

# Exam Questions AWS-Certified-Developer-Associate

Amazon AWS Certified Developer - Associate

<https://www.2passeasy.com/dumps/AWS-Certified-Developer-Associate/>



### NEW QUESTION 1

A company has a multi-node Windows legacy application that runs on premises. The application uses a network shared folder as a centralized configuration repository to store configuration files in .xml format. The company is migrating the application to Amazon EC2 instances. As part of the migration to AWS, a developer must identify a solution that provides high availability for the repository.

Which solution will meet this requirement MOST cost-effectively?

- A. Mount an Amazon Elastic Block Store (Amazon EBS) volume onto one of the EC2 instance
- B. Deploy a file system on the EBS volum
- C. Use the host operating system to share a folde shared folder.
- D. Update the application code to read and write configuration files from the shared folder.
- E. Deploy a micro EC2 instance with an instance store volum
- F. Use the host operating system to share a folde
- G. Update the application code to read and write configuration files from the shared folder.
- H. Create an Amazon S3 bucket to host the repositor
- I. Migrate the existing .xml files to the S3 bucke
- J. Update the application code to use the AWS SDK to read and write configuration files from Amazon S3.
- K. Create an Amazon S3 bucket to host the repositor
- L. Migrate the existing .xml files to the S3 bucke
- M. Mount the S3 bucket to the EC2 instances as a local volum
- N. Update the application code to read and write configuration files from the disk.

**Answer: C**

#### Explanation:

Amazon S3 is a service that provides highly scalable, durable, and secure object storage. The developer can create an S3 bucket to host the repository and migrate the existing .xml files to the S3 bucket. The developer can update the application code to use the AWS SDK to read and write configuration files from S3. This solution will meet the requirement of high availability for the repository in a cost-effective way.

References:

? [Amazon Simple Storage Service (S3)]

? [Using AWS SDKs with Amazon S3]

### NEW QUESTION 2

A company needs to deploy all its cloud resources by using AWS CloudFormation templates A developer must create an Amazon Simple Notification Service (Amazon SNS) automatic notification to help enforce this rule. The developer creates an SNS topic and subscribes the email address of the company's security team to the SNS topic.

The security team must receive a notification immediately if an IAM role is created without the use of CloudFormation.

Which solution will meet this requirement?

- A. Create an AWS Lambda function to filter events from CloudTrail if a role was created without CloudFormation Configure the Lambda function to publish to the SNS topi
- B. Create an Amazon EventBridge schedule to invoke the Lambda function every 15 minutes
- C. Create an AWS Fargate task in Amazon Elastic Container Service (Amazon ECS) to filter events from CloudTrail if a role was created without CloudFormation Configure the Fargate task to publish to the SNS topic Create an Amazon EventBridge schedule to run the Fargate task every 15 minutes
- D. Launch an Amazon EC2 instance that includes a script to filter events from CloudTrail if a role was created without CloudFormatio
- E. Configure the script to publish to the SNS topi
- F. Create a cron job to run the script on the EC2 instance every 15 minutes.
- G. Create an Amazon EventBridge rule to filter events from CloudTrail if a role was created without CloudFormation Specify the SNS topic as the target of the EventBridge rule.

**Answer: D**

#### Explanation:

Creating an Amazon EventBridge rule is the most efficient and scalable way to monitor and react to events from CloudTrail, such as the creation of an IAM role without CloudFormation. EventBridge allows you to specify a filter pattern to match the events you are interested in, and then specify an SNS topic as the target to send notifications. This solution does not require any additional resources or code, and it can trigger notifications in near real-time. The other solutions involve creating and managing additional resources, such as Lambda functions, Fargate tasks, or EC2 instances, and they rely on polling CloudTrail events every 15 minutes, which can introduce delays and increase costs. References

? Using Amazon EventBridge rules to process AWS CloudTrail events

? Using AWS CloudFormation to create and manage AWS Batch resources

? How to use AWS CloudFormation to configure auto scaling for Amazon Cognito and AWS AppSync

? Using AWS CloudFormation to automate the creation of AWS WAF web ACLs, rules, and conditions

### NEW QUESTION 3

A developer has observed an increase in bugs in the AWS Lambda functions that a development team has deployed in its Node.js application. To minimize these bugs, the developer wants to implement automated testing of Lambda functions in an environment that closely simulates the Lambda environment.

The developer needs to give other developers the ability to run the tests locally. The developer also needs to integrate the tests into the team's continuous integration and continuous delivery (CI/CD) pipeline before the AWS Cloud Development Kit (AWS CDK) deployment.

Which solution will meet these requirements?

- A. Create sample events based on the Lambda documentatio
- B. Create automated test scripts that use the cdk local invoke command to invoke the Lambda function
- C. Check the response Document the test scripts for the other developers on the team Update the CI/CD pipeline to run the test scripts.
- D. Install a unit testing framework that reproduces the Lambda execution environmen
- E. Create sample events based on the Lambda Documentation Invoke the handler function by using a unit testing framewor framework for the other developers on the tea
- F. Check the response Document how to run the unit testing.
- G. Update the CI/CD pipeline to run the unit testing framewor
- H. Install the AWS Serverless Application Model (AWS SAM) CLI tool Use the Sam local generate-event command to generate sample events for me automated test

- I. Create automated test scripts that use the Sam local invoke command to invoke the Lambda function
- J. Check the response Document the test scripts for the other developers on the team Update the CI/CD pipeline to run the test scripts.
- K. Create sample events based on the Lambda documentatio
- L. Create a Docker container from the Node is base image to invoke the Lambda function
- M. Check the response Document how to run the Docker container for the more developers on the team update the CI/CD pipeline to run the Docker container.

**Answer:** C

**Explanation:**

This solution will meet the requirements by using AWS SAM CLI tool, which is a command line tool that lets developers locally build, test, debug, and deploy serverless applications defined by AWS SAM templates. The developer can use sam local generate-event command to generate sample events for different event sources such as API Gateway or S3. The developer can create automated test scripts that use sam local invoke command to invoke Lambda functions locally in an environment that closely simulates Lambda environment. The developer can check the response from Lambda functions and document how to run the test scripts for other developers on the team. The developer can also update CI/CD pipeline to run these test scripts before deploying with AWS CDK. Option A is not optimal because it will use cdk local invoke command, which does not exist in AWS CDK CLI tool. Option B is not optimal because it will use a unit testing framework that reproduces Lambda execution environment, which may not be accurate or consistent with Lambda environment. Option D is not optimal because it will create a Docker container from Node.js base image to invoke Lambda functions, which may introduce additional overhead and complexity for creating and running Docker containers.

References: [AWS Serverless Application Model (AWS SAM)], [AWS Cloud Development Kit (AWS CDK)]

**NEW QUESTION 4**

A company runs a payment application on Amazon EC2 instances behind an Application Load Balance The EC2 instances run in an Auto Scaling group across multiple Availability Zones. The application needs to retrieve application secrets during the application startup and export the secrets as environment variables. These secrets must be encrypted at rest and need to be rotated every month. Which solution will meet these requirements with the LEAST development effort?

- A. Save the secrets in a text file and store the text file in Amazon S3 Provision a customer managed key Use the key for secret encryption in Amazon S3 Read the contents of the text file and read the export as environment variables Configure S3 Object Lambda to rotate the text file every month
- B. Save the secrets as strings in AWS Systems Manager Parameter Store and use the default AWS Key Management Service (AWS KMS) key Configure an Amazon EC2 user data script to retrieve the secrets during the startup and export as environment variables Configure an AWS Lambda function to rotate the secrets in Parameter Store every month.
- C. Save the secrets as base64 encoded environment variables in the application propertie
- D. Retrieve the secrets during the application startu
- E. Reference the secrets in the application cod
- F. Write a script to rotate the secrets saved as environment variables.
- G. Store the secrets in AWS Secrets Manager Provision a new customer master key Use the key to encrypt the secrets Enable automatic rotation Configure an Amazon EC2 user data script to programmatically retrieve the secrets during the startup and export as environment variables

**Answer:** D

**Explanation:**

AWS Secrets Manager is a service that enables the secure management and rotation of secrets, such as database credentials, API keys, or passwords. By using Secrets Manager, the company can avoid hardcoding secrets in the application code or properties files, and instead retrieve them programmatically during the application startup. Secrets Manager also supports automatic rotation of secrets by using AWS Lambda functions or built-in rotation templates. The company can provision a customer master key (CMK) to encrypt the secrets and use the AWS SDK or CLI to export the secrets as environment variables. References:

- ? What Is AWS Secrets Manager? - AWS Secrets Manager
- ? Rotating Your AWS Secrets Manager Secrets - AWS Secrets Manager
- ? Retrieving a Secret - AWS Secrets Manager

**NEW QUESTION 5**

A developer is creating a simple proof-of-concept demo by using AWS CloudFormation and AWS Lambda functions The demo will use a CloudFormation template to deploy an existing Lambda function The Lambda function uses deployment packages and dependencies stored in Amazon S3 The developer defined an AWS Lambda Function resource in a CloudFormation template. The developer needs to add the S3 bucket to the CloudFormation template. What should the developer do to meet these requirements with the LEAST development effort?

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

**Answer:** D

**Explanation:**

The easiest way to add the S3 bucket to the CloudFormation template is to use the S3Bucket and S3Key properties of the AWS::Lambda::Function resource. These properties specify the name of the S3 bucket and the location of the .zip file that contains the function code and dependencies. This way, the developer does not need to modify the function code or upload it to a different location. The other options are either not feasible or not efficient. The code property can only be used for inline code, not for code stored in S3. The ZipFile property can only be used for code that is less than 4096 bytes, not for code that has dependencies. Finding the S3 key for the Lambda function and adding it as the ZipFile property would not work, as the ZipFile property expects a base64-encoded .zip file, not an S3 location. References

- ? AWS::Lambda::Function - AWS CloudFormation
- ? Deploying Lambda functions as .zip file archives
- ? AWS Lambda Function Code - AWS CloudFormation

**NEW QUESTION 6**

A company has a web application that is hosted on Amazon EC2 instances The EC2 instances are configured to stream logs to Amazon CloudWatch Logs The company needs to receive an Amazon Simple Notification Service (Amazon SNS) notification when the number of application error messages exceeds a defined threshold within a 5-minute period Which solution will meet these requirements?

- A. Rewrite the application code to stream application logs to Amazon SNS Configure an SNS topic to send a notification when the number of errors exceeds the

defined threshold within a 5-minute period

B. Configure a subscription filter on the CloudWatch Logs log group

C. Configure the filter to send an SNS notification when the number of errors exceeds the defined threshold within a 5-minute period.

D. Install and configure the Amazon Inspector agent on the EC2 instances to monitor for errors. Configure Amazon Inspector to send an SNS notification when the number of errors exceeds the defined threshold within a 5-minute period

Set up a CloudWatch alarm based on the new

E. Create a CloudWatch metric filter to match the application error pattern in the log data.

custom metric

F. Configure the alarm to send an SNS notification when the number of errors exceeds the defined threshold within a 5-minute period.

**Answer: D**

#### Explanation:

The best solution is to create a CloudWatch metric filter to match the application error pattern in the log data. This will allow you to create a custom metric that tracks the number of errors in your application. You can then set up a CloudWatch alarm based on this metric and configure it to send an SNS notification when the number of errors exceeds a defined threshold within a 5-minute period. This solution does not require any changes to your application code or installing any additional agents on your EC2 instances. It also leverages the existing integration between CloudWatch and SNS for sending notifications. References

? Create Metric Filters - Amazon CloudWatch Logs

? Creating Amazon CloudWatch Alarms - Amazon CloudWatch

? How to send alert based on log message on CloudWatch - Stack Overflow

#### NEW QUESTION 7

A developer is creating an application that will store personal health information (PHI). The PHI needs to be encrypted at all times. An encrypted Amazon RDS for MySQL DB instance is storing the data. The developer wants to increase the performance of the application by caching frequently accessed data while adding the ability to sort or rank the cached datasets.

Which solution will meet these requirements?

A. Create an Amazon ElastiCache for Redis instance

B. Enable encryption of data in transit and at rest

C. Store frequently accessed data in the cache.

D. Create an Amazon ElastiCache for Memcached instance

E. Enable encryption of data in transit and at rest

F. Store frequently accessed data in the cache.

G. Create an Amazon RDS for MySQL read replica

H. Connect to the read replica by using SSL

I. Configure the read replica to store frequently accessed data.

J. Create an Amazon DynamoDB table and a DynamoDB Accelerator (DAX) cluster for the table

K. Store frequently accessed data in the DynamoDB table.

**Answer: A**

#### Explanation:

Amazon ElastiCache is a service that offers fully managed in-memory data stores that are compatible with Redis or Memcached. The developer can create an ElastiCache for Redis instance and enable encryption of data in transit and at rest. This will ensure that the PHI is encrypted at all times. The developer can store frequently accessed data in the cache and use Redis features such as sorting and ranking to enhance the performance of the application.

References:

? [What Is Amazon ElastiCache? - Amazon ElastiCache]

? [Encryption in Transit - Amazon ElastiCache for Redis]

? [Encryption at Rest - Amazon ElastiCache for Redis]

#### NEW QUESTION 8

A company is using an AWS Lambda function to process records from an Amazon Kinesis data stream. The company recently observed slow processing of the records. A developer notices that the iterator age metric for the function is increasing and that the Lambda run duration is constantly above normal.

Which actions should the developer take to increase the processing speed? (Choose two.)

Increase the number of shards of the Kinesis data stream.

A. Decrease the timeout of the Lambda function.

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

C. Decrease the number of shards of the Kinesis data stream.

D. Increase the timeout of the Lambda function.

**Answer: AC**

#### Explanation:

Increasing the number of shards of the Kinesis data stream will increase the throughput and parallelism of the data processing. Increasing the memory that is allocated to the Lambda function will also increase the CPU and network performance of the function, which will reduce the run duration and improve the processing speed. Option B is not correct because decreasing the timeout of the Lambda function will not affect the processing speed, but may cause some records to fail if they exceed the timeout limit. Option D is not correct because decreasing the number of shards of the Kinesis data stream will decrease the throughput and parallelism of the data processing, which will slow down the processing speed. Option E is not correct because increasing the timeout of the Lambda function will not affect the processing speed, but may increase the cost of running the function.

References: [Amazon Kinesis Data Streams Scaling], [AWS Lambda Performance Tuning]

#### NEW QUESTION 9

A developer is creating an AWS CloudFormation template to deploy Amazon EC2 instances across multiple AWS accounts. The developer must choose the EC2 instances from a list of approved instance types.

How can the developer incorporate the list of approved instance types in the CloudFormation template?

A. Create a separate CloudFormation template for each EC2 instance type in the list.

B. In the Resources section of the CloudFormation template, create resources for each EC2 instance type in the list.

C. In the CloudFormation template, create a separate parameter for each EC2 instance type in the list.

D. In the CloudFormation template, create a parameter with the list of EC2 instance types as AllowedValues.

**Answer:** D

**Explanation:**

In the CloudFormation template, the developer should create a parameter with the list of approved EC2 instance types as AllowedValues. This way, users can select the instance type they want to use when launching the CloudFormation stack, but only from the approved list.

**NEW QUESTION 10**

A developer has an application that makes batch requests directly to Amazon DynamoDB by using the BatchGetItem low-level API operation. The responses frequently return values in the UnprocessedKeys element.

Which actions should the developer take to increase the resiliency of the application when the batch response includes values in UnprocessedKeys? (Choose two.)

- A. Retry the batch operation immediately.
- B. Retry the batch operation with exponential backoff and randomized delay.
- C. Update the application to use an AWS software development kit (AWS SDK) to make the requests.
- D. Increase the provisioned read capacity of the DynamoDB tables that the operation accesses.
- E. Increase the provisioned write capacity of the DynamoDB tables that the operation accesses.

**Answer:** BC

**Explanation:**

The UnprocessedKeys element indicates that the BatchGetItem operation did not process all of the requested items in the current response. This can happen if the

response size limit is exceeded or if the table's provisioned throughput is exceeded. To handle this situation, the developer should retry the batch operation with exponential backoff and randomized delay to avoid throttling errors and reduce the load on the table. The developer should also use an AWS SDK to make the requests, as the SDKs automatically retry requests that return UnprocessedKeys.

References:

- ? [BatchGetItem - Amazon DynamoDB]
- ? [Working with Queries and Scans - Amazon DynamoDB]
- ? [Best Practices for Handling DynamoDB Throttling Errors]

**NEW QUESTION 10**

A developer is using AWS Step Functions to automate a workflow. The workflow defines each step as an AWS Lambda function task. The developer notices that runs of the Step Functions state machine fail in the GetResource task with either an ULegalArgumentException error or a TooManyRequestsException error. The developer wants the state machine to stop running when the state machine encounters a ULegalArgumentException error. The state machine needs to retry the GetResource task one additional time after 10 seconds if the state machine encounters a TooManyRequestsException error. If the second attempt fails, the developer wants the state machine to stop running.

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

- A. Add a Delay task after the GetResource task.
- B. Add a catcher to the GetResource task.
- C. Configure the catcher with an error type of TooManyRequestsException.
- D. Configure the next step to be the Delay task. Configure the Delay task to wait for an interval of 10 seconds. Configure the next step to be the GetResource task.
- E. Add a catcher to the GetResource task. Configure the catcher with an error type of TooManyRequestsException.
- F. an interval of 10 seconds, and a maximum attempts value of 1. Configure the next step to be the GetResource task.
- G. Add a retrier to the GetResource task. Configure the retrier with an error type of TooManyRequestsException, an interval of 10 seconds, and a maximum attempts value of 1.
- H. Duplicate the GetResource task. Rename the new GetResource task to TryAgain. Add a catcher to the original GetResource task.
- I. Configure the catcher with an error type of TooManyRequestsException.
- J. Configure the next step to be TryAgain.

**Answer:** C

**Explanation:**

The best way to implement the Lambda retry functionality is to use the Retry field in the state definition of the GetResource task. The Retry field allows the developer to specify an array of retriers, each with an error type, an interval, and a maximum number of attempts. By setting the error type to TooManyRequestsException, the interval to 10 seconds, and the maximum attempts to 1, the developer can achieve the desired behavior of retrying the GetResource task once after 10 seconds if it encounters

a TooManyRequestsException error. If the retry fails, the state machine will stop running. If the GetResource task encounters an ULegalArgumentException error, the state machine will also stop running without retrying, as this error type is not specified in the Retry field. References

- ? Error handling in Step Functions
- ? Handling Errors, Retries, and adding Alerting to Step Function State Machine Executions
- ? The Jitter Strategy for Step Functions Error Retries on the New Workflow Studio

**NEW QUESTION 14**

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

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

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

- A. Use Amazon Cognito user pools to manage user account.
- B. Create an Amazon Cognito user pool authorizer in API Gateway to control access to the AP
- C. Use the Lambda function to store the photos and details in the DynamoDB tabl
- D. Retrieve previously uploaded photos directly from the DynamoDB table.
- E. Use Amazon Cognito user pools to manage user account.
- F. Create an Amazon Cognito user pool authorizer in API Gateway to control access to the AP
- G. Use the Lambda function to store the photos in Amazon S3. Store the object's S3 key as part of the photo details in the DynamoDB tabl
- H. Retrieve previously uploaded photos by querying DynamoDB for the S3 key.

- I. Create an IAM user for each user of the application during the sign-up process
- J. Use IAM authentication to access the API Gateway AP

DynamoDB

- K. Use the Lambda function to store the photos in Amazon S3. Store the object's S3 key as part of the photo details in the DynamoDB table.
- L. Retrieve previously uploaded photos by querying DynamoDB for the S3 key.
- M. Create a users table in DynamoDB
- N. Use the table to manage user accounts
- O. Create a Lambda authorizer that validates user credentials against the users table
- P. Integrate the Lambda authorizer with API Gateway to control access to the API
- Q. Use the Lambda function to store the photos in Amazon S3. Store the object's S3 key as part of the photo details in the DynamoDB table
- R. Retrieve previously uploaded photos by querying DynamoDB for the S3 key.

**Answer: B**

**Explanation:**

Amazon Cognito user pools is a service that provides a secure user directory that scales to hundreds of millions of users. The developer can use Amazon Cognito user pools to manage user accounts and create an Amazon Cognito user pool authorizer in API Gateway to control access to the API. The developer can use the Lambda function to store the photos in Amazon S3, which is a highly scalable, durable, and secure object storage service. The developer can store the object's S3 key as part of the photo details in the DynamoDB table, which is a fast and flexible NoSQL database service. The developer can retrieve previously uploaded photos by querying DynamoDB for the S3 key and fetching the photos from S3. This solution will meet the requirements with the least operational overhead.

References:

- ? [Amazon Cognito User Pools]
- ? [Use Amazon Cognito User Pools - Amazon API Gateway]
- ? [Amazon Simple Storage Service (S3)]
- ? [Amazon DynamoDB]

**NEW QUESTION 17**

A developer is creating an application that will be deployed on IoT devices. The application will send data to a RESTful API that is deployed as an AWS Lambda function. The application will assign each API request a unique identifier. The volume of API requests from the application can randomly increase at any given time of day.

During periods of request throttling, the application might need to retry requests. The API must be able to handle duplicate requests without inconsistencies or data loss.

Which solution will meet these requirements?

- A. Create an Amazon RDS for MySQL DB instance
- B. Store the unique identifier for each request in a database table
- C. Modify the Lambda function to check the table for the identifier before processing the request.
- D. Create an Amazon DynamoDB table
- E. Store the unique identifier for each request in the table
- F. Modify the Lambda function to check the table for the identifier before processing the request.
- G. Create an Amazon DynamoDB table
- H. Store the unique identifier for each request in the table

receives a duplicate request.

- I. Modify the Lambda function to return a client error response when the function receives a duplicate request.
- J. Create an Amazon ElastiCache for Memcached instance
- K. Store the unique identifier for each request in the cache
- L. Modify the Lambda function to check the cache for the identifier before processing the request.

**Answer: B**

**Explanation:**

Amazon DynamoDB is a fully managed NoSQL database service that can store and retrieve any amount of data with high availability and performance. DynamoDB can handle concurrent requests from multiple IoT devices without throttling or data loss. To prevent duplicate requests from causing inconsistencies or data loss, the Lambda function can use DynamoDB conditional writes to check if the unique identifier for each request already exists in the table before processing the request. If the identifier exists, the function can skip or abort the request; otherwise, it can process the request and store the identifier in the table. Reference: Using conditional writes

**NEW QUESTION 20**

A company has an existing application that has hardcoded database credentials. A developer needs to modify the existing application. The application is deployed in two AWS Regions with an active-passive failover configuration to meet company's disaster recovery strategy.

The developer needs a solution to store the credentials outside the code. The solution must comply with the company's disaster recovery strategy.

Which solution will meet these requirements in the MOST secure way?

- A. Store the credentials in AWS Secrets Manager in the primary Region
- B. Enable secret replication to the secondary Region. Update the application to use the Amazon Resource Name (ARN) based on the Region.
- C. Store credentials in AWS Systems Manager Parameter Store in the primary Region
- D. Enable parameter replication to the secondary Region
- E. Update the application to use the Amazon Resource Name (ARN) based on the Region.
- F. Store credentials in a config file
- G. Upload the config file to an S3 bucket in the primary Region
- H. Enable Cross-Region Replication (CRR) to an S3 bucket in the secondary region
- I. Update the application to access the config file from the S3 bucket based on the Region.
- J. Store credentials in a config file
- K. Upload the config file to an Amazon Elastic File System (Amazon EFS) file system
- L. Update the application to use the Amazon EFS file system Regional endpoints to access the config file in the primary and secondary Regions.

**Answer: A**

**Explanation:**

AWS Secrets Manager is a service that allows you to store and manage secrets, such as database credentials, API keys, and passwords, in a secure and centralized way. It also provides features such as automatic secret rotation, auditing, and monitoring. By using AWS Secrets Manager, you can avoid hardcoding credentials in your code, which is a bad security practice and makes it difficult to update them. You can also replicate your secrets to another Region, which is

useful for disaster recovery purposes<sup>2</sup>. To access your secrets from your application, you can use the ARN of the secret, which is a unique identifier that includes the Region name. This way, your application can use the appropriate secret based on the Region where it is deployed<sup>3</sup>.

References:

- ? AWS Secrets Manager
- ? Replicating and sharing secrets
- ? Using your own encryption keys

#### NEW QUESTION 25

A developer must use multi-factor authentication (MFA) to access data in an Amazon S3 bucket that is in another AWS account. Which AWS Security Token Service (AWS STS) API operation should the developer use with the MFA information to meet this requirement?

- A. AssumeRoleWithWebIdentity
- B. GetFederationToken
- C. AssumeRoleWithSAML
- D. AssumeRole

**Answer:** D

#### Explanation:

The AssumeRole API operation returns a set of temporary security credentials that can be used to access resources in another AWS account. The developer can specify the MFA device serial number and the MFA token code in the request parameters. This option enables the developer to use MFA to access data in an S3 bucket that is in another AWS account. The other options are not relevant or effective for this scenario. References

- ? AssumeRole
- ? Requesting Temporary Security Credentials

#### NEW QUESTION 27

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

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

Which solution will meet these requirements?

- A. Set up an Amazon API Gateway REST API with a gateway response configured for status code 200. Add response templates that contain sample responses captured from the real third-party API.
- B. Set up an AWS AppSync GraphQL API with a data source configured for each third-party API. Specify an integration type of Mock. Configure integration responses by using sample responses captured from the real third-party API.
- C. Create an AWS Lambda function for each third-party API.
- D. Embed responses captured from the real third-party API.
- E. Configure Amazon Route 53 Resolver with an inbound endpoint for each Lambda function's Amazon Resource Name (ARN).
- F. Set up an Amazon API Gateway REST API for each third-party API. Specify an integration request type of Mock. Configure integration responses by using sample responses captured from the real third-party API.

**Answer:** D

#### Explanation:

Amazon API Gateway can mock responses for testing purposes without requiring any integration backend. This allows the developer to test the API integration code without invoking the third-party payment processing APIs. The developer can configure integration responses by using sample responses captured from the real third-party API. References:

- ? Mocking Integration Responses in API Gateway
- ? Set up Mock Integrations for an API in API Gateway

#### NEW QUESTION 30

A company is building a web application on AWS. When a customer sends a request, the application will generate reports and then make the reports available to the customer within one hour. Reports should be accessible to the customer for 8 hours. Some reports are larger than 1 MB. Each report is unique to the customer. The application should delete all reports that are older than 2 days.

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

- A. Generate the reports and then store the reports as Amazon DynamoDB items that have a specified TTL.
- B. Generate a URL that retrieves the reports from DynamoDB.
- C. Provide the URL to customers through the web application.
- D. Generate the reports and then store the reports in an Amazon S3 bucket that uses server-side encryption.
- E. Attach the reports to an Amazon Simple Notification Service (Amazon SNS) message.
- F. Subscribe the customer to email notifications from Amazon SNS.
- G. Generate the reports and then store the reports in an Amazon S3 bucket that uses server-side encryption.
- H. Generate a presigned URL that contains an expiration date. Provide the URL to customers through the web application.
- I. Add S3 Lifecycle configuration rules to the S3 bucket to delete old reports.
- J. Generate the reports and then store the reports in an Amazon RDS database with a date stamp.
- K. Generate a URL that retrieves the reports from the RDS database.
- L. Provide the URL to customers through the web application.
- M. Schedule an hourly AWS Lambda function to delete database records that have expired date stamps.

**Answer:** C

#### Explanation:

This solution will meet the requirements with the least operational overhead because it uses Amazon S3 as a scalable, secure, and durable storage service for the reports. The presigned URL will allow customers to access their reports for a limited time (8 hours) without requiring additional authentication. The S3 Lifecycle configuration rules will automatically delete the reports that are older than 2 days, reducing storage costs and complying with the data retention policy. Option A is not optimal because it will incur additional costs and complexity to store the reports as DynamoDB items, which have a size limit of 400 KB. Option B is not optimal because it will not provide customers with access to their reports within one hour, as Amazon SNS email delivery is not guaranteed. Option D is not optimal

because it will require more operational overhead to manage an RDS database and a Lambda function for storing and deleting the reports.  
 References: Amazon S3 Presigned URLs, Amazon S3 Lifecycle

#### NEW QUESTION 34

For a deployment using AWS Code Deploy, what is the run order of the hooks for in-place deployments?

- A. BeforeInstall -> ApplicationStop -> ApplicationStart -> AfterInstall
- B. ApplicationStop -> BeforeInstall -> AfterInstall -> ApplicationStart
- C. BeforeInstall -> ApplicationStop -> ValidateService -> ApplicationStart
- D. ApplicationStop -> BeforeInstall -> ValidateService -> ApplicationStart

**Answer: B**

#### Explanation:

For in-place deployments, AWS CodeDeploy uses a set of predefined hooks that run in a specific order during each deployment lifecycle event. The hooks are ApplicationStop, BeforeInstall, AfterInstall, ApplicationStart, and ValidateService. The run order of the hooks for in-place deployments is as follows:

? ApplicationStop: This hook runs first on all instances and stops the current application that is running on the instances.

? BeforeInstall: This hook runs after ApplicationStop on all instances and performs any tasks required before installing the new application revision.

? AfterInstall: This hook runs after BeforeInstall on all instances and performs any tasks required after installing the new application revision.

? ApplicationStart: This hook runs after AfterInstall on all instances and starts the new application that has been installed on the instances.

? ValidateService: This hook runs last on all instances and verifies that the new application is running properly on the instances.

Reference: [AWS CodeDeploy lifecycle event hooks reference]

#### NEW QUESTION 37

A developer at a company needs to create a small application that makes the same API call once each day at a designated time. The company does not have infrastructure in the AWS Cloud yet, but the company wants to implement this functionality on AWS.

Which solution meets these requirements in the MOST operationally efficient manner?

A. Use a Kubernetes cron job that runs on Amazon Elastic Kubernetes Service (Amazon EKS).

B. Use an Amazon Linux crontab scheduled job that runs on Amazon EC2.

C. Use an AWS Lambda function that is invoked by an Amazon EventBridge scheduled event.

D. Use an AWS Batch job that is submitted to an AWS Batch job queue.

**Answer: C**

#### Explanation:

The correct answer is C. Use an AWS Lambda function that is invoked by an Amazon EventBridge scheduled event.

\* C. Use an AWS Lambda function that is invoked by an Amazon EventBridge scheduled event. This is correct. AWS Lambda is a serverless compute service that lets you run code without provisioning or managing servers. Lambda runs your code on a high-availability compute infrastructure and performs all of the administration of the compute resources, including server and operating system maintenance, capacity provisioning and automatic scaling, and logging<sup>1</sup>. Amazon EventBridge is a serverless event bus service that enables you to connect your applications with data from a variety of sources<sup>2</sup>. EventBridge can create rules that run on a schedule, either at regular intervals or at specific times and dates, and invoke targets such as Lambda functions<sup>3</sup>. This solution meets the requirements of creating a small application that makes the same API call once each day at a designated time, without requiring any infrastructure in the AWS Cloud or any operational overhead.

\* A. Use a Kubernetes cron job that runs on Amazon Elastic Kubernetes Service (Amazon EKS). This is incorrect. Amazon EKS is a fully managed Kubernetes service that allows you to run containerized applications on AWS<sup>4</sup>. Kubernetes cron jobs are tasks that run periodically on a given schedule<sup>5</sup>. This solution could meet the functional requirements of creating a small application that makes the same API call once each day at a designated time, but it would not be the most operationally efficient manner. The company would need to provision and manage an EKS cluster, which would incur additional costs and complexity.

\* B. Use an Amazon Linux crontab scheduled job that runs on Amazon EC2. This is incorrect. Amazon EC2 is a web service that provides secure, resizable compute capacity in the cloud<sup>6</sup>. Crontab is a Linux utility that allows you to schedule commands or scripts to run automatically at a specified time or date<sup>7</sup>. This solution could meet the functional requirements of creating a small application that makes the same API call once each day at a designated time, but it would not be the most operationally efficient manner. The company would need to provision and manage an EC2 instance, which would incur additional costs and complexity.

\* D. Use an AWS Batch job that is submitted to an AWS Batch job queue. This is incorrect. AWS Batch enables you to run batch computing workloads on the AWS Cloud<sup>8</sup>. Batch jobs are units of work that can be submitted to job queues, where they are executed in parallel or sequentially on compute environments<sup>9</sup>. This solution could meet the functional requirements of creating a small application that makes the same API call once each day at a designated time, but it would not be the most operationally efficient manner. The company would need to configure and manage an AWS Batch environment, which would incur additional costs and complexity.

References:

? 1: What is AWS Lambda? - AWS Lambda

? 2: What is Amazon EventBridge? - Amazon EventBridge

? 3: Creating an Amazon EventBridge rule that runs on a schedule - Amazon EventBridge

? 4: What is Amazon EKS? - Amazon EKS

? 5: CronJob - Kubernetes

? 6: What is Amazon EC2? - Amazon EC2

? 7: Crontab in Linux with 20 Useful Examples to Schedule Jobs - Tecmint

? 8: What is AWS Batch? - AWS Batch

? 9: Jobs - AWS Batch

#### NEW QUESTION 40

A company wants to automate part of its deployment process. A developer needs to automate the process of checking for and deleting unused resources that supported previously deployed stacks but that are no longer used.

The company has a central application that uses the AWS Cloud Development Kit (AWS CDK) to manage all deployment stacks. The stacks are spread out across multiple accounts. The developer's solution must integrate as seamlessly as possible within the current deployment process.

Which solution will meet these requirements with the LEAST amount of configuration?

A. In the central AWS CDK application, write a handler function in the code that uses AWS SDK calls to check for and delete unused

A.

resource

- B. Create an AWS CloudFormation template from a JSON file
- C. Use the template to attach the function code to an AWS Lambda function and to invoke the Lambda function when the deployment stack runs.
- D. In the central AWS CDK application
- E. write a handler function in the code that uses AWS SDK calls to check for and delete unused resource
- F. Create an AWS CDK custom resource Use the custom resource to attach the function code to an AWS Lambda function and to invoke the Lambda function when the deployment stack runs.
- G. In the central AWS CDK, write a handler function in the code that uses AWS SDK calls to check for and delete unused resource
- H. Create an API in AWS Amplify Use the API to attach the function code to an AWS Lambda function and to invoke the Lambda function when the deployment stack runs.
- I. In the AWS Lambda console write a handler function in the code that uses AWS SDK calls to check for and delete unused resource
- J. Create an AWS CDK custom resource
- K. Use the custom resource to import the Lambda function into the stack and to invoke the Lambda function when the deployment stack runs.

**Answer: B**

**Explanation:**

This solution meets the requirements with the least amount of configuration because it uses a feature of AWS CDK that allows custom logic to be executed during stack deployment or deletion. The AWS Cloud Development Kit (AWS CDK) is a software development framework that allows you to define cloud infrastructure as code and provision it through CloudFormation. An AWS CDK custom resource is a construct that enables you to create resources that are not natively supported by CloudFormation or perform tasks that are not supported by CloudFormation during stack deployment or deletion. The developer can write a handler function in the code that uses AWS SDK calls to check for and delete unused resources, and create an AWS CDK custom resource that attaches the function code to a Lambda function and invokes it when the deployment stack runs. This way, the developer can automate the cleanup process without requiring additional configuration or integration. Creating a CloudFormation template from a JSON file will require additional configuration and integration with the central AWS CDK application. Creating an API in AWS Amplify will require additional configuration and integration with the central AWS CDK application and may not provide optimal performance or availability. Writing a handler function in the AWS Lambda console will require additional configuration and integration with the central AWS CDK application.

Reference: [AWS Cloud Development Kit (CDK)], [Custom Resources]

**NEW QUESTION 43**

A developer at a company needs to create a small application that makes the same API call once each day at a designated time. The company does not have infrastructure in the AWS Cloud yet, but the company wants to implement this functionality on AWS. Which solution meets these requirements in the MOST operationally efficient manner?

- A. Use a Kubernetes cron job that runs on Amazon Elastic Kubernetes Service (Amazon EKS)
- B. Use an Amazon Linux cron job scheduled job that runs on Amazon EC2
- C. Use an AWS Lambda function that is invoked by an Amazon EventBridge scheduled event.
- D. Use an AWS Batch job that is submitted to an AWS Batch job queue.

**Answer: C**

**Explanation:**

This solution meets the requirements in the most operationally efficient manner because it does not require any infrastructure provisioning or management. The developer can create a Lambda function that makes the API call and configure an EventBridge rule that triggers the function once a day at a designated time. This is a serverless solution that scales automatically and only charges for the execution time of the function.

Reference: [Using AWS Lambda with Amazon EventBridge], [Schedule Expressions for Rules]

**NEW QUESTION 45**

A company hosts a client-side web application for one of its subsidiaries on Amazon S3. The web application can be accessed through Amazon CloudFront from <https://www.example.com>. After a successful rollout, the company wants to host three more client-side web applications for its remaining subsidiaries on three separate S3 buckets.

To achieve this goal, a developer moves all the common JavaScript files and web fonts to a central S3 bucket that serves the web applications. However, during testing, the developer notices that the browser blocks the JavaScript files and web fonts.

What should the developer do to prevent the browser from blocking the JavaScript files and web fonts?

- A. Create four access points that allow access to the central S3 bucket
- B. Assign an access point to each web application bucket.
- C. Create a bucket policy that allows access to the central S3 bucket
- D. Attach the bucket policy to the central S3 bucket.
- E. Create a cross-origin resource sharing (CORS) configuration that allows access to the central S3 bucket
- F. Add the CORS configuration to the central S3 bucket.
- G. Create a Content-MD5 header that provides a message integrity check for the central S3 bucket
- H. Insert the Content-MD5 header for each web application request.

**Answer: C**

**Explanation:**

This is a frequent trouble. Web applications cannot access the resources in other domains by default, except some exceptions. You must configure CORS on the resources to be accessed. <https://docs.aws.amazon.com/AmazonS3/latest/userguide/cors.html>

**NEW QUESTION 46**

A company has multiple Amazon VPC endpoints in the same VPC. A developer needs to configure an Amazon S3 bucket policy so users can access an S3 bucket only by using these VPC endpoints.

Which solution will meet these requirements?

- A. Create multiple S3 bucket policies by using each VPC endpoint ID that have the `aws:SourceVpce` value in the `StringNotEquals` condition.
- B. Create a single S3 bucket policy that has the `aws:SourceVpc` value and in the `StringNotEquals` condition to use VPC ID.
- C. Create a single S3 bucket policy that has the multiple `aws:SourceVpce` value and in the `StringNotEquals` condition to use `vpce`.

- D. Create a single S3 bucket policy that has multiple aws sourceVpce value in the StingNotEquale conditio
- E. Repeat for all the VPC endpoint IDs.

**Answer:** D

**Explanation:**

This solution will meet the requirements by creating a single S3 bucket policy that denies access to the S3 bucket unless the request comes from one of the specified VPC endpoints. The aws:SourceVpce condition key is used to match the ID of the VPC endpoint that is used to access the S3 bucket. The StringNotEquals condition operator is used to negate the condition, so that only requests from the listed VPC endpoints are allowed. Option A is not optimal because it will create multiple S3 bucket policies, which is not possible as only one bucket policy can be attached to an S3 bucket. Option B is not optimal because it will use the aws:SourceVpc condition key, which matches the ID of the VPC that is used to access the S3 bucket, not the VPC endpoint. Option C is not optimal because it will use the StringNotEquals condition operator with a single value, which will deny access to the S3 bucket from all VPC endpoints except one.

References: Using Amazon S3 Bucket Policies and User Policies, AWS Global Condition Context Keys

**NEW QUESTION 51**

A developer is migrating an application to Amazon Elastic Kubernetes Service (Amazon EKS). The developer migrates the application to Amazon Elastic Container Registry (Amazon ECR) with an EKS cluster.

As part of the application migration to a new backend, the developer creates a new AWS account. The developer makes configuration changes to the application to point the application to the new AWS account and to use new backend resources. The developer successfully tests the changes within the application by deploying the pipeline.

The Docker image build and the pipeline deployment are successful, but the application is still connecting to the old backend. The developer finds that the application's configuration is still referencing the original EKS cluster and not referencing the new backend resources.

Which reason can explain why the application is not connecting to the new resources?

- A. The developer did not successfully create the new AWS account.
- B. The developer added a new tag to the Docker image.
- C. The developer did not update the Docker image tag to a new version.
- D. The developer pushed the changes to a new Docker image tag.

**Answer:** C

**Explanation:**

The correct answer is C. The developer did not update the Docker image tag to a new version.

\* C. The developer did not update the Docker image tag to a new version. This is correct. When deploying an application to Amazon EKS, the developer needs to specify the Docker image tag that contains the application code and configuration. If the developer does not update the Docker image tag to a new version after making changes to the application, the EKS cluster will continue to use the old Docker image tag that references the original backend resources. To fix this issue, the developer should update the Docker image tag to a new version and redeploy the application to the EKS cluster.

\* A. The developer did not successfully create the new AWS account. This is incorrect. The creation of a new AWS account is not related to the application's connection to the

backend resources. The developer can use any AWS account to host the EKS cluster and the backend resources, as long as they have the proper permissions and configurations.

\* B. The developer added a new tag to the Docker image. This is incorrect. Adding a new tag to the Docker image is not enough to deploy the changes to the application. The developer also needs to update the Docker image tag in the EKS cluster configuration, so that the EKS cluster can pull and run the new Docker image.

\* D. The developer pushed the changes to a new Docker image tag. This is incorrect. Pushing the changes to a new Docker image tag is not enough to deploy the changes to the application. The developer also needs to update the Docker image tag in the EKS cluster configuration, so that the EKS cluster can pull and run the new Docker image. References:

? 1: Amazon EKS User Guide, "Deploying applications to your Amazon EKS cluster", <https://docs.aws.amazon.com/eks/latest/userguide/deploying-applications.html>

? 2: Amazon ECR User Guide, "Pushing an image", <https://docs.aws.amazon.com/AmazonECR/latest/userguide/docker-push-ecr-image.html>

? 3: Amazon EKS User Guide, "Updating an Amazon EKS cluster", <https://docs.aws.amazon.com/eks/latest/userguide/update-cluster.html>

**NEW QUESTION 54**

A developer needs to store configuration variables for an application. The developer needs to set an expiration date and time for me configuration. The developer wants to receive notifications. Before the configuration expires. Which solution will meet these requirements with the LEAST operational overhead?

- A. Create a standard parameter in AWS Systems Manager Parameter Store Set Expiation and Expiration Notification policy types.
- B. Create a standard parameter in AWS Systems Manager Parameter Store Create an AWS Lambda function to expire the configuration and to send Amazon Simple Notification Service (Amazon SNS) notifications.
- C. Create an advanced parameter in AWS Systems Manager Parameter Store Set Expiration and Expiration Notification policy types.
- D. Create an advanced parameter in AWS Systems Manager Parameter Store Create an Amazon EC2 instance with a corn job to expire the configuration and to send notifications.

**Answer:** C

**Explanation:**

This solution will meet the requirements by creating an advanced parameter in AWS Systems Manager Parameter Store, which is a secure and scalable service for storing and managing configuration data and secrets. The advanced parameter allows setting expiration and expiration notification policy types, which enable specifying an expiration date and time for the configuration and receiving notifications before the configuration expires. The Lambda code will be refactored to load the Root CA Cert from the parameter store and modify the runtime trust store outside the Lambda function handler, which will improve performance and reduce latency by avoiding repeated calls to Parameter Store and trust store modifications for each invocation of the Lambda function. Option A is not optimal because it will create a standard parameter in AWS Systems Manager Parameter Store, which does not support expiration and expiration notification policy types. Option B is not optimal because it will create a secret access key and access key ID with permission to access the S3 bucket, which will introduce additional security risks and complexity for storing and managing credentials. Option D is not optimal because it will create a Docker container from Node.js base image to invoke Lambda functions, which will incur additional costs and overhead for creating and running Docker containers. References: AWS Systems Manager Parameter Store, [Using SSL/TLS to Encrypt a Connection to a DB Instance]

**NEW QUESTION 57**

A development team maintains a web application by using a single AWS CloudFormation template. The template defines web servers and an Amazon RDS database. The team uses the Cloud Formation template to deploy the Cloud Formation stack to different environments. During a recent application deployment, a developer caused the primary development database to be dropped and recreated. The result of this incident was a loss of data. The team needs to avoid accidental database deletion in the future. Which solutions will meet these requirements? (Choose two.)

- A. Add a CloudFormation Deletion Policy attribute with the Retain value to the database resource.
- B. Update the CloudFormation stack policy to prevent updates to the database.  
Modify the database to use a Multi-AZ deployment.
- C. Create a CloudFormation stack set for the web application and database deployments.**
- E. Add a Cloud Formation DeletionPolicy attribute with the Retain value to the stack.

**Answer: AB****Explanation:**

AWS CloudFormation is a service that enables developers to model and provision AWS resources using templates. The developer can add a CloudFormation Deletion Policy attribute with the Retain value to the database resource. This will prevent the database from being deleted when the stack is deleted or updated. The developer can also update the CloudFormation stack policy to prevent updates to the database. This will prevent accidental changes to the database configuration or properties.

References:

- ? [What Is AWS CloudFormation? - AWS CloudFormation]
- ? [DeletionPolicy Attribute - AWS CloudFormation]
- ? [Protecting Resources During Stack Updates - AWS CloudFormation]

**NEW QUESTION 59**

A developer is working on a serverless application that needs to process any changes to an Amazon DynamoDB table with an AWS Lambda function. How should the developer configure the Lambda function to detect changes to the DynamoDB table?

- A. Create an Amazon Kinesis data stream, and attach it to the DynamoDB tabl
- B. Create a trigger to connect the data stream to the Lambda function.
- C. Create an Amazon EventBridge rule to invoke the Lambda function on a regular** schedul
- ~~D. Connect to the DynamoDB table from the Lambda function to detect changes.~~
- E. Enable DynamoDB Streams on the tabl
- F. Create a trigger to connect the DynamoDB stream to the Lambda function.
- G. Create an Amazon Kinesis Data Firehose delivery stream, and attach it to the DynamoDB tabl
- H. Configure the delivery stream destination as the Lambda function.

**Answer: C****Explanation:**

Amazon DynamoDB is a fully managed NoSQL database service that provides fast and consistent performance with seamless scalability. DynamoDB Streams is a feature that captures data modification events in DynamoDB tables. The developer can enable DynamoDB Streams on the table and create a trigger to connect the DynamoDB stream to the Lambda function. This solution will enable the Lambda function to detect changes to the DynamoDB table in near real time.

References:

- ? [Amazon DynamoDB]
- ? [DynamoDB Streams - Amazon DynamoDB]
- ? [Using AWS Lambda with Amazon DynamoDB - AWS Lambda]

**NEW QUESTION 64**

A developer is trying get data from an Amazon DynamoDB table called demoman-table. The developer configured the AWS CLI to use a specific IAM use's credentials and ran the following command.

```
aws dynamodb get-item --table-name demoman-table --key '{"id": {"N": "1993"}}'
```

The command returned errors and no rows were returned. What is the MOST likely cause of these issues?

- A. The command is incorrect; it should be rewritten to use put-item with a string argument
- B. The developer needs to log a ticket with AWS Support to enable access to the demoman-table
- C. Amazon DynamoOB cannot be accessed from the AWS CLI and needs to called via the REST API
- D. The IAM user needs an associated policy with read access to demoman-table**

**Answer: D****Explanation:**

This solution will most likely solve the issues because it will grant the IAM user the necessary permission to access the DynamoDB table using the AWS CLI command. The error message indicates that the IAM user does not have sufficient access rights to perform the scan operation on the table. Option A is not optimal because it will change the command to use put-item instead of scan, which will not achieve the desired result of getting data from the table. Option B is not optimal because it will involve contacting AWS Support, which may not be necessary or efficient for this issue. Option C is not optimal because it will state that DynamoDB cannot be accessed from the AWS CLI, which is incorrect as DynamoDB supports AWS CLI commands.

References: AWS CLI for DynamoDB, [IAM Policies for DynamoDB]

**NEW QUESTION 69**

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

- A. Set the Deployment Preference Type to Canary10Percent10Minute**

AutoPublishAlias property to the Lambda alias.

- B. Set the Deployment Preference Type to LinearIOPercentEvery10Minute
- D. Set AutoPublishAlias property to the Lambda alias.
- E. Set the Deployment Preference Type to CanaryIOPercentIOMinute
- F. Set the PreTraffic and PostTraffic properties to the Lambda alias.
- G. Set the Deployment Preference Type to LinearIOPercentEveryIOMinute
- H. Set PreTraffic and Post Traffic properties to the Lambda alias.

**Answer:** A

**Explanation:**

The AWS Serverless Application Model (AWS SAM) comes built-in with CodeDeploy to provide gradual AWS Lambda deployments<sup>1</sup>. The DeploymentPreference property in AWS SAM allows you to specify the type of deployment that you want. The Canary10Percent10Minutes option means that 10 percent of your customer traffic is immediately shifted to your new version. After 10 minutes, all traffic is shifted to the new version<sup>1</sup>. The AutoPublishAlias property in AWS SAM allows AWS SAM to automatically create an alias that points to the updated version of the Lambda function<sup>1</sup>. Therefore, option A is correct.

**NEW QUESTION 74**

A developer is writing an application that will retrieve sensitive data from a third-party system. The application will format the data into a PDF file. The PDF file could be more than 1 MB. The application will encrypt the data to disk by using AWS Key Management Service (AWS KMS). The application will decrypt the file when a user requests to download it. The retrieval and formatting portions of the application are complete.

The developer needs to use the GenerateDataKey API to encrypt the PDF file so that the PDF file can be decrypted later. The developer needs to use an AWS KMS symmetric customer managed key for encryption.

Which solutions will meet these requirements?

- A. Write the encrypted key from the GenerateDataKey API to disk for later use and use the plaintext key from the GenerateDataKey API and a symmetric encryption algorithm to encrypt the file.
- B. Use the plaintext key from the GenerateDataKey API to encrypt the file.
- D. Use the encrypted key from the GenerateDataKey API and a symmetric encryption algorithm to encrypt the file.
- E. Write the encrypted key from the GenerateDataKey API to disk for later use.
- F. Use the plaintext key from the GenerateDataKey API to encrypt the file by using the KMS Encrypt API.
- G. Write the plaintext key from the GenerateDataKey API to disk for later use.
- H. Use the encrypted key from the GenerateDataKey API to encrypt the file by using the KMS Encrypt API.

**Answer:** A

**Explanation:**

? The GenerateDataKey API returns a data key that is encrypted under a symmetric encryption KMS key that you specify, and a plaintext copy of the same data key<sup>1</sup>. The data key is a random byte string that can be used with any standard encryption algorithm, such as AES or SM42. The plaintext data key can be used to encrypt or decrypt data outside of AWS KMS, while the encrypted data key can be stored with the encrypted data and later decrypted by AWS KMS<sup>1</sup>.

? In this scenario, the developer needs to use the GenerateDataKey API to encrypt the PDF file so that it can be decrypted later. The developer also needs to use an AWS KMS symmetric customer managed key for encryption. To achieve this, the developer can follow these steps:

**NEW QUESTION 79**

A company wants to deploy and maintain static websites on AWS. Each website's source code is hosted in one of several version control systems, including AWS CodeCommit, Bitbucket, and GitHub.

The company wants to implement phased releases by using development, staging, user acceptance testing, and production environments in the AWS Cloud. Deployments to each environment must be started by code merges on the relevant Git branch. The company wants to use HTTPS for all data exchange. The company needs a solution that does not require servers to run continuously.

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

- A. Host each website by using AWS Amplify with a serverless backend.
- B. Connect the repository branches that correspond to each of the desired environment.
- C. Start deployments by merging code changes to a desired branch.
- D. Host each website in AWS Elastic Beanstalk with multiple environment.
- E. Use the EB CLI to link each repository branch.
- F. Integrate AWS CodePipeline to automate deployments from version control code merges.
- G. Host each website in different Amazon S3 buckets for each environment.
- H. Configure AWS CodePipeline to pull source code from version control.
- I. Add an AWS CodeBuild stage to copy source code to Amazon S3.
- J. Host each website on its own Amazon EC2 instance.
- K. Write a custom deployment script to bundle each website's static asset.
- L. Copy the assets to Amazon EC2. Set up a workflow to run the script when code is merged.

**Answer:** A

**Explanation:**

AWS Amplify is a set of tools and services that enables developers to build and deploy full-stack web and mobile applications that are powered by AWS. AWS Amplify supports hosting static websites on Amazon S3 and Amazon CloudFront, with HTTPS enabled by default. AWS Amplify also integrates with various version control systems, such as AWS CodeCommit, Bitbucket, and GitHub, and allows developers to connect different branches to different environments. AWS Amplify automatically builds and deploys the website whenever code changes are merged to a connected branch, enabling phased releases with minimal operational overhead. Reference: AWS Amplify Console

**NEW QUESTION 80**

A company's developer has deployed an application in AWS by using AWS CloudFormation. The CloudFormation stack includes parameters in AWS Systems Manager Parameter Store that the application uses as configuration settings. The application can modify the parameter values.

When the developer updated the stack to create additional resources with tags, the developer noted that the parameter values were reset and that the values ignored the latest changes made by the application. The developer needs to change the way the company deploys the CloudFormation stack. The developer also needs to avoid resetting the parameter values outside the stack.

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

- A. Modify the CloudFormation stack to set the deletion policy to Retain for the Parameter Store parameters.
- B. Create an Amazon DynamoDB table as a resource in the CloudFormation stack to hold configuration data for the application. Migrate the parameters that the application is modifying from Parameter Store to the DynamoDB table.
- C. Create an Amazon RDS DB instance as a resource in the CloudFormation stack.
- D. Create a table in the database for parameter configuration.
- E. Migrate the parameters that the application is modifying from Parameter Store to the configuration table.
- F. Modify the CloudFormation stack policy to deny updates on Parameter Store parameters.

**Answer:** D

**Explanation:**

<https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/protect-stack-resources.html#stack-policy-samples>

**NEW QUESTION 85**

A company needs to set up secure database credentials for all its AWS Cloud resources. The company's resources include Amazon RDS DB instances, Amazon DocumentDB clusters, and Amazon Aurora DB instances. The company's security policy mandates that database credentials be encrypted at rest and rotated at a regular interval.

Which solution will meet these requirements MOST securely?

- A. Set up IAM database authentication for token-based access.
- B. Generate user tokens to provide centralized access to RDS DB instances.
- C. Amazon DocumentDB clusters and Aurora DB instances.
- D. Create parameters for the database credentials in AWS Systems Manager Parameter Store. Set the Type parameter to SecureString.
- E. Set up automatic rotation on the parameters.
- F. Store the database access credentials as an encrypted Amazon S3 object in an S3 bucket. Block all public access on the S3 bucket. Enable automatic rotation on the encryption key.
- G. Use S3 server-side encryption to set up the parameters.
- H. Create an AWS Lambda function by using the SecretsManagerRotationTemplate template in the AWS Secrets Manager console.
- I. Create secrets for the database credentials in Secrets Manager. Set up secrets rotation on a schedule.

**Answer:** D

**Explanation:**

This solution will meet the requirements by using AWS Secrets Manager, which is a service that helps protect secrets such as database credentials by encrypting them with AWS Key Management Service (AWS KMS) and enabling automatic rotation of secrets. The developer can create an AWS Lambda function by using the SecretsManagerRotationTemplate template in the AWS Secrets Manager console, which provides a sample code for rotating secrets for RDS DB instances, Amazon DocumentDB clusters, and Amazon Aurora DB instances. The developer can also create secrets for the database credentials in Secrets Manager, which encrypts them at rest and provides secure access to them. The developer can set up secrets rotation on a schedule, which changes the database credentials periodically according to a specified interval or event. Option A is not optimal because it will set up IAM database authentication for token-based access, which may not be compatible with all database engines and may require additional configuration and management of IAM roles or users. Option B is not optimal because it will create parameters for the database credentials in AWS Systems Manager Parameter Store, which does not support automatic rotation of secrets. Option C is not optimal because it will store the database access credentials as an encrypted Amazon S3 object in an S3 bucket, which may introduce additional costs and complexity for accessing and securing the data.

References: [AWS Secrets Manager], [Rotating Your AWS Secrets Manager Secrets]

**NEW QUESTION 87**

A company has an ecommerce application. To track product reviews, the company's development team uses an Amazon DynamoDB table.

Every record includes the following

- A Review ID, a 16-digit universally unique identifier (UUID)
- A Product ID and User ID, 16-digit UUIDs that reference other tables
- A Product Rating on a scale of 1-5
- An optional comment from the user

The table's partition key is the Review ID. The most performed query against the table is to find the 10 reviews with the highest rating for a given product.

Which index will provide the FASTEST response for this query?

- A. A global secondary index (GSI) with Product ID as the partition key and Product Rating as the sort key.
- B. A global secondary index (GSI) with Product ID as the partition key and Review ID as the sort key.
- C. A local secondary index (LSI) with Product ID as the partition key and Product Rating as the sort key.
- D. A local secondary index (LSI) with Review ID as the partition key and Product ID as the sort key.

**Answer:** A

**Explanation:**

This solution allows the fastest response for the query because it enables the query to use a single partition key value (the Product ID) and a range of sort key values (the Product Rating) to find the matching items. A global secondary index (GSI) is an index that has a partition key and an optional sort key that are different from those on the base table. A GSI can be created at any time and can be queried or scanned independently of the base table. A local secondary index (LSI) is an index that has the same partition key as the base table, but a different sort key. An LSI can only be created when the base table is created and must be queried together with the base table's partition key. Using a GSI with Product ID as the partition key and Review ID as the sort key will not allow the query to use a range of sort key values to find the highest ratings. Using an LSI with Product ID as the partition key and Product Rating as the sort key will not work because Product ID is not the partition key of the base table. Using an LSI with Review ID as the partition key and Product ID as the sort key will not allow the query to use a single partition key value to find the matching items.

Reference: [Global Secondary Indexes], [Querying]

**NEW QUESTION 91**

A company is migrating legacy internal applications to AWS. Leadership wants to rewrite the internal employee directory to use native AWS services. A developer needs to create a solution for storing employee contact details and high-resolution photos for use with the new application.

Which solution will enable the search and retrieval of each employee's individual details and high-resolution photos using AWS APIs?

- A. Encode each employee's contact information and photos using Base64. Store the information in an Amazon DynamoDB table using a sort key.  
 B. Store each employee's contact information in an Amazon DynamoDB table along with the object keys for the photos stored in Amazon S3.  
 C. Use Amazon Cognito user pools to implement the employee directory in a fully managed software-as-a-service (SaaS) method.  
 D. Store employee contact information in an Amazon RDS DB instance with the photos stored in Amazon Elastic File System (Amazon EFS).

**Answer: B**

**Explanation:**

Amazon DynamoDB is a fully managed NoSQL database service that provides fast and consistent performance with seamless scalability. The developer can store each employee's contact information in a DynamoDB table along with the object keys for the photos stored in Amazon S3. Amazon S3 is an object storage service that offers industry-leading scalability, data availability, security, and performance. The developer can use AWS APIs to search and retrieve the employee details and photos from DynamoDB and S3.

References:

- ? [Amazon DynamoDB]
- ? [Amazon Simple Storage Service (S3)]

**NEW QUESTION 92**

A developer uses AWS CloudFormation to deploy an Amazon API Gateway API and an AWS Step Functions state machine. The state machine must reference the API Gateway API after the CloudFormation template is deployed. The developer needs a solution that uses the state machine to reference the API Gateway endpoint.

Which solution will meet these requirements MOST cost-effectively?

- A. Configure the CloudFormation template to reference the API endpoint in the DefinitionSubstitutions property for the AWS StepFunctions StateMachine resource.  
 B. Configure the CloudFormation template to store the API endpoint in an environment variable for the AWS::StepFunctions::StateMachine resource. Configure the state machine to reference the environment variable.  
 C. Configure the CloudFormation template to store the API endpoint in a standard AWS::SecretsManager::Secret resource. Configure the state machine to reference the resource.  
 D. Configure the CloudFormation template to store the API endpoint in a standard AWS::AppConfig::ConfigurationProfile resource. Configure the state machine to reference the resource.

**Answer: A**

**Explanation:**

The most cost-effective solution is to use the DefinitionSubstitutions property of the AWS::StepFunctions::StateMachine resource to inject the API endpoint as a variable in the state machine definition. This way, the developer can use the intrinsic function

Fn::GetAtt to get the API endpoint from the AWS::ApiGateway::RestApi resource, and pass it to the state machine without creating any additional resources or environment variables. The other solutions involve creating and managing extra resources, such as Secrets Manager secrets or AppConfig configuration profiles, which incur additional costs and complexity. References:

- ? AWS::StepFunctions::StateMachine - AWS CloudFormation
- ? Call API Gateway with Step Functions - AWS Step Functions
- ? amazon-web-services aws-api-gateway terraform aws-step-functions

**NEW QUESTION 96**

A developer deployed an application to an Amazon EC2 instance. The application needs to know the public IPv4 address of the instance. How can the application find this information?

- A. Query the instance metadata from `http://169.254.169.254/latest/meta-data/`.  
 B. Query the instance user data from `http://169.254.169.254/latest/user-data/`.  
 C. Query the Amazon Machine Image (AMI) information from `http://169.254.169.254/latest/meta-data/ami/`.  
 D. Check the hosts file of the operating system.

**Answer: A**

**Explanation:**

The instance metadata service provides information about the EC2 instance, including the public IPv4 address, which can be obtained by querying the endpoint `http://169.254.169.254/latest/meta-data/public-ipv4`. References:

- ? Instance metadata and user data
- ? Get Public IP Address on current EC2 Instance
- ? Get the public ip address of your EC2 instance quickly

**NEW QUESTION 101**

A company's website runs on an Amazon EC2 instance and uses Auto Scaling to scale the environment during peak times. Website users across the world are experiencing high latency due to static content on the EC2 instance, even during non-peak hours.

When combination of steps will resolve the latency issue? (Select TWO)

- A. Double the Auto Scaling group's maximum number of servers.  
 B. Host the application code on AWS Lambda.  
 C. Scale vertically by resizing the EC2 instances.  
 D. Create an Amazon CloudFront distribution to cache the static content.  
 E. Store the application's static content in Amazon S3.

**Answer: DE**

**Explanation:**

The combination of steps that will resolve the latency issue is to create an Amazon CloudFront distribution to cache the static content and store the application's static content in Amazon S3. This way, the company can use CloudFront to deliver the static content from edge locations that are closer to the website users, reducing latency and improving performance. The company can also use S3 to store the static content reliably and cost-effectively, and integrate it with CloudFront easily. The other options either do not address the latency issue, or are not necessary or feasible for the given scenario.

Reference: Using Amazon S3 Origins and Custom Origins for Web Distributions

#### NEW QUESTION 102

A developer wants to add request validation to a production environment Amazon API Gateway API. The developer needs to test the changes before the API is deployed to the production environment. For the least the developer will send test requests to the API through a testing tool. Which solution will meet these requirements with the LEAST operational overhead?

- A. Export the existing API to an OpenAPI file
- B. Create a new API Import the OpenAPI file Modify the new API to add request validation
- C. Perform the tests Modify the existing API to add request validation
- D. Deploy the existing API to production.
- E. Modify the existing API to add request validation
- F. Deploy the updated API to a new API Gateway stage Perform the tests Deploy the updated API to the API Gateway production stage.
- G. Create a new API Add the necessary resources and methods including new request validation
- H. Perform the tests Modify the existing API to add request validation
- I. Deploy the existing API to production.
- J. Clone the existing API Modify the new API to add request validation  
 Modify the existing API to add request validation Deploy the existing API to production.
- K. Perform the tests

**Answer:** D

#### Explanation:

This solution allows the developer to test the changes without affecting the production environment. Cloning an API creates a copy of the API definition that can be modified independently. The developer can then add request validation to the new API and test it using a testing tool. After verifying that the changes work as expected, the developer can apply the same changes to the existing API and deploy it to production.

Reference: Clone an API, [Enable Request Validation for an API in API Gateway]

#### NEW QUESTION 106

A developer is planning to migrate on-premises company data to Amazon S3. The data must be encrypted, and the encryption Keys must support automate annual rotation. The company must use AWS Key Management Service (AWS KMS) to encrypt the data. When type of keys should the developer use to meet these requirements?

- A. Amazon S3 managed keys
- B. Symmetric customer managed keys with key material that is generated by AWS
- C. Asymmetric customer managed keys with key material that generated by AWS
- D. Symmetric customer managed keys with imported key material

**Answer:** B

#### Explanation:

The type of keys that the developer should use to meet the requirements is symmetric customer managed keys with key material that is generated by AWS. This way, the developer can use AWS Key Management Service (AWS KMS) to encrypt the data with a symmetric key that is managed by the developer. The developer can also enable automatic annual rotation for the key, which creates new key material for the key every year. The other options either involve using Amazon S3 managed keys, which do not support automatic annual rotation, or using asymmetric keys or imported key material, which are not supported by S3 encryption.

Reference: Using AWS KMS keys to encrypt S3 objects

#### NEW QUESTION 111

A company is using Amazon RDS as the Backend database for its application. After a recent marketing campaign, a surge of read requests to the database increased the latency of data retrieval from the database.

The company has decided to implement a caching layer in front of the database. The cached content must be encrypted and must be highly available. Which solution will meet these requirements?

- A. Amazon Cloudfront
- B. Amazon ElastiCache to Memcached
- C. Amazon ElastiCache for Redis in cluster mode
- D. Amazon DynamoDB Accelerate (DAX)

**Answer:** C

#### Explanation:

This solution meets the requirements because it provides a caching layer that can store and retrieve encrypted data from multiple nodes. Amazon ElastiCache for Redis supports encryption at rest and in transit, and can scale horizontally to increase the cache capacity and availability. Amazon ElastiCache for Memcached does not support encryption, Amazon CloudFront is a content delivery network that is not suitable for caching database queries, and Amazon DynamoDB Accelerator (DAX) is a caching service that only works with DynamoDB tables.

Reference: [Amazon ElastiCache for Redis Features], [Choosing a Cluster Engine]

#### NEW QUESTION 115

A developer is working on an ecommerce website The developer wants to review server logs without logging in to each of the application servers individually. The website runs on multiple Amazon EC2 instances, is written in Python, and needs to be highly available

How can the developer update the application to meet these requirements with MINIMUM changes?

- A. Rewrite the application to be cloud native and to run on AWS Lambda, where the logs can be reviewed in Amazon CloudWatch
- B. Set up centralized logging by using Amazon OpenSearch Service, Logstash, and OpenSearch Dashboards
- C. Scale down the application to one larger EC2 instance where only one instance is recording logs
- D. Install the unified Amazon CloudWatch agent on the EC2 instances Configure the agent to push the application logs to CloudWatch

**Answer:** D

#### Explanation:

The unified Amazon CloudWatch agent can collect both system metrics and log files from Amazon EC2 instances and on-premises servers. By installing and configuring the agent on the EC2 instances, the developer can easily access and analyze the application logs in CloudWatch without logging in to each server individually. This option requires minimum changes to the existing application and does not affect its availability or scalability. References

? Using the CloudWatch Agent

? Collecting Metrics and Logs from Amazon EC2 Instances and On-Premises Servers with the CloudWatch Agent

#### NEW QUESTION 120

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

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

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

**Answer:** D

#### Explanation:

When creating an Amazon DynamoDB table using the AWS CLI, server-side encryption with an AWS owned encryption key is enabled by default. Therefore, the developer does not need to create an AWS KMS key or specify the `KMSMasterKeyId` parameter. Option A and B are incorrect because they suggest creating customer-managed and AWS-managed KMS keys, which are not needed in this scenario. Option C is also incorrect because AWS owned keys are automatically used for server-side encryption by default.

#### NEW QUESTION 121

A developer is designing a serverless application with two AWS Lambda functions to process photos. One Lambda function stores objects in an Amazon S3 bucket and stores the associated metadata in an Amazon DynamoDB table. The other Lambda function fetches the objects from the S3 bucket by using the metadata from the DynamoDB table. Both Lambda functions use the same Python library to perform complex computations and are approaching the quota for the maximum size of zipped deployment packages.

What should the developer do to reduce the size of the Lambda deployment packages with the LEAST operational overhead?

- A. Package each Python library in its own .zip file archive.
- B. Deploy each Lambda function with its own copy of the library.
- C. Create a Lambda layer with the required Python library.
- D. Use the Lambda layer in both Lambda functions.
- E. Combine the two Lambda functions into one Lambda function.
- F. Deploy the Lambda function as a single .zip file archive.
- G. Download the Python library to an S3 bucket.
- H. Program the Lambda functions to reference the object URLs.

**Answer:** B

#### Explanation:

AWS Lambda is a service that lets developers run code without provisioning or managing servers. Lambda layers are a distribution mechanism for libraries, custom runtimes, and other dependencies. The developer can create a Lambda layer with the

required Python library and use the layer in both Lambda functions. This will reduce the size of the Lambda deployment packages and avoid reaching the quota for the maximum size of zipped deployment packages. The developer can also benefit from using layers to manage dependencies separately from function code.

References:

? [What Is AWS Lambda? - AWS Lambda]

? [AWS Lambda Layers - AWS Lambda]

#### NEW QUESTION 122

A developer is troubleshooting an application in an integration environment. In the application, an Amazon Simple Queue Service (Amazon SQS) queue consumes messages and then an AWS Lambda function processes the messages. The Lambda function transforms the messages and makes an API call to a third-party

service.  
There has been an increase in application usage. The third-party API frequently returns an HTTP 429 Too Many Requests error message. The error message prevents a significant number of messages from being processed successfully.  
How can the developer resolve this issue?

- A. Increase the SQS event source's batch size setting.
- B. Configure provisioned concurrency for the Lambda function based on the third-party API's documented rate limits.
- C. Increase the retry attempts and maximum event age in the Lambda function's asynchronous configuration.
- D. Configure maximum concurrency on the SQS event source based on the third-party service's documented rate limits.

**Answer: D**

**Explanation:**

? Maximum concurrency for SQS as an event source allows customers to control the maximum concurrent invokes by the SQS event source<sup>1</sup>. When multiple SQS event sources are configured to a function, customers can control the maximum concurrent invokes of individual SQS event source<sup>1</sup>.  
? In this scenario, the developer needs to resolve the issue of the third-party API frequently returning an HTTP 429 Too Many Requests error message, which prevents a significant number of messages from being processed successfully. To achieve this, the developer can follow these steps:  
? By using this solution, the developer can reduce the frequency of HTTP 429 errors and improve the message processing success rate. The developer can also avoid throttling or blocking by the third-party API.

**NEW QUESTION 126**

An ecommerce application is running behind an Application Load Balancer. A developer observes some unexpected load on the application during non-peak hours. The developer wants to analyze patterns for the client IP addresses that use the application. Which HTTP header should the developer use for this analysis?

- A. The X-Forwarded-Proto header
- B. The X-F Forwarded-Host header
- C. The X-Forwarded-For header
- D. The X-Forwarded-Port header

**Answer: C**

**Explanation:**

The HTTP header that the developer should use for this analysis is the X- Forwarded-For header. This header contains the IP address of the client that made the request to the Application Load Balancer. The developer can use this header to analyze patterns for the client IP addresses that use the application. The other headers either contain information about the protocol, host, or port of the request, which are not relevant for the analysis.  
Reference: How Application Load Balancer works with your applications

**NEW QUESTION 127**

A developer is creating a template that uses AWS CloudFormation to deploy an application. The application is serverless and uses Amazon API Gateway, Amazon DynamoDB, and AWS Lambda.  
Which AWS service or tool should the developer use to define serverless resources in YAML?

- A. CloudFormation serverless intrinsic functions
- B. AWS Elastic Beanstalk
- C. AWS Serverless Application Model (AWS SAM)
- D. AWS Cloud Development Kit (AWS CDK)

**Answer: C**

**Explanation:**

AWS Serverless Application Model (AWS SAM) is an open-source framework that enables developers to build and deploy serverless applications on AWS. AWS SAM uses a template specification that extends AWS CloudFormation to simplify the

definition of serverless resources such as API Gateway, DynamoDB, and Lambda. The developer can use AWS SAM to define serverless resources in YAML and deploy them using the AWS SAM CLI.

References:

? [What Is the AWS Serverless Application Model (AWS SAM)? - AWS Serverless Application Model]

? [AWS SAM Template Specification - AWS Serverless Application Model]

#### NEW QUESTION 129

A developer is storing sensitive data generated by an application in Amazon S3. The developer wants to encrypt the data at rest. A company policy requires an audit trail of when the AWS Key Management Service (AWS KMS) key was used and by whom.

Which encryption option will meet these requirements?

- A. Server-side encryption with Amazon S3 managed keys (SSE-S3)
- B. Server-side encryption with AWS KMS managed keys (SSE-KMS)
- C. Server-side encryption with customer-provided keys (SSE-C)
- D. Server-side encryption with self-managed keys

**Answer: B**

#### Explanation:

This solution meets the requirements because it encrypts data at rest using AWS KMS keys and provides an audit trail of when and by whom they were used. Server-side encryption with AWS KMS managed keys (SSE-KMS) is a feature of Amazon S3 that encrypts data using keys that are managed by AWS KMS. When SSE-KMS is enabled for an S3 bucket or object, S3 requests AWS KMS to generate data keys and encrypts data using these keys. AWS KMS logs every use of its keys in AWS CloudTrail, which records all API calls to AWS KMS as events. These events include information such as who made the request, when it was made, and which key was used. The company policy can use CloudTrail logs to audit critical events related to their data encryption and access. Server-side encryption with Amazon S3 managed keys (SSE-S3) also encrypts data at rest using keys that are managed by S3, but does not provide an audit trail of key usage. Server-side encryption with customer-provided keys (SSE-C) and server-side encryption with self-managed keys also encrypt data at rest using keys that are provided or managed by customers, but do not provide an audit trail of key usage and require additional overhead for key management.

Reference: [Protecting Data Using Server-Side Encryption with AWS KMS–Managed Encryption Keys (SSE-KMS)], [Logging AWS KMS API calls with AWS CloudTrail]

#### NEW QUESTION 131

A developer wants to add request validation to a production environment Amazon API Gateway API. The developer needs to test the changes before the API is deployed to the production environment. For the test, the developer will send test requests to the API through a testing tool.

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

- A. Export the existing API to an OpenAPI file
- B. Create a new AP
- C. Import the OpenAPI file.
- D. Perform the test
- E. Modify the existing API to add request validation
- F. Deploy the existing API to production.
- G. Modify the existing API to add request validation
- H. Deploy the updated API to a new API Gateway stage
- I. Perform the test
- J. Deploy the updated API to the API Gateway production stage.
- K. Create a new AP
- L. Add the necessary resources and methods, including new request validation
- M. Perform the test
- N. Modify the existing API to add request validation
- O. Deploy the existing API to production.
- P. Clone the existing AP
- Q. Modify the new API to add request validation
- R. Perform the test
- S. Modify the existing API to add request validation
- T. Deploy the existing API to production.

**Answer: B**

**Explanation:**

Amazon API Gateway allows you to create, deploy, and manage a RESTful API to expose backend HTTP endpoints, AWS Lambda functions, or other AWS services<sup>1</sup>. You can use API Gateway to perform basic validation of an API request before proceeding with the integration request<sup>1</sup>. When the validation fails, API Gateway immediately fails the request, returns a 400 error response to the caller, and publishes the validation results in CloudWatch Logs<sup>1</sup>. To test changes before deploying to a production environment, you can modify the existing API to add request validation and deploy the updated API to a new API Gateway stage<sup>1</sup>. This allows you to perform tests without affecting the production environment. Once testing is complete and successful, you can then deploy the updated API to the API Gateway production stage<sup>1</sup>. This approach has the least operational overhead as it avoids unnecessary creation of new APIs or exporting and importing of APIs. It leverages the existing infrastructure and only requires changes in the configuration of the existing API<sup>1</sup>.

**NEW QUESTION 136**

A developer is incorporating AWS X-Ray into an application that handles personal identifiable information (PII). The application is hosted on Amazon EC2 instances. The application trace messages include encrypted PII and go to Amazon CloudWatch. The developer needs to ensure that no PII goes outside of the EC2 instances. Which solution will meet these requirements?

- A. Manually instrument the X-Ray SDK in the application code.
- B. Use the X-Ray auto-instrumentation agent.
- C. Use Amazon Macie to detect and hide PII
- D. Call the X-Ray API from AWS Lambda.
- E. Use AWS Distro for Open Telemetry.

**Answer:** A

**Explanation:**

This solution will meet the requirements by allowing the developer to control what data is sent to X-Ray and CloudWatch from the application code. The developer can filter out any PII from the trace messages before sending them to X-Ray and CloudWatch, ensuring that no PII goes outside of the EC2 instances. Option B is not optimal because it will automatically instrument all incoming and outgoing requests from the application, which may include PII in the trace messages. Option C is not optimal because it will require additional services and costs to use Amazon Macie and AWS Lambda, which may not be able to detect and hide all PII from the trace messages. Option D is not optimal because it will use Open Telemetry instead of X-Ray, which may not be compatible with CloudWatch and other AWS services.

References: [AWS X-Ray SDKs]

**NEW QUESTION 138**

A developer is building an application that uses AWS API Gateway APIs, AWS Lambda function, and AWS Dynamic DB tables. The developer uses the AWS Serverless Application Model (AWS SAM) to build and run serverless applications on AWS. Each time the developer pushes changes for only to the Lambda functions, all the artifacts in the application are rebuilt. The developer wants to implement AWS SAM Accelerate by running a command to only redeploy the Lambda functions that have changed. Which command will meet these requirements?

- A. `sam deploy -force-upload`
- B. `sam deploy -no-execute-changeset`
- C. `sam package`
- D. `sam sync -watch`

**Answer:** D

**Explanation:**

The command that will meet the requirements is `sam sync -watch`. This command enables AWS SAM Accelerate mode, which allows the developer to only redeploy the Lambda functions that have changed. The `-watch` flag enables file watching, which automatically detects changes in the source code and triggers a redeployment. The other commands either do not enable AWS SAM Accelerate mode, or do not redeploy the Lambda functions automatically.

Reference: AWS SAM Accelerate

**NEW QUESTION 141**

A development team wants to build a continuous integration/continuous delivery (CI/CD) pipeline. The team is using AWS CodePipeline to automate the code build and deployment. The team wants to store the program code to prepare for the CI/CD pipeline. Which AWS service should the team use to store the program code?

- A. AWS CodeDeploy
- B. AWS CodeArtifact
- C. AWS CodeCommit
- D. Amazon CodeGuru

**Answer:** C

**Explanation:**

AWS CodeCommit is a service that provides fully managed source control for hosting secure and scalable private Git repositories. The development team can use CodeCommit to store the program code and prepare for the CI/CD pipeline. CodeCommit integrates with other AWS services such as CodePipeline, CodeBuild, and CodeDeploy to automate the code build and deployment process.

References:

? [What Is AWS CodeCommit? - AWS CodeCommit]

? [AWS CodePipeline - AWS CodeCommit]

**NEW QUESTION 143**

A company is using Amazon OpenSearch Service to implement an audit monitoring system. A developer needs to create an AWS CloudFormation custom resource that is associated with an AWS Lambda function to configure the OpenSearch Service domain. The Lambda function must access the

OpenSearch Service domain by using Open Search Service internal master user credentials.  
 What is the MOST secure way to pass these credentials to the Lambdas function?

- A. Use a CloudFormation parameter to pass the master user credentials at deployment to the OpenSearch Service domain's MasterUserOptions and the Lambda function's environment variable
- B. Set the No Echo attribute to true.
- C. Use a CloudFormation parameter to pass the master user credentials at deployment to the OpenSearch Service domain's MasterUserOptions and to create a parameter
- D. In AWS Systems Manager Parameter Store
- E. Set the No Echo attribute to true
- F. Create an IAM role that has the ssm GetParameter permission
- G. Assign the role to the Lambda function
- H. Store the parameter name as the Lambda function's environment variable
- I. Resolve the parameter's value at runtime.
- J. Use a CloudFormation parameter to pass the master user credentials at deployment to the OpenSearch Service domain's MasterUserOptions and the Lambda function's environment variable. We Encrypt the parameter's value by using the AWS Key Management Service (AWS KMS) encrypt command.
- K. Use CloudFormation to create an AWS Secrets Manager Secret
- L. Use a CloudFormation dynamic reference to retrieve the secret's value for the OpenSearch Service domain's MasterUserOptions
- M. Create an IAM role that has the secretsmanager:GetSecretValue permission
- N. GetSecretValue permission
- O. Assign the role to the Lambda Function Store the secret's name as the Lambda function's environment variable
- P. Resolve the secret's value at runtime.

**Answer: D**

**Explanation:**

The solution that will meet the requirements is to use CloudFormation to create an AWS Secrets Manager secret. Use a CloudFormation dynamic reference to retrieve the secret's value for the OpenSearch Service domain's MasterUserOptions. Create an IAM role that has the secretsmanager:GetSecretValue permission. Assign the role to the Lambda function. Store the secret's name as the Lambda function's environment variable. Resolve the secret's value at runtime. This way, the developer can pass the credentials to the Lambda function in a secure way, as AWS Secrets Manager encrypts and manages the secrets. The developer can also use a dynamic reference to avoid exposing the secret's value in plain text in the CloudFormation template. The other options either involve passing the credentials as plain text parameters, which is not secure, or encrypting them with AWS KMS, which is less convenient than using AWS Secrets Manager.

Reference: Using dynamic references to specify template values

**NEW QUESTION 146**

A developer needs to deploy an application running on AWS Fargate using Amazon ECS. The application has environment variables that must be passed to a container for the application to initialize.  
 How should the environment variables be passed to the container?

- A. Define an array that includes the environment variables under the environment parameter within the service definition.
- B. Define an array that includes the environment variables under the environment parameter within the task definition.
- C. Define an array that includes the environment variables under the entryPoint parameter within the task definition.
- D. Define an array that includes the environment variables under the entryPoint parameter within the service definition.

**Answer: B**

**Explanation:**

This solution allows the environment variables to be passed to the container when it is launched by AWS Fargate using Amazon ECS. The task definition is a text file that describes one or more containers that form an application. It contains various parameters for configuring the containers, such as CPU and memory requirements, network mode, and environment variables. The environment parameter is an array of key-value pairs that specify environment variables to pass to a container. Defining an array that includes the environment variables under the entryPoint parameter within the task definition

will not pass them to the container, but use them as command-line arguments for overriding the default entry point of a container.

Defining an array that includes the environment variables under the environment or entryPoint parameter within the service definition will not pass them to the container, but cause an error because these parameters are not valid for a service definition.

Reference: [Task Definition Parameters], [Environment Variables]

**NEW QUESTION 151**

A company runs an application on AWS. The application uses an AWS Lambda function that is configured with an Amazon Simple Queue Service (Amazon SQS) queue called high priority queue as the event source. A developer is updating the Lambda function with another SQS queue called low priority queue as the event source. The Lambda function must always read up to 10 simultaneous messages from the high priority queue before processing messages from low priority queue. The Lambda function must be limited to 100 simultaneous invocations.  
 Which solution will meet these requirements'?

- A. Set the event source mapping batch size to 10 for the high priority queue and to 90 for the low priority queue
- B. Set the delivery delay to 0 seconds for the high priority queue and to 10 seconds for the low priority queue
- C. Set the event source mapping maximum concurrency to 10 for the high priority queue and to 90 for the low priority queue
- D. Set the event source mapping batch window to 10 for the high priority queue and to 90 for the low priority queue

**Answer: C**

**Explanation:**

Setting the event source mapping maximum concurrency is the best way to control how many messages from each queue are processed by the Lambda function at a time. The maximum concurrency setting limits the number of batches that can be processed concurrently from the same event source.

By setting it to 10 for the high priority queue and to 90 for the low priority queue, the developer can ensure that the Lambda function always reads up to 10 simultaneous messages from the high priority queue before processing messages from the low priority queue, and that the total number of concurrent invocations does not exceed 100. The other solutions are either not effective or not relevant. The batch size setting controls how many messages are sent to the Lambda function in a single invocation, not how many invocations are allowed at a time. The delivery delay setting controls how long a message is invisible in the queue after it is sent, not how often it is processed by the Lambda function. The batch window setting controls how long the event source mapping can buffer messages before sending a batch, not how many batches are processed concurrently. References

? Using AWS Lambda with Amazon SQS

- ? AWS Lambda Event Source Mapping - Examples and best practices | Shisho Dojo
- ? Lambda event source mappings - AWS Lambda
- ? aws\_lambda\_event\_source\_mapping - Terraform Registry

**NEW QUESTION 156**

A developer is testing a RESTful application that is deployed by using Amazon API Gateway and AWS Lambda. When the developer tests the user login by using credentials that are not valid, the developer receives an HTTP 405 METHOD\_NOT\_ALLOWED error. The developer has verified that the test is sending the correct request for the resource.

Which HTTP error should the application return in response to the request?

- A. HTTP 401
- B. HTTP 404
- C. HTTP 503
- D. HTTP 505

**Answer:** A

**Explanation:**

The HTTP 401 error indicates that the request has not been applied because it lacks valid authentication credentials for the target resource. This is the appropriate error code to return when the user login fails due to invalid credentials. The HTTP 405 error means that the method specified in the request is not allowed for the resource identified by the request URI, which is not the case here. The other error codes are not relevant to the authentication failure scenario.

References

? HTTP Status Codes

? AWS Lambda Function Errors in API Gateway

**NEW QUESTION 160**

A mobile app stores blog posts in an Amazon DynamoDB table. Millions of posts are added every day and each post represents a single item in the table. The mobile app requires only recent posts. Any post that is older than 48 hours can be removed.

What is the MOST cost-effective way to delete posts that are older than 48 hours?

- A. For each item add a new attribute of type String that has a timestamp that is set to the blog post creation time.
- B. Create a script to find old posts with a table scan and remove posts that are older than 48 hours by using the Batch Write Item API operation.
- C. Schedule a cron job on an Amazon EC2 instance once an hour to start the script.
- D. For each item add a new attribute of type Number that has a timestamp that is set to the blog post creation time.
- E. Create a script to find old posts with a table scan and remove posts that are older than 48 hours by using the Batch Write item API operation.
- F. Place the script in a container image.
- G. Schedule an Amazon Elastic Container Service (Amazon ECS) task on AWS Fargate that invokes the container every 5 minutes.
- H. For each item, add a new attribute of type Date that has a timestamp that is set to 48 hours after the blog post creation time.
- I. Create a global secondary index (GSI) that uses the new attribute as a sort key.
- J. Create an AWS Lambda function that references the GSI and removes expired items by using the Batch Write item API operation. Schedule the function with an Amazon CloudWatch event every minute.
- K. For each item add a new attribute of type Number that has a timestamp that is set to 48 hours after the blog post creation time.
- L. Create a global secondary index (GSI) that uses the expiration time as a sort key, which may consume more storage space and incur more costs.
- M. Configure the DynamoDB table with a TTL that references the new attribute.
- N. Create a script to find old posts with a table scan and remove posts that are older than 48 hours by using the Batch Write item API operation.

**Answer:** D

**Explanation:**

This solution will meet the requirements by using the Time to Live (TTL) feature of DynamoDB, which enables automatically deleting items from a table after a certain time period. The developer can add a new attribute of type Number that has a timestamp that is set to 48 hours after the blog post creation time, which represents the expiration time of the item. The developer can configure the DynamoDB table with a TTL that references the new attribute, which instructs DynamoDB to delete the item when the current time is greater than or equal to the expiration time. This solution is also cost-effective as it does not incur any additional charges for deleting expired items.

Option A is not optimal because it will create a script to find and remove old posts with a table scan and a batch write item API operation, which may consume more read and write capacity units and incur more costs. Option B is not optimal because it will use Amazon Elastic Container Service (Amazon ECS) and AWS Fargate to run the script, which may introduce additional costs and complexity for managing and scaling containers. Option C is not optimal because it will create a global secondary index (GSI) that uses the expiration time as a sort key, which may consume more storage space and incur more costs.

References: Time To Live, Managing DynamoDB Time To Live (TTL)

**NEW QUESTION 162**

A developer needs to build an AWS CloudFormation template that self-populates the AWS Region variable that deploys the CloudFormation template.

What is the MOST operationally efficient way to determine the Region in which the template is being deployed?

- A. Use the AWS::Region pseudo parameter.
- B. Require the Region as a CloudFormation parameter.
- C. Find the Region from the AWS::StackId pseudo parameter by using the Fn::Split intrinsic function.
- D. Dynamically import the Region by referencing the relevant parameter in AWS Systems Manager Parameter Store.

**Answer:** A

**Explanation:**

<https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/mappings-section-structure.html>

<https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/pseudo-parameter-reference.html>

<https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/pseudo-parameter-reference.html>

**NEW QUESTION 166**

A company is building an application for stock trading. The application needs sub-millisecond latency for processing trade requests. The company uses Amazon DynamoDB to store all the trading data that is used to process each trading request. A development team performs load testing on the application and finds that the

data retrieval time is higher

than expected. The development team needs a solution that reduces the data retrieval time with the least possible effort.

Which solution meets these requirements'?

- A. Add local secondary indexes (LSIs) for the trading data.
- B. Store the trading data in Amazon S3 and use S3 Transfer Acceleration.
- C. Add retries with exponential back off for DynamoDB queries.
- D. Use DynamoDB Accelerator (DAX) to cache the trading data.

**Answer:** D

**Explanation:**

This solution will meet the requirements by using DynamoDB Accelerator (DAX), which is a fully managed, highly available, in-memory cache for DynamoDB that delivers up to a 10 times performance improvement - from milliseconds to microseconds - even at millions of requests per second. The developer can use DAX to cache the trading data that is used to process each trading request, which will reduce the data retrieval time with the least possible effort. Option A is not optimal because it will add local secondary indexes (LSIs) for the trading data, which may not improve the performance or reduce the latency of data retrieval, as LSIs are stored on the same partition as the base table and share the same provisioned throughput. Option B is not optimal because it will store the trading data in Amazon S3 and use S3 Transfer Acceleration, which is a feature that enables fast, easy, and secure transfers of files over long distances between S3 buckets and clients, not between DynamoDB and clients. Option C is not optimal because it will add retries with exponential backoff for DynamoDB queries, which is a strategy to handle transient errors by retrying failed requests with increasing delays, not by reducing data retrieval time.

References: [DynamoDB Accelerator (DAX)], [Local Secondary Indexes]

**NEW QUESTION 168**

A developer is investigating an issue in part of a company's application. In the application messages are sent to an Amazon Simple Queue Service (Amazon SQS) queue. The AWS Lambda function polls messages from the SQS queue and sends email messages by using Amazon Simple Email Service (Amazon SES). Users have been receiving duplicate email messages during periods of high traffic.

Which reasons could explain the duplicate email messages? (Select TWO.)

- A. Standard SQS queues support at-least-once message delivery
- B. Standard SQS queues support exactly-once processing, so the duplicate email messages are because of user error.
- C. Amazon SES has the DomainKeys Identified Mail (DKIM) authentication incorrectly configured
- D. The SQS queue's visibility timeout is lower than or the same as the Lambda function's timeout.
- E. The Amazon SES bounce rate metric is too high.

**Answer:** AD

**Explanation:**

Standard SQS queues support at-least-once message delivery, which means that a message can be delivered more than once to the same or different consumers. This can happen if the message is not deleted from the queue before the visibility timeout expires, or if there is a network issue or a system failure. The SQS queue's visibility timeout is the period of time that a message is invisible to other consumers after it is received by one consumer. If the visibility timeout is lower than or the same as the Lambda function's timeout, the Lambda function might not be able to process and delete the message before it becomes visible again, leading to duplicate processing and email messages. To avoid this, the visibility timeout should be set to at least 6 times the length of the Lambda function's timeout. The other options are not related to the issue of duplicate email messages. References

? Using the Amazon SQS message deduplication ID

? Exactly-once processing - Amazon Simple Queue Service

? Amazon SQS duplicated messages in queue - Stack Overflow

? amazon web services - How long can duplicate SQS messages persist ...

? Standard SQS - Duplicate message | AWS re:Post - Amazon Web Services, Inc.

**NEW QUESTION 170**

A developer is building a microservices-based application by using Python on AWS and several AWS services. The developer must use AWS X-Ray. The developer views the service map by using the console to view the service dependencies. During testing, the developer notices that some services are missing from the service map.

What can the developer do to ensure that all services appear in the X-Ray service map?

- A. Modify the X-Ray Python agent configuration in each service to increase the sampling rate
- B. Instrument the application by using the X-Ray SDK for Python
- C. Install the X-Ray SDK for all the services that the application uses
- D. Enable X-Ray data aggregation in Amazon CloudWatch Logs for all the services that the application uses
- E. Increase the X-Ray service map timeout value in the X-Ray console

**Answer:** B

**Explanation:**

The X-Ray SDK for Python provides libraries and tools for instrumenting Python applications that use AWS services and other AWS X-Ray integrations. By installing the X-Ray SDK for all the services that the application uses, the developer can ensure that all the service dependencies are captured and displayed in the X-Ray service map. The other options are not relevant or effective for this scenario. References

? AWS X-Ray SDK for Python

? Instrumenting a Python Application

**NEW QUESTION 174**

.....

## THANKS FOR TRYING THE DEMO OF OUR PRODUCT

Visit Our Site to Purchase the Full Set of Actual AWS-Certified-Developer-Associate Exam Questions With Answers.

We Also Provide Practice Exam Software That Simulates Real Exam Environment And Has Many Self-Assessment Features. Order the AWS-Certified-Developer-Associate Product From:

<https://www.2passeasy.com/dumps/AWS-Certified-Developer-Associate/>

## Money Back Guarantee

### **AWS-Certified-Developer-Associate Practice Exam Features:**

- \* AWS-Certified-Developer-Associate Questions and Answers Updated Frequently
- \* AWS-Certified-Developer-Associate Practice Questions Verified by Expert Senior Certified Staff
- \* AWS-Certified-Developer-Associate Most Realistic Questions that Guarantee you a Pass on Your FirstTry
- \* AWS-Certified-Developer-Associate Practice Test Questions in Multiple Choice Formats and Updatesfor 1 Year