect. Disabling AWS X-Ray tracing does not prevent sensitive execution data from being written to the configured CloudWatch Logs destination. The logging configuration must set IncludeExecutionData to false.

https://docs.aws.amazon.com/step-functions/latest/dg/cw-logs.html

Question 18

Single choice · Domain: D1 Development with AWS Services › Configure an API Gateway mock integration to return predefined API responses

A company is using Amazon API Gateway to develop an API for its application on AWS. A developer needs to test and generate API responses. Other teams are required to test the API immediately.

What should the developer do to meet these requirements?

A. Set up a mock integration request in API Gateway. Configure the method's integration request and integration response to associate a response with a given status code.

B. Set up the request validators in the API's OpenAPI definition file. Import the OpenAPI definitions into API Gateway to test the API.

C. Set up a gateway response for the API in API Gateway. Configure response headers with hardcoded HTTP status codes and responses.

D. Set up a request parameter-based Lambda authorizer to control access to the API. Configure the Lambda function with the necessary mapping template.

Answer: A. Set up a mock integration request in API Gateway. Configure the method's integration request and integration response to associate a response with a given status code.

Explanation

This question tests the ability to use API Gateway mock integrations to generate responses without invoking a backend service, allowing teams to test the API immediately.

✓ **A** Set up a mock integration request in API Gateway. Configure the method's integration request and integration response to associate a response with a given status code.
Correct. An API Gateway mock integration lets the developer generate predefined API responses without calling a backend, making the API available for immediate testing.

https://docs.aws.amazon.com/apigateway/latest/developerguide/how-to-mock-integration.html

✗ **B** Set up the request validators in the API's OpenAPI definition file. Import the OpenAPI definitions into API Gateway to test the API.

Free demo · full exam: 595 questions

Incorrect. Request validators check incoming request parameters and payloads; they do not generate API responses or provide a mock backend for testing.

https://docs.aws.amazon.com/apigateway/latest/developerguide/api-gateway-request-validation-set-up.html

✗ **C** Set up a gateway response for the API in API Gateway. Configure response headers with hardcoded HTTP status codes and responses.
Incorrect. Gateway responses customize responses for API Gateway-generated errors, such as authentication failures or throttling, but they are not a general mechanism for generating successful API responses for testing.

https://docs.aws.amazon.com/apigateway/latest/developerguide/supported-gateway-response-types.html

✗ **D** Set up a request parameter-based Lambda authorizer to control access to the API. Configure the Lambda function with the necessary mapping template.
Incorrect. A Lambda authorizer controls access by evaluating authorization information; it does not generate the API's application responses or replace the backend integration.

https://docs.aws.amazon.com/apigateway/latest/developerguide/apigateway-use-lambda-authorizer.html

---

Question 19

Single choice · Domain: D1 Development with AWS Services › Select and query the appropriate DynamoDB secondary index for a new access pattern

A developer manages an application that writes customer orders to an Amazon DynamoDB table.

The orders use customerId as the partition key, orderId as the sort key, and order_date as an attribute. A new access pattern requires accessing data by order_date and orderId. The developer needs to implement a new AWS Lambda function to support the new access pattern.

How should the developer support the new access pattern in the MOST operationally efficient way?

A. Add a new local secondary index (LSI) to the DynamoDB table that specifies order_date as the partition key and orderId as the sort key. Write the new Lambda function to query the new LSI index.

B. Write the new Lambda function to scan the DynamoDB table. In the Lambda function, write a method to retrieve and combine results by order_date and orderId.

C. Add a new global secondary index (GSI) to the DynamoDB table that specifies order_date as the partition key and orderId as the sort key. Write the new Lambda function to query the new GSI index.

D. Enable DynamoDB Streams on the table. Choose the new and old images information to write to the DynamoDB stream. Write the new Lambda function to query the DynamoDB stream.

Answer: C. Add a new global secondary index (GSI) to the DynamoDB table that specifies order_date as the partition key and orderId as the sort key. Write the new Lambda function to query the new GSI index.

Explanation

This question tests the ability to design a DynamoDB index that supports querying by attributes different from the table's primary key. A global secondary index can use order_date as its partition key and orderId as its sort key, enabling efficient queries from the Lambda function.

✗ **A** Add a new local secondary index (LSI) to the DynamoDB table that specifies order_date as the partition key and orderId as the sort key. Write the new Lambda function to query the new LSI index.
Incorrect. A local secondary index must use the same partition key as the base table, which is customerId.

Free demo · full exam: 595 questions

It cannot use order_date as a different partition key, and LSIs must be created when the table is created.
https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/LSI.html

✗ **B** Write the new Lambda function to scan the DynamoDB table. In the Lambda function, write a method to retrieve and combine results by order_date and orderId.
Incorrect. A table scan examines every item and is inefficient for an access pattern that can be supported by an index. It can consume substantial read capacity and requires application-side filtering or processing.
https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/Scan.html

✓ **C** Add a new global secondary index (GSI) to the DynamoDB table that specifies order_date as the partition key and orderId as the sort key. Write the new Lambda function to query the new GSI index.
Correct. A global secondary index can use a partition key and sort key that differ from those of the base table. Defining a GSI with order_date as the partition key and orderId as the sort key allows the Lambda function to efficiently query the required access pattern.
https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/GSI.html

✗ **D** Enable DynamoDB Streams on the table. Choose the new and old images information to write to the DynamoDB stream. Write the new Lambda function to query the DynamoDB stream.
Incorrect. DynamoDB Streams capture item-level changes for processing and are not queryable indexes for retrieving orders by order_date and orderId. A global secondary index is the appropriate mechanism for this access pattern.
https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/Streams.html

Question 20

Single choice · Domain: D1 Development with AWS Services › Use DynamoDB condition expressions with PutItem to prevent overwriting an existing item

A developer is writing an application that stores data in an Amazon DynamoDB table by using the PutItem API operation. The table has a partition key of streamID and has a sort key of seqID. The developer needs to make sure that the PutItem invocation does not overwrite the existing partition key and sort key.

Which condition expression will maintain the uniqueness of the partition key and the sort key?

A. condition = attribute_not_exists(streamID) AND attribute_not_exists(seqID)
B. condition = attribute_not_exists(PARTITION) AND attribute_not_exists(SORT)
C. condition = attribute_exists(streamID) AND attribute_exists(seqID)
D. condition = attribute_exists(PARTITION) AND attribute_exists(SORT)

Answer: A. condition = attribute_not_exists(streamID) AND attribute_not_exists(seqID)

Explanation

This question tests the ability to apply attribute_not_exists conditions to a DynamoDB PutItem operation so that an item with the same partition key and sort key cannot already exist.

✓ **A** condition = attribute_not_exists(streamID) AND attribute_not_exists(seqID)
Correct. The condition succeeds only when the specified key attributes do not already exist on an item, preventing PutItem from overwriting an existing item with the same composite primary key.
https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/Expressions.ConditionExpressions.html

✗ **B** condition = attribute_not_exists(PARTITION) AND attribute_not_exists(SORT)
Incorrect. PARTITION and SORT are not the table's attribute names. The condition expression must reference the actual key attributes, streamID and seqID.
https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/Expressions.ConditionExpressions.html

✗ **C** condition = attribute_exists(streamID) AND attribute_exists(seqID)
Incorrect. attribute_exists requires the attributes to exist, which would allow the condition only for an existing item and does not enforce uniqueness for a new PutItem operation.
https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/Expressions.ConditionExpressions.html

✗ **D** condition = attribute_exists(PARTITION) AND attribute_exists(SORT)
Incorrect. This uses attribute_exists instead of attribute_not_exists and also references names that are not the table's key attributes. It does not prevent overwriting the specified composite key.
https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/Expressions.ConditionExpressions.html

Question 21

Single choice · Domain: D1 Development with AWS Services › Publish high-resolution custom metrics to Amazon CloudWatch with dimensions

A developer needs to add additional monitoring to an application that runs in an Amazon ECS cluster.

The solution must publish and store custom metrics for the application. The developer must be able to save metadata about the custom metrics and store data with a granularity of 1 second.

Which solution will meet these requirements?

A. Instrument the application to use AWS X-Ray by installing and running the X-Ray daemon. Use the X-Ray SDK to upload custom metrics and metadata through traces. Segment and subsegment data.

B. Use Amazon ECS Container Insights to publish custom metrics from within the application. Specify high resolution and use the metadata feature to store more data.

C. Use the Amazon CloudWatch PutMetricData API to publish the custom metrics. Set the StorageResolution field to 1. Use the dimensions parameter to assign name-value pairs to the metrics.

D. Monitor the containers by using Amazon ECS Exec. Use Amazon EventBridge Scheduler to run the ECS execute-command CLI command on the cluster to publish custom metrics on a schedule.

Answer: C. Use the Amazon CloudWatch PutMetricData API to publish the custom metrics. Set the StorageResolution field to 1. Use the dimensions parameter to assign name-value pairs to the metrics.

Explanation

This question tests the ability to use the CloudWatch PutMetricData API to publish application metrics, configure one-second storage resolution, and associate metadata-like name-value pairs through dimensions.

✗ **A** Instrument the application to use AWS X-Ray by installing and running the X-Ray daemon. Use the X-Ray SDK to upload custom metrics and metadata through traces. Segment and subsegment data.
Incorrect. AWS X-Ray is used to collect trace data, segments, subsegments, annotations, and metadata; it is not the service or API for publishing custom CloudWatch metrics with one-second storage resolution.
https://docs.aws.amazon.com/xray/latest/devguide/aws-xray.html

✗ **B** Use Amazon ECS Container Insights to publish custom metrics from within the application. Specify high resolution and use the metadata feature to store more data.
Incorrect. Container Insights collects and publishes operational metrics for ECS clusters, services, tasks, and containers, but it does not provide the application custom metric publishing mechanism described in the question or a metadata feature for custom metrics.

https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/ContainerInsights.html

✓ **C** Use the Amazon CloudWatch PutMetricData API to publish the custom metrics. Set the StorageResolution field to 1. Use the dimensions parameter to assign name-value pairs to the metrics.
Correct. PutMetricData publishes custom metrics to CloudWatch. Setting StorageResolution to 1 creates a high-resolution metric with one-second granularity, and dimensions provide name-value pairs that can be used to associate and distinguish metric data.

https://docs.aws.amazon.com/AmazonCloudWatch/latest/APIReference/API_PutMetricData.html

✗ **D** Monitor the containers by using Amazon ECS Exec. Use Amazon EventBridge Scheduler to run the ECS execute-command CLI command on the cluster to publish custom metrics on a schedule.
Incorrect. ECS Exec provides interactive command execution inside running containers; it is not a custom metrics publishing service and cannot provide one-second metric storage resolution. Custom metrics should be sent directly to CloudWatch using PutMetricData.

https://docs.aws.amazon.com/AmazonECS/latest/developerguide/ecs-exec-run.html

Question 22

Single choice · Domain: D1 Development with AWS Services › Use AWS Lambda layers to manage and share common function dependencies

A company is developing a set of AWS Lambda functions to process data. The Lambda functions need to use a common third-party library as a dependency. The library is frequently updated with new features and bug fixes. The company wants to ensure that the Lambda functions always use the latest version of the library.

Which solution will meet these requirements in the MOST operationally efficient way?

A. Store the dependency and the function code in an Amazon S3 bucket.

B. Create a Lambda layer that includes the library. Attach the layer to each Lambda function.

C. Install the dependency in an Amazon EFS file system. Attach the file system to each Lambda function.

D. Create a new Lambda function to load the library. Configure the existing Lambda functions to invoke the new Lambda function when the existing functions need to use the library.

Answer: B. Create a Lambda layer that includes the library. Attach the layer to each Lambda function.

Explanation

This question tests the ability to select Lambda layers for centrally managing a shared third-party library across multiple functions. A new layer version can be published when the library changes and associated functions can be updated to use it.

✗ **A** Store the dependency and the function code in an Amazon S3 bucket.
Incorrect. Storing code and dependencies in Amazon S3 does not automatically make the dependency available to Lambda functions at runtime or keep deployed functions synchronized with updates. Lambda

layers are designed to package and share common dependencies.

https://docs.aws.amazon.com/lambda/latest/dg/chapter-layers.html

✓ **B** Create a Lambda layer that includes the library. Attach the layer to each Lambda function.
Correct. Lambda layers provide a centralized way to package and share libraries across multiple functions. When the library is updated, the company can publish a new layer version and update the functions to use that version, avoiding separate dependency packages for every function.

https://docs.aws.amazon.com/lambda/latest/dg/chapter-layers.html

✗ **C** Install the dependency in an Amazon EFS file system. Attach the file system to each Lambda function.
Incorrect. Lambda can access dependencies on Amazon EFS, but this requires additional file-system, network, mount-target, and VPC configuration. EFS is not the most operationally efficient solution for a commonly shared library when Lambda layers directly support this use case.

https://docs.aws.amazon.com/lambda/latest/dg/configuration-filesystem.html

✗ **D** Create a new Lambda function to load the library. Configure the existing Lambda functions to invoke the new Lambda function when the existing functions need to use the library.
Incorrect. Invoking another Lambda function does not directly provide its library to the caller's execution environment. This approach adds unnecessary invocations, coordination, latency, and operational complexity instead of using a shared dependency mechanism such as a Lambda layer.

https://docs.aws.amazon.com/lambda/latest/dg/chapter-layers.html

Question 23

Single choice · Domain: D1 Development with AWS Services › Use AWS Lambda layers to share reusable code and dependencies across Lambda functions

A developer created reusable code that several AWS Lambda functions need to use. The developer bundled the code into a .zip archive. The developer needs to deploy the code to AWS and update the Lambda functions to use the code.

Which solution will meet this requirement in the MOST operationally efficient way?

A. Upload the .zip archive to Amazon S3. Configure an import path on the Lambda functions to point to the .zip archive.

B. Create a new Lambda function that contains and runs the shared code. Update the existing Lambda functions to invoke the new Lambda function synchronously.

C. Create a Lambda layer that contains the .zip archive. Attach the Lambda layer to the Lambda functions.

D. Create a Lambda container image that includes the shared code. Use the container image as a Lambda base image for all the functions.

Answer: C. Create a Lambda layer that contains the .zip archive. Attach the Lambda layer to the Lambda functions.

Explanation

This question tests the ability to select Lambda layers as the operationally efficient mechanism for packaging and attaching common code to multiple Lambda functions. Layers allow shared libraries and other dependencies to be managed separately from function code.

✗ **A** Upload the .zip archive to Amazon S3. Configure an import path on the Lambda functions to point to

Free demo · full exam: 595 questions

the .zip archive.
Incorrect. Lambda functions cannot import shared code merely by configuring an import path to a ZIP archive stored in Amazon S3. The code must be included in the deployment package or provided through a Lambda layer.

https://docs.aws.amazon.com/lambda/latest/dg/chapter-layers.html

✗ **B** Create a new Lambda function that contains and runs the shared code. Update the existing Lambda functions to invoke the new Lambda function synchronously.
Incorrect. Although functions can invoke other Lambda functions, this introduces additional synchronous invocations, coupling, latency, and operational overhead. A Lambda layer is intended for sharing common code and dependencies.

https://docs.aws.amazon.com/lambda/latest/dg/chapter-layers.html

✓ **C** Create a Lambda layer that contains the .zip archive. Attach the Lambda layer to the Lambda functions.
Correct. Lambda layers package libraries and other dependencies separately from function code and can be attached to multiple functions, making them the most operationally efficient solution for reusable shared code.

https://docs.aws.amazon.com/lambda/latest/dg/chapter-layers.html

✗ **D** Create a Lambda container image that includes the shared code. Use the container image as a Lambda base image for all the functions.
Incorrect. Lambda container images can package function code and dependencies, but creating and maintaining container images for shared code is more operationally complex than using a Lambda layer for this requirement.

https://docs.aws.amazon.com/lambda/latest/dg/images-create.html

Question 24

Single choice · Domain: D1 Development with AWS Services › Implement a circuit breaker to prevent repeated Lambda calls to an unavailable third-party service and automatically resume calls when the service recovers

A developer maintains an AWS Lambda function that is invoked synchronously through an API Gateway API. The Lambda function processes payments by calling a third-party platform. The Lambda function includes retry logic that will make five attempts for each invocation.

The third-party platform occasionally experiences outages that last 10–20 minutes. During outages, the Lambda function makes five retry attempts for each incoming request before failing. The request retry process generates thousands of wasted API calls and increases Lambda compute costs.

The developer needs a solution to detect third-party platform outages and to ensure that invocations of the Lambda function stop when a platform outage is detected. The developer must also ensure that the invocations resume when the platform is operational and accessible.

Which solution will meet these requirements?

A. Configure a dead-letter queue to capture events after the Lambda function's five retry attempts are exhausted.

B. Increase the Lambda function's timeout allocation and memory allocation so that retry attempts run longer.

C. Add exponential backoff to the Lambda function's retry logic to space out retry attempts during failures.

D. Implement a circuit breaker that stops invocations after failures exceed a threshold.

Answer: D. Implement a circuit breaker that stops invocations after failures exceed a threshold.

Explanation

This question tests the ability to design resilient Lambda integrations with external services by detecting repeated failures, stopping unnecessary requests during an outage, and allowing requests to resume after recovery.

✗ **A** Configure a dead-letter queue to capture events after the Lambda function's five retry attempts are exhausted.
Incorrect. A dead-letter queue stores failed asynchronous invocation events after Lambda's retry processing; it does not detect a third-party outage or stop synchronous API Gateway invocations from repeatedly executing the function.

https://docs.aws.amazon.com/lambda/latest/dg/invocation-async-retain-records.html

✗ **B** Increase the Lambda function's timeout allocation and memory allocation so that retry attempts run longer.
Incorrect. Increasing timeout or memory does not detect an outage or prevent repeated calls. It can instead allow the function to consume more compute resources while retries continue.

https://docs.aws.amazon.com/lambda/latest/dg/configuration-timeout.html

✗ **C** Add exponential backoff to the Lambda function's retry logic to space out retry attempts during failures.
Incorrect. Exponential backoff reduces the frequency of retries, but it does not stop invocations after an outage threshold is reached or provide a mechanism to resume calls based on service recovery.

https://docs.aws.amazon.com/prescriptive-guidance/latest/cloud-design-patterns/retry-backoff.html

✓ **D** Implement a circuit breaker that stops invocations after failures exceed a threshold.
Correct. A circuit breaker tracks failures and opens the circuit when the failure threshold is exceeded, preventing additional calls to the unavailable platform and avoiding wasted retries and Lambda compute. After a recovery interval, it can test the dependency and close the circuit when the platform becomes available again.

https://docs.aws.amazon.com/prescriptive-guidance/latest/cloud-design-patterns/circuit-breaker.html

Question 25

Single choice · Domain: D1 Development with AWS Services › Implement cursor-based pagination for DynamoDB query results using LastEvaluatedKey and ExclusiveStartKey

A company built an application on a SQL database. The company then migrated the application to Amazon DynamoDB. The company needs to ensure that the application supports infinite scrolling for a news feed that displays user-generated posts. The application previously relied on numeric page-based pagination. The application must now rely on cursor-based pagination in DynamoDB.

Which solution will meet these requirements in the MOST operationally efficient way?

A. Use the LastEvaluatedKey value returned from each DynamoDB query as the ExclusiveStartKey value in subsequent queries.

B. Perform a full table scan to calculate page offsets before returning results.

C. Retrieve all news feed items from DynamoDB in a single query. Paginate entirely on the client side.

D. Store the page numbers directly in DynamoDB items. Query based on the page numbers to retrieve paginated results.

Answer: A. Use the LastEvaluatedKey value returned from each DynamoDB query as the ExclusiveStartKey value in subsequent queries.

Explanation

This question tests the ability to implement DynamoDB pagination by using the LastEvaluatedKey returned by one Query operation as the ExclusiveStartKey for the next operation. This approach supports efficient infinite scrolling without calculating numeric page offsets or retrieving the entire result set.

✓ **A** Use the LastEvaluatedKey value returned from each DynamoDB query as the ExclusiveStartKey value in subsequent queries.
Correct. DynamoDB returns LastEvaluatedKey when more results are available, and the application can pass that value as ExclusiveStartKey in the next Query request. This is the native cursor-based pagination mechanism and is operationally efficient.

https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/Query.Pagination.html

✗ **B** Perform a full table scan to calculate page offsets before returning results.
Incorrect. A full table scan is inefficient because it reads items across the table and does not provide native cursor-based pagination. It can consume substantial read capacity and increase latency.

https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/Scan.html

✗ **C** Retrieve all news feed items from DynamoDB in a single query. Paginate entirely on the client side.
Incorrect. Retrieving all items is inefficient, may exceed DynamoDB response size limits, and increases network usage, memory consumption, and application latency. DynamoDB Query operations should be paginated using LastEvaluatedKey when additional results exist.

https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/Query.Pagination.html

✗ **D** Store the page numbers directly in DynamoDB items. Query based on the page numbers to retrieve paginated results.
Incorrect. Storing numeric page numbers recreates offset-based pagination and introduces maintenance and consistency problems when posts are added or removed. DynamoDB provides cursor-based pagination through LastEvaluatedKey and ExclusiveStartKey instead.

https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/Query.Pagination.html

Free demo · full exam: 595 questions

Question 26

Single choice · Domain: D1 Development with AWS Services › Configure Amazon SNS message filtering with numeric message attributes

A company is building a new application that processes customer transactions by using an event-driven architecture. The company needs to receive a notification that includes transaction details when the transactions are above a price threshold.

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

A. Create an Amazon SQS standard queue and an SQS dead-letter queue (DLQ). Configure a redrive policy for the standard queue to send failed messages to the DLQ based on the price threshold. Use an Amazon EventBridge event to notify the company when a message is sent to the DLQ.

B. Create an Amazon SNS topic and an Amazon SQS dead-letter queue (DLQ). Create a subscription and configure a redrive policy for the subscription to send failed messages to the DLQ when the transaction price is above the threshold. Use an Amazon CloudWatch alarm to notify the company when a message is sent to the DLQ.

C. Create an Amazon SQS queue and an event source mapping with an AWS Lambda function. Use the body key in the Amazon SQS message record and a FilterCriteria object to process only the transactions that are above the threshold price. Configure an Amazon CloudWatch alarm to invoke the Lambda function to notify the company.

D. Create an Amazon SNS topic. Include the MessageAttributes field to list attributes that describe the transaction, including the price. Subscribe the company to the SNS topic. Create a topic filter on the subscription for the MessageAttributes field. Use the numeric value range matching policy.

Answer: D. Create an Amazon SNS topic. Include the MessageAttributes field to list attributes that describe the transaction, including the price. Subscribe the company to the SNS topic. Create a topic filter on the subscription for the MessageAttributes field. Use the numeric value range matching policy.

Explanation

This question tests the ability to implement event-driven notification logic by using Amazon SNS subscription filter policies to deliver only transactions whose numeric price attribute meets a specified threshold.

✗ **A** Create an Amazon SQS standard queue and an SQS dead-letter queue (DLQ). Configure a redrive policy for the standard queue to send failed messages to the DLQ based on the price threshold. Use an Amazon EventBridge event to notify the company when a message is sent to the DLQ.
Incorrect. An SQS redrive policy sends messages to a DLQ based on the maximum receive count, not a transaction price threshold. This approach also adds unnecessary queue and DLQ components.

https://docs.aws.amazon.com/AWSSimpleQueueService/latest/SQSDeveloperGuide/sqs-dead-letter-queues.html

✗ **B** Create an Amazon SNS topic and an Amazon SQS dead-letter queue (DLQ). Create a subscription and configure a redrive policy for the subscription to send failed messages to the DLQ when the transaction price is above the threshold. Use an Amazon CloudWatch alarm to notify the company when a message is sent to the DLQ.
Incorrect. An SNS subscription redrive policy handles delivery failures and does not evaluate message prices. A DLQ is intended for undeliverable messages, not for selecting transactions based on message content.

https://docs.aws.amazon.com/sns/latest/dq/sns-configure-dead-letter-queue.html

✗ **C** Create an Amazon SQS queue and an event source mapping with an AWS Lambda function. Use the

Free demo · full exam: 595 questions

body key in the Amazon SQS message record and a FilterCriteria object to process only the transactions that are above the threshold price. Configure an Amazon CloudWatch alarm to invoke the Lambda function to notify the company.
Incorrect. Lambda event filtering can filter SQS records based on message contents, but this solution still requires Lambda processing and a separate notification mechanism. A CloudWatch alarm is not the simplest way to notify based on each matching transaction, so it has more operational overhead than SNS subscription filtering.

https://docs.aws.amazon.com/lambda/latest/dg/with-sqs-filtering.html

✓ **D** Create an Amazon SNS topic. Include the MessageAttributes field to list attributes that describe the transaction, including the price. Subscribe the company to the SNS topic. Create a topic filter on the subscription for the MessageAttributes field. Use the numeric value range matching policy.
Correct. Amazon SNS subscription filter policies can evaluate message attributes, including numeric attributes, and deliver messages only when the price satisfies the configured numeric range. This directly implements threshold-based notifications with the least operational overhead.

https://docs.aws.amazon.com/sns/latest/dg/sns-message-filtering.html

Question 27

Single choice · Domain: D1 Development with AWS Services › Use Amazon S3 Batch Operations to encrypt existing S3 objects with SSE-KMS

A company runs an application that currently uploads content to an unencrypted Amazon S3 bucket.

The S3 bucket contains millions of objects that total more than 2 TB in size. A new company policy requires all data to be encrypted. The company creates an AWS KMS key to use for encryption.

Which solution will meet this requirement MOST cost-effectively?

A. Modify the properties of the S3 bucket. Enable default encryption, select SSE-KMS, and specify the Amazon Resource Name (ARN) of the provided KMS key.

B. Create an S3 bucket policy that denies any PutObject requests that do not include the x-amz-server-side-encryption header that specifies the Amazon Resource Name (ARN) of the provided KMS key. Apply this policy to the bucket.

C. Configure an Amazon S3 Batch Operations job. Choose the source bucket that contains the unencrypted objects. Select the COPY operation. Specify the source bucket as the destination. In the copy operation parameters, specify the Amazon Resource Name (ARN) of the provided KMS key.

D. Develop a script by using the AWS SDK. Configure the script to iterate through all objects in the bucket and download each object locally. Ensure that the script then re-uploads each object to the same object key name and specifies server-side encryption with the KMS key during the re-upload.

Answer: C. Configure an Amazon S3 Batch Operations job. Choose the source bucket that contains the unencrypted objects. Select the COPY operation. Specify the source bucket as the destination. In the copy operation parameters, specify the Amazon Resource Name (ARN) of the provided KMS key.

Explanation

This question tests the ability to apply server-side encryption with AWS KMS to a large existing S3 data set while avoiding unnecessary downloads and local re-uploads. S3 Batch Operations can copy objects in place and apply SSE-KMS encryption at scale.

Free demo · full exam: 595 questions

✗ **A** Modify the properties of the S3 bucket. Enable default encryption, select SSE-KMS, and specify the Amazon Resource Name (ARN) of the provided KMS key.
Incorrect. S3 default encryption automatically encrypts new objects but does not encrypt the millions of objects that already exist in the bucket.

https://docs.aws.amazon.com/AmazonS3/latest/userguide/default-bucket-encryption.html

✗ **B** Create an S3 bucket policy that denies any PutObject requests that do not include the x-amz- server-side-encryption header that specifies the Amazon Resource Name (ARN) of the provided KMS key. Apply this policy to the bucket.
Incorrect. This policy can require future uploads to use SSE-KMS, but it does not encrypt the existing objects in the bucket.

https://docs.aws.amazon.com/AmazonS3/latest/userguide/amazon-s3-policy-keys.html

✓ **C** Configure an Amazon S3 Batch Operations job. Choose the source bucket that contains the unencrypted objects. Select the COPY operation. Specify the source bucket as the destination. In the copy operation parameters, specify the Amazon Resource Name (ARN) of the provided KMS key.
Correct. An S3 Batch Operations COPY job can copy the existing objects within the same bucket and apply SSE-KMS encryption using the specified KMS key. This avoids downloading and re-uploading more than 2 TB through a locally managed script, making it the most cost-effective option for encrypting the existing data at scale.

https://docs.aws.amazon.com/AmazonS3/latest/userguide/batch-ops-copy.html

✗ **D** Develop a script by using the AWS SDK. Configure the script to iterate through all objects in the bucket and download each object locally. Ensure that the script then re-uploads each object to the same object key name and specifies server-side encryption with the KMS key during the re-upload.
Incorrect. Although downloading and re-uploading each object can apply SSE-KMS encryption, transferring more than 2 TB through local infrastructure adds unnecessary data transfer, compute, storage, and operational costs compared with an S3 Batch Operations copy job.

https://docs.aws.amazon.com/AmazonS3/latest/userguide/UsingKMSEncryption.html

Question 28

Single choice · Domain: D1 Development with AWS Services › Configure Amazon SNS subscription filter policies to route only matching messages to a Lambda subscriber

An ecommerce company integrates with a payment processing service that publishes payment activity messages to an Amazon SNS topic. A developer is building an AWS Lambda function to process refund events. The Lambda function must process only refund-related messages.

Which solution will meet this requirement in the MOST operationally efficient way?

A. Configure a Lambda event filter to allow only refund messages to invoke the function.

B. Add logic in the Lambda function to ignore non-refund messages.

C. Use Amazon SNS subscription filter policies to deliver only refund messages to the Lambda function.

D. Increase batch size and batching window parameters for the Lambda function.

Answer: C. Use Amazon SNS subscription filter policies to deliver only refund messages to the Lambda function.

Explanation

Free demo · full exam: 595 questions

This question tests the ability to filter SNS messages before they invoke a Lambda function. SNS subscription filter policies are the most operationally efficient choice because non-refund messages are not delivered to or processed by Lambda.

✗ **A** Configure a Lambda event filter to allow only refund messages to invoke the function.
Incorrect. Lambda event filtering is designed for supported event source mappings and does not provide the appropriate filtering mechanism for messages delivered directly from an Amazon SNS subscription.

https://docs.aws.amazon.com/lambda/latest/dg/invocation-eventfiltering.html

✗ **B** Add logic in the Lambda function to ignore non-refund messages.
Incorrect. Although the function could discard non-refund messages, every message would still invoke Lambda, consuming invocation and compute resources and increasing operational overhead.

https://docs.aws.amazon.com/lambda/latest/dg/with-sns.html

✓ **C** Use Amazon SNS subscription filter policies to deliver only refund messages to the Lambda function.
Correct. An SNS subscription filter policy evaluates message attributes and delivers only matching messages to the subscribed Lambda function. This prevents non-refund messages from invoking Lambda and is the most operationally efficient solution.

https://docs.aws.amazon.com/sns/latest/dg/sns-message-filtering.html

✗ **D** Increase batch size and batching window parameters for the Lambda function.
Incorrect. Batch size and batching window settings control how supported event source mappings batch records; they do not filter SNS messages by event type and would not prevent non-refund messages from invoking the function.

https://docs.aws.amazon.com/lambda/latest/dg/invocation-eventsourcemapping.html

Dumpsgate

Question 29

Single choice · Domain: D1 Development with AWS Services › Design a DynamoDB partition key strategy to distribute read traffic and avoid hot partitions

A company is developing a weather forecast application that displays forecasts for cities within a single country. Weather data is stored in an Amazon DynamoDB table named Forecasts with attributes CityId, Temperature, and ForecastDate.

The application is read heavy. Most read requests target the current day's forecast for a small number of very large cities. The company must design a partition key strategy that ensures the most even distribution of read traffic.

Which solution will meet these requirements?

A. Use CityId with a calculated suffix as the partition key and ForecastDate as the sort key.
B. Use a numeric index as the partition key and ForecastDate as the sort key.
C. Use ForecastDate as the partition key and CityId as the sort key.
D. Use ForecastDate with a calculated suffix as the partition key and CityId as the sort key.

Answer: D. Use ForecastDate with a calculated suffix as the partition key and CityId as the sort key.

Explanation

This question tests the ability to select a DynamoDB key design that distributes heavily concentrated read traffic across multiple partitions. Adding a calculated suffix to the date-based partition key shards

requests for the current date, while CityId remains available as the sort key.

✗ **A** Use CityId with a calculated suffix as the partition key and ForecastDate as the sort key.
Incorrect. A CityId-based key concentrates requests for each large city on the corresponding city partition unless the application deliberately distributes and queries multiple suffixes. It does not directly address the concentration of requests for the current date across a small number of cities.

https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/bp-partition-key-design.html

✗ **B** Use a numeric index as the partition key and ForecastDate as the sort key.
Incorrect. A numeric index is not an appropriate access-pattern-based partition key and may result in uneven or concentrated traffic. DynamoDB partition keys should distribute requests as evenly as possible across partition key values.

https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/bp-partition-key-design.html

✗ **C** Use ForecastDate as the partition key and CityId as the sort key.
Incorrect. Using ForecastDate alone creates a low-cardinality partition key. Most requests for the current date would target the same partition key value, creating a hot partition.

https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/bp-partition-key-design.html

✓ **D** Use ForecastDate with a calculated suffix as the partition key and CityId as the sort key.
Correct. Adding a calculated suffix shards the current date across multiple partition key values, distributing read traffic that would otherwise concentrate on one ForecastDate value. CityId as the sort key supports city-level access within each shard. This is a form of write or read sharding used to mitigate hot partitions.

https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/bp-partition-key-sharding.html

Dumpsgate

Question 30

Single choice · Domain: D1 Development with AWS Services › Configure VPC connectivity for AWS Lambda functions that access private resources and public internet services

A company runs a serverless application that uses several AWS Lambda functions. The existing Lambda functions run in a VPC. The Lambda functions query public APIs successfully.

To add a new feature to the application, a developer creates a new Lambda function to query external public APIs. The new Lambda function must store aggregated results in an Amazon RDS database that is in a private subnet of the VPC. The developer configures VPC access for the new Lambda function and sets up a working connection to the RDS database. The requests that the new Lambda function makes to the external APIs fail. However, requests from the developer's local workstation to the same APIs are successful.

Which solution will meet this requirement?

A. Provision an elastic network interface for the new Lambda function.
B. Provision a NAT gateway in a public subnet in the VPC.
C. Provision an outbound rule for the new Lambda function's security group to grant internet access.
D. Provision a gateway VPC endpoint in a public subnet in the VPC.

Answer: B. Provision a NAT gateway in a public subnet in the VPC.

Explanation

This question tests how to provide a VPC-connected Lambda function with access to both a private

Amazon RDS database and external public APIs. The function requires a route from its private subnet through a NAT gateway in a public subnet.

✗ **A** Provision an elastic network interface for the new Lambda function.
Incorrect. AWS Lambda creates and manages the elastic network interfaces required for VPC connectivity. An elastic network interface alone does not provide a route to the public internet.

https://docs.aws.amazon.com/lambda/latest/dg/configuration-vpc.html

✓ **B** Provision a NAT gateway in a public subnet in the VPC.
Correct. A Lambda function attached to private subnets can access external public APIs when those subnets have a route to a NAT gateway in a public subnet, and the NAT gateway has a route to an internet gateway.

https://docs.aws.amazon.com/vpc/latest/userguide/vpc-nat-gateway.html

✗ **C** Provision an outbound rule for the new Lambda function's security group to grant internet access.
Incorrect. An outbound security group rule permits traffic at the virtual firewall, but it does not provide the required network route or public source address for internet access from a private subnet. A NAT gateway and appropriate route table configuration are also required.

https://docs.aws.amazon.com/vpc/latest/userguide/vpc-security-groups.html

✗ **D** Provision a gateway VPC endpoint in a public subnet in the VPC.
Incorrect. Gateway VPC endpoints provide private connectivity to supported AWS services such as Amazon S3 and DynamoDB; they do not provide general access to external public APIs. Gateway endpoints are also associated with route tables rather than provisioned in a public subnet.

https://docs.aws.amazon.com/vpc/latest/privatelink/vpc-endpoints.html

Question 31

Single choice · Domain: D1 Development with AWS Services › Configure Amazon API Gateway stages and deployments to manage REST API environments and promote versions

A company wants to launch a REST API across multiple environments. The company needs to configure staging, testing, and production environments. The configuration must provide built-in support for version control. The team must be able to promote the test environment to the production environment when required. Which solution will meet these requirements in the MOST operationally efficient way?

A. Use Amazon API Gateway to create a REST API. Set up an API method for each environment. Deploy the REST API. Associate the REST API deployment with the appropriate method.

B. Create a REST API for each environment in Amazon API Gateway. Deploy the APIs and distribute the URL of the API to the corresponding users.

C. Identify the API methods required for the configuration. Create an AWS Lambda function for each required API method. Use Lambda canary deployments with a weighted alias for new releases and version control.

D. Use Amazon API Gateway to create a REST API. Set up an API method for each environment. Deploy the REST API and associate the deployment with the appropriate stage.

Answer: D. Use Amazon API Gateway to create a REST API. Set up an API method for each environment. Deploy the REST API and associate the deployment with the appropriate stage.

Explanation

This question tests the ability to use API Gateway stages to represent environments such as staging, testing, and production. Stages reference deployments, which provide versioned snapshots that can be promoted between environments by updating the stage deployment.

✗ **A** Use Amazon API Gateway to create a REST API. Set up an API method for each environment. Deploy the REST API. Associate the REST API deployment with the appropriate method.
Incorrect. Environments should be represented by API Gateway stages, not by separate API methods. Deployments are associated with stages rather than individual methods.

https://docs.aws.amazon.com/apigateway/latest/developerguide/set-up-deployments.html

✗ **B** Create a REST API for each environment in Amazon API Gateway. Deploy the APIs and distribute the URL of the API to the corresponding users.
Incorrect. Creating separate REST APIs for each environment increases operational overhead and does not use API Gateway's built-in stage and deployment model for version promotion.

https://docs.aws.amazon.com/apigateway/latest/developerguide/stages.html

✗ **C** Identify the API methods required for the configuration. Create an AWS Lambda function for each required API method. Use Lambda canary deployments with a weighted alias for new releases and version control.
Incorrect. Lambda versions, aliases, and weighted canary deployments manage Lambda function releases, but they do not provide the required environment management for the REST API itself. API Gateway stages and deployments are the appropriate mechanism.

https://docs.aws.amazon.com/lambda/latest/dg/configuration-aliases.html

✓ **D** Use Amazon API Gateway to create a REST API. Set up an API method for each environment. Deploy the REST API and associate the deployment with the appropriate stage.
Correct. API Gateway stages can represent testing, staging, and production environments. Each stage points to a deployment, which is a versioned snapshot of the API. The team can promote a tested deployment by associating it with the production stage.

https://docs.aws.amazon.com/apigateway/latest/developerguide/stages.html

Question 32

Single choice · Domain: D1 Development with AWS Services › Use AWS AppConfig feature flags to control the availability of application features

A developer is working on a web application that requires selective activation of specific features.

The developer wants to keep the features hidden from end users until the features are ready for public access. Which solution will meet these requirements?

A. Create a feature flag configuration profile in AWS AppSync. Store the feature flag values in the configuration profile. Activate and deactivate feature flags as needed.

B. Store prerelease data in an Amazon DynamoDB table. Enable Amazon DynamoDB Streams in the table. Toggle between hidden and visible states by using DynamoDB Streams.

C. Create a feature flag configuration profile in AWS AppConfig. Store the feature flag values in the configuration profile. Activate and deactivate feature flags as needed.

D. Store prerelease data in AWS Amplify DataStore. Toggle between hidden and visible states by using Amplify DataStore cloud synchronization.

Answer: C. Create a feature flag configuration profile in AWS AppConfig. Store the feature flag values in the configuration profile. Activate and deactivate feature flags as needed.

Free demo · full exam: 595 questions

Dumpsgate

Question 33

Single choice · Domain: D1 Development with AWS Services › Use Amazon Q Developer to generate unit tests for application code

A developer is building an application and wants to quickly test the effectiveness of a function before committing the code. Which solution will meet this requirement with the LEAST effort?

A. Add a report group that has test case information to the buildspec file of an AWS CodeBuild project.

B. Highlight the function within the application code. Use the Amazon Q Developer agent to generate a unit test.

C. Add a report group that has test case information to the appspec file of an AWS CodeBuild project.

D. Use Amazon Q Developer to review the code. Use the results of the review as context in a follow-up prompt to generate a unit test.

Answer: B. Highlight the function within the application code. Use the Amazon Q Developer agent to generate a unit test.

Explanation

Free demo · full exam: 595 questions

This question tests the ability to use Amazon Q Developer in an IDE to quickly generate a unit test for a selected function with minimal developer effort.

✗ **A** Add a report group that has test case information to the buildspec file of an AWS CodeBuild project.
Incorrect. CodeBuild report groups configure how test results are collected and reported after tests run; they do not generate a unit test for a selected function.

https://docs.aws.amazon.com/codebuild/latest/userguide/test-reporting.html

✓ **B** Highlight the function within the application code. Use the Amazon Q Developer agent to generate a unit test.
Correct. Amazon Q Developer can generate unit tests for selected application code directly in supported development environments, making this the solution requiring the least effort.

https://docs.aws.amazon.com/amazonq/latest/qdeveloper-ug/test-generation.html

✗ **C** Add a report group that has test case information to the appspec file of an AWS CodeBuild project.
Incorrect. An AppSpec file is associated with AWS CodeDeploy deployments, not CodeBuild test-result configuration. In addition, configuring a report group would not generate a unit test.

https://docs.aws.amazon.com/codedeploy/latest/userguide/reference-appspec-file.html

✗ **D** Use Amazon Q Developer to review the code. Use the results of the review as context in a follow- up prompt to generate a unit test.
Incorrect. Amazon Q Developer can assist with code reviews and test generation, but performing a separate review and then using its results in a follow-up prompt requires more steps than directly selecting the function and requesting a unit test.

https://docs.aws.amazon.com/amazonq/latest/qdeveloper-ug/test-generation.html

Dumpsgate

Question 34

Single choice · Domain: D1 Development with AWS Services › Implement envelope encryption for large client-side encrypted data using AWS KMS data keys

A developer is working on an application that handles 10 MB documents that contain highly sensitive data. The application will use AWS KMS to perform client-side encryption.

What steps must be followed?

A. Invoke the Encrypt API, passing the plaintext data that must be encrypted, then reference the customer managed key ARN in the KeyId parameter.

B. Invoke the GenerateRandom API to get a data encryption key, then use the data encryption key to encrypt the data.

C. Invoke the GenerateDataKey API to retrieve the encrypted version of the data encryption key to encrypt the data.

D. Invoke the GenerateDataKey API to retrieve the plaintext version of the data encryption key to encrypt the data.

Answer: D. Invoke the GenerateDataKey API to retrieve the plaintext version of the data encryption key to encrypt the data.

Explanation

This question tests selecting the AWS KMS data-key workflow for encrypting data larger than the KMS direct-encryption plaintext limit. The application must use the plaintext data key locally for encryption and

retain the encrypted data key for later decryption.

✗ **A** Invoke the Encrypt API, passing the plaintext data that must be encrypted, then reference the customer managed key ARN in the KeyId parameter.
Incorrect. The KMS Encrypt API is intended for plaintext data up to 4,096 bytes, so it cannot directly encrypt a 10 MB document. Large data should be encrypted locally with a data key generated by KMS.
https://docs.aws.amazon.com/kms/latest/APIReference/API_Encrypt.html

✗ **B** Invoke the GenerateRandom API to get a data encryption key, then use the data encryption key to encrypt the data.
Incorrect. GenerateRandom returns random bytes and does not create a KMS-encrypted data key. GenerateDataKey returns a plaintext data key for local encryption together with an encrypted copy protected by the KMS key.
https://docs.aws.amazon.com/kms/latest/APIReference/API_GenerateRandom.html

✗ **C** Invoke the GenerateDataKey API to retrieve the encrypted version of the data encryption key to encrypt the data.
Incorrect. The encrypted data key is not used directly to encrypt the document. It is stored with the ciphertext and later decrypted by KMS. The plaintext version of the data key is used locally for encryption.
https://docs.aws.amazon.com/kms/latest/APIReference/API_GenerateDataKey.html

✓ **D** Invoke the GenerateDataKey API to retrieve the plaintext version of the data encryption key to encrypt the data.
Correct. For envelope encryption, GenerateDataKey returns both a plaintext data key and an encrypted copy of that key. The application uses the plaintext data key with a local cryptographic library to encrypt the 10 MB document, then stores the encrypted data key with the encrypted document and securely erases the plaintext key.
https://docs.aws.amazon.com/kms/latest/developerguide/concepts.html#enveloping

Question 35

Multiple choice · Domain: D1 Development with AWS Services › Select DynamoDB capacity management strategies to handle write traffic and prevent throttling

A development team uses an Amazon DynamoDB table as a database for an application. The team notices errors and slowdowns in the application during peak usage hours. The slowdowns and errors occur during a surge of user logins. The application receives frequent write requests. Application logs indicate that write requests are being throttled.

The development team needs to reduce the application latency and resolve the throttling errors.

Which solutions will meet these requirements? (Select TWO.)

A. Create a DynamoDB Accelerator (DAX) cluster. Update the application to send read requests to the DAX endpoint.

B. Increase the provisioned throughput of the table.

C. Reduce the frequency of write requests by using error retries and exponential backoff.

D. Reduce control plane operations that occur during peak usage hours by consolidating the DynamoDB tables and indexes.

E. Change the table's capacity mode to on-demand.

Answer

- B. **Increase the provisioned throughput of the table.**
- E. **Change the table's capacity mode to on-demand.**

Free demo · full exam: 595 questions

Dumpsgate

Question 36

Single choice · Domain: D1 Development with AWS Services › Configure Amazon S3 Lifecycle expiration rules for objects matching a prefix

A developer manages an application that stores user objects in an Amazon S3 bucket without versioning enabled. The application has premium users and basic users.

After premium users upload objects, the premium users have unlimited downloads of their objects.

Their objects are stored with a premium/ prefix. After basic users upload objects, the basic users can download their objects for 90 days. Their objects are stored with a basic/ prefix.

The developer needs to implement a solution to automatically delete objects for the basic users after 90 days.

Which solution will meet these requirements with the LEAST development effort?

A. Create an AWS Lambda function that removes any objects in the S3 bucket that have the basic/ prefix and are more than 90 days old. Set up an Amazon EventBridge schedule to invoke the Lambda function every day.

B. Set up an S3 Lifecycle rule that applies to the objects that have the premium/ prefix. Set the S3 Lifecycle rule action to expire the current version of the objects that have the premium/ prefix after 90 days.

C. Set up an S3 Lifecycle rule that applies to the objects that have the basic/ prefix. Set the S3 Lifecycle rule action to expire the current version of the objects that have the basic/ prefix after 90 days.

D. Create a rule for the S3 bucket to identify objects that have the basic/ prefix. Set the rule action to delete any objects that have object delete markers and unfinished multipart uploads after 90 days.

Answer: C. Set up an S3 Lifecycle rule that applies to the objects that have the basic/ prefix. Set the S3 Lifecycle rule action to expire the current version of the objects that have the basic/ prefix after 90 days.

Explanation

This question tests the ability to use an S3 Lifecycle rule to automatically expire objects based on their key prefix and age. A prefix-filtered expiration rule provides the required 90-day deletion with less development effort than implementing scheduled Lambda code.

✗ **A** Create an AWS Lambda function that removes any objects in the S3 bucket that have the basic/ prefix and are more than 90 days old. Set up an Amazon EventBridge schedule to invoke the Lambda function every day.
This could delete the required objects, but it requires custom Lambda code, scheduling, permissions, and ongoing operation. S3 Lifecycle rules provide a managed alternative with less development effort.

https://docs.aws.amazon.com/AmazonS3/latest/userguide/object-lifecycle-mgmt.html

✗ **B** Set up an S3 Lifecycle rule that applies to the objects that have the premium/ prefix. Set the S3 Lifecycle rule action to expire the current version of the objects that have the premium/ prefix after 90 days.
This applies expiration to premium objects, which must have unlimited downloads and therefore must not be deleted after 90 days. The rule must target the basic/ prefix.

https://docs.aws.amazon.com/AmazonS3/latest/userguide/lifecycle-expire-general-considerations.html

✓ **C** Set up an S3 Lifecycle rule that applies to the objects that have the basic/ prefix. Set the S3 Lifecycle rule action to expire the current version of the objects that have the basic/ prefix after 90 days.

Free demo · full exam: 595 questions

Correct. An S3 Lifecycle rule can filter objects by the basic/ prefix and expire the current version after 90 days. Because versioning is not enabled, expiration deletes the objects, meeting the requirement without custom application code.

https://docs.aws.amazon.com/AmazonS3/latest/userguide/object-lifecycle-mgmt.html

✗ **D** Create a rule for the S3 bucket to identify objects that have the basic/ prefix. Set the rule action to delete any objects that have object delete markers and unfinished multipart uploads after 90 days.
This does not configure expiration of the basic/ objects themselves. Delete-marker expiration and incomplete multipart-upload cleanup address different lifecycle items and do not meet the requirement to delete basic-user objects after 90 days.

https://docs.aws.amazon.com/AmazonS3/latest/userguide/mpu-abort-incomplete-mpu-lifecycle-config.html

---

Question 37

Single choice · Domain: D1 Development with AWS Services › Configure an AWS Lambda function timeout to accommodate longer-running processing tasks

A developer is building an image-processing application that includes an AWS Lambda function. The Lambda function moves images from one AWS service to another AWS service for image processing.

For images that are larger than 2 MB, the Lambda function returns the following error: "Task timed out after 3.01 seconds." The developer needs to resolve the error without modifying the Lambda function code.

Which solution will meet these requirements?

A. Increase the Lambda function’s timeout value.
B. Configure the Lambda function to not move images that are larger than 2 MB.
C. Request a concurrency quota increase for the Lambda function.
D. Configure provisioned concurrency for the Lambda function.

Answer: A. Increase the Lambda function’s timeout value.

Explanation

The question tests the ability to identify a Lambda execution timeout and select the appropriate configuration change without modifying application code. Increasing the function timeout allows image transfers and processing that require more than the current three-second limit to complete.

✓ **A** Increase the Lambda function’s timeout value.
Correct. The error indicates that the function exceeded its configured three-second timeout. Increasing the Lambda timeout allows the image transfer and processing to run longer without changing the code.

https://docs.aws.amazon.com/lambda/latest/dg/configuration-timeout.html

✗ **B** Configure the Lambda function to not move images that are larger than 2 MB.
Incorrect. This avoids processing some images rather than resolving the timeout problem, and it does not meet the requirement to process larger images. Lambda's timeout can be configured to allow longer executions.

https://docs.aws.amazon.com/lambda/latest/dg/configuration-timeout.html

✗ **C** Request a concurrency quota increase for the Lambda function.
Incorrect. Concurrency quotas control how many function invocations can run simultaneously; they do not increase the maximum execution duration of an invocation. The execution timeout must be increased

instead.

https://docs.aws.amazon.com/lambda/latest/dg/lambda-concurrency.html

✗ **D** Configure provisioned concurrency for the Lambda function.
Incorrect. Provisioned concurrency keeps execution environments initialized to reduce cold-start latency, but it does not extend the function's execution timeout.

https://docs.aws.amazon.com/lambda/latest/dg/provisioned-concurrency.html

Question 38

Single choice · Domain: D1 Development with AWS Services › Use Amazon Q Developer Pro in an IDE to generate unit tests and perform automated code reviews

A company is building a serverless application composed of multiple AWS Lambda functions. The company wants to accelerate feature development without adding staff or reducing quality. The solution must improve unit tests and code reviews, integrate with the existing IDE, and require no new infrastructure.

Which solution will meet these requirements?

A. Use AWS CodeBuild with custom scripts for unit tests and Amazon CodeGuru Reviewer for code reviews.

B. Use Amazon CodeGuru for code reviews and a custom Lambda-based solution for unit tests.

C. Use AWS CodeBuild for tests and rely on manual pull request reviews.

D. Use Amazon Q Developer Pro to generate unit tests and perform code reviews directly in the IDE.

Answer: D. Use Amazon Q Developer Pro to generate unit tests and perform code reviews directly in the IDE.

Explanation

This question tests the ability to select an AWS developer productivity tool that integrates directly with existing IDEs and supports both unit-test generation and code reviews without requiring additional infrastructure.

✗ **A** Use AWS CodeBuild with custom scripts for unit tests and Amazon CodeGuru Reviewer for code reviews.
Incorrect. Although CodeBuild can run unit tests and CodeGuru Reviewer can analyze code, this combination does not provide the required direct IDE-based workflow for generating tests and performing reviews.

https://aws.amazon.com/codeguru/

✗ **B** Use Amazon CodeGuru for code reviews and a custom Lambda-based solution for unit tests.
Incorrect. A custom Lambda-based testing solution adds development and maintenance effort, does not inherently integrate with the existing IDE, and does not provide the unified productivity features required.

https://aws.amazon.com/q/developer/

✗ **C** Use AWS CodeBuild for tests and rely on manual pull request reviews.
Incorrect. CodeBuild can automate test execution, but manual reviews do not accelerate code reviews through AI assistance or provide direct IDE integration.

https://aws.amazon.com/codebuild/

✓ **D** Use Amazon Q Developer Pro to generate unit tests and perform code reviews directly in the IDE.

Free demo · full exam: 595 questions

Correct. Amazon Q Developer integrates with supported IDEs and can assist developers by generating unit tests and performing automated code reviews, improving development speed without requiring new application infrastructure.
https://aws.amazon.com/q/developer/

Question 39

Single choice · Domain: D1 Development with AWS Services › Use Amazon CloudWatch Logs Live Tail with a filter pattern to troubleshoot an AWS Lambda function in near real time

An ecommerce company uses a set of AWS Lambda functions to process orders. The Lambda functions send logs to an Amazon CloudWatch Logs log group. The company observes timeout issues for one recently deployed processing function. The company needs to debug and identify the root cause of the timeout issue. The Lambda function is already in production. The company wants to have a live feed of filtered logs that start with the word "ERROR" to identify the root cause of the issue. The company wants to review only relevant log lines in near real time. Which solution will meet these requirements?

A. Run an Amazon CloudWatch Logs Insights query with a filter expression after the function finishes running.
B. Create an Amazon CloudWatch Logs subscription filter to a new log group. Apply a metric filter for "ERROR."
C. Use the live tail feature in the Lambda console with a filter for "ERROR."
D. Use Amazon Athena to query the log data that is stored in the Amazon CloudWatch log group.

Answer: C. Use the live tail feature in the Lambda console with a filter for "ERROR."

Explanation

This question tests the ability to select the appropriate CloudWatch Logs capability for viewing filtered Lambda logs as they are generated. Live Tail is designed for near-real-time log monitoring and supports filtering relevant log events.

✗ **A** Run an Amazon CloudWatch Logs Insights query with a filter expression after the function finishes running.
Incorrect. CloudWatch Logs Insights is useful for querying and analyzing logs after they have been ingested, but running a query after the function finishes does not provide a live near-real-time feed.
https://docs.aws.amazon.com/AmazonCloudWatch/latest/logs/AnalyzingLogData.html

✗ **B** Create an Amazon CloudWatch Logs subscription filter to a new log group. Apply a metric filter for "ERROR."
Incorrect. Subscription filters deliver log events to supported destinations such as Amazon Kinesis, Lambda, or Amazon Data Firehose; they do not directly create a live filtered view in a new log group. Metric filters extract metric observations from logs rather than displaying the relevant log lines.
https://docs.aws.amazon.com/AmazonCloudWatch/latest/logs/SubscriptionFilters.html

✓ **C** Use the live tail feature in the Lambda console with a filter for "ERROR."
Correct. CloudWatch Logs Live Tail, available from the Lambda console, displays log events in near real time and allows filtering so that only relevant events, such as messages containing or beginning with ERROR according to the filter pattern, are shown.
https://docs.aws.amazon.com/AmazonCloudWatch/latest/logs/CloudWatchLogs_LiveTail.html

✗ **D** Use Amazon Athena to query the log data that is stored in the Amazon CloudWatch log group.
Incorrect. Amazon Athena queries data in supported sources such as Amazon S3; it does not directly provide a near-real-time query interface for CloudWatch Logs log groups. CloudWatch Logs must first be exported or delivered to an appropriate data store.
https://docs.aws.amazon.com/athena/latest/ug/what-is.html

Question 40

Single choice · Domain: D1 Development with AWS Services › Configure timeout settings for an Amazon API Gateway integration with AWS Lambda

An application uses an Amazon API Gateway API that is integrated with an AWS Lambda function.

Specific API requests are returning the HTTP 504 (Gateway Timeout) error message. The Lambda function is configured with a 20-second timeout. The API Gateway integration is configured with a 15-second timeout and a rate of 10 requests every second. There are no errors in the logs for the Lambda function. Which solution will prevent the HTTP 504 error responses?

A. Increase the reserved concurrency of the Lambda function.
B. Increase the timeout of the Lambda function.
C. Increase the timeout of the API Gateway integration.
D. Increase the per-client throttling limit.

Answer: C. Increase the timeout of the API Gateway integration.

Explanation

This question tests the ability to identify and resolve a timeout mismatch between API Gateway and the Lambda function. API Gateway's integration timeout is 15 seconds, which is shorter than the Lambda timeout, so API Gateway can return HTTP 504 before Lambda finishes processing.

✗ **A** Increase the reserved concurrency of the Lambda function.
Incorrect. Reserved concurrency controls the maximum number of concurrent Lambda invocations and does not increase the amount of time API Gateway waits for a response. A concurrency problem would more likely result in throttling or invocation errors, not resolve an API Gateway integration timeout.
https://docs.aws.amazon.com/lambda/latest/dg/configuration-concurrency.html

✗ **B** Increase the timeout of the Lambda function.
Incorrect. The Lambda timeout is already 20 seconds, which is longer than the API Gateway integration timeout of 15 seconds. Increasing the Lambda timeout does not cause API Gateway to wait longer for the response.
https://docs.aws.amazon.com/lambda/latest/dg/configuration-timeout.html

✓ **C** Increase the timeout of the API Gateway integration.
Correct. API Gateway is configured to wait only 15 seconds, while the Lambda function can run for up to 20 seconds. Increasing the API Gateway integration timeout, within the applicable API Gateway limit, allows the request to wait long enough for the Lambda function to respond and prevents the 504 caused by the timeout mismatch.
https://docs.aws.amazon.com/apigateway/latest/developerguide/api-gateway-execution-service-limits-table.html

✗ **D** Increase the per-client throttling limit.
Incorrect. Throttling limits control request rates and are intended to prevent excessive traffic. Exceeding a throttling limit typically produces a 429 Too Many Requests response; it does not increase the time API

Free demo · full exam: 595 questions

Gateway waits for Lambda to complete.
https://docs.aws.amazon.com/apigateway/latest/developerguide/api-gateway-request-throttling.html

Question 41

Single choice · Domain: D1 Development with AWS Services › Configure an API Gateway mock integration with predefined HTTP status codes and JSON responses

A developer is building an application that will use an Amazon API Gateway API with an AWS Lambda backend. The team that will develop the frontend requires immediate access to the API endpoints to build the UI. To prepare the backend application for integration, the developer needs to set up endpoints. The endpoints need to return predefined HTTP status codes and JSON responses for the frontend team. The developer creates a method for an API resource.

Which solution will meet these requirements?

A. Set the integration type to AWS_PROXY. Provision Lambda functions to return hardcoded JSON data.

B. Set the integration type to MOCK. Configure the method's integration request and integration response to associate JSON responses with specific HTTP status codes.

C. Set the integration type to HTTP PROXY. Configure API Gateway to pass all requests to an external placeholder API, which the team will build.

D. Set the integration type to MOCK. Use a method request to define HTTP status codes. Use an integration request to define JSON responses.

Answer: B. Set the integration type to MOCK. Configure the method's integration request and integration response to associate JSON responses with specific HTTP status codes.

Explanation

This question tests the ability to configure API Gateway mock integrations for frontend development before the Lambda backend is implemented. It specifically assesses the use of integration requests and integration responses to return predefined responses.

✗ **A** Set the integration type to AWS_PROXY. Provision Lambda functions to return hardcoded JSON data.
Incorrect. An AWS_PROXY integration invokes a Lambda function and requires the Lambda backend to generate the response. A mock integration is designed to return predefined responses without invoking a backend.

https://docs.aws.amazon.com/apigateway/latest/developerguide/how-to-mock-integration.html

✓ **B** Set the integration type to MOCK. Configure the method's integration request and integration response to associate JSON responses with specific HTTP status codes.
Correct. A MOCK integration allows API Gateway to return predefined responses without invoking a backend. The integration request can provide the data used for request mapping, and integration responses can map selected outcomes to HTTP status codes and JSON response bodies.

https://docs.aws.amazon.com/apigateway/latest/developerguide/how-to-mock-integration.html

✗ **C** Set the integration type to HTTP PROXY. Configure API Gateway to pass all requests to an external placeholder API, which the team will build.
Incorrect. An HTTP proxy integration forwards requests to an existing HTTP endpoint. It does not provide predefined responses independently of a backend, and the placeholder API does not yet exist. A MOCK

Free demo · full exam: 595 questions

integration is the appropriate choice.

https://docs.aws.amazon.com/apigateway/latest/developerguide/api-gateway-api-integration-types.html

✗ **D** Set the integration type to MOCK. Use a method request to define HTTP status codes. Use an integration request to define JSON responses.
Incorrect. Although MOCK is the correct integration type, the configuration roles are reversed. Integration responses define the returned status codes and response bodies, while the integration request is used to configure the mock response selection or mapping. Method requests define how API Gateway validates incoming requests, not the response status codes or JSON response bodies.

https://docs.aws.amazon.com/apigateway/latest/developerguide/how-to-mock-integration.html

---

Question 42

Single choice · Domain: D1 Development with AWS Services › Select and implement an appropriate caching strategy and TTL for a read-heavy application

A company has a two-tier application that consists of a frontend application and an Amazon RDS for MySQL database. The frontend application retrieves data from the database. The data consists of postal codes and their associated geographic coordinates. The application performs many more database reads than writes. The application retrieves each postal code from the database thousands of times each day.

A developer must improve the performance of the application to decrease latency.

Which change should the developer make to the application to meet these requirements?

A. Modify the application to use a lazy loading caching strategy with a small TTL value.
B. Modify the application to use a write-behind caching strategy.
C. Modify the application to use a write-through caching strategy.
D. Modify the application to use a lazy loading caching strategy with a large TTL value.

Answer: D. Modify the application to use a lazy loading caching strategy with a large TTL value.

Explanation

This question tests the ability to improve application latency by caching frequently read, infrequently changed database data. A lazy loading strategy with a large TTL reduces repeated database reads while allowing data to be loaded into the cache on demand.

✗ **A** Modify the application to use a lazy loading caching strategy with a small TTL value.
Incorrect. Lazy loading retrieves data from the database on a cache miss, but a small TTL causes frequently accessed postal code data to expire often, resulting in unnecessary database reads and reduced cache effectiveness.

https://docs.aws.amazon.com/AmazonElastiCache/latest/dg/Strategies.html

✗ **B** Modify the application to use a write-behind caching strategy.
Incorrect. Write-behind caching is intended to improve write performance by writing updates to the cache first and asynchronously persisting them to the database. This application is read-heavy and the requirement is to reduce read latency.

https://docs.aws.amazon.com/AmazonElastiCache/latest/dg/Strategies.html

✗ **C** Modify the application to use a write-through caching strategy.
Incorrect. Write-through caching updates the cache whenever data is written to the database, which is

Free demo · full exam: 595 questions

useful for keeping cached data current after writes. It does not best address this application's repeated reads of relatively stable data because the data can be loaded on demand and retained with a suitable TTL.

https://docs.aws.amazon.com/AmazonElastiCache/latest/dg/Strategies.html

✓ **D** Modify the application to use a lazy loading caching strategy with a large TTL value.
Correct. Lazy loading retrieves a postal code from the database only when it is not already cached. A large TTL keeps frequently requested postal code and coordinate data in the cache longer, reducing repeated database reads and decreasing latency.

https://docs.aws.amazon.com/AmazonElastiCache/latest/dg/Strategies.html

---

Question 43

Single choice · Domain: D1 Development with AWS Services › Select and configure Amazon S3 storage classes and lifecycle transitions based on access frequency, retrieval latency, and cost requirements

A developer is building a video search application. Video files average 2.5 TB in size. Files must have instant access for the first 90 days. After 90 days, files can take more than 10 minutes to load.

Which solution will meet these requirements in the MOST cost-effective way?

A. Store files in Amazon EFS Standard, then transition to EFS Standard-IA.

B. Store files in Amazon S3 Glacier Deep Archive for 90 days, then transition to S3 Glacier Flexible Retrieval.

C. Store files in Amazon EBS for 90 days, then transition to S3 Glacier Deep Archive.

D. Store files in Amazon S3 Glacier Instant Retrieval for 90 days, then transition to S3 Glacier Flexible Retrieval.

Answer: D. Store files in Amazon S3 Glacier Instant Retrieval for 90 days, then transition to S3 Glacier Flexible Retrieval.

Explanation

This question tests the ability to match S3 storage classes to an initial period requiring instant access and a later period where retrieval can take several minutes. It also tests using an S3 lifecycle rule to transition objects between storage classes.

✗ **A** Store files in Amazon EFS Standard, then transition to EFS Standard-IA.
Incorrect. Amazon EFS is a file system service and is not the most cost-effective choice for multi-terabyte video objects that can tolerate delayed retrieval after 90 days. EFS Standard-IA still provides file-system access and does not provide the archival cost and retrieval characteristics of S3 Glacier Flexible Retrieval.

https://docs.aws.amazon.com/efs/latest/ug/storage-classes.html

✗ **B** Store files in Amazon S3 Glacier Deep Archive for 90 days, then transition to S3 Glacier Flexible Retrieval.
Incorrect. S3 Glacier Deep Archive does not provide instant access during the first 90 days; it is intended for long-term archival data with retrieval times measured in hours. Transitioning later to Glacier Flexible Retrieval also reverses the intended cost and access pattern.

https://aws.amazon.com/s3/storage-classes/

✗ **C** Store files in Amazon EBS for 90 days, then transition to S3 Glacier Deep Archive.
Incorrect. Amazon EBS is block storage for EC2 and does not provide a direct lifecycle transition from EBS

Free demo · full exam: 595 questions

volumes to S3 Glacier Deep Archive. The files would need to be copied or exported to Amazon S3, and Deep Archive would not meet the first 90 days requirement for instant access.

https://docs.aws.amazon.com/ebs/latest/userguide/ebs-volumes.html

✓ **D** Store files in Amazon S3 Glacier Instant Retrieval for 90 days, then transition to S3 Glacier Flexible Retrieval.
Correct. S3 Glacier Instant Retrieval is designed for data that requires immediate access, satisfying the first 90 days. After 90 days, an S3 lifecycle rule can transition the objects to S3 Glacier Flexible Retrieval, which has lower storage cost and retrieval times ranging from minutes to hours, matching the requirement that retrieval can take more than 10 minutes. and.

https://aws.amazon.com/s3/storage-classes/
https://docs.aws.amazon.com/AmazonS3/latest/userguide/lifecycle-transition-general-considerations.html

---

Question 44

Single choice · Domain: D1 Development with AWS Services › Configure AWS Amplify feature branch deployments from a Git repository

A developer is building a web and mobile application and needs a solution to deploy the application code. The solution must be compatible with the developer's Git source control repository. When the developer adds a new branch, the solution must create a separate deployment.

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

A. Push the code to an Amazon ECR repository. Deploy the application on Amazon ECS. Set up a GitHub Actions workflow to create new branches.

B. Use AWS Elastic Beanstalk to deploy the application. Create new branches by uploading the application's source bundle code to create new application versions.

C. Deploy the application code to an AWS Lambda function. Publish a new version of the Lambda function and point an alias to the new version. Create a new branch in GitHub that connects to the Lambda alias.

D. Use AWS Amplify to deploy the application. Use feature branch deployments to connect and manage Git branches.

Answer: D. Use AWS Amplify to deploy the application. Use feature branch deployments to connect and manage Git branches.

Explanation

This question tests the ability to select a managed deployment solution that integrates with Git repositories and automatically creates separate environments for new branches. AWS Amplify provides feature branch deployments with less operational overhead than manually configuring container, Elastic Beanstalk, or Lambda deployments.

✗ **A** Push the code to an Amazon ECR repository. Deploy the application on Amazon ECS. Set up a GitHub Actions workflow to create new branches.
Incorrect. Amazon ECR and Amazon ECS can deploy containerized applications, but this approach requires container image management, ECS configuration, and a separate GitHub Actions workflow. It does not provide the integrated feature branch deployment capability with the least operational overhead.

https://docs.aws.amazon.com/AmazonECS/latest/developerguide/Welcome.html

Free demo · full exam: 595 questions

✗ **B** Use AWS Elastic Beanstalk to deploy the application. Create new branches by uploading the application's source bundle code to create new application versions.
Incorrect. Elastic Beanstalk supports application versions, but creating versions by uploading source bundles is a manual process and does not automatically create separate deployments when Git branches are added.

https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/using-features.deploy-existing-version.html

✗ **C** Deploy the application code to an AWS Lambda function. Publish a new version of the Lambda function and point an alias to the new version. Create a new branch in GitHub that connects to the Lambda alias.
Incorrect. Lambda versions and aliases manage function versions and traffic routing, but they do not automatically map Git branches to separate deployments. This option requires additional deployment automation and is not the lowest-overhead solution for a web and mobile application.

https://docs.aws.amazon.com/lambda/latest/dg/configuration-versions.html

✓ **D** Use AWS Amplify to deploy the application. Use feature branch deployments to connect and manage Git branches.
Correct. AWS Amplify integrates with Git repositories and supports feature branch deployments, allowing each connected branch to have its own deployment with minimal operational effort.

https://docs.aws.amazon.com/amplify/latest/userguide/feature-branch-deployments.html

---

Question 45

Single choice · Domain: D1 Development with AWS Services › Generate temporary presigned URLs to securely serve private Amazon S3 objects

A company is developing a web application that allows its employees to upload a profile picture to a private Amazon S3 bucket. There is no size limit for the profile pictures, which should be displayed every time an employee logs in. For security reasons, the pictures cannot be publicly accessible.

What is a viable long-term solution for this scenario?

A. Generate a presigned URL when a picture is uploaded. Save the URL in an Amazon DynamoDB table. Return the URL to the browser when the employee logs in.

B. Save the picture's S3 key in an Amazon DynamoDB table. Create an Amazon S3 VPC endpoint to allow the employees to download pictures once they log in.

C. Encode a picture using base64. Save the base64 string in an Amazon DynamoDB table. Allow the browser to retrieve the string and convert it to a picture.

D. Save the picture's S3 key in an Amazon DynamoDB table. Use a function to generate a presigned URL every time an employee logs in. Return the URL to the browser.

Answer: D. Save the picture's S3 key in an Amazon DynamoDB table. Use a function to generate a presigned URL every time an employee logs in. Return the URL to the browser.

Explanation

This question tests the ability to design an application that stores S3 object references and generates time-limited access for authenticated users without making the objects public.

✗ **A** Generate a presigned URL when a picture is uploaded. Save the URL in an Amazon DynamoDB table. Return the URL to the browser when the employee logs in.
Incorrect. A presigned URL has a limited expiration period, so storing it for long-term reuse can cause the

Free demo · full exam: 595 questions

URL to expire before the employee logs in. The application should store the S3 object key and generate a new presigned URL when access is needed.

https://docs.aws.amazon.com/AmazonS3/latest/userguide/using-presigned-url.html

✗ **B** Save the picture's S3 key in an Amazon DynamoDB table. Create an Amazon S3 VPC endpoint to allow the employees to download pictures once they log in.
Incorrect. An S3 VPC endpoint provides private network connectivity from resources in a VPC to Amazon S3; it does not by itself provide browser users with authorized access to private objects. The application still needs to authenticate the user and provide a suitable access mechanism, such as a presigned URL.

https://docs.aws.amazon.com/vpc/latest/privatelink/vpc-endpoints-s3.html

✗ **C** Encode a picture using base64. Save the base64 string in an Amazon DynamoDB table. Allow the browser to retrieve the string and convert it to a picture.
Incorrect. Storing images as Base64 strings in DynamoDB is inefficient and is constrained by DynamoDB's item size limit, which is 400 KB. Amazon S3 is designed for storing objects such as profile pictures, while DynamoDB should store the S3 key or other metadata.

https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/Constraints.html

✓ **D** Save the picture's S3 key in an Amazon DynamoDB table. Use a function to generate a presigned URL every time an employee logs in. Return the URL to the browser.
Correct. Storing the S3 key avoids storing an expiring URL. When the authenticated employee logs in, the application can generate a new time-limited presigned URL that grants access to the private S3 object without making it publicly accessible.

https://docs.aws.amazon.com/AmazonS3/latest/userguide/using-presigned-url.html

Question 46

Single choice · Domain: D1 Development with AWS Services › Generate and use Amazon S3 presigned URLs for temporary, secure object access

A developer is storing JSON files in an Amazon S3 bucket. The developer wants to securely share an object with a specific group of people.

How can the developer securely provide temporary access to the objects that are stored in the S3 bucket?

A. Set object retention on the files. Use the AWS SDK to restore the object before subsequent requests. Provide the bucket's S3 URL.

B. Use the AWS SDK to generate a presigned URL. Provide the presigned URL.

C. Set a bucket policy that restricts access after a period of time. Provide the bucket's S3 URL.

D. Configure static web hosting on the S3 bucket. Provide the bucket's web URL.

Answer: B. Use the AWS SDK to generate a presigned URL. Provide the presigned URL.

Explanation

This question tests the ability to provide time-limited access to private S3 objects for specific recipients. Presigned URLs grant temporary access to an object without requiring the recipients to have AWS credentials.

✗ **A** Set object retention on the files. Use the AWS SDK to restore the object before subsequent requests. Provide the bucket's S3 URL.
Incorrect. S3 Object Lock retention protects objects from deletion or overwrite; it does not provide

temporary read access to a specific group, and an S3 URL alone does not grant authorization.

https://docs.aws.amazon.com/AmazonS3/latest/userguide/object-lock.html

✓ **B** Use the AWS SDK to generate a presigned URL. Provide the presigned URL.
Correct. A presigned URL grants temporary access to a specific S3 object using the permissions of the principal that generated it. The developer can set an expiration time and share the URL with the intended recipients.

https://docs.aws.amazon.com/AmazonS3/latest/userguide/using-presigned-url.html

✗ **C** Set a bucket policy that restricts access after a period of time. Provide the bucket's S3 URL.
Incorrect. A bucket policy can control access based on identities, conditions, and other request attributes, but providing the bucket URL does not itself give recipients authorized temporary access to private objects. Presigned URLs are the standard mechanism for time-limited object sharing.

https://docs.aws.amazon.com/AmazonS3/latest/userguide/using-presigned-url.html

✗ **D** Configure static web hosting on the S3 bucket. Provide the bucket's web URL.
Incorrect. S3 static website hosting is intended to serve website content and does not provide secure, recipient-specific temporary access. A website endpoint does not replace authorization for private object sharing.

https://docs.aws.amazon.com/AmazonS3/latest/userguide/WebsiteAccessPermissionsReqd.html

Question 47

Single choice · Domain: D1 Development with AWS Services › Use DynamoDB condition expressions to control conditional item updates

An application makes requests to an Amazon DynamoDB table that is configured with provisioned throughput. The table has a partition key of UserID.

A developer needs to identify items in the table that have a status attribute of "Inactive". The developer must add a specific email address to the appropriate attribute for all inactive items. The developer attempts to use an AWS Lambda function to update the table items. However, the function fails with the following error: ConditionalCheckFailedException.

How can the developer resolve this error?

A. Update the condition expression section of the request to check whether the status attribute equals "Inactive" before making the update request.

B. Update the condition expression section of the request to check whether the email attribute exists in the item before making the update request.

C. Include error-handling logic in the Lambda function to retry the update operation when a ConditionalCheckFailedException error occurs.

D. Update the condition expression in the request to check whether the email address is in the expected format before making the update request.

Answer: A. Update the condition expression section of the request to check whether the status attribute equals "Inactive" before making the update request.

Explanation

This question tests the ability to diagnose a ConditionalCheckFailedException and configure a DynamoDB condition expression that matches the intended update criteria. The update should proceed only when

Free demo · full exam: 595 questions

the item's status is Inactive.

✓ **A** Update the condition expression section of the request to check whether the status attribute equals "Inactive" before making the update request.
Correct. A ConditionalCheckFailedException occurs when the condition expression evaluates to false. Checking that status equals Inactive ensures the update targets the required items.

https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/Expressions.ConditionExpressions.html

✗ **B** Update the condition expression section of the request to check whether the email attribute exists in the item before making the update request.
Incorrect. Checking whether the email attribute exists does not identify items whose status is Inactive and could prevent updates to items that do not yet have an email attribute.

https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/Expressions.ConditionExpressions.html

✗ **C** Include error-handling logic in the Lambda function to retry the update operation when a ConditionalCheckFailedException error occurs.
Incorrect. Retrying does not correct a condition expression that evaluates to false. The request must use a condition that matches the intended status of the item; retries are not an appropriate resolution for this conditional failure.

https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/Expressions.ConditionExpressions.html

✗ **D** Update the condition expression in the request to check whether the email address is in the expected format before making the update request.
Incorrect. Validating the email format does not identify inactive items and does not address why the current condition expression failed. The condition should check that the status attribute equals Inactive.

https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/Expressions.ConditionExpressions.html

Question 48

Single choice · Domain: D1 Development with AWS Services › Configure API Gateway stages and stage variables to test updated Lambda integrations without affecting production traffic

A developer updates an AWS Lambda function that is integrated with an Amazon API Gateway API.

The API serves as the backend for a web application. The developer must test the updated Lambda function without affecting production users.

Which solution will meet these requirements in the MOST operationally efficient way?

A. Create a canary deployment on the existing API stage and test by using the production URL.
B. Change the API endpoint type to private and test by using the production URL.
C. Create a new API Gateway stage for testing and use stage variables to route traffic to the updated Lambda function.
D. Deploy a separate CloudFormation stack that duplicates the production API and Lambda function.

Answer: C. Create a new API Gateway stage for testing and use stage variables to route traffic to the updated Lambda function.

Explanation

This question tests the ability to isolate testing from production by creating a separate API Gateway stage and using stage variables to route that stage to the updated Lambda function or version.

✗ **A** Create a canary deployment on the existing API stage and test by using the production URL.

Free demo · full exam: 595 questions

Incorrect. An API Gateway canary deployment on the production stage can route a percentage of live production traffic to the updated integration, which could affect production users.

https://docs.aws.amazon.com/apigateway/latest/developerguide/canary-release.html

✗ **B** Change the API endpoint type to private and test by using the production URL.
Incorrect. Changing the endpoint type does not create an isolated test environment and can disrupt how production clients access the API. Private APIs are accessible only through an interface VPC endpoint.

https://docs.aws.amazon.com/apigateway/latest/developerguide/api-gateway-private-apis.html

✓ **C** Create a new API Gateway stage for testing and use stage variables to route traffic to the updated Lambda function.
Correct. A separate stage provides an isolated stage-specific invoke URL, while stage variables can select the Lambda function, alias, or version used by that stage. Production traffic remains on the production stage.

https://docs.aws.amazon.com/apigateway/latest/developerguide/stage-variables.html

✗ **D** Deploy a separate CloudFormation stack that duplicates the production API and Lambda function.
Incorrect. A duplicate CloudFormation stack can isolate testing, but it requires provisioning and maintaining duplicate infrastructure, making it less operationally efficient than creating a test stage in the existing API.

https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/stacks.html

Question 49

Single choice · Domain: D1 Development with AWS Services › Configure Amazon SQS visibility timeout for an AWS Lambda event source mapping

A developer is testing an AWS Lambda function that has an event source of an Amazon SQS queue.

The developer notices that some of the messages the Lambda function processes reappear in the queue while the messages are being processed. The developer must correct this behavior. Which solution will meet this requirement?

A. Increase the timeout of the Lambda function.
B. Increase the visibility timeout of the SQS queue.
C. Increase the memory allocation of the Lambda function.
D. Increase the batch size in the event source mapping.

Answer: B. Increase the visibility timeout of the SQS queue.

Explanation

This question tests the ability to configure SQS and Lambda integration so that a message remains invisible for the entire duration of Lambda processing. The visibility timeout must be long enough to prevent messages from becoming available again before processing completes.

✗ **A** Increase the timeout of the Lambda function.
Incorrect. Increasing the Lambda timeout does not directly prevent an SQS message from reappearing. The SQS visibility timeout controls how long a received message remains hidden from other consumers.

https://docs.aws.amazon.com/AWSSimpleQueueService/latest/SQSDeveloperGuide/sqs-visibility-timeout.html

✓ **B** Increase the visibility timeout of the SQS queue.
Correct. If the SQS visibility timeout expires while Lambda is still processing a message, the message

Free demo · full exam: 595 questions

becomes visible again and can be received again. Increasing the visibility timeout to exceed the expected processing duration prevents this behavior.

https://docs.aws.amazon.com/lambda/latest/dg/with-sqs.html

✗ **C** Increase the memory allocation of the Lambda function.
Incorrect. More memory can provide additional CPU capacity and may reduce execution time, but it does not change the SQS visibility timeout. The message can still reappear if the visibility timeout expires during processing.

https://docs.aws.amazon.com/lambda/latest/dg/configuration-memory.html

✗ **D** Increase the batch size in the event source mapping.
Incorrect. Batch size controls the number of messages Lambda retrieves per invocation; it does not extend the period during which each message is hidden in the queue. The visibility timeout must be configured appropriately.

https://docs.aws.amazon.com/lambda/latest/dg/with-sqs.html

Question 50

Single choice · Domain: D1 Development with AWS Services › Design DynamoDB partition keys to distribute read traffic and avoid hot partitions

A company is developing a weather forecast application that displays forecasts for towns and cities in a single country for a specified date range. The application stores weather data in an Amazon DynamoDB table named Forecasts. The Forecasts table has attributes named CityId, Temperature, and ForecastDate. The company expects the application to be read-heavy and that the vast majority of reads will target the current day's forecast for one of a small number of very large cities. The company needs to design a partition key configuration that will yield the most even distribution of read requests for the application's expected use case. Which solution will meet these requirements?

A. Use CityId with a calculated suffix as the partition key. Use ForecastDate as the sort key.
B. Use a numeric index as the partition key. Use ForecastDate as the sort key.
C. Use ForecastDate as the partition key. Use CityId as the sort key.
D. Use ForecastDate with a calculated suffix as the partition key. Use CityId as the sort key.

Answer: D. Use ForecastDate with a calculated suffix as the partition key. Use CityId as the sort key.

Explanation

This question tests the ability to select a DynamoDB composite key that distributes heavily concentrated reads across multiple partitions. A calculated suffix added to the frequently accessed forecast date provides read sharding, while CityId supports item lookup within each shard.

✗ **A** Use CityId with a calculated suffix as the partition key. Use ForecastDate as the sort key.
Incorrect. Although a suffix can increase the number of partitions, using CityId as the primary partition-key basis does not directly distribute the dominant current-date access pattern as effectively as sharding the date-based workload. The application would also need to know or query the appropriate suffixes for a city.

https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/bp-partition-key-sharding.html

✗ **B** Use a numeric index as the partition key. Use ForecastDate as the sort key.
Incorrect. A numeric index is not a defined application attribute or an appropriate high-cardinality access

key in this scenario. The partition key should be based on the application's access pattern and should distribute requests evenly.

https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/bp-partition-key-design.html

✗ **C** Use ForecastDate as the partition key. Use CityId as the sort key.
Incorrect. ForecastDate has low cardinality, and most requests target the current date. This would concentrate a large proportion of read traffic on the same partition key and could create a hot partition.

https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/bp-partition-key-design.html

✓ **D** Use ForecastDate with a calculated suffix as the partition key. Use CityId as the sort key.
Correct. Adding a calculated suffix to ForecastDate shards the heavily accessed current-date data across multiple partition-key values, distributing read requests more evenly. CityId as the sort key supports retrieving a city's forecast within each shard.

https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/bp-partition-key-sharding.html

---

Question 51

Single choice · Domain: D1 Development with AWS Services › Design DynamoDB secondary indexes and use Query operations for efficient retrieval

An application uses an Amazon DynamoDB table to manage user profiles. A UserID attribute is the primary key of the table. The table also includes columns named Username, EmailAddress, RegistrationDate, Location, and Status.

The application needs to display a list of users from a specific location who registered after a specific date. Queries on the table must be optimized for efficiency.

Which solution will meet these requirements?

A. Create a global secondary index (GSI). Use Location as the partition key and RegistrationDate as the sort key. Use the Query operation to retrieve the specified users.

B. Use the Scan operation to retrieve the specified users. Use a filter expression for a value in the RegistrationDate column that is greater than the date required by the application.

C. Create a local secondary index (LSI). Use Location as the partition key and RegistrationDate as the sort key. Use the Query operation to retrieve the specified users.

D. Use the BatchGetItem operation with a filter on the RegistrationDate column for a value that is greater than the required date to retrieve the specified users.

Answer: A. Create a global secondary index (GSI). Use Location as the partition key and RegistrationDate as the sort key. Use the Query operation to retrieve the specified users.

Explanation

This question tests the ability to select and query a DynamoDB secondary index based on an alternate partition key and sort key. A GSI supports Location as the partition key and RegistrationDate as the sort key, enabling efficient retrieval of users from a location after a specified date.

✓ **A** Create a global secondary index (GSI). Use Location as the partition key and RegistrationDate as the sort key. Use the Query operation to retrieve the specified users.
Correct. A GSI can use a partition key different from the base table's UserID, and RegistrationDate can be used as the sort key to apply a range condition for dates after the specified date. The Query operation efficiently retrieves items for the specified Location.

Free demo · full exam: 595 questions

https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/GSI.html

✗ **B** Use the Scan operation to retrieve the specified users. Use a filter expression for a value in the RegistrationDate column that is greater than the date required by the application.
Incorrect. Scan examines every item in the table or index, and a filter expression is applied after the items are read, making this inefficient for targeted queries. A GSI and Query operation are appropriate for this access pattern.

https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/Scan.html

✗ **C** Create a local secondary index (LSI). Use Location as the partition key and RegistrationDate as the sort key. Use the Query operation to retrieve the specified users.
Incorrect. An LSI must use the same partition key as the base table, which is UserID. Because this solution requires Location as an alternate partition key, a GSI is required.

https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/LSI.html

✗ **D** Use the BatchGetItem operation with a filter on the RegistrationDate column for a value that is greater than the required date to retrieve the specified users.
Incorrect. BatchGetItem retrieves items by explicitly specified primary keys and does not support filter expressions for selecting items by RegistrationDate. It cannot efficiently find all users in a location who registered after a date.

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

Question 52

Single choice · Domain: D1 Development with AWS Services › Design DynamoDB partition and sort keys to distribute write traffic and avoid hot partitions

A food-delivery company has an application that uses an Amazon DynamoDB table for customer orders. The table is in on-demand mode. The table uses order_date as the partition key. The table uses a combination of customer_id and order_id as the sort key (customer_id#order_id). During peak order periods, write operations to the table are being throttled. What should a developer do to resolve this issue?

A. Use a sequential number for the partition key. Continue to use order_id as the sort key.

B. Change the capacity mode of the table to provisioned mode. Increase the write capacity units (WCUs) to meet the application demand.

C. Transition the table to Amazon Aurora.

D. Use customer_id for the partition key. Use order_id as the sort key.

Answer: D. Use customer_id for the partition key. Use order_id as the sort key.

Explanation

This question tests the ability to identify a hot partition caused by using order_date as the partition key and to redesign the key schema so writes are distributed across multiple partitions.

✗ **A** Use a sequential number for the partition key. Continue to use order_id as the sort key.
Incorrect. A sequential partition key can concentrate recent writes on the same partition and create a hot partition, so it does not reliably resolve throttling. DynamoDB recommends designing partition keys with high cardinality and evenly distributed access patterns.

https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/bp-partition-key-design.html

✗ **B** Change the capacity mode of the table to provisioned mode. Increase the write capacity units (WCUs)

Free demo · full exam: 595 questions

to meet the application demand.
Incorrect. Increasing capacity does not resolve throttling caused by an uneven access pattern or a hot partition. The partition key design must distribute writes across partitions.

https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/bp-partition-key-design.html

✗ **C** Transition the table to Amazon Aurora.
Incorrect. Moving to Aurora is unnecessary and does not address the DynamoDB partition-key design problem. The application can resolve the issue by distributing writes more evenly across DynamoDB partitions.

https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/bp-partition-key-design.html

✓ **D** Use customer_id for the partition key. Use order_id as the sort key.
Correct. Using customer_id as the partition key distributes orders across customers instead of concentrating all orders for the same date on one partition. Using order_id as the sort key preserves the ability to store and query multiple orders for each customer.

https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/bp-partition-key-design.html

---

Question 53

Single choice · Domain: D1 Development with AWS Services › Configure IAM authorization for an Amazon API Gateway HTTP API and grant invocation permissions through a Lambda execution role

A developer is working on an application that uses a microservices architecture. The developer needs to grant an AWS Lambda function access to make calls to an Amazon API Gateway HTTP API. The HTTP API requires authentication.

Which solution will meet these requirements?

A. Create a JSON Web Token (JWT) authorizer for the HTTP API. Create a new IAM identity provider (IdP). Configure the Lambda function's execution role to allow the Lambda function to access the IdP.

B. Enable IAM authorization for the HTTP API route. Configure the Lambda function's IAM role policy to allow the Lambda function to access the API.

C. Configure the HTTP API route to require API keys. Add an API key to the Lambda function environment variables.

D. Create a resource-based policy for the Lambda function that allows the Lambda function to access the HTTP API.

Answer: B. Enable IAM authorization for the HTTP API route. Configure the Lambda function's IAM role policy to allow the Lambda function to access the API.

Explanation

This question tests selecting an authentication method that allows a Lambda function to invoke a protected API Gateway HTTP API. IAM authorization uses AWS Signature Version 4 and the Lambda execution role must allow the execute-api:Invoke action.

✗ **A** Create a JSON Web Token (JWT) authorizer for the HTTP API. Create a new IAM identity provider (IdP). Configure the Lambda function's execution role to allow the Lambda function to access the IdP.
Incorrect. A JWT authorizer requires the caller to provide a valid JWT issued by a configured identity provider; creating an IAM identity provider and granting the Lambda role access to it does not by itself authenticate the API request.

Free demo · full exam: 595 questions

https://docs.aws.amazon.com/apigateway/latest/developerguide/http-api-jwt-authorizer.html

✓ **B** Enable IAM authorization for the HTTP API route. Configure the Lambda function's IAM role policy to allow the Lambda function to access the API.
Correct. With IAM authorization enabled, the Lambda function can sign the API request with AWS Signature Version 4 using its execution role. The role policy must allow execute-api:Invoke for the API route.

https://docs.aws.amazon.com/apigateway/latest/developerguide/http-api-access-control-iam.html

✗ **C** Configure the HTTP API route to require API keys. Add an API key to the Lambda function environment variables.
Incorrect. API keys are not an appropriate IAM-based authorization mechanism for this requirement, and storing a key in environment variables does not automatically cause the Lambda request to include the required authentication header. HTTP API authorization options include IAM, JWT authorizers, and Lambda authorizers.

https://docs.aws.amazon.com/apigateway/latest/developerguide/http-api-access-control.html

✗ **D** Create a resource-based policy for the Lambda function that allows the Lambda function to access the HTTP API.
Incorrect. A Lambda resource-based policy controls which principals can invoke the Lambda function; it does not grant the function permission to invoke an API Gateway HTTP API. The Lambda execution role must instead be granted execute-api:Invoke permission when IAM authorization is used.

https://docs.aws.amazon.com/lambda/latest/dg/access-control-resource-based.html

Question 54

Single choice · Domain: D1 Development with AWS Services › Configure Lambda asynchronous invocation destinations for successful and failed invocations

A developer is building a serverless application that uses asynchronous AWS Lambda functions. The developer needs a solution to capture records of every Lambda function invocation. Each function must have multiple destinations based on whether each invocation is successful. The solution must record function responses in JSON format.

Which solution will meet these requirements?

A. Amazon CloudWatch Logs log groups that use the default log format for Lambda functions. Route each invocation to the appropriate log group by using a Lambda canary deployment and weighted aliases. Set the appropriate log group as the target for each function.

B. Set up an S3 bucket as an on-failure destination for the Lambda function. Configure an Amazon SNS topic as the destination for successful Lambda function invocations.

C. Configure an Amazon SQS dead-letter queue as an event source for the Lambda function to store failed invocations. In the Lambda function code, use the PutItem Amazon DynamoDB API call to add the successful invocation information to the database.

D. Set up an Amazon SQS queue as an on-failure destination for the Lambda function. Configure an Amazon OpenSearch Service cluster as the destination for the Lambda function for successful invocations.

Answer: B. Set up an S3 bucket as an on-failure destination for the Lambda function. Configure an Amazon SNS topic as the destination for successful Lambda function invocations.

Free demo · full exam: 595 questions

Dumpsgate

Question 55

Single choice · Domain: D2 Security › Restore access to data encrypted with an AWS KMS key after imported key material is deleted

A company has an application that runs on Amazon EC2 instances and stores sensitive data. The application encrypts the data by using an AWS KMS customer managed key with key material that has been imported from outside of AWS.

A developer accidentally deletes the key material, and the application is unable to read the encrypted data.

What should the developer do to fix the application with the LEAST amount of effort?

A. Create new key material from the original source outside of AWS. Reimport the key material to the existing KMS key.

B. Recover the deleted key material by opening a request with AWS Support.

C. Generate an AWS managed KMS key. Reconfigure the application to use this key.

D. Reimport the key material from the original source outside of AWS to the existing KMS key.

Answer: D. Reimport the key material from the original source outside of AWS to the existing KMS key.

Explanation

This question tests the ability to manage imported key material in AWS KMS and understand that deleted imported material can be reimported into the existing customer managed key. Reusing the existing key preserves the key ID and avoids reconfiguring the application.

✗ **A** Create new key material from the original source outside of AWS. Reimport the key material to the existing KMS key.
Incorrect. Creating new key material is unnecessary and would not decrypt data encrypted with the original key material. The original key material must be reimported into the existing KMS key.

https://docs.aws.amazon.com/kms/latest/developerguide/importing-keys.html

✗ **B** Recover the deleted key material by opening a request with AWS Support.
Incorrect. AWS KMS does not recover deleted imported key material through AWS Support. The customer must retain the key material and reimport it when necessary.

https://docs.aws.amazon.com/kms/latest/developerguide/importing-keys.html

✗ **C** Generate an AWS managed KMS key. Reconfigure the application to use this key.
Incorrect. A different KMS key cannot decrypt ciphertext encrypted with the original key, and reconfiguring the application would require re-encrypting or otherwise restoring access to the existing data.

https://docs.aws.amazon.com/kms/latest/developerguide/concepts.html

✓ **D** Reimport the key material from the original source outside of AWS to the existing KMS key.
Correct. When imported key material is deleted, the same key material can be reimported into the existing KMS key. This preserves the existing key ID and requires the least application change, allowing the application to decrypt the data again.

https://docs.aws.amazon.com/kms/latest/developerguide/importing-keys.html

Free demo · full exam: 595 questions

Question 56

Single choice · Domain: D2 Security › Configure a least-privilege IAM execution role for a Lambda function to access Amazon S3

A company needs a new REST API that can return information about the contents of an Amazon S3 bucket, such as a count of the objects stored in it. The company has decided that the new API should be written as a microservice using AWS Lambda and Amazon API Gateway.

How should the developer ensure that the microservice has the necessary access to the Amazon S3 bucket, while adhering to security best practices?

A. Create an IAM user that has permissions to access the Amazon S3 bucket, and store the IAM user credentials in the Lambda function source code.

B. Create an IAM role that has permissions to access the Amazon S3 bucket and assign it to the Lambda function as its execution role.

C. Create an Amazon S3 bucket policy that specifies the Lambda service as its principal and assign it to the Amazon S3 bucket.

D. Create an IAM role, attach the AmazonS3FullAccess managed policy to it, and assign the role to the Lambda function as its execution role.

Answer: B. Create an IAM role that has permissions to access the Amazon S3 bucket and assign it to the Lambda function as its execution role.

Explanation

This question tests the ability to grant a Lambda function secure access to AWS resources by using an IAM execution role with only the permissions required by the function.

✗ **A** Create an IAM user that has permissions to access the Amazon S3 bucket, and store the IAM user credentials in the Lambda function source code.
Incorrect. Credentials must not be embedded in Lambda source code, because this creates a risk of credential exposure and complicates credential rotation. Lambda functions should use an execution role to obtain temporary credentials.

https://docs.aws.amazon.com/lambda/latest/dg/lambda-intro-execution-role.html

✓ **B** Create an IAM role that has permissions to access the Amazon S3 bucket and assign it to the Lambda function as its execution role.
Correct. A Lambda execution role grants the function temporary credentials and permissions to access AWS services. The role should contain only the required S3 permissions, following the principle of least privilege.

https://docs.aws.amazon.com/lambda/latest/dg/lambda-intro-execution-role.html

✗ **C** Create an Amazon S3 bucket policy that specifies the Lambda service as its principal and assign it to the Amazon S3 bucket.
Incorrect. Lambda functions access AWS resources using the function's execution role, not by specifying the Lambda service as the bucket policy principal. A bucket policy can be used for cross-account or resource-based access scenarios, but it does not replace the required Lambda execution role permissions.

https://docs.aws.amazon.com/AmazonS3/latest/userguide/access-policy-language-overview.html

✗ **D** Create an IAM role, attach the AmazonS3FullAccess managed policy to it, and assign the role to the Lambda function as its execution role.
Incorrect. Although an IAM role is the correct mechanism, AmazonS3FullAccess grants permissions far beyond what is needed to count or read objects in a specific bucket. The execution role should use a

narrowly scoped policy that grants only the required S3 actions and resources.
https://docs.aws.amazon.com/IAM/latest/UserGuide/best-practices.html

Question 57

Single choice · Domain: D2 Security › Configure CloudWatch Logs data protection policies to mask sensitive data and control unmasking permissions

A healthcare company discovers that one of the company's AWS Lambda functions is improperly sending customer personal health information (PHI) and personally identifiable information (PII) to an Amazon CloudWatch Logs log group.

The company needs a solution to automatically mask PHI and PII across all log events in the log group. The company must ensure that masked values cannot be accidentally revealed through CloudWatch Logs Insights queries or subscription filters. The solution must allow only specific security engineers to view the original unmasked values when required for investigations.

Which solution will meet these requirements?

A. Create a CloudWatch Logs data protection policy for the log group. Add managed data identifiers for PHI and PII. Grant the logs:Unmask IAM permission only to security engineers.

B. Update the Lambda function to replace PHI and PII with placeholder characters before the function writes logs to the log group.

C. Use AWS KMS to encrypt the log group. Give the security engineers KMS decrypt permissions.

D. Use Amazon Data Firehose to stream the logs to an Amazon S3 bucket. Use Amazon Athena to query the logs. Use Athena views to filter out PHI and PII.

Answer: A. Create a CloudWatch Logs data protection policy for the log group. Add managed data identifiers for PHI and PII. Grant the logs:Unmask IAM permission only to security engineers.

Explanation

This question tests the ability to protect PHI and PII in CloudWatch Logs by using managed data identifiers, while restricting access to original values through the logs:Unmask IAM permission.

✓ **A** Create a CloudWatch Logs data protection policy for the log group. Add managed data identifiers for PHI and PII. Grant the logs:Unmask IAM permission only to security engineers.
Correct. CloudWatch Logs data protection policies can identify and mask sensitive data such as PHI and PII in log events. Masked data remains protected when accessed through CloudWatch Logs Insights and subscription filters, while authorized principals with the logs:Unmask permission can view the original values.

https://docs.aws.amazon.com/AmazonCloudWatch/latest/logs/mask-sensitive-log-data.html

✗ **B** Update the Lambda function to replace PHI and PII with placeholder characters before the function writes logs to the log group.
Incorrect. Updating one Lambda function does not automatically protect sensitive data from all log events in the log group, and it would not provide CloudWatch Logs-controlled unmasking for authorized security engineers.

https://docs.aws.amazon.com/AmazonCloudWatch/latest/logs/mask-sensitive-log-data.html

✗ **C** Use AWS KMS to encrypt the log group. Give the security engineers KMS decrypt permissions.
Incorrect. KMS encryption protects CloudWatch Logs data at rest but does not identify or mask PHI and

Free demo · full exam: 595 questions

PII in log events, nor does it provide the required logs:Unmask behavior.

https://docs.aws.amazon.com/AmazonCloudWatch/latest/logs/encrypt-log-data-kms.html

✗ **D** Use Amazon Data Firehose to stream the logs to an Amazon S3 bucket. Use Amazon Athena to query the logs. Use Athena views to filter out PHI and PII.
Incorrect. Athena views can filter query results but do not automatically mask sensitive values across the original log events or prevent exposure through other CloudWatch Logs access paths such as subscription filters.

https://docs.aws.amazon.com/AmazonCloudWatch/latest/logs/mask-sensitive-log-data.html

Question 58

Single choice · Domain: D2 Security › Configure cross-account access to a customer managed AWS KMS key for Amazon S3 encryption

A company uses two AWS accounts: production and development. The company stores data in an Amazon S3 bucket that is in the production account. The data is encrypted with an AWS KMS customer managed key. The company plans to copy the data to another S3 bucket that is in the development account.

A developer needs to use a KMS key to encrypt the data in the S3 bucket that is in the development account. The KMS key in the development account must be accessible from the production account.

Which solution will meet these requirements?

A. Replicate the customer managed KMS key from the production account to the development account. Specify the production account in the key policy.

B. Create a new customer managed KMS key in the development account. Specify the production account in the key policy.

C. Create a new AWS managed KMS key for Amazon S3 in the development account. Specify the production account in the key policy.

D. Replicate the default AWS managed KMS key for Amazon S3 from the production account to the development account. Specify the production account in the key policy.

Answer: B. Create a new customer managed KMS key in the development account. Specify the production account in the key policy.

Explanation

This question tests the ability to select and authorize a customer managed KMS key in one AWS account for use by principals in another account. It also tests the distinction between customer managed keys, which support customized key policies, and AWS managed keys, which cannot be used for this cross-account access pattern.

✗ **A** Replicate the customer managed KMS key from the production account to the development account. Specify the production account in the key policy.
Incorrect. KMS keys cannot be replicated between AWS accounts in this manner. A customer managed key must be created in the development account, and its key policy must allow the production account to use it.

https://docs.aws.amazon.com/kms/latest/developerguide/multi-region-keys-overview.html

✓ **B** Create a new customer managed KMS key in the development account. Specify the production

account in the key policy.
Correct. A customer managed KMS key can be created in the development account with a key policy that grants the production account access. The appropriate IAM permissions in the production account must also be configured for the cross-account use.

https://docs.aws.amazon.com/kms/latest/developerguide/key-policy-modifying.html

✗ **C** Create a new AWS managed KMS key for Amazon S3 in the development account. Specify the production account in the key policy.
Incorrect. AWS managed KMS key policies cannot be customized to grant the required cross-account access. A customer managed key is required for this use case.

https://docs.aws.amazon.com/kms/latest/developerguide/concepts.html#aws-managed-cmk

✗ **D** Replicate the default AWS managed KMS key for Amazon S3 from the production account to the development account. Specify the production account in the key policy.
Incorrect. AWS managed keys cannot be replicated between accounts or modified to add the required cross-account key policy permissions.

https://docs.aws.amazon.com/kms/latest/developerguide/concepts.html#aws-managed-cmk

Question 59

Single choice · Domain: D2 Security › Provision an ACM public certificate with DNS validation for automatic renewal and CloudFront use

A developer wants to use an Amazon CloudFront distribution to deliver a web application to users.

Users will access the application through the internet.

The developer needs to create an SSL/TLS certificate to use with the CloudFront distribution. The developer wants to minimize operational overhead by ensuring that the certificate is renewed automatically.

Which solution will meet these requirements?

A. Create a public certificate that uses DNS validation in IAM.
B. Import a certificate into IAM. Configure managed renewal for the certificate.
C. Create a public certificate that uses DNS validation in AWS Certificate Manager (ACM).
D. Create a public certificate that uses email validation in AWS Certificate Manager (ACM).

Answer: C. Create a public certificate that uses DNS validation in AWS Certificate Manager (ACM).

Explanation

This question tests the ability to select the AWS Certificate Manager validation method that supports automatic renewal for a public certificate used by an internet-facing CloudFront distribution. DNS validation allows ACM to renew the certificate automatically while the DNS validation record remains in place.

✗ **A** Create a public certificate that uses DNS validation in IAM.
Incorrect. IAM does not issue ACM public certificates, and IAM is not the service used to create a publicly trusted certificate with DNS validation.

https://docs.aws.amazon.com/acm/latest/userguide/acm-overview.html

✗ **B** Import a certificate into IAM. Configure managed renewal for the certificate.
Incorrect. Imported certificates are not eligible for ACM managed renewal, and IAM is not the service for

Free demo · full exam: 595 questions

configuring ACM certificate renewal.

https://docs.aws.amazon.com/acm/latest/userguide/import-certificate.html

✓ **C** Create a public certificate that uses DNS validation in AWS Certificate Manager (ACM).
Correct. ACM public certificates validated through DNS can be renewed automatically when the ACM validation CNAME record remains in DNS. ACM certificates can be used with CloudFront, and CloudFront certificates must be requested or imported in the US East (N. Virginia) Region. and.

https://docs.aws.amazon.com/acm/latest/userguide/dns-validation.html
https://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/cnames-and-https-procedures.html

✗ **D** Create a public certificate that uses email validation in AWS Certificate Manager (ACM).
Incorrect. Email validation requires approval through validation emails during certificate issuance and renewal, which introduces operational overhead. DNS validation is preferred when automatic renewal is required.

https://docs.aws.amazon.com/acm/latest/userguide/email-validation.html

Question 60

Single choice · Domain: D2 Security › Implement secure storage and automatic rotation of database credentials using AWS Secrets Manager

A company runs a critical application on AWS. The application uses credentials to access an Amazon RDS database.

As part of a risk assessment to analyze security vulnerabilities, the company found that database credentials are not frequently rotated. A developer needs to implement a solution that securely stores and rotates the credentials. The solution must not expose the credentials to anyone who should not have access to the credentials.

Which solution will meet these requirements?

A. Configure AWS Secrets Manager to store the database credentials. Schedule managed rotation for Secrets Manager secrets.

B. Configure the Systems Manager parameters with an AWS KMS AWS managed key to encrypt the database credentials at rest. Create an AWS Lambda function to rotate the KMS key. Set up an Amazon EventBridge rule to invoke the Lambda function on a schedule.

C. Configure an Amazon S3 bucket with S3 Bucket Keys for server-side encryption with AWS KMS keys (SSE-KMS) to store the database credentials.

D. Configure the Systems Manager parameters with an AWS KMS customer managed key to encrypt the database credentials at rest. Create an AWS Lambda function to periodically rotate the KMS key. Use a cron expression to invoke the Lambda function on a schedule.

Answer: A. Configure AWS Secrets Manager to store the database credentials. Schedule managed rotation for Secrets Manager secrets.

Explanation

This question tests the ability to select an AWS service that securely stores sensitive database credentials and performs managed rotation without exposing the credentials to unauthorized users.

✓ **A** Configure AWS Secrets Manager to store the database credentials. Schedule managed rotation for Secrets Manager secrets.

Free demo · full exam: 595 questions

Correct. AWS Secrets Manager is designed to securely store credentials and supports scheduled, managed rotation for supported Amazon RDS databases. Applications can retrieve the secret without embedding credentials in code or exposing them to unauthorized users.

https://docs.aws.amazon.com/secretsmanager/latest/userguide/rotating-secrets.html

✗ **B** Configure the Systems Manager parameters with an AWS KMS AWS managed key to encrypt the database credentials at rest. Create an AWS Lambda function to rotate the KMS key. Set up an Amazon EventBridge rule to invoke the Lambda function on a schedule.
Incorrect. Encrypting a Systems Manager parameter and rotating a KMS key does not rotate the database credentials. KMS key rotation changes the cryptographic key material, not the password stored in the database.

https://docs.aws.amazon.com/kms/latest/developerguide/rotate-keys.html

✗ **C** Configure an Amazon S3 bucket with S3 Bucket Keys for server-side encryption with AWS KMS keys (SSE-KMS) to store the database credentials.
Incorrect. S3 with SSE-KMS can encrypt objects at rest, but this solution does not provide automatic database credential rotation and is not the purpose-built service for managing application secrets.

https://docs.aws.amazon.com/AmazonS3/latest/userguide/UsingKMSEncryption.html

✗ **D** Configure the Systems Manager parameters with an AWS KMS customer managed key to encrypt the database credentials at rest. Create an AWS Lambda function to periodically rotate the KMS key. Use a cron expression to invoke the Lambda function on a schedule.
Incorrect. Rotating a customer managed KMS key does not rotate the database password or credentials. The solution would require custom logic to change the database credentials and update the stored value, whereas Secrets Manager provides managed rotation for supported Amazon RDS databases.

https://docs.aws.amazon.com/secretsmanager/latest/userguide/rotating-secrets.html

Question 61

Single choice · Domain: D2 Security › Securely manage and retrieve sensitive credentials used by AWS Lambda functions

A company operates a web-based loan processing application. The application's UI was implemented in JavaScript. The frontend transmits application data securely by using HTTPS to Amazon API Gateway, which initiates an AWS Lambda function in private subnets. The Lambda function interacts with third-party credit check APIs that require persistent API keys. The company enforces strict policies to ensure that personally identifiable information (PII) and sensitive credentials are never exposed in client code, request paths, headers, or logs. The company needs a solution to manage the API keys that the Lambda function needs to use. Which solution will meet this requirement in the MOST secure way?

A. Store the API keys as encrypted environment variables by using an AWS KMS key. Configure the execution role of the Lambda function to have permissions to securely decrypt the environment variables at runtime.

B. Pass the API keys to the Lambda function by including the keys as URL query parameters in each HTTPS request.

C. Bundle the API keys inside the minified client-side JavaScript.

D. Store the API keys as resource metadata tags on the Lambda function. Configure the Lambda function to read its own tags at startup by using the AWS SDK.

Free demo · full exam: 595 questions

Answer: A. Store the API keys as encrypted environment variables by using an AWS KMS key. Configure the execution role of the Lambda function to have permissions to securely decrypt the environment variables at runtime.

Explanation

This question tests the ability to protect persistent third-party API keys from exposure in client-side code, request data, and logs while allowing a Lambda function to decrypt and use them at runtime.

✓ **A** Store the API keys as encrypted environment variables by using an AWS KMS key. Configure the execution role of the Lambda function to have permissions to securely decrypt the environment variables at runtime.
Correct. Lambda environment variables can be encrypted at rest with a customer managed AWS KMS key, and the function's execution role can be granted permission to decrypt them at runtime. The keys are therefore not exposed in client-side code, request paths, headers, or logs.

https://docs.aws.amazon.com/lambda/latest/dg/configuration-envvars.html

✗ **B** Pass the API keys to the Lambda function by including the keys as URL query parameters in each HTTPS request.
Incorrect. URL query parameters can be recorded in browser history, web server logs, API Gateway logs, monitoring systems, and other intermediaries. This violates the requirement that sensitive credentials never be exposed in request paths or logs.

https://docs.aws.amazon.com/apigateway/latest/developerguide/api-gateway-known-issues.html

✗ **C** Bundle the API keys inside the minified client-side JavaScript.
Incorrect. Any secret included in client-side JavaScript can be downloaded, inspected, or extracted by users of the application. Minification does not provide confidentiality.

https://docs.aws.amazon.com/secretsmanager/latest/userguide/best-practices.html

✗ **D** Store the API keys as resource metadata tags on the Lambda function. Configure the Lambda function to read its own tags at startup by using the AWS SDK.
Incorrect. Resource tags are metadata and are not intended to store secrets. They can be exposed through APIs and administrative tools and do not provide the secure secret storage and controlled decryption required for API keys.

https://docs.aws.amazon.com/tag-editor/latest/userguide/tagging.html

Free demo · full exam: 595 questions

Question 62

Multiple choice · Domain: D2 Security › Configure encryption and IAM transport-security controls for Lambda-to-SNS communication

A developer creates an AWS Lambda function to publish a message to an Amazon SNS topic. All message content must be encrypted in transit and at rest between Lambda and Amazon SNS. A part of the Lambda execution role is as follows:

"Effect": "Allow", "Action": "SNS:Publish", "Resource": "arn:aws:sns:us-east-1:1234567890:secure-topic" Which combination of steps should the developer take to meet these requirements? (Select TWO.)

A. Enable server-side encryption on the SNS topic.

B. Add a Deny statement to the Lambda execution role. Specify the SNS topic ARN as the resource. Specify "aws:SecureTransport": "true" as the condition.

C. Create a VPC endpoint for Amazon SNS.

D. Add a StringEquals condition of "sns:Protocol": "https" to the Lambda execution role.

E. Add a Deny statement to the Lambda execution role. Specify the SNS topic ARN as the resource. Specify "aws:SecureTransport": "false" as the condition.

Answer

- A. **Enable server-side encryption on the SNS topic.**
- E. **Add a Deny statement to the Lambda execution role. Specify the SNS topic ARN as the resource. Specify "aws:SecureTransport": "false" as the condition.**

Explanation

This question tests selecting Amazon SNS server-side encryption for data at rest and enforcing encrypted transport with the aws:SecureTransport IAM condition. It also tests distinguishing transport encryption controls from networking configuration and SNS subscription protocol conditions.

✓ **A** Enable server-side encryption on the SNS topic.
Correct. Amazon SNS server-side encryption protects published messages while they are stored at rest by using AWS KMS.

https://docs.aws.amazon.com/sns/latest/dg/sns-server-side-encryption.html

✗ **B** Add a Deny statement to the Lambda execution role. Specify the SNS topic ARN as the resource. Specify "aws:SecureTransport": "true" as the condition.
Incorrect. An explicit Deny with aws:SecureTransport set to true would deny requests that use secure transport, which is the opposite of the requirement. The policy should deny requests when aws:SecureTransport is false.

https://docs.aws.amazon.com/IAM/latest/UserGuide/reference_policies_condition-keys.html#condition-keys-securetransport

✗ **C** Create a VPC endpoint for Amazon SNS.
Incorrect. An Amazon SNS VPC endpoint provides private network connectivity but does not enable SNS server-side encryption for data at rest and is not the IAM control used to require HTTPS transport.

https://docs.aws.amazon.com/sns/latest/dg/sns-vpc-endpoints.html

✗ **D** Add a StringEquals condition of "sns:Protocol": "https" to the Lambda execution role.
Incorrect. sns:Protocol is associated with SNS subscription protocol controls and does not enforce the transport security of the Lambda Publish API call. The aws:SecureTransport condition key is used to require secure transport.

https://docs.aws.amazon.com/service-authorization/latest/reference/list_amazonsimplenotificationservice.html

Free demo · full exam: 595 questions

✓ **E** Add a Deny statement to the Lambda execution role. Specify the SNS topic ARN as the resource. Specify "aws:SecureTransport": "false" as the condition.
Correct. An explicit Deny conditioned on aws:SecureTransport being false prevents the Lambda function from publishing over an insecure connection. AWS service API calls should use HTTPS, and this policy condition enforces that requirement.
https://docs.aws.amazon.com/IAM/latest/UserGuide/reference_policies_condition-keys.html#condition-keys-securetransport

---

Question 63

Single choice · Domain: D2 Security › Select an AWS solution that encrypts sensitive application data in transit between EC2-based workloads

A developer is building an application on a fleet of Amazon EC2 Linux instances that run Apache web server. The application needs to send API calls that contain sensitive customer data to a second Linux fleet that also runs Apache within a peered VPC in the same AWS account and AWS Region as the original cluster. All sensitive data must be encrypted in transit. Instances in both fleets have Amazon EBS volumes attached.

Which solution will meet these requirements in the MOST operationally efficient way?

A. Create a security group in each VPC that allows traffic only from the other fleet's security group. Assign each security group to the corresponding instances.

B. Create an AWS Site-to-Site VPN connection between the two peered VPCs. Pass the sensitive data in the API calls through the Site-to-Site VPN connection.

C. Create a customer managed key in AWS KMS to encrypt all EBS volumes. Create an IAM instance profile that has permission to use the customer managed key and attach the key to all instances in both fleets.

D. Request a valid certificate through AWS Certificate Manager (ACM) and redeploy both fleets as AWS Nitro Enclaves. Use the ACM for Nitro Enclaves Amazon Machine Image (AMI) and specify the requested certificate for all instances.

Answer: D. Request a valid certificate through AWS Certificate Manager (ACM) and redeploy both fleets as AWS Nitro Enclaves. Use the ACM for Nitro Enclaves Amazon Machine Image (AMI) and specify the requested certificate for all instances.

Explanation

This question tests the ability to distinguish network access controls and encryption at rest from a solution that provides TLS encryption for API traffic. It also evaluates knowledge of using AWS Certificate Manager with Nitro Enclaves to provision certificates for encrypted communications.

✗ **A** Create a security group in each VPC that allows traffic only from the other fleet's security group. Assign each security group to the corresponding instances.
Incorrect. Security groups provide stateful network access control, but they do not encrypt traffic in transit.
https://docs.aws.amazon.com/vpc/latest/userguide/vpc-security-groups.html

✗ **B** Create an AWS Site-to-Site VPN connection between the two peered VPCs. Pass the sensitive data in the API calls through the Site-to-Site VPN connection.
Incorrect. AWS Site-to-Site VPN is designed to connect a VPC or transit gateway to an external network

such as an on-premises network; it is not created directly between two peered VPCs. Additionally, VPC peering itself does not provide application-level TLS encryption.

https://docs.aws.amazon.com/vpn/latest/s2svpn/VPC_VPN.html

✗ **C** Create a customer managed key in AWS KMS to encrypt all EBS volumes. Create an IAM instance profile that has permission to use the customer managed key and attach the key to all instances in both fleets.
Incorrect. Encrypting EBS volumes protects data at rest on the volumes, not sensitive data transmitted by API calls between the fleets.

https://docs.aws.amazon.com/ebs/latest/userguide/ebs-encryption.html

✓ **D** Request a valid certificate through AWS Certificate Manager (ACM) and redeploy both fleets as AWS Nitro Enclaves. Use the ACM for Nitro Enclaves Amazon Machine Image (AMI) and specify the requested certificate for all instances.
Correct. ACM for Nitro Enclaves enables certificates and private keys to be securely provisioned to Nitro Enclaves, allowing applications to establish TLS-protected communications and encrypt sensitive data in transit.

https://docs.aws.amazon.com/acm/latest/userguide/acm-for-nitro-enclaves.html

Question 64

Single choice · Domain: D2 Security › Securely manage and retrieve application credentials using AWS managed secret stores

A developer owns and supports an application that has database credentials that are stored in environment variables for AWS Lambda functions. The developer needs an alternate storage method for the credentials instead of environment variables in plaintext. Which solution will handle the credentials MOST securely?

A. Store the database credentials as a secret in AWS Secrets Manager. Set the secret's Amazon Resource Name (ARN) as the value of an environment variable. Use the AWS Parameters and Secrets Lambda Extension to retrieve the credentials in the Lambda function.

B. Use base64 encoding for the database credentials. Include the database credentials into the Lambda function's source code as part of a build process. Update the Lambda function's code to decode the credentials before the credentials are used.

C. Store the database credentials as a string type parameter in the AWS Systems Manager Parameter Store. Set the parameter's Amazon Resource Name (ARN) as the value of an environment variable. Use the AWS Parameters and Secrets Lambda Extension to retrieve the credentials from the Lambda function.

D. Use AWS CloudFormation to deploy the application. Ensure that the NoEcho property is set to true for the parameters that contain the database credentials.

Answer: A. Store the database credentials as a secret in AWS Secrets Manager. Set the secret's Amazon Resource Name (ARN) as the value of an environment variable. Use the AWS Parameters and Secrets Lambda Extension to retrieve the credentials in the Lambda function.

Explanation

This question tests the ability to select the most secure AWS service and retrieval method for database credentials used by Lambda functions. Secrets Manager is designed for sensitive credentials and

Free demo · full exam: 595 questions

supports secure retrieval and secret rotation.

✓ **A** Store the database credentials as a secret in AWS Secrets Manager. Set the secret's Amazon Resource Name (ARN) as the value of an environment variable. Use the AWS Parameters and Secrets Lambda Extension to retrieve the credentials in the Lambda function.
Correct. AWS Secrets Manager is designed to securely store sensitive information such as database credentials. Storing only the secret ARN in the environment variable avoids placing the plaintext credentials there, and the Lambda extension retrieves the secret securely at runtime.

https://docs.aws.amazon.com/secretsmanager/latest/userguide/retrieving-secrets_lambda.html

✗ **B** Use base64 encoding for the database credentials. Include the database credentials into the Lambda function's source code as part of a build process. Update the Lambda function's code to decode the credentials before the credentials are used.
Incorrect. Base64 is an encoding mechanism, not encryption, so anyone who obtains the source package can decode the credentials. Embedding credentials in application source code also increases the risk of exposure.

https://docs.aws.amazon.com/lambda/latest/dg/configuration-envvars.html

✗ **C** Store the database credentials as a string type parameter in the AWS Systems Manager Parameter Store. Set the parameter's Amazon Resource Name (ARN) as the value of an environment variable. Use the AWS Parameters and Secrets Lambda Extension to retrieve the credentials from the Lambda function.
Incorrect. A String parameter stores the credentials as plaintext in Parameter Store. A SecureString parameter would provide encryption, but Secrets Manager is the purpose-built service for securely storing and managing credentials, including rotation capabilities.

https://docs.aws.amazon.com/systems-manager/latest/userguide/systems-manager-parameter-store.html

✗ **D** Use AWS CloudFormation to deploy the application. Ensure that the NoEcho property is set to true for the parameters that contain the database credentials.
Incorrect. NoEcho masks parameter values in CloudFormation stack descriptions and events, but it does not provide a dedicated secure credential-management solution or prevent all forms of exposure. Secrets Manager is more appropriate for storing and retrieving database credentials.

https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/parameters-section-structure.html

Question 65

Single choice · Domain: D2 Security › Enforce encrypted HTTPS transport for Amazon S3 requests using the aws:SecureTransport condition key

An application that is running on Amazon EC2 instances stores data in an Amazon S3 bucket. All the data must be encrypted in transit.

How can a developer ensure that all traffic to the S3 bucket is encrypted?

A. Install certificates on the EC2 instances.

B. Create a private VPC endpoint.

C. Configure the S3 bucket with server-side encryption with AWS KMS managed encryption keys (SSE-KMS).

D. Create an S3 bucket policy that denies traffic when the value for the aws:SecureTransport condition key is false.

Answer: D. Create an S3 bucket policy that denies traffic when the value for the aws:SecureTransport condition key is false.

Dumpsgate

Question 66

Single choice · Domain: D2 Security › Securely manage and access third-party API credentials used by AWS Lambda

A company operates a web-based loan processing application. The application's UI is implemented in JavaScript. The frontend transmits application data securely by using HTTPS to Amazon API Gateway, which invokes an AWS Lambda function in private subnets. The Lambda function interacts with third-party credit check APIs that require persistent API keys. The company enforces strict policies to ensure that personally identifiable information (PII) and sensitive credentials are never exposed in client code, request paths, headers, or logs. The company needs a solution to manage the API keys that the Lambda function must use. Which solution will meet this requirement in the MOST secure way?

A. Store the API keys as encrypted environment variables by using an AWS KMS key. Configure the execution role of the Lambda function to have permissions to securely decrypt the environment variables at runtime.

B. Pass the API keys to the Lambda function by including the keys as URL query parameters in each HTTPS request. Rely on TLS for encryption of the payload and response. Use API Gateway logging controls to manage what query parameters are logged.

C. Bundle the API keys inside the minified client-side JavaScript. Configure the web application to call the Lambda function by using an API Gateway HTTP API, cross-origin resource sharing (CORS) restrictions, domain allowlists, and frequent rotation.

D. Store the API keys as resource metadata tags on the Lambda function. Configure the Lambda function to read its own tags at startup by using the AWS SDK. Use IAM conditions to control access to the keys when the function retrieves the tags.

Answer: A. Store the API keys as encrypted environment variables by using an AWS KMS key. Configure the execution role of the Lambda function to have permissions to securely decrypt the environment variables at runtime.

Explanation

This question tests selecting a method that keeps persistent API keys out of client-side code, request data, and logs while allowing the Lambda function to decrypt and use them at runtime.

✓ **A** Store the API keys as encrypted environment variables by using an AWS KMS key. Configure the execution role of the Lambda function to have permissions to securely decrypt the environment variables at runtime.
Correct. Lambda environment variables can be encrypted with an AWS KMS customer managed key, and the function's execution role can be granted permission to decrypt them at runtime. The keys are not exposed in client code, request paths, headers, or ordinary application payloads. For production secret management, AWS also documents Secrets Manager as a purpose-built option for storing credentials. and.

https://docs.aws.amazon.com/lambda/latest/dg/configuration-envvars.html
https://docs.aws.amazon.com/secretsmanager/latest/userguide/intro.html

✗ **B** Pass the API keys to the Lambda function by including the keys as URL query parameters in each HTTPS request. Rely on TLS for encryption of the payload and response. Use API Gateway logging controls to manage what query parameters are logged.
Incorrect. HTTPS protects the request in transit, but credentials in URL query parameters can be exposed through access logs, tracing, browser history, proxies, monitoring systems, and other request metadata. Logging controls reduce some exposure but do not make transmitting persistent credentials in URLs secure.

https://docs.aws.amazon.com/apigateway/latest/developerguide/api-gateway-known-issues.html

✗ **C** Bundle the API keys inside the minified client-side JavaScript. Configure the web application to call the Lambda function by using an API Gateway HTTP API, cross-origin resource sharing (CORS) restrictions, domain allowlists, and frequent rotation.
Incorrect. Any credential delivered in client-side JavaScript can be retrieved by users or attackers through browser developer tools, downloaded assets, source inspection, or runtime instrumentation. CORS, domain restrictions, minification, and rotation do not keep the API keys confidential.

https://docs.aws.amazon.com/secretsmanager/latest/userguide/best-practices.html

✗ **D** Store the API keys as resource metadata tags on the Lambda function. Configure the Lambda function to read its own tags at startup by using the AWS SDK. Use IAM conditions to control access to the keys when the function retrieves the tags.
Incorrect. Lambda tags are resource metadata for organization and access-control purposes, not a secure secret-storage mechanism. Tags can be exposed through resource-management APIs and may be visible to principals with tagging or describe permissions; they also do not provide dedicated secret encryption, rotation, or auditing features.

https://docs.aws.amazon.com/lambda/latest/dg/tagging.html

Question 67

Single choice · Domain: D2 Security › Configure server-side encryption for Amazon SQS queues

A developer is building two microservices that use an Amazon SQS queue to communicate. The messages that the microservices send to one another contain sensitive information. The developer must ensure the messages are stored and are encrypted at rest.

Which solution will meet these requirements?

A. Add a policy to the SQS queue that sets the aws:SecureTransport condition.

B. Configure the microservices to use the server-side encryption (SSE) option within the messages to send messages to the SQS queue.

C. Enable the server-side encryption (SSE) option on the SQS queue. Ensure the microservices contain the sensitive information within the body of the messages.

D. Transmit sensitive information as part of the attributes of the messages that the microservices send.

Answer: C. Enable the server-side encryption (SSE) option on the SQS queue. Ensure the microservices contain the sensitive information within the body of the messages.

Explanation

This question tests the ability to protect sensitive SQS message data at rest by enabling server-side encryption on the queue. SQS server-side encryption encrypts message bodies and message attributes while they are stored.

✗ **A** Add a policy to the SQS queue that sets the aws:SecureTransport condition.
Incorrect. The aws:SecureTransport condition requires requests to use HTTPS/TLS, protecting data in transit, but it does not enable encryption of messages at rest.

https://docs.aws.amazon.com/AWSSimpleQueueService/latest/SQSDeveloperGuide/sqs-using-identity-based-policies.html

✗ **B** Configure the microservices to use the server-side encryption (SSE) option within the messages to

send messages to the SQS queue.
Incorrect. Amazon SQS server-side encryption is configured on the queue, not within individual messages. SQS then encrypts messages while they are stored.

https://docs.aws.amazon.com/AWSSimpleQueueService/latest/SQSDeveloperGuide/sqs-server-side-encryption.html

✓ **C** Enable the server-side encryption (SSE) option on the SQS queue. Ensure the microservices contain the sensitive information within the body of the messages.
Correct. Enabling server-side encryption on the SQS queue encrypts message data at rest, including message bodies and attributes. The sensitive information can remain in the message body.

https://docs.aws.amazon.com/AWSSimpleQueueService/latest/SQSDeveloperGuide/sqs-server-side-encryption.html

✗ **D** Transmit sensitive information as part of the attributes of the messages that the microservices send.
Incorrect. Placing sensitive information in message attributes does not itself encrypt the data at rest. Encryption must be enabled on the SQS queue; server-side encryption protects both message bodies and attributes.

https://docs.aws.amazon.com/AWSSimpleQueueService/latest/SQSDeveloperGuide/sqs-server-side-encryption.html

---

Question 68

Multiple choice · Domain: D2 Security › Implement tenant isolation with IAM session tags and attribute-based access control

A developer is building a multi-tenant application using AWS Lambda, Amazon S3, and Amazon DynamoDB. Each S3 object prefix represents a tenant name, and DynamoDB uses the tenant name as the partition key.

The developer must prevent cross-tenant data access during processing.

Which combination of actions will meet this requirement? (Select THREE.)

A. Create a data access IAM role that allows the sts:TagSession action.

B. Allow the Lambda execution role to assume the data access role.

C. Configure IAM policies on the data access role to allow S3 and DynamoDB access only when resource attributes match the tenant session tag.

D. Create a resource-based policy on DynamoDB based on principal tags.

E. Create a resource control policy (RCP) for the S3 bucket.

F. Configure the Lambda function to assume the data access role and pass the tenant name as a session tag.

Answer

- B. **Allow the Lambda execution role to assume the data access role.**
- C. **Configure IAM policies on the data access role to allow S3 and DynamoDB access only when resource attributes match the tenant session tag.**
- F. **Configure the Lambda function to assume the data access role and pass the tenant name as a session tag.**

Explanation

This question tests the ability to use temporary role sessions and tenant session tags to enforce attribute-based access controls for Amazon S3 prefixes and DynamoDB partition keys.

Free demo · full exam: 595 questions

✗ **A** Create a data access IAM role that allows the sts:TagSession action.
Incorrect as stated. Permission to pass session tags must be granted to the calling principal, and the target role's trust policy must allow sts:TagSession. Merely creating a role that allows this action does not establish the complete required trust and caller permissions.

https://docs.aws.amazon.com/IAM/latest/UserGuide/id_session-tags.html

✓ **B** Allow the Lambda execution role to assume the data access role.
Correct. The Lambda execution role must have permission to call sts:AssumeRole on the tenant data access role before the function can obtain temporary credentials for tenant-scoped access.

https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_temp_request.html

✓ **C** Configure IAM policies on the data access role to allow S3 and DynamoDB access only when resource attributes match the tenant session tag.
Correct. Policies on the assumed role can use the tenant session tag as an IAM policy variable or condition value, restricting S3 object prefixes and DynamoDB partition-key access to the current tenant.

https://docs.aws.amazon.com/IAM/latest/UserGuide/id_session-tags.html

✗ **D** Create a resource-based policy on DynamoDB based on principal tags.
Incorrect. The required control is an identity-based policy on the tenant-scoped access role using the session tag. A DynamoDB resource-based policy based on principal tags is not the appropriate mechanism for enforcing these per-tenant partition-key restrictions.

https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/specifying-conditions.html

✗ **E** Create a resource control policy (RCP) for the S3 bucket.
Incorrect. An Organizations resource control policy is an organization-level guardrail and is not the mechanism for dynamically restricting each Lambda invocation to the tenant represented by a session tag.

https://docs.aws.amazon.com/organizations/latest/userguide/orgs_manage_policies_rcps.html

✓ **F** Configure the Lambda function to assume the data access role and pass the tenant name as a session tag.
Correct. The function must obtain temporary credentials for the data access role and pass the tenant name as a session tag. The role policies can then use that tag to limit access to the corresponding S3 prefix and DynamoDB partition key.

https://docs.aws.amazon.com/IAM/latest/UserGuide/id_session-tags.html

Question 69

Single choice · Domain: D2 Security › Use IAM policies to grant permissions to an individual user without modifying group policies

A company wants to ensure that only one user from its Admin group has the permanent right to delete an Amazon EC2 resource. The company must not modify the existing Admin group policy.

What should a developer use to meet these requirements?

A. AWS managed policy
B. Inline policy
C. IAM trust relationship
D. AWS STS

Answer: B. Inline policy

Dumpsgate

Question 70

Single choice · Domain: D3 Deployment › Configure Elastic Beanstalk application version lifecycle settings to retain source bundles in Amazon S3

A development team uses AWS Elastic Beanstalk for application deployment. The development team has configured the application version lifecycle policy to limit the number of application versions to What should the development team do in the Elastic Beanstalk application version lifecycle settings to retain the source code in the S3 bucket?

25. However, even with the application version lifecycle policy, the source bundle is deleted from the Amazon S3 source bucket.

A. Enable versioning on the source bundle S3 bucket.

B. Disable the S3 bucket lifecycle policy to avoid the archiving of the source bundle.

C. Update the Elastic Beanstalk application version lifecycle policy to increase the version quota to

E. 50.

D. Update the Elastic Beanstalk application version lifecycle policy to retain the source bundle in Amazon S3.

Answer: D. Update the Elastic Beanstalk application version lifecycle policy to retain the source bundle in Amazon S3.

Explanation

Free demo · full exam: 595 questions

This question tests configuring an Elastic Beanstalk application version lifecycle policy so that source bundles are retained in the associated Amazon S3 bucket when application versions are deleted. The setting is independent of the number of versions retained or S3 bucket versioning.

✗ **A** Enable versioning on the source bundle S3 bucket.
Incorrect. S3 versioning does not change Elastic Beanstalk's lifecycle behavior for deleting source bundles. The Elastic Beanstalk lifecycle policy must be configured to retain the source bundle.

https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/applications-versions.html

✗ **B** Disable the S3 bucket lifecycle policy to avoid the archiving of the source bundle.
Incorrect. An S3 bucket lifecycle policy is separate from the Elastic Beanstalk application version lifecycle policy, and disabling it does not configure Elastic Beanstalk to retain source bundles.

https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/applications-versions.html

✗ **C** Update the Elastic Beanstalk application version lifecycle policy to increase the version quota to
Incorrect. Increasing the number of application versions retained does not ensure that source bundles are retained in Amazon S3. The policy must explicitly retain the source bundle.

https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/applications-versions.html

✓ **D** Update the Elastic Beanstalk application version lifecycle policy to retain the source bundle in Amazon S3.
Correct. Elastic Beanstalk application version lifecycle settings include an option to delete or retain the source bundle in Amazon S3 when an application version is deleted. Selecting the retention option preserves the source bundle.

https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/applications-versions.html

✗ **E** 50.
Incorrect. Increasing the application version quota to 50 does not configure Elastic Beanstalk to retain source bundles in Amazon S3. Source bundle retention must be enabled directly in the application version lifecycle policy.

https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/applications-versions.html

Question 71

Single choice · Domain: D3 Deployment › Select an AWS CodeDeploy deployment configuration for gradual traffic shifting to AWS Lambda versions

A company is planning to use AWS CodeDeploy to deploy an application to AWS Lambda. 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, every 10 minutes, the company must route another 10% of live traffic to the new version of the deployed application until all live traffic is routed to the new version.

Which CodeDeploy predefined configuration will meet these requirements?

A. CodeDeployDefault.OneAtATime
B. CodeDeployDefault.LambdaCanary10Percent10Minutes
C. CodeDeployDefault.LambdaLinear10PercentEvery10Minutes
D. CodeDeployDefault.ECSLinear10PercentEvery3Minutes

Answer: C. CodeDeployDefault.LambdaLinear10PercentEvery10Minutes

Dumpsgate

Question 72

Single choice · Domain: D3 Deployment › Use AWS Lambda versions and aliases to promote and roll back function deployments across environments

A developer is designing a new feature for an existing application. The new feature uses an AWS Lambda function. The developer wants to test the Lambda function safely in a development AWS account and a testing AWS account before deploying the function into a production AWS account.

The developer must roll back the function if issues are found. Which solution will meet these requirements with the LEAST operational overhead?

A. Create new Lambda function versions and aliases for the development, testing, and production accounts. After successful testing, update the production alias to point to the function. Roll back to the most recent stable function version if issues are discovered.

B. Deploy the Lambda function to the development, testing, and production accounts separately after successfully testing the function in each environment.

C. Use Lambda layers to separate the code and the libraries that are associated with each AWS account. Deploy and test the Lambda function with each Lambda layer in each environment.

D. Update the Lambda function code to use environment variables for the development, testing, and production accounts. Use the environment variables to make code updates available to specific environments.

Answer: A. Create new Lambda function versions and aliases for the development, testing, and production accounts. After successful testing, update the production alias to point to the function. Roll back to the most recent stable function version if issues are discovered.

Explanation

This question tests the ability to deploy immutable Lambda function versions, use aliases to identify stable environments, and roll back by repointing an alias to a previously tested version with minimal operational overhead.

✓ **A** Create new Lambda function versions and aliases for the development, testing, and production accounts. After successful testing, update the production alias to point to the function. Roll back to the most recent stable function version if issues are discovered.
Correct. Published Lambda versions are immutable, and aliases can point to a specific version. Updating the production alias promotes the tested version, while repointing it to the previous stable version provides a fast rollback with minimal operational effort.

https://docs.aws.amazon.com/lambda/latest/dg/configuration-versions.html

✗ **B** Deploy the Lambda function to the development, testing, and production accounts separately after successfully testing the function in each environment.
Incorrect. Separate deployments in each account require additional deployment and synchronization work and do not inherently provide the simple version-and-alias rollback mechanism described in the correct solution.

https://docs.aws.amazon.com/lambda/latest/dg/configuration-versions.html

✗ **C** Use Lambda layers to separate the code and the libraries that are associated with each AWS account. Deploy and test the Lambda function with each Lambda layer in each environment.
Incorrect. Lambda layers package reusable code and dependencies; they are not a deployment promotion or rollback mechanism for application function versions.

https://docs.aws.amazon.com/lambda/latest/dg/chapter-layers.html

✗ **D** Update the Lambda function code to use environment variables for the development, testing, and production accounts. Use the environment variables to make code updates available to specific environments.
Incorrect. Environment variables provide configuration values to a function but do not create immutable application versions or provide a reliable mechanism to promote and roll back code.

https://docs.aws.amazon.com/lambda/latest/dg/configuration-envvars.html

Free demo · full exam: 595 questions

Question 73

Single choice · Domain: D3 Deployment › Identify the rollback behavior of AWS CodeDeploy for failed in-place deployments

A company has implemented AWS CodeDeploy as part of its continuous integration and continuous delivery (CI/CD) stack in the cloud. The company sets up automatic rollbacks while initiating an in-place deployment of a new version of its web application on Amazon EC2 instances. Which result occurs if the deployment of the new version fails validation?

A. CodeDeploy automatically restores the last known successful deployment from a snapshot that is stored in Amazon S3.

B. CodeDeploy switches the Amazon Route 53 alias records back to the last known successful green deployment and terminates the failed blue deployment.

C. CodeDeploy redeploys the last known stable version of the application as a new deployment with a new deployment ID.

D. AWS CodePipeline promotes the most recent deployment that has a SUCCEEDED status to production.

Answer: C. CodeDeploy redeploys the last known stable version of the application as a new deployment with a new deployment ID.

Explanation

This question tests knowledge of how CodeDeploy handles automatic rollback when a deployment fails validation. For an in-place deployment, CodeDeploy redeploys the previously deployed version as a new deployment rather than switching traffic between blue and green environments.

✗ **A** CodeDeploy automatically restores the last known successful deployment from a snapshot that is stored in Amazon S3.
Incorrect. CodeDeploy does not restore the application from an S3 snapshot. Automatic rollback starts a new deployment of the last known good application revision.

https://docs.aws.amazon.com/codedeploy/latest/userguide/deployments-rollback-and-redeploy.html

✗ **B** CodeDeploy switches the Amazon Route 53 alias records back to the last known successful green deployment and terminates the failed blue deployment.
Incorrect. This describes blue/green deployment traffic management, not an in-place deployment. An in-place deployment updates the existing instances and does not use separate blue and green environments or Route 53 alias switching.

https://docs.aws.amazon.com/codedeploy/latest/userguide/deployments-in-place.html

✓ **C** CodeDeploy redeploys the last known stable version of the application as a new deployment with a new deployment ID.
Correct. When automatic rollback is configured and a deployment fails, CodeDeploy deploys the last known good revision again as a new deployment, which receives a new deployment ID.

https://docs.aws.amazon.com/codedeploy/latest/userguide/deployments-rollback-and-redeploy.html

✗ **D** AWS CodePipeline promotes the most recent deployment that has a SUCCEEDED status to production.
Incorrect. CodePipeline does not perform the CodeDeploy rollback operation by promoting a prior successful deployment. CodeDeploy itself handles automatic rollback by initiating a new deployment of the last known good revision.

https://docs.aws.amazon.com/codedeploy/latest/userguide/deployments-rollback-and-redeploy.html

Free demo · full exam: 595 questions

Question 74

Single choice · Domain: D3 Deployment › Configure an Amazon API Gateway canary release to progressively route production traffic to a new deployment

A company uses AWS CloudFormation to deploy an application that uses an Amazon API Gateway REST API with AWS Lambda function integration. The application uses Amazon DynamoDB for data persistence. The application has three stages: development, testing, and production. Each stage uses its own DynamoDB table. The company has encountered unexpected issues when promoting changes to the production stage. A developer needs to route 20% of the traffic to the new production stage API with the next production release. The developer needs to route the remaining 80% of the traffic to the existing production stage. The solution must minimize the number of errors that any single customer experiences. Which approach should the developer take to meet these requirements?

A. Update 20% of the planned changes to the production stage. Deploy the new production stage. Monitor the results. Repeat this process five times to test all planned changes.

B. Update the Amazon Route 53 DNS record entry for the production stage API to use a weighted routing policy. Set the weight to 80. Add a second record for the production domain name. Change the second routing policy to a weighted routing policy. Set the weight of the second policy to 20. Change the alias of the second policy to use the testing stage API.

C. Deploy an Application Load Balancer (ALB) in front of the REST API. Change the production API Amazon Route 53 record to point traffic to the ALB. Register the production and testing stages as targets of the ALB with weights of 80% and 20%, respectively.

D. Configure canary settings for the production stage API. Change the percentage of traffic directed to canary deployment to 20%. Make the planned updates to the production stage. Deploy the changes.

Answer: D. Configure canary settings for the production stage API. Change the percentage of traffic directed to canary deployment to 20%. Make the planned updates to the production stage. Deploy the changes.

Explanation

This question tests the ability to use API Gateway stage canary settings to direct a controlled percentage of production requests to a new deployment while retaining the existing deployment for the remaining traffic. Canary deployments are designed to reduce production risk during application releases.

✗ **A** Update 20% of the planned changes to the production stage. Deploy the new production stage. Monitor the results. Repeat this process five times to test all planned changes.
Incorrect. This approach incrementally changes the application configuration but does not route 20% of production traffic to a separate new deployment while keeping 80% on the existing deployment. It also does not provide API Gateway canary controls for comparing deployments.

https://docs.aws.amazon.com/apigateway/latest/developerguide/api-gateway-deploy-api.html

✗ **B** Update the Amazon Route 53 DNS record entry for the production stage API to use a weighted routing policy. Set the weight to 80. Add a second record for the production domain name. Change the second routing policy to a weighted routing policy. Set the weight of the second policy to 20. Change the alias of the second policy to use the testing stage API.
Incorrect. Route 53 weighted DNS routing is not the appropriate mechanism for API Gateway stage

canary deployments. DNS caching and resolver behavior can prevent the intended per-request traffic split and can cause customers to remain directed to one endpoint. API Gateway provides native canary settings for routing a percentage of stage traffic to a new deployment.

https://docs.aws.amazon.com/apigateway/latest/developerguide/api-gateway-deploy-api.html#api-gateway-deploy-api-canary

✗ **C** Deploy an Application Load Balancer (ALB) in front of the REST API. Change the production API Amazon Route 53 record to point traffic to the ALB. Register the production and testing stages as targets of the ALB with weights of 80% and 20%, respectively.
Incorrect. An ALB is unnecessary and does not directly provide the required API Gateway stage canary deployment behavior. API Gateway canary settings can route a specified percentage of traffic between the existing stage deployment and the canary deployment without introducing another load-balancing layer.

https://docs.aws.amazon.com/apigateway/latest/developerguide/api-gateway-deploy-api.html#api-gateway-deploy-api-canary

✓ **D** Configure canary settings for the production stage API. Change the percentage of traffic directed to canary deployment to 20%. Make the planned updates to the production stage. Deploy the changes.
Correct. API Gateway canary deployments allow a specified percentage of traffic to be sent to a new deployment while the remaining traffic continues to use the current deployment. Setting the canary percentage to 20% limits exposure to the new production version and helps reduce the number of errors experienced by any individual customer during the release.

https://docs.aws.amazon.com/apigateway/latest/developerguide/api-gateway-deploy-api.html#api-gateway-deploy-api-canary

Question 75

Single choice · Domain: D3 Deployment › Configure an automated CI/CD pipeline to build and deploy AWS Lambda applications from source control

A developer is writing a new serverless application for a company. Several other developers must collaborate on the code for this application, and the company expects frequent changes to the code.

The developer needs to deploy the code from source control to AWS Lambda with the fewest number of manual steps.

Which strategy for the build and deployment should the developer use to meet these requirements?

A. Build the code locally, and then upload the code into the source control system. When a release is needed, run AWS CodePipeline to extract the uploaded build and deploy the resources.

B. Use the AWS SAM CLI to build and deploy the application from the developer's local machine with the latest version checked out locally.

C. Use AWS CodeBuild and AWS CodePipeline to invoke builds and corresponding deployments when configured source-controlled branches have pull requests merged into them.

D. Use the Lambda console to upload a .zip file of the application that is created by the AWS SAM CLI build command.

Answer: C. Use AWS CodeBuild and AWS CodePipeline to invoke builds and corresponding deployments when configured source-controlled branches have pull requests merged into them.

Explanation

Free demo · full exam: 595 questions

This question tests the ability to select a managed, event-driven build and deployment workflow that automatically builds and deploys Lambda code after changes are merged into a source-controlled branch.

✗ **A** Build the code locally, and then upload the code into the source control system. When a release is needed, run AWS CodePipeline to extract the uploaded build and deploy the resources.
Incorrect. Building locally introduces manual steps and does not provide a consistent automated build process for collaborators. CodeBuild is designed to compile, test, and package source code as part of an automated pipeline.

https://docs.aws.amazon.com/codebuild/latest/userguide/welcome.html

✗ **B** Use the AWS SAM CLI to build and deploy the application from the developer's local machine with the latest version checked out locally.
Incorrect. Although the SAM CLI can build and deploy applications, this approach depends on a developer's local machine and requires manual execution, so it does not provide the requested automated collaboration workflow.

https://docs.aws.amazon.com/serverless-application-model/latest/developerguide/using-sam-cli-deploy.html

✓ **C** Use AWS CodeBuild and AWS CodePipeline to invoke builds and corresponding deployments when configured source-controlled branches have pull requests merged into them.
Correct. CodePipeline can detect source changes and orchestrate build and deployment stages, while CodeBuild performs the repeatable build process. Configuring the pipeline to run after changes are merged provides automated deployment with minimal manual steps. and.

https://docs.aws.amazon.com/codepipeline/latest/userguide/welcome.html
https://docs.aws.amazon.com/codebuild/latest/userguide/welcome.html

✗ **D** Use the Lambda console to upload a .zip file of the application that is created by the AWS SAM CLI build command.
Incorrect. Uploading a package through the Lambda console is a manual deployment process and does not integrate source control, automated builds, or continuous deployment.

https://docs.aws.amazon.com/lambda/latest/dg/configuration-function-zip.html

Question 76

Single choice · Domain: D3 Deployment › Use AWS Lambda versions and aliases to promote and roll back function deployments

A developer is designing a new feature for an existing application that uses an AWS Lambda function.

The developer wants to test the Lambda function safely in development and testing AWS accounts before deploying the function to a production AWS account. The developer must be able to roll back the function if issues are discovered.

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

A. Create new Lambda function versions and aliases for the development, testing, and production accounts. After successful testing, update the production alias to point to the function version. Roll back to the most recent stable version if issues are discovered.

B. Deploy the Lambda function separately to the development, testing, and production accounts after testing the function in each environment.

C. Use Lambda layers to separate the code and libraries for each AWS account and deploy the function with different layers.

D. Use environment variables to differentiate development, testing, and production behavior.

Answer: A. Create new Lambda function versions and aliases for the development, testing, and production accounts. After successful testing, update the production alias to point to the function version. Roll back to the most recent stable version if issues are discovered.

Explanation

This question tests selecting a low-overhead Lambda deployment strategy that supports testing immutable function versions and rolling back by moving an alias to a previously stable version.

✓ **A** Create new Lambda function versions and aliases for the development, testing, and production accounts. After successful testing, update the production alias to point to the function version. Roll back to the most recent stable version if issues are discovered.
Correct. Lambda versions provide immutable snapshots of function code and configuration, while aliases point to specific versions and can be moved to a previous stable version for rollback. This avoids maintaining separate function implementations and has the least operational overhead.

https://docs.aws.amazon.com/lambda/latest/dg/configuration-versions.html

✗ **B** Deploy the Lambda function separately to the development, testing, and production accounts after testing the function in each environment.
Incorrect. Separately deploying and managing the function in each account increases operational overhead and does not itself provide the straightforward version-based rollback mechanism offered by Lambda aliases.

https://docs.aws.amazon.com/lambda/latest/dg/configuration-aliases.html

✗ **C** Use Lambda layers to separate the code and libraries for each AWS account and deploy the function with different layers.
Incorrect. Lambda layers package shared dependencies and libraries; they are not a deployment promotion or rollback mechanism for function versions.

https://docs.aws.amazon.com/lambda/latest/dg/chapter-layers.html

✗ **D** Use environment variables to differentiate development, testing, and production behavior.
Incorrect. Environment variables can configure runtime behavior, but they do not provide immutable deployment versions or an alias-based rollback process.

https://docs.aws.amazon.com/lambda/latest/dg/configuration-envvars.html

Free demo · full exam: 595 questions

Question 77

Single choice · Domain: D4 Troubleshooting and Optimization › Implement a consistent caching strategy to reduce DynamoDB read latency and read capacity consumption

A healthcare company develops a patient monitoring application that uses AWS AppSync and an AWS Lambda function. The application stores patient data in an Amazon DynamoDB database. The application processes real-time patient vital signs from thousands of IoT devices. The application must support queries from medical staff who access patient historical data. A developer notices significant latency when medical staff query patient histories, specifically for frequently-accessed records of patients currently under observation. The developer observes that the Lambda functions are consuming high read capacity units (RCUs) from DynamoDB. The developer needs to optimize the application's performance and maintain data consistency. Which solution will meet these requirements?

A. Enable response caching in AppSync with a TTL value of 1 hour for all queries. Update the Lambda code to use in-memory caching.

B. Increase the Lambda function memory allocation to maximum. Configure provisioned concurrency for the function.

C. Create a caching Lambda function that aggregates patient data periodically and stores it in Amazon S3. Implement a read-through cache pattern.

D. Implement in-memory caching by using Amazon ElastiCache for frequently-accessed patient records. Configure the Lambda functions to check the cache before querying the database. Set up cache invalidation when patient data is updated.

Answer: D. Implement in-memory caching by using Amazon ElastiCache for frequently-accessed patient records. Configure the Lambda functions to check the cache before querying the database. Set up cache invalidation when patient data is updated.

Explanation

This question tests the ability to diagnose high DynamoDB read utilization and latency for frequently accessed data, then select a shared cache with an appropriate invalidation strategy to preserve consistency.

✗ **A** Enable response caching in AppSync with a TTL value of 1 hour for all queries. Update the Lambda code to use in-memory caching.
Incorrect. A one-hour TTL for all queries can return stale patient information, which is unsuitable when data consistency is required. Lambda in-memory caches are also isolated to individual execution environments and are not a reliable shared cache. AppSync caching documentation.

https://docs.aws.amazon.com/appsync/latest/devguide/enabling-caching.html

✗ **B** Increase the Lambda function memory allocation to maximum. Configure provisioned concurrency for the function.
Incorrect. More Lambda memory and provisioned concurrency can improve function startup and execution performance, but they do not substantially reduce DynamoDB read capacity consumption or provide a shared cache for frequently accessed records. Lambda memory documentation.

https://docs.aws.amazon.com/lambda/latest/dg/configuration-memory.html

✗ **C** Create a caching Lambda function that aggregates patient data periodically and stores it in Amazon S3. Implement a read-through cache pattern.
Incorrect. Amazon S3 is object storage and is not appropriate as a low-latency cache for interactive medical-history queries. Periodic aggregation can also cause stale data and does not provide the required

fine-grained invalidation behavior. Amazon S3 performance guidance.

https://docs.aws.amazon.com/AmazonS3/latest/userguide/optimizing-performance.html

✓ **D** Implement in-memory caching by using Amazon ElastiCache for frequently-accessed patient records. Configure the Lambda functions to check the cache before querying the database. Set up cache invalidation when patient data is updated.
Correct. ElastiCache provides a shared, low-latency in-memory cache that can reduce repeated DynamoDB reads and lower RCU consumption. Having Lambda check the cache first and invalidate or update cached records when the source data changes helps maintain consistency. ElastiCache caching strategies documentation.

https://docs.aws.amazon.com/AmazonElastiCache/latest/dg/Strategies.html

---

Question 78

Single choice · Domain: D4 Troubleshooting and Optimization › Use CloudWatch Logs Insights to query and analyze application logs

A developer is debugging an issue with an application that is based on an AWS Lambda function. The Lambda function intermittently fails during a 1-hour window. The developer needs to investigate the root cause of the intermittent failures. The application sends logs to an Amazon CloudWatch Logs log group. The developer must be able to collect logs that relate to Lambda function failures. The developer must capture the dates and times at which the failures occurred. Which solution will meet these requirements in the MOST operationally efficient way?

A. Use the AWS CLI in AWS CloudShell to browse the CloudWatch Logs log group to search for the error messages.

B. Use CloudWatch Logs Insights to run a query on the log group that searches for specific patterns that match the error messages.

C. Download log files from the CloudWatch Logs log group to the developer's local machine. Use a local text search tool to search for the error messages.

D. Export the CloudWatch Logs log group to an Amazon S3 bucket. Use Amazon Athena to run a SQL query against the bucket to search for the error messages.

Answer: B. Use CloudWatch Logs Insights to run a query on the log group that searches for specific patterns that match the error messages.

Explanation

This question tests the ability to efficiently investigate intermittent Lambda failures by querying CloudWatch Logs for error patterns and identifying the timestamps of matching log events.

✗ **A** Use the AWS CLI in AWS CloudShell to browse the CloudWatch Logs log group to search for the error messages.
Incorrect. Although the AWS CLI can retrieve and filter log events, manually browsing and searching logs is less operationally efficient than using CloudWatch Logs Insights for an interactive, time-bounded analysis.

https://docs.aws.amazon.com/cli/latest/reference/logs/filter-log-events.html

✓ **B** Use CloudWatch Logs Insights to run a query on the log group that searches for specific patterns that match the error messages.
Correct. CloudWatch Logs Insights is designed to interactively query and analyze log data, filter events by

error patterns, and return timestamps and other relevant fields within a selected time range.
https://docs.aws.amazon.com/AmazonCloudWatch/latest/logs/AnalyzingLogData.html

✗ **C** Download log files from the CloudWatch Logs log group to the developer's local machine. Use a local text search tool to search for the error messages.
Incorrect. Downloading logs and searching them locally introduces unnecessary data transfer and manual processing, making it less operationally efficient than querying the logs directly with CloudWatch Logs Insights.
https://docs.aws.amazon.com/AmazonCloudWatch/latest/logs/AnalyzingLogData.html

✗ **D** Export the CloudWatch Logs log group to an Amazon S3 bucket. Use Amazon Athena to run a SQL query against the bucket to search for the error messages.
Incorrect. Exporting logs to Amazon S3 and querying them with Athena adds extra export, storage, and query configuration steps and is not the most operationally efficient approach for investigating a one-hour window of intermittent failures. CloudWatch Logs Insights can query the log group directly.
https://docs.aws.amazon.com/AmazonCloudWatch/latest/logs/S3Export.html

Question 79

Single choice · Domain: D4 Troubleshooting and Optimization › Configure Auto Scaling health checks to replace EC2 instances that fail Application Load Balancer target health checks

A company has a two-tier application that runs on Amazon EC2 instances behind an Application Load Balancer (ALB). The EC2 instances are in an Auto Scaling group. The Auto Scaling group is configured with a lifecycle hook that uses the default health checks for the EC2 instances.

During a recent change deployment, the ALB reported HTTP 502 errors. When a developer checked the target group, some of the EC2 instances were marked as unhealthy. However, the Auto Scaling group was not replacing the unhealthy EC2 instances.

Which configuration change should the developer make to replace the unhealthy EC2 instances?

A. Increase the cooldown period of the Auto Scaling group from its default value.
B. Update the lifecycle hook to wait for the application to install.
C. Update the Auto Scaling group's health check from Amazon EC2 to Elastic Load Balancing (ELB).
D. Set the health check grace period for the Auto Scaling group.

Answer: C. Update the Auto Scaling group's health check from Amazon EC2 to Elastic Load Balancing (ELB).

Explanation

This question tests the ability to diagnose a mismatch between ALB target health and Auto Scaling group health monitoring. The developer must configure the Auto Scaling group to use Elastic Load Balancing health checks so that unhealthy targets can be replaced.

✗ **A** Increase the cooldown period of the Auto Scaling group from its default value.
Incorrect. The cooldown period controls how long Amazon EC2 Auto Scaling waits after a scaling activity before starting another scaling activity. It does not cause the Auto Scaling group to use ALB target health status when evaluating instance health.
https://docs.aws.amazon.com/autoscaling/ec2/userguide/ec2-auto-scaling-scaling-cooldowns.html

✗ **B** Update the lifecycle hook to wait for the application to install.

Incorrect. Lifecycle hooks pause instances during launch or termination transitions so that initialization or cleanup tasks can run. They do not change the health-check source used to determine whether an instance should be replaced.

https://docs.aws.amazon.com/autoscaling/ec2/userguide/lifecycle-hooks.html

✓ **C** Update the Auto Scaling group's health check from Amazon EC2 to Elastic Load Balancing (ELB).
Correct. By default, an Auto Scaling group uses EC2 status checks. Configuring ELB health checks allows the group to use the load balancer target group's health status and replace instances that are unhealthy according to the ALB.

https://docs.aws.amazon.com/autoscaling/ec2/userguide/health-checks-overview.html

✗ **D** Set the health check grace period for the Auto Scaling group.
Incorrect. The health check grace period delays health evaluation for newly launched instances, allowing applications time to start. It does not configure the Auto Scaling group to monitor ALB health checks or replace instances based on target group status.

https://docs.aws.amazon.com/autoscaling/ec2/userguide/health-check-grace-period.html

---

Question 80

Single choice · Domain: D4 Troubleshooting and Optimization › Use CloudWatch Logs Insights to query and correlate Lambda logs across multiple log groups

A developer is building a process flow that invokes two AWS Lambda functions. The Lambda functions write logs to Amazon CloudWatch. Each run of the process has a unique request ID that flows to both Lambda functions.

The developer encounters a failure in the process flow. The developer wants to use the request IDs to analyze the flow logs.

Which solution will meet these requirements with the LEAST development effort?

A. Use an AWS SDK to query the logs from Amazon CloudWatch.

B. Export the log data for a specific time range to an Amazon S3 bucket. Use Amazon Athena to query the S3 bucket.

C. Use Amazon CloudWatch Logs Insights to query log groups for the Lambda functions. Filter on the request IDs.

D. Use Amazon CloudWatch Live Tail to examine log groups for both Lambda functions, and check for the error.

Answer: C. Use Amazon CloudWatch Logs Insights to query log groups for the Lambda functions. Filter on the request IDs.

Explanation

This question tests the ability to troubleshoot a distributed Lambda process by searching related log events with a shared request ID. CloudWatch Logs Insights can query multiple log groups and filter results directly, requiring less development effort than exporting or programmatically querying logs.

✗ **A** Use an AWS SDK to query the logs from Amazon CloudWatch.
Incorrect. An SDK-based solution requires writing code to call CloudWatch Logs APIs, retrieve events, and correlate or filter them by request ID. CloudWatch Logs Insights provides this query capability without custom application code.

Free demo · full exam: 595 questions

https://docs.aws.amazon.com/AmazonCloudWatch/latest/logs/AnalyzingLogData.html

✗ **B** Export the log data for a specific time range to an Amazon S3 bucket. Use Amazon Athena to query the S3 bucket.
Incorrect. Exporting logs to Amazon S3 and querying them with Athena adds unnecessary data-export, storage, and query configuration steps. CloudWatch Logs Insights can query the Lambda log groups directly.

https://docs.aws.amazon.com/AmazonCloudWatch/latest/logs/S3Export.html

✓ **C** Use Amazon CloudWatch Logs Insights to query log groups for the Lambda functions. Filter on the request IDs.
Correct. CloudWatch Logs Insights can query one or more log groups and filter log events using the shared request ID, making it the solution with the least development effort for analyzing the process flow.

https://docs.aws.amazon.com/AmazonCloudWatch/latest/logs/AnalyzingLogData.html

✗ **D** Use Amazon CloudWatch Live Tail to examine log groups for both Lambda functions, and check for the error.
Incorrect. CloudWatch Logs Live Tail is intended for viewing and filtering logs as they are ingested in near real time. It is not the best tool for retrospectively querying and correlating completed process-flow logs by request ID.

https://docs.aws.amazon.com/AmazonCloudWatch/latest/logs/CloudWatchLogs_LiveTail.html

Dumpsgate