
QUESTION 1

A company is implementing an application on Amazon EC2 instances. The application needs to process incoming transactions. When the application detects a transaction that is not valid, the application must send a chat message to the company's support team. To send the message, the application needs to retrieve the access token to authenticate by using the chat API.

A developer needs to implement a solution to store the access token. The access token must be encrypted at rest and in transit. The access token must also be accessible from other AWS accounts. Which solution will meet these requirements with the LEAST management overhead?

- A. Use an AWS Systems Manager Parameter Store SecureString parameter that uses an AWS Key Management Service (AWS KMS) AWS managed key to store the access token. Add a resource-based policy to the parameter to allow access from other accounts. Update the IAM role of the EC2 instances with permissions to access Parameter Store. Retrieve the token from Parameter Store with the decrypt flag enabled. Use the decrypted access token to send the message to the chat.
- B. Encrypt the access token by using an AWS Key Management Service (AWS KMS) customer managed key. Store the access token in an Amazon DynamoDB table. Update the IAM role of the EC2 instances with permissions to access DynamoDB and AWS KMS. Retrieve the token from DynamoDB. Decrypt the token by using AWS KMS on the EC2 instances. Use the decrypted access token to send the message to the chat.
- C. Use AWS Secrets Manager with an AWS Key Management Service (AWS KMS) customer managed key to store the access token. Add a resource-based policy to the secret to allow access from other accounts. Update the IAM role of the EC2 instances with permissions to access Secrets Manager. Retrieve the token from Secrets Manager. Use the decrypted access token to send the message to the chat.
- D. Encrypt the access token by using an AWS Key Management Service (AWS KMS) AWS managed key. Store the access token in an Amazon S3 bucket. Add a bucket policy to the S3 bucket to allow access from other accounts. Update the IAM role of the EC2 instances with permissions to access Amazon S3 and AWS KMS. Retrieve the token from the S3 bucket. Decrypt the token by using AWS KMS on the EC2 instances. Use the decrypted access token to send the message to the chat.

Answer: C

Explanation:

<https://aws.amazon.com/premiumsupport/knowledge-center/secrets-manager-share-betweenaccounts/>
https://docs.aws.amazon.com/secretsmanager/latest/userguide/auth-and-access_examples_cross.html

QUESTION 2

A company is running Amazon EC2 instances in multiple AWS accounts. A developer needs to implement an application that collects all the lifecycle events of the EC2 instances. The application needs to store the lifecycle events in a single Amazon Simple Queue Service (Amazon SQS) queue in

the company's main AWS account for further processing.
Which solution will meet these requirements?

- A. Configure Amazon EC2 to deliver the EC2 instance lifecycle events from all accounts to the Amazon EventBridge event bus of the main account. Add an EventBridge rule to the event bus of the main account that matches all EC2 instance lifecycle events. Add the SQS queue as a target of the rule.
- B. Use the resource policies of the SQS queue in the main account to give each account permissions to write to that SQS queue. Add to the Amazon EventBridge event bus of each account an EventBridge rule that matches all EC2 instance lifecycle events. Add the SQS queue in the main account as a target of the rule.
- C. Write an AWS Lambda function that scans through all EC2 instances in the company accounts to detect EC2 instance lifecycle changes. Configure the Lambda function to write a notification message to the SQS queue in the main account if the function detects an EC2 instance lifecycle change. Add an Amazon EventBridge scheduled rule that invokes the Lambda function every minute.
- D. Configure the permissions on the main account event bus to receive events from all accounts. Create an Amazon EventBridge rule in each account to send all the EC2 instance lifecycle events to the main account event bus. Add an EventBridge rule to the main account event bus that matches all EC2 instance lifecycle events. Set the SQS queue as a target for the rule.

Answer: D

Explanation:

Amazon EC2 instances can send the state-change notification events to Amazon EventBridge.

<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/monitoring-instance-state-changes.html>

Amazon EventBridge can send and receive events between event buses in AWS accounts.

<https://docs.aws.amazon.com/eventbridge/latest/userguide/eb-cross-account.html>

QUESTION 3

An application is using Amazon Cognito user pools and identity pools for secure access. A developer wants to integrate the user-specific file upload and download features in the application with Amazon S3. The developer must ensure that the files are saved and retrieved in a secure manner and that users can access only their own files. The file sizes range from 3 KB to 300 MB.
Which option will meet these requirements with the HIGHEST level of security?

- A. Use S3 Event Notifications to validate the file upload and download requests and update the user interface (UI).
- B. Save the details of the uploaded files in a separate Amazon DynamoDB table. Filter the list of files in the user interface (UI) by comparing the current user ID with the user ID associated with the file in the table.
- C. Use Amazon API Gateway and an AWS Lambda function to upload and download files. Validate each request in the Lambda function before performing the requested operation.
- D. Use an IAM policy within the Amazon Cognito identity prefix to restrict users to use their own

folders in Amazon S3.

Answer: D

Explanation:

<https://docs.aws.amazon.com/cognito/latest/developerguide/amazon-cognito-integrating-userpools-with-identity-pools.html>

QUESTION 4

A company is building a scalable data management solution by using AWS services to improve the speed and agility of development. The solution will ingest large volumes of data from various sources and will process this data through multiple business rules and transformations.

The solution requires business rules to run in sequence and to handle reprocessing of data if errors occur when the business rules run. The company needs the solution to be scalable and to require the least possible maintenance.

Which AWS service should the company use to manage and automate the orchestration of the data flows to meet these requirements?

- A. AWS Batch
- B. AWS Step Functions
- C. AWS Glue
- D. AWS Lambda

Answer: B

Explanation:

<https://docs.aws.amazon.com/step-functions/latest/dg/welcome.html>

QUESTION 5

A developer has created an AWS Lambda function that is written in Python. The Lambda function reads data from objects in Amazon S3 and writes data to an Amazon DynamoDB table. The function is successfully invoked from an S3 event notification when an object is created. However, the function fails when it attempts to write to the DynamoDB table.

What is the MOST likely cause of this issue?

- A. The Lambda function's concurrency limit has been exceeded.
- B. DynamoDB table requires a global secondary index (GSI) to support writes.
- C. The Lambda function does not have IAM permissions to write to DynamoDB.
- D. The DynamoDB table is not running in the same Availability Zone as the Lambda function.

Answer: C

Explanation:

https://docs.aws.amazon.com/IAM/latest/UserGuide/reference_policies_examples_lambda-accessdynamodb.html

QUESTION 6

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.

QUESTION 7

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: B, C

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 tables 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`.

Reference:

[BatchGetItem - Amazon DynamoDB]

[Working with Queries and Scans - Amazon DynamoDB]

[Best Practices for Handling DynamoDB Throttling Errors]

QUESTION 8

A company is running a custom application on a set of on-premises Linux servers that are accessed using Amazon API Gateway. AWS X-Ray tracing has been enabled on the API test stage. How can a developer enable X-Ray tracing on the on-premises servers with the LEAST amount of configuration?

- A. Install and run the X-Ray SDK on the on-premises servers to capture and relay the data to the XRay service.
- B. Install and run the X-Ray daemon on the on-premises servers to capture and relay the data to the X-Ray service.
- C. Capture incoming requests on-premises and configure an AWS Lambda function to pull, process, and relay relevant data to X-Ray using the `PutTraceSegments` API call.
- D. Capture incoming requests on-premises and configure an AWS Lambda function to pull, process, and relay relevant data to X-Ray using the `PutTelemetryRecords` API call.

Answer: B

Explanation:

The X-Ray daemon is a software that collects trace data from the X-Ray SDK and relays it to the X-Ray service. The X-Ray daemon can run on any platform that supports Go, including Linux, Windows, and macOS. The developer can install and run the X-Ray daemon on the on-premises servers to capture and relay the data to the X-Ray service with minimal configuration. The X-Ray SDK is used to instrument the application code, not to capture and relay data. The Lambda function solutions are more complex and require additional configuration.

Reference:

[AWS X-Ray concepts - AWS X-Ray]

[Setting up AWS X-Ray - AWS X-Ray]

QUESTION 9

A company wants to share information with a third party. The third party has an HTTP API endpoint that the company can use to share the information. The company has the required API key to access

the HTTP API.

The company needs a way to manage the API key by using code. The integration of the API key with the application code cannot affect application performance.

Which solution will meet these requirements MOST securely?

- A. Store the API credentials in AWS Secrets Manager. Retrieve the API credentials at runtime by using the AWS SDK. Use the credentials to make the API call.
- B. Store the API credentials in a local code variable. Push the code to a secure Git repository. Use the local code variable at runtime to make the API call.
- C. Store the API credentials as an object in a private Amazon S3 bucket. Restrict access to the S3 object by using IAM policies. Retrieve the API credentials at runtime by using the AWS SDK. Use the credentials to make the API call.
- D. Store the API credentials in an Amazon DynamoDB table. Restrict access to the table by using resource-based policies. Retrieve the API credentials at runtime by using the AWS SDK. Use the credentials to make the API call.

Answer: A

Explanation:

AWS Secrets Manager is a service that helps securely store, rotate, and manage secrets such as API keys, passwords, and tokens. The developer can store the API credentials in AWS Secrets Manager and retrieve them at runtime by using the AWS SDK. This solution will meet the requirements of security, code management, and performance. Storing the API credentials in a local code variable or an S3 object is not secure, as it exposes the credentials to unauthorized access or leakage. Storing the API credentials in a DynamoDB table is also not secure, as it requires additional encryption and access control measures. Moreover, retrieving the credentials from S3 or DynamoDB may affect application performance due to network latency.

Reference:

[What Is AWS Secrets Manager? - AWS Secrets Manager]

[Retrieving a Secret - AWS Secrets Manager]

QUESTION 10

A developer is deploying a new application to Amazon Elastic Container Service (Amazon ECS). The developer needs to securely store and retrieve different types of variables. These variables include authentication information for a remote API, the URL for the API, and credentials. The authentication information and API URL must be available to all current and future deployed versions of the application across development, testing, and production environments.

How should the developer retrieve the variables with the FEWEST application changes?

- A. Update the application to retrieve the variables from AWS Systems Manager Parameter Store. Use unique paths in Parameter Store for each variable in each environment. Store the credentials in AWS Secrets Manager in each environment.

- B. Update the application to retrieve the variables from AWS Key Management Service (AWS KMS). Store the API URL and credentials as unique keys for each environment.
- C. Update the application to retrieve the variables from an encrypted file that is stored with the application. Store the API URL and credentials in unique files for each environment.
- D. Update the application to retrieve the variables from each of the deployed environments. Define the authentication information and API URL in the ECS task definition as unique names during the deployment process.

Answer: A

Explanation:

AWS Systems Manager Parameter Store is a service that provides secure, hierarchical storage for configuration data management and secrets management. The developer can update the application to retrieve the variables from Parameter Store by using the AWS SDK or the AWS CLI. The developer can use unique paths in Parameter Store for each variable in each environment, such as /dev/api-url, /test/api-url, and /prod/api-url. The developer can also store the credentials in AWS Secrets Manager, which is integrated with Parameter Store and provides additional features such as automatic rotation and encryption.

Reference:

[What Is AWS Systems Manager? - AWS Systems Manager]

[Parameter Store - AWS Systems Manager]

[What Is AWS Secrets Manager? - AWS Secrets Manager]

QUESTION 11

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.

Reference:

[Amazon DynamoDB]

[Amazon Simple Storage Service (S3)]

QUESTION 12

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 accounts. Create an Amazon Cognito user pool authorizer in API Gateway to control access to the API. Use the Lambda function to store the photos and details in the DynamoDB table. Retrieve previously uploaded photos directly from the DynamoDB table.

B. Use Amazon Cognito user pools to manage user accounts. Create an Amazon Cognito user pool authorizer in API Gateway to control access to the API. 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. Retrieve previously uploaded photos by querying DynamoDB for the S3 key.

C. Create an IAM user for each user of the application during the sign-up process. Use IAM authentication to access the API Gateway API. 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. Retrieve previously uploaded photos by querying DynamoDB for the S3 key.

D. Create a users table in DynamoDB. Use the table to manage user accounts. Create a Lambda authorizer that validates user credentials against the users table. Integrate the Lambda authorizer with API Gateway to control access to the API. 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. 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 objects 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.

Reference:

[Amazon Cognito User Pools]

[Use Amazon Cognito User Pools - Amazon API Gateway]

[Amazon Simple Storage Service (S3)]

[Amazon DynamoDB]

QUESTION 13

A company receives food orders from multiple partners. The company has a microservices application that uses Amazon API Gateway APIs with AWS Lambda integration. Each partner sends orders by calling a customized API that is exposed through API Gateway. The API call invokes a shared Lambda function to process the orders.

Partners need to be notified after the Lambda function processes the orders. Each partner must receive updates for only the partner's own orders. The company wants to add new partners in the future with the fewest code changes possible.

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

- A. Create a different Amazon Simple Notification Service (Amazon SNS) topic for each partner. Configure the Lambda function to publish messages for each partner to the partner's SNS topic.
- B. Create a different Lambda function for each partner. Configure the Lambda function to notify each partner's service endpoint directly.
- C. Create an Amazon Simple Notification Service (Amazon SNS) topic. Configure the Lambda function to publish messages with specific attributes to the SNS topic. Subscribe each partner to the SNS topic. Apply the appropriate filter policy to the topic subscriptions.
- D. Create one Amazon Simple Notification Service (Amazon SNS) topic. Subscribe all partners to the SNS topic.

Answer: C

Explanation:

Amazon Simple Notification Service (Amazon SNS) is a fully managed messaging service that enables pub/sub communication between distributed systems. The developer can create an SNS topic and configure the Lambda function to publish messages with specific attributes to the topic. The developer can subscribe each partner to the SNS topic and apply the appropriate filter policy to the topic subscriptions. This way, each partner will receive updates for only their own orders based on

the message attributes. This solution will meet the requirements in the most scalable way and allow adding new partners in the future with minimal code changes.

Reference:

[Amazon Simple Notification Service (SNS)]

[Filtering Messages with Attributes - Amazon Simple Notification Service]

QUESTION 14

A financial company must store original customer records for 10 years for legal reasons. A complete record contains personally identifiable information (PII). According to local regulations, PII is available to only certain people in the company and must not be shared with third parties. The company needs to make the records available to third-party organizations for statistical analysis without sharing the PII.

A developer wants to store the original immutable record in Amazon S3. Depending on who accesses the S3 document, the document should be returned as is or with all the PII removed. The developer has written an AWS Lambda function to remove the PII from the document. The function is named `removePii`.

What should the developer do so that the company can meet the PII requirements while maintaining only one copy of the document?

- A. Set up an S3 event notification that invokes the `removePii` function when an S3 GET request is made. Call Amazon S3 by using a GET request to access the object without PII.
- B. Set up an S3 event notification that invokes the `removePii` function when an S3 PUT request is made. Call Amazon S3 by using a PUT request to access the object without PII.
- C. Create an S3 Object Lambda access point from the S3 console. Select the `removePii` function. Use S3 Access Points to access the object without PII.
- D. Create an S3 access point from the S3 console. Use the access point name to call the `GetObjectLegalHold` S3 API function. Pass in the `removePii` function name to access the object without PII.

Answer: C

Explanation:

S3 Object Lambda allows you to add your own code to process data retrieved from S3 before returning it to an application. You can use an AWS Lambda function to modify the data, such as removing PII, redacting confidential information, or resizing images. You can create an S3 Object Lambda access point and associate it with your Lambda function. Then, you can use the access point to request objects from S3 and get the modified data back. This way, you can maintain only one copy of the original document in S3 and apply different transformations depending on who accesses it.

Reference: Using AWS Lambda with Amazon S3

QUESTION 15

A developer is deploying an AWS Lambda function. The developer wants the ability to return to older

versions of the function quickly and seamlessly.

How can the developer achieve this goal with the LEAST operational overhead?

- A. Use AWS OpsWorks to perform blue/green deployments.
- B. Use a function alias with different versions.
- C. Maintain deployment packages for older versions in Amazon S3.
- D. Use AWS CodePipeline for deployments and rollbacks.

Answer: B

Explanation:

A function alias is a pointer to a specific Lambda function version. You can use aliases to create different environments for your function, such as development, testing, and production. You can also use aliases to perform blue/green deployments by shifting traffic between two versions of your function gradually. This way, you can easily roll back to a previous version if something goes wrong, without having to redeploy your code or change your configuration. Reference: AWS Lambda function aliases

QUESTION 16

A developer has written an AWS Lambda function. The function is CPU-bound. The developer wants to ensure that the function returns responses quickly.

How can the developer improve the function's performance?

- A. Increase the function's CPU core count.
- B. Increase the function's memory.
- C. Increase the function's reserved concurrency.
- D. Increase the function's timeout.

Answer: B

Explanation:

The amount of memory you allocate to your Lambda function also determines how much CPU and network bandwidth it gets. Increasing the memory size can improve the performance of CPU-bound functions by giving them more CPU power. The CPU allocation is proportional to the memory allocation, so a function with 1 GB of memory has twice the CPU power of a function with 512 MB of memory. Reference: AWS Lambda execution environment

QUESTION 17

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]

QUESTION 18

A company is building a serverless application on AWS. The application uses an AWS Lambda function to process customer orders 24 hours a day, 7 days a week. The Lambda function calls an external vendor's HTTP API to process payments.

During load tests, a developer discovers that the external vendor payment processing API occasionally times out and returns errors. The company expects that some payment processing API calls will return errors.

The company wants the support team to receive notifications in near real time only when the payment processing external API error rate exceed 5% of the total number of transactions in an hour. Developers need to use an existing Amazon Simple Notification Service (Amazon SNS) topic that is configured to notify the support team.

Which solution will meet these requirements?

- A. Write the results of payment processing API calls to Amazon CloudWatch. Use Amazon CloudWatch Logs Insights to query the CloudWatch logs. Schedule the Lambda function to check the CloudWatch logs and notify the existing SNS topic.
- B. Publish custom metrics to CloudWatch that record the failures of the external payment processing API calls. Configure a CloudWatch alarm to notify the existing SNS topic when error rate exceeds the specified rate.

- C. Publish the results of the external payment processing API calls to a new Amazon SNS topic. Subscribe the support team members to the new SNS topic.
- D. Write the results of the external payment processing API calls to Amazon S3. Schedule an Amazon Athena query to run at regular intervals. Configure Athena to send notifications to the existing SNS topic when the error rate exceeds the specified rate.

Answer: B

Explanation:

Amazon CloudWatch is a service that monitors AWS resources and applications. The developer can publish custom metrics to CloudWatch that record the failures of the external payment processing API calls. The developer can configure a CloudWatch alarm to notify the existing SNS topic when the error rate exceeds 5% of the total number of transactions in an hour. This solution will meet the requirements in a near real-time and scalable way.

Reference:

[What Is Amazon CloudWatch? - Amazon CloudWatch]

[Publishing Custom Metrics - Amazon CloudWatch]

[Creating Amazon CloudWatch Alarms - Amazon CloudWatch]

QUESTION 19

A company is offering APIs as a service over the internet to provide unauthenticated read access to statistical information that is updated daily. The company uses Amazon API Gateway and AWS Lambda to develop the APIs. The service has become popular, and the company wants to enhance the responsiveness of the APIs.

Which action can help the company achieve this goal?

- A. Enable API caching in API Gateway.
- B. Configure API Gateway to use an interface VPC endpoint.
- C. Enable cross-origin resource sharing (CORS) for the APIs.
- D. Configure usage plans and API keys in API Gateway.

Answer: A

Explanation:

Amazon API Gateway is a service that enables developers to create, publish, maintain, monitor, and secure APIs at any scale. The developer can enable API caching in API Gateway to cache responses from the backend integration point for a specified time-to-live (TTL) period. This can improve the responsiveness of the APIs by reducing the number of calls made to the backend service.

Reference:

[What Is Amazon API Gateway? - Amazon API Gateway]

[Enable API Caching to Enhance Responsiveness - Amazon API Gateway]

QUESTION 20

A developer wants to store information about movies. Each movie has a title, release year, and genre. The movie information also can include additional properties about the cast and production crew. This additional information is inconsistent across movies. For example, one movie might have an assistant director, and another movie might have an animal trainer.

The developer needs to implement a solution to support the following use cases:

For a given title and release year, get all details about the movie that has that title and release year.

For a given title, get all details about all movies that have that title.

For a given genre, get all details about all movies in that genre.

Which data store configuration will meet these requirements?

A. Create an Amazon DynamoDB table. Configure the table with a primary key that consists of the title as the partition key and the release year as the sort key. Create a global secondary index that uses the genre as the partition key and the title as the sort key.

B. Create an Amazon DynamoDB table. Configure the table with a primary key that consists of the genre as the partition key and the release year as the sort key. Create a global secondary index that uses the title as the partition key.

C. On an Amazon RDS DB instance, create a table that contains columns for title, release year, and genre. Configure the title as the primary key.

D. On an Amazon RDS DB instance, create a table where the primary key is the title and all other data is encoded into JSON format as one additional column.

Answer: A

Explanation:

Amazon DynamoDB is a fully managed NoSQL database service that provides fast and consistent performance with seamless scalability. The developer can create a DynamoDB table and configure the table with a primary key that consists of the title as the partition key and the release year as the sort key. This will enable querying for a given title and release year efficiently. The developer can also create a global secondary index that uses the genre as the partition key and the title as the sort key. This will enable querying for a given genre efficiently. The developer can store additional properties about the cast and production crew as attributes in the DynamoDB table. These attributes can have different data types and structures, and they do not need to be consistent across items.

Reference:

[Amazon DynamoDB]

[Working with Queries - Amazon DynamoDB]

[Working with Global Secondary Indexes - Amazon DynamoDB]

QUESTION 21

A developer maintains an Amazon API Gateway REST API. Customers use the API through a frontend UI and Amazon Cognito authentication.

The developer has a new version of the API that contains new endpoints and backward-incompatible

interface changes. The developer needs to provide beta access to other developers on the team without affecting customers.

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

- A. Define a development stage on the API Gateway API. Instruct the other developers to point the endpoints to the development stage.
- B. Define a new API Gateway API that points to the new API application code. Instruct the other developers to point the endpoints to the new API.
- C. Implement a query parameter in the API application code that determines which code version to call.
- D. Specify new API Gateway endpoints for the API endpoints that the developer wants to add.

Answer: A

Explanation:

Amazon API Gateway is a service that enables developers to create, publish, maintain, monitor, and secure APIs at any scale. The developer can define a development stage on the API Gateway API and instruct the other developers to point the endpoints to the development stage. This way, the developer can provide beta access to the new version of the API without affecting customers who use the production stage. This solution will meet the requirements with the least operational overhead.

Reference:

[What Is Amazon API Gateway? - Amazon API Gateway]

[Set up a Stage in API Gateway - Amazon API Gateway]

QUESTION 22

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

- a. 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. Enable encryption of data in transit and at rest. Store frequently accessed data in the cache.
- B. Create an Amazon ElastiCache for Memcached instance. Enable encryption of data in transit and at rest. Store frequently accessed data in the cache.
- C. Create an Amazon RDS for MySQL read replica. Connect to the read replica by using SSL. Configure the read replica to store frequently accessed data.
- D. Create an Amazon DynamoDB table and a DynamoDB Accelerator (DAX) cluster for the table. 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.

Reference:

[What Is Amazon ElastiCache? - Amazon ElastiCache]

[Encryption in Transit - Amazon ElastiCache for Redis]

[Encryption at Rest - Amazon ElastiCache for Redis]

QUESTION 23

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 instances. Deploy a file system on the EBS volume. Use the host operating system to share a folder. Update the application code to read and write configuration files from the shared folder.
- B. Deploy a micro EC2 instance with an instance store volume. Use the host operating system to share a folder. Update the application code to read and write configuration files from the shared folder.
- C. Create an Amazon S3 bucket to host the repository. Migrate the existing .xml files to the S3 bucket. Update the application code to use the AWS SDK to read and write configuration files from Amazon S3.
- D. Create an Amazon S3 bucket to host the repository. Migrate the existing .xml files to the S3 bucket. Mount the S3 bucket to the EC2 instances as a local volume. 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.

Reference:

QUESTION 24

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. Connect the repository branches that correspond to each of the desired environments. Start deployments by merging code changes to a desired branch.
- B. Host each website in AWS Elastic Beanstalk with multiple environments. Use the EB CLI to link each repository branch. Integrate AWS CodePipeline to automate deployments from version control code merges.
- C. Host each website in different Amazon S3 buckets for each environment. Configure AWS CodePipeline to pull source code from version control. Add an AWS CodeBuild stage to copy source code to Amazon S3.
- D. Host each website on its own Amazon EC2 instance. Write a custom deployment script to bundle each website's static assets. 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

QUESTION 25

A company is migrating an on-premises database to Amazon RDS for MySQL. The company has read-heavy workloads. The company wants to refactor the code to achieve optimum read performance for queries.

Which solution will meet this requirement with LEAST current and future effort?

- A. Use a multi-AZ Amazon RDS deployment. Increase the number of connections that the code makes to the database or increase the connection pool size if a connection pool is in use.
- B. Use a multi-AZ Amazon RDS deployment. Modify the code so that queries access the secondary RDS instance.
- C. Deploy Amazon RDS with one or more read replicas. Modify the application code so that queries use the URL for the read replicas.
- D. Use open source replication software to create a copy of the MySQL database on an Amazon EC2 instance. Modify the application code so that queries use the IP address of the EC2 instance.

Answer: C

Explanation:

Amazon RDS for MySQL supports read replicas, which are copies of the primary database instance that can handle read-only queries. Read replicas can improve the read performance of the database by offloading the read workload from the primary instance and distributing it across multiple replicas. To use read replicas, the application code needs to be modified to direct read queries to the URL of the read replicas, while write queries still go to the URL of the primary instance. This solution requires less current and future effort than using a multi-AZ deployment, which does not provide read scaling benefits, or using open source replication software, which requires additional configuration and maintenance. Reference: Working with read replicas

QUESTION 26

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. Store the unique identifier for each request in a database table. Modify the Lambda function to check the table for the identifier before processing the request.
- B. Create an Amazon DynamoDB table. Store the unique identifier for each request in the table. Modify the Lambda function to check the table for the identifier before processing the request.
- C. Create an Amazon DynamoDB table. Store the unique identifier for each request in the table. Modify the Lambda function to return a client error response when the function receives a duplicate request.
- D. Create an Amazon ElastiCache for Memcached instance. Store the unique identifier for each request in the cache. 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

QUESTION 27

A developer wants to expand an application to run in multiple AWS Regions. The developer wants to copy Amazon Machine Images (AMIs) with the latest changes and create a new application stack in the destination Region. According to company requirements, all AMIs must be encrypted in all Regions. However, not all the AMIs that the company uses are encrypted. How can the developer expand the application to run in the destination Region while meeting the encryption requirement?

- A. Create new AMIs, and specify encryption parameters. Copy the encrypted AMIs to the destination Region. Delete the unencrypted AMIs.
- B. Use AWS Key Management Service (AWS KMS) to enable encryption on the unencrypted AMIs. Copy the encrypted AMIs to the destination Region.
- C. Use AWS Certificate Manager (ACM) to enable encryption on the unencrypted AMIs. Copy the encrypted AMIs to the destination Region.
- D. Copy the unencrypted AMIs to the destination Region. Enable encryption by default in the destination Region.

Answer: A

Explanation:

Amazon Machine Images (AMIs) are encrypted snapshots of EC2 instances that can be used to launch new instances. The developer can create new AMIs from the existing instances and specify encryption parameters. The developer can copy the encrypted AMIs to the destination Region and use them to create a new application stack. The developer can delete the unencrypted AMIs after the encryption process is complete. This solution will meet the encryption requirement and allow the developer to expand the application to run in the destination Region.

Reference:

[Amazon Machine Images (AMI) - Amazon Elastic Compute Cloud]

[Encrypting an Amazon EBS Snapshot - Amazon Elastic Compute Cloud]

[Copying an AMI - Amazon Elastic Compute Cloud]

QUESTION 28

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. Assign an access point to each web application bucket.
- B. Create a bucket policy that allows access to the central S3 bucket. Attach the bucket policy to the central S3 bucket.
- C. Create a cross-origin resource sharing (CORS) configuration that allows access to the central S3 bucket. Add the CORS configuration to the central S3 bucket.
- D. Create a Content-MD5 header that provides a message integrity check for the central S3 bucket. 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>

QUESTION 29

An application is processing clickstream data using Amazon Kinesis. The clickstream data feed into Kinesis experiences periodic spikes. The PutRecords API call occasionally fails and the logs show that the failed call returns the response shown below:

```

{
  "FailedRecordCount": 1,
  "Records": [
    {
      "SequenceNumber": "21269319989900637946712965403778482371",
      "ShardId": "shardId-000000000001"
    },
    {
      "ErrorCode": "ProvisionedThroughputExceededException",
      "ErrorMessage": "Rate exceeded for shard shardId-000000000001 in
stream exampleStreamName under account 123456789."
    },
    {
      "SequenceNumber": "21269319989999637946712965403778482985",
      "ShardId": "shardId-000000000002"
    }
  ]
}

```

Which techniques will help mitigate this exception? (Choose two.)

- A. Implement retries with exponential backoff.
- B. Use a PutRecord API instead of PutRecords.
- C. Reduce the frequency and/or size of the requests.
- D. Use Amazon SNS instead of Kinesis.
- E. Reduce the number of KCL consumers.

Answer: AC

Explanation:

The response from the API call indicates that the ProvisionedThroughputExceededException exception has occurred. This exception means that the rate of incoming requests exceeds the throughput limit for one or more shards in a stream. To mitigate this exception, the developer can use one or more of the following techniques:

Implement retries with exponential backoff. This will introduce randomness in the retry intervals and avoid overwhelming the shards with retries.

Reduce the frequency and/or size of the requests. This will reduce the load on the shards and avoid throttling errors.

Increase the number of shards in the stream. This will increase the throughput capacity of the stream and accommodate higher request rates.

Use a PutRecord API instead of PutRecords. This will reduce the number of records per request and avoid exceeding the payload limit.

Reference:

[ProvisionedThroughputExceededException - Amazon Kinesis Data Streams Service API Reference]

[Best Practices for Handling Kinesis Data Streams Errors]

QUESTION 30

A company has an application that uses Amazon Cognito user pools as an identity provider. The company must secure access to user records. The company has set up multi-factor authentication (MFA). The company also wants to send a login activity notification by email every time a user logs in.

What is the MOST operationally efficient solution that meets this requirement?

A. Create an AWS Lambda function that uses Amazon Simple Email Service (Amazon SES) to send the email notification. Add an Amazon API Gateway API to invoke the function. Call the API from the client side when login confirmation is received.

B. Create an AWS Lambda function that uses Amazon Simple Email Service (Amazon SES) to send the email notification. Add an Amazon Cognito post authentication Lambda trigger for the function.

C. Create an AWS Lambda function that uses Amazon Simple Email Service (Amazon SES) to send the email notification. Create an Amazon CloudWatch Logs log subscription filter to invoke the function based on the login status.

D. Configure Amazon Cognito to stream all logs to Amazon Kinesis Data Firehose. Create an AWS Lambda function to process the streamed logs and to send the email notification based on the login status of each user.

Answer: B

Explanation:

Amazon Cognito user pools support Lambda triggers, which are custom functions that can be executed at various stages of the user pool workflow. A post authentication Lambda trigger can be used to perform custom actions after a user is authenticated, such as sending an email notification. Amazon SES is a cloud-based email sending service that can be used to send transactional or marketing emails. A Lambda function can use the Amazon SES API to send an email to the users email address after the user logs in successfully. Reference: Post authentication Lambda trigger

QUESTION 31

A developer has an application that stores data in an Amazon S3 bucket. The application uses an HTTP API to store and retrieve objects. When the PutObject API operation adds objects to the S3 bucket the developer must encrypt these objects at rest by using server-side encryption with

Amazon S3 managed keys (SSE-S3).
Which solution will meet this requirement?

- A. Create an AWS Key Management Service (AWS KMS) key. Assign the KMS key to the S3 bucket.
- B. Set the x-amz-server-side-encryption header when invoking the PutObject API operation.
- C. Provide the encryption key in the HTTP header of every request.
- D. Apply TLS to encrypt the traffic to the S3 bucket.

Answer: B

Explanation:

Amazon S3 supports server-side encryption, which encrypts data at rest on the server that stores the data. One of the encryption options is SSE-S3, which uses keys managed by S3. To use SSE-S3, the x-amz-server-side-encryption header must be set to AES256 when invoking the PutObject API operation. This instructs S3 to encrypt the object data with SSE-S3 before saving it on disks in its data centers and decrypt it when it is downloaded. Reference: Protecting data using server-side encryption with Amazon S3-managed encryption keys (SSE-S3)

QUESTION 32

A developer needs to perform geographic load testing of an API. The developer must deploy resources to multiple AWS Regions to support the load testing of the API.
How can the developer meet these requirements without additional application code?

- A. Create and deploy an AWS Lambda function in each desired Region. Configure the Lambda function to create a stack from an AWS CloudFormation template in that Region when the function is invoked.
- B. Create an AWS CloudFormation template that defines the load test resources. Use the AWS CLI create-stack-set command to create a stack set in the desired Regions.
- C. Create an AWS Systems Manager document that defines the resources. Use the document to create the resources in the desired Regions.
- D. Create an AWS CloudFormation template that defines the load test resources. Use the AWS CLI deploy command to create a stack from the template in each Region.

Answer: B

Explanation:

AWS CloudFormation is a service that allows developers to model and provision AWS resources using templates. A CloudFormation template can define the load test resources, such as EC2 instances, load balancers, and Auto Scaling groups. A CloudFormation stack set is a collection of stacks that can be created and managed from a single template in multiple Regions and accounts. The AWS CLI create-stack-set command can be used to create a stack set from a template and specify the Regions where the stacks should be created. Reference: Working with AWS CloudFormation stack sets

QUESTION 33

A developer is creating an application that includes an Amazon API Gateway REST API in the us-east-2 Region. The developer wants to use Amazon CloudFront and a custom domain name for the API. The developer has acquired an SSL/TLS certificate for the domain from a third-party provider. How should the developer configure the custom domain for the application?

- A. Import the SSL/TLS certificate into AWS Certificate Manager (ACM) in the same Region as the API. Create a DNS A record for the custom domain.
- B. Import the SSL/TLS certificate into CloudFront. Create a DNS CNAME record for the custom domain.
- C. Import the SSL/TLS certificate into AWS Certificate Manager (ACM) in the same Region as the API. Create a DNS CNAME record for the custom domain.
- D. Import the SSL/TLS certificate into AWS Certificate Manager (ACM) in the us-east-1 Region. Create a DNS CNAME record for the custom domain.

Answer: D

Explanation:

Amazon API Gateway is a service that enables developers to create, publish, maintain, monitor, and secure APIs at any scale. Amazon CloudFront is a content delivery network (CDN) service that can improve the performance and security of web applications. The developer can use CloudFront and a custom domain name for the API Gateway REST API. To do so, the developer needs to import the SSL/TLS certificate into AWS Certificate Manager (ACM) in the us-east-1 Region. This is because CloudFront requires certificates from ACM to be in this Region. The developer also needs to create a DNS CNAME record for the custom domain that points to the CloudFront distribution.

Reference:

[What Is Amazon API Gateway? - Amazon API Gateway]

[What Is Amazon CloudFront? - Amazon CloudFront]

[Custom Domain Names for APIs - Amazon API Gateway]

QUESTION 34

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.

Reference:

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

[AWS SAM Template Specification - AWS Serverless Application Model]

QUESTION 35

A developer wants to insert a record into an Amazon DynamoDB table as soon as a new file is added to an Amazon S3 bucket.

Which set of steps would be necessary to achieve this?

- A. Create an event with Amazon EventBridge that will monitor the S3 bucket and then insert the records into DynamoDB.
- B. Configure an S3 event to invoke an AWS Lambda function that inserts records into DynamoDB.
- C. Create an AWS Lambda function that will poll the S3 bucket and then insert the records into DynamoDB.
- D. Create a cron job that will run at a scheduled time and insert the records into DynamoDB.

Answer: B

Explanation:

Amazon S3 is a service that provides highly scalable, durable, and secure object storage. Amazon DynamoDB is a fully managed NoSQL database service that provides fast and consistent performance with seamless scalability. AWS Lambda is a service that lets developers run code without provisioning or managing servers. The developer can configure an S3 event to invoke a Lambda function that inserts records into DynamoDB whenever a new file is added to the S3 bucket. This solution will meet the requirement of inserting a record into DynamoDB as soon as a new file is added to S3.

Reference:

[Amazon Simple Storage Service (S3)]

[Amazon DynamoDB]

[What Is AWS Lambda? - AWS Lambda]

[Using AWS Lambda with Amazon S3 - AWS Lambda]

QUESTION 36

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.

a. 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.
- C. Modify the database to use a Multi-AZ deployment.
- D. 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: A, B

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.

Reference:

[What Is AWS CloudFormation? - AWS CloudFormation]

[DeletionPolicy Attribute - AWS CloudFormation]

[Protecting Resources During Stack Updates - AWS CloudFormation]

QUESTION 37

A company has an Amazon S3 bucket that contains sensitive data.

a. The data must be encrypted in transit and at rest. The company encrypts the data in the S3 bucket by using an AWS Key Management Service (AWS KMS) key. A developer needs to grant several other AWS accounts the permission to use the S3 GetObject operation to retrieve the data from the S3 bucket.

How can the developer enforce that all requests to retrieve the data provide encryption in transit?

- A. Define a resource-based policy on the S3 bucket to deny access when a request meets the condition `aws:SecureTransport : false`.
- B. Define a resource-based policy on the S3 bucket to allow access when a request meets the condition `aws:SecureTransport : false`.
- C. Define a role-based policy on the other accounts' roles to deny access when a request meets the condition of `aws:SecureTransport : false`.
- D. Define a resource-based policy on the KMS key to deny access when a request meets the condition of `aws:SecureTransport : false`.

Answer: A

Explanation:

Amazon S3 supports resource-based policies, which are JSON documents that specify the permissions for accessing S3 resources. A resource-based policy can be used to enforce encryption in transit by denying access to requests that do not use HTTPS. The condition key `aws:SecureTransport` can be used to check if the request was sent using SSL. If the value of this key is false, the request is denied; otherwise, the request is allowed. Reference: How do I use an S3 bucket policy to require requests to use Secure Socket Layer (SSL)?

QUESTION 38

An application that is hosted on an Amazon EC2 instance needs access to files that are stored in an Amazon S3 bucket. The application lists the objects that are stored in the S3 bucket and displays a table to the user. During testing, a developer discovers that the application does not show any objects in the list.

What is the MOST secure way to resolve this issue?

- A. Update the IAM instance profile that is attached to the EC2 instance to include the `S3:*` permission for the S3 bucket.
- B. Update the IAM instance profile that is attached to the EC2 instance to include the `S3:ListBucket` permission for the S3 bucket.
- C. Update the developer's user permissions to include the `S3:ListBucket` permission for the S3 bucket.
- D. Update the S3 bucket policy by including the `S3:ListBucket` permission and by setting the Principal element to specify the account number of the EC2 instance.

Answer: B

Explanation:

IAM instance profiles are containers for IAM roles that can be associated with EC2 instances. An IAM role is a set of permissions that grant access to AWS resources. An IAM role can be used to allow an EC2 instance to access an S3 bucket by including the appropriate permissions in the roles policy. The `S3:ListBucket` permission allows listing the objects in an S3 bucket. By updating the IAM instance profile with this permission, the application on the EC2 instance can retrieve the objects from the S3 bucket and display them to the user. Reference: Using an IAM role to grant permissions to applications running on Amazon EC2 instances

QUESTION 39

A company is planning to securely manage one-time fixed license keys in AWS. The company's development team needs to access the license keys in automaton scripts that run in Amazon EC2 instances and in AWS CloudFormation stacks.

Which solution will meet these requirements MOST cost-effectively?

- A. Amazon S3 with encrypted files prefixed with æconfig
- B. AWS Secrets Manager secrets with a tag that is named SecretString
- C. AWS Systems Manager Parameter Store SecureString parameters
- D. CloudFormation NoEcho parameters

Answer: C

Explanation:

AWS Systems Manager Parameter Store is a service that provides secure, hierarchical storage for configuration data and secrets. Parameter Store supports SecureString parameters, which are encrypted using AWS Key Management Service (AWS KMS) keys. SecureString parameters can be used to store license keys in AWS and retrieve them securely from automation scripts that run in EC2 instances or CloudFormation stacks. Parameter Store is a cost-effective solution because it does not charge for storing parameters or API calls. Reference: Working with Systems Manager parameters

QUESTION 40

A company has deployed infrastructure on AWS. A development team wants to create an AWS Lambda function that will retrieve data from an Amazon Aurora database. The Amazon Aurora database is in a private subnet in company's VPC. The VPC is named VPC1. The data is relational in nature. The Lambda function needs to access the data securely. Which solution will meet these requirements?

- A. Create the Lambda function. Configure VPC1 access for the function. Attach a security group named SG1 to both the Lambda function and the database. Configure the security group inbound and outbound rules to allow TCP traffic on Port 3306.
- B. Create and launch a Lambda function in a new public subnet that is in a new VPC named VPC2. Create a peering connection between VPC1 and VPC2.
- C. Create the Lambda function. Configure VPC1 access for the function. Assign a security group named SG1 to the Lambda function. Assign a second security group named SG2 to the database. Add an inbound rule to SG1 to allow TCP traffic from Port 3306.
- D. Export the data from the Aurora database to Amazon S3. Create and launch a Lambda function in VPC1. Configure the Lambda function query the data from Amazon S3.

Answer: A

Explanation:

AWS Lambda is a service that lets you run code without provisioning or managing servers. Lambda functions can be configured to access resources in a VPC, such as an Aurora database, by specifying one or more subnets and security groups in the VPC settings of the function. A security group acts as a virtual firewall that controls inbound and outbound traffic for the resources in a VPC. To allow a

Lambda function to communicate with an Aurora database, both resources need to be associated with the same security group, and the security group rules need to allow TCP traffic on Port 3306, which is the default port for MySQL databases. Reference: [Configuring a Lambda function to access resources in a VPC]

QUESTION 41

A developer is building a web application that uses Amazon API Gateway to expose an AWS Lambda function to process requests from clients. During testing, the developer notices that the API Gateway times out even though the Lambda function finishes under the set time limit.

Which of the following API Gateway metrics in Amazon CloudWatch can help the developer troubleshoot the issue? (Choose two.)

- A. CacheHitCount
- B. IntegrationLatency
- C. CacheMissCount
- D. Latency
- E. Count

Answer: B, D

Explanation:

Amazon API Gateway is a service that enables developers to create, publish, maintain, monitor, and secure APIs at any scale. Amazon CloudWatch is a service that monitors AWS resources and applications. API Gateway provides several CloudWatch metrics to help developers troubleshoot issues with their APIs. Two of the metrics that can help the developer troubleshoot the issue of API Gateway timing out are:

IntegrationLatency: This metric measures the time between when API Gateway relays a request to the backend and when it receives a response from the backend. A high value for this metric indicates that the backend is taking too long to respond and may cause API Gateway to time out.

Latency: This metric measures the time between when API Gateway receives a request from a client and when it returns a response to the client. A high value for this metric indicates that either the integration latency is high or API Gateway is taking too long to process the request or response.

Reference:

[What Is Amazon API Gateway? - Amazon API Gateway]

[Amazon API Gateway Metrics and Dimensions - Amazon CloudWatch]

[Troubleshooting API Errors - Amazon API Gateway]

QUESTION 42

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.

Reference:

[What Is AWS CodeCommit? - AWS CodeCommit]

[AWS CodePipeline - AWS CodeCommit]

QUESTION 43

A developer is designing an AWS Lambda function that creates temporary files that are less than 10 MB during invocation. The temporary files will be accessed and modified multiple times during invocation. The developer has no need to save or retrieve these files in the future.

Where should the temporary files be stored?

- A. the /tmp directory
- B. Amazon Elastic File System (Amazon EFS)
- C. Amazon Elastic Block Store (Amazon EBS)
- D. Amazon S3

Answer: A

Explanation:

AWS Lambda is a service that lets developers run code without provisioning or managing servers. Lambda provides a local file system that can be used to store temporary files during invocation. The local file system is mounted under the /tmp directory and has a limit of 512 MB. The temporary files are accessible only by the Lambda function that created them and are deleted after the function execution ends. The developer can store temporary files that are less than 10 MB in the /tmp directory and access and modify them multiple times during invocation.

Reference:

[What Is AWS Lambda? - AWS Lambda]

[AWS Lambda Execution Environment - AWS Lambda]

QUESTION 44

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. Deploy each Lambda function with its own copy of the library.
- B. Create a Lambda layer with the required Python library. Use the Lambda layer in both Lambda functions.
- C. Combine the two Lambda functions into one Lambda function. Deploy the Lambda function as a single .zip file archive.
- D. Download the Python library to an S3 bucket. 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.

Reference:

[What Is AWS Lambda? - AWS Lambda]

[AWS Lambda Layers - AWS Lambda]

QUESTION 45

A developer is writing an AWS Lambda function. The developer wants to log key events that occur while the Lambda function runs. The developer wants to include a unique identifier to associate the events with a specific function invocation. The developer adds the following code to the Lambda function:

```
function handler(event, context) {  
    }  
}
```

Which solution will meet this requirement?

- A. Obtain the request identifier from the AWS request ID field in the context object. Configure the application to write logs to standard output.
- B. Obtain the request identifier from the AWS request ID field in the event object. Configure the application to write logs to a file.
- C. Obtain the request identifier from the AWS request ID field in the event object. Configure the application to write logs to standard output.
- D. Obtain the request identifier from the AWS request ID field in the context object. Configure the application to write logs to a file.

Answer: A

Explanation:

<https://docs.aws.amazon.com/lambda/latest/dg/nodejs-context.html>

<https://docs.aws.amazon.com/lambda/latest/dg/nodejs-logging.html>

There is no explicit information for the runtime, the code is written in Node.js.

AWS Lambda is a service that lets developers run code without provisioning or managing servers.

The developer can use the AWS request ID field in the context object to obtain a unique identifier for each function invocation. The developer can configure the application to write logs to standard output, which will be captured by Amazon CloudWatch Logs. This solution will meet the requirement of logging key events with a unique identifier.

Reference:

[What Is AWS Lambda? - AWS Lambda]

[AWS Lambda Function Handler in Node.js - AWS Lambda]

[Using Amazon CloudWatch - AWS Lambda]

QUESTION 46

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 table. Create a trigger to connect the data stream to the Lambda function.
- B. Create an Amazon EventBridge rule to invoke the Lambda function on a regular schedule. Connect to the DynamoDB table from the Lambda function to detect changes.
- C. Enable DynamoDB Streams on the table. Create a trigger to connect the DynamoDB stream to the

Lambda function.

D. Create an Amazon Kinesis Data Firehose delivery stream, and attach it to the DynamoDB table. 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.

Reference:

[Amazon DynamoDB]

[DynamoDB Streams - Amazon DynamoDB]

[Using AWS Lambda with Amazon DynamoDB - AWS Lambda]

QUESTION 47

An application uses an Amazon EC2 Auto Scaling group. A developer notices that EC2 instances are taking a long time to become available during scale-out events. The UserData script is taking a long time to run.

The developer must implement a solution to decrease the time that elapses before an EC2 instance becomes available. The solution must make the most recent version of the application available at all times and must apply all available security updates. The solution also must minimize the number of images that are created. The images must be validated.

Which combination of steps should the developer take to meet these requirements? (Choose two.)

A. Use EC2 Image Builder to create an Amazon Machine Image (AMI). Install all the patches and agents that are needed to manage and run the application. Update the Auto Scaling group launch configuration to use the AMI.

B. Use EC2 Image Builder to create an Amazon Machine Image (AMI). Install the latest version of the application and all the patches and agents that are needed to manage and run the application. Update the Auto Scaling group launch configuration to use the AMI.

C. Set up AWS CodeDeploy to deploy the most recent version of the application at runtime.

D. Set up AWS CodePipeline to deploy the most recent version of the application at runtime.

E. Remove any commands that perform operating system patching from the UserData script.

Answer: BE

Explanation:

AWS CloudFormation is a service that enables developers to model and provision AWS resources using templates. The developer can use the following steps to avoid accidental database deletion in

the future:

Set up AWS CodeDeploy to deploy the most recent version of the application at runtime. This will ensure that the application code is always up to date and does not depend on the AMI.

Remove any commands that perform operating system patching from the UserData script. This will reduce the time that the UserData script takes to run and speed up the instance launch process.

Reference:

[What Is AWS CloudFormation? - AWS CloudFormation]

[What Is AWS CodeDeploy? - AWS CodeDeploy]

[Running Commands on Your Linux Instance at Launch - Amazon Elastic Compute Cloud]

QUESTION 48

A developer is creating an AWS Lambda function that needs credentials to connect to an Amazon RDS for MySQL database. An Amazon S3 bucket currently stores the credentials. The developer needs to improve the existing solution by implementing credential rotation and secure storage. The developer also needs to provide integration with the Lambda function.

Which solution should the developer use to store and retrieve the credentials with the LEAST management overhead?

A. Store the credentials in AWS Systems Manager Parameter Store. Select the database that the parameter will access. Use the default AWS Key Management Service (AWS KMS) key to encrypt the parameter. Enable automatic rotation for the parameter. Use the parameter from Parameter Store on the Lambda function to connect to the database.

B. Encrypt the credentials with the default AWS Key Management Service (AWS KMS) key. Store the credentials as environment variables for the Lambda function. Create a second Lambda function to generate new credentials and to rotate the credentials by updating the environment variables of the first Lambda function. Invoke the second Lambda function by using an Amazon EventBridge rule that runs on a schedule. Update the database to use the new credentials. On the first Lambda function, retrieve the credentials from the environment variables. Decrypt the credentials by using AWS KMS, Connect to the database.

C. Store the credentials in AWS Secrets Manager. Set the secret type to Credentials for Amazon RDS database. Select the database that the secret will access. Use the default AWS Key Management Service (AWS KMS) key to encrypt the secret. Enable automatic rotation for the secret. Use the secret from Secrets Manager on the Lambda function to connect to the database.

D. Encrypt the credentials by using AWS Key Management Service (AWS KMS). Store the credentials in an Amazon DynamoDB table. Create a second Lambda function to rotate the credentials. Invoke the second Lambda function by using an Amazon EventBridge rule that runs on a schedule. Update the DynamoDB table. Update the database to use the generated credentials. Retrieve the credentials from DynamoDB with the first Lambda function. Connect to the database.

Answer: C

Explanation:

AWS Secrets Manager is a service that helps you protect secrets needed to access your applications, services, and IT resources. Secrets Manager enables you to store, retrieve, and rotate secrets such as database credentials, API keys, and passwords. Secrets Manager supports a secret type for RDS databases, which allows you to select an existing RDS database instance and generate credentials for it. Secrets Manager encrypts the secret using AWS Key Management Service (AWS KMS) keys and enables automatic rotation of the secret at a specified interval. A Lambda function can use the AWS SDK or CLI to retrieve the secret from Secrets Manager and use it to connect to the database.

Reference: Rotating your AWS Secrets Manager secrets

QUESTION 49

A developer has written the following IAM policy to provide access to an Amazon S3 bucket:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:PutObject"
      ],
      "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"
    },
    {
      "Effect": "Deny",
      "Action": "s3:*",
      "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET/secrets*"
    }
  ]
}
```

Which access does the policy allow regarding the s3:GetObject and s3:PutObject actions?

- A. Access on all buckets except the æDOC-EXAMPLE-BUCKET bucket
- B. Access on all buckets that start with æDOC-EXAMPLE-BUCKET except the æDOC-EXAMPLEBUCKET/secrets bucket
- C. Access on all objects in the æDOC-EXAMPLE-BUCKET bucket along with access to all S3 actions for objects in the æDOC-EXAMPLE-BUCKET bucket that start with æsecrets
- D. Access on all objects in the æDOC-EXAMPLE-BUCKET bucket except on objects that start with æsecrets

Answer: D

Explanation:

The IAM policy shown in the image is a resource-based policy that grants or denies access to an S3 bucket based on certain conditions. The first statement allows access to any S3 action on any object in the `DOC-EXAMPLE-BUCKET` bucket when the request is made over HTTPS (the value of `aws:SecureTransport` is true). The second statement denies access to the `s3:GetObject` and `s3:PutObject` actions on any object in the `DOC-EXAMPLE-BUCKET/secrets` prefix when the request is made over HTTP (the value of `aws:SecureTransport` is false). Therefore, the policy allows access on all objects in the `DOC-EXAMPLE-BUCKET` bucket except on objects that start with `secrets`.

Reference: Using IAM policies for Amazon S3

QUESTION 50

A developer is creating a mobile app that calls a backend service by using an Amazon API Gateway REST API. For integration testing during the development phase, the developer wants to simulate different backend responses without invoking the backend service.

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

- A. Create an AWS Lambda function. Use API Gateway proxy integration to return constant HTTP responses.
- B. Create an Amazon EC2 instance that serves the backend REST API by using an AWS CloudFormation template.
- C. Customize the API Gateway stage to select a response type based on the request.
- D. Use a request mapping template to select the mock integration response.

Answer: D

Explanation:

Amazon API Gateway supports mock integration responses, which are predefined responses that can be returned without sending requests to a backend service. Mock integration responses can be used for testing or prototyping purposes, or for simulating different backend responses based on certain conditions. A request mapping template can be used to select a mock integration response based on an expression that evaluates some aspects of the request, such as headers, query strings, or body content. This solution does not require any additional resources or code changes and has the least operational overhead. Reference: Set up mock integrations for an API Gateway REST API <https://docs.aws.amazon.com/apigateway/latest/developerguide/how-to-mock-integration.html>

QUESTION 51

A developer has a legacy application that is hosted on-premises. Other applications hosted on AWS depend on the on-premises application for proper functioning. In case of any application errors, the developer wants to be able to use Amazon CloudWatch to monitor and troubleshoot all applications

from one place.

How can the developer accomplish this?

- A. Install an AWS SDK on the on-premises server to automatically send logs to CloudWatch.
- B. Download the CloudWatch agent to the on-premises server. Configure the agent to use IAM user credentials with permissions for CloudWatch.
- C. Upload log files from the on-premises server to Amazon S3 and have CloudWatch read the files.
- D. Upload log files from the on-premises server to an Amazon EC2 instance and have the instance forward the logs to CloudWatch.

Answer: B

Explanation:

Amazon CloudWatch is a service that monitors AWS resources and applications. The developer can use CloudWatch to monitor and troubleshoot all applications from one place. To do so, the developer needs to download the CloudWatch agent to the on-premises server and configure the agent to use IAM user credentials with permissions for CloudWatch. The agent will collect logs and metrics from the on-premises server and send them to CloudWatch.

Reference:

[What Is Amazon CloudWatch? - Amazon CloudWatch]

[Installing and Configuring the CloudWatch Agent - Amazon CloudWatch]

QUESTION 52

An Amazon Kinesis Data Firehose delivery stream is receiving customer data that contains personally identifiable information. A developer needs to remove pattern-based customer identifiers from the data and store the modified data in an Amazon S3 bucket.

What should the developer do to meet these requirements?

- A. Implement Kinesis Data Firehose data transformation as an AWS Lambda function. Configure the function to remove the customer identifiers. Set an Amazon S3 bucket as the destination of the delivery stream.
- B. Launch an Amazon EC2 instance. Set the EC2 instance as the destination of the delivery stream. Run an application on the EC2 instance to remove the customer identifiers. Store the transformed data in an Amazon S3 bucket.
- C. Create an Amazon OpenSearch Service instance. Set the OpenSearch Service instance as the destination of the delivery stream. Use search and replace to remove the customer identifiers. Export the data to an Amazon S3 bucket.
- D. Create an AWS Step Functions workflow to remove the customer identifiers. As the last step in the workflow, store the transformed data in an Amazon S3 bucket. Set the workflow as the destination of the delivery stream.

Answer: A

Explanation:

Amazon Kinesis Data Firehose is a service that delivers real-time streaming data to destinations such as Amazon S3, Amazon Redshift, Amazon OpenSearch Service, and Amazon Kinesis Data Analytics. The developer can implement Kinesis Data Firehose data transformation as an AWS Lambda function. The function can remove pattern-based customer identifiers from the data and return the modified data to Kinesis Data Firehose. The developer can set an Amazon S3 bucket as the destination of the delivery stream.

Reference:

[What Is Amazon Kinesis Data Firehose? - Amazon Kinesis Data Firehose]

[Data Transformation - Amazon Kinesis Data Firehose]

QUESTION 53

A developer is using an AWS Lambda function to generate avatars for profile pictures that are uploaded to an Amazon S3 bucket. The Lambda function is automatically invoked for profile pictures that are saved under the /original/ S3 prefix. The developer notices that some pictures cause the Lambda function to time out. The developer wants to implement a fallback mechanism by using another Lambda function that resizes the profile picture.

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

- A. Set the image resize Lambda function as a destination of the avatar generator Lambda function for the events that fail processing.
- B. Create an Amazon Simple Queue Service (Amazon SQS) queue. Set the SQS queue as a destination with an on failure condition for the avatar generator Lambda function. Configure the image resize Lambda function to poll from the SQS queue.
- C. Create an AWS Step Functions state machine that invokes the avatar generator Lambda function and uses the image resize Lambda function as a fallback. Create an Amazon EventBridge rule that matches events from the S3 bucket to invoke the state machine.
- D. Create an Amazon Simple Notification Service (Amazon SNS) topic. Set the SNS topic as a destination with an on failure condition for the avatar generator Lambda function. Subscribe the image resize Lambda function to the SNS topic.

Answer: A

Explanation:

The solution that will meet the requirements with the least development effort is to set the image resize Lambda function as a destination of the avatar generator Lambda function for the events that fail processing. This way, the fallback mechanism is automatically triggered by the Lambda service without requiring any additional components or configuration. The other options involve creating and managing additional resources such as queues, topics, state machines, or rules, which would increase the complexity and cost of the solution.

Reference: Using AWS Lambda destinations

QUESTION 54

A developer needs to migrate an online retail application to AWS to handle an anticipated increase in traffic. The application currently runs on two servers: one server for the web application and another server for the database. The web server renders webpages and manages session state in memory. The database server hosts a MySQL database that contains order details. When traffic to the application is heavy, the memory usage for the web server approaches 100% and the application slows down considerably.

The developer has found that most of the memory increase and performance decrease is related to the load of managing additional user sessions. For the web server migration, the developer will use Amazon EC2 instances with an Auto Scaling group behind an Application Load Balancer. Which additional set of changes should the developer make to the application to improve the application's performance?

- A. Use an EC2 instance to host the MySQL database. Store the session data and the application data in the MySQL database.
- B. Use Amazon ElastiCache for Memcached to store and manage the session data. Use an Amazon RDS for MySQL DB instance to store the application data.
- C. Use Amazon ElastiCache for Memcached to store and manage the session data and the application data.
- D. Use the EC2 instance store to manage the session data. Use an Amazon RDS for MySQL DB instance to store the application data.

Answer: B

Explanation:

Using Amazon ElastiCache for Memcached to store and manage the session data will reduce the memory load and improve the performance of the web server. Using Amazon RDS for MySQL DB instance to store the application data will provide a scalable, reliable, and managed database service. Option A is not optimal because it does not address the memory issue of the web server. Option C is not optimal because it does not provide a persistent storage for the application data. Option D is not optimal because it does not provide a high availability and durability for the session data.

Reference: Amazon ElastiCache, Amazon RDS

QUESTION 55

An application uses Lambda functions to extract metadata from files uploaded to an S3 bucket; the metadata is stored in Amazon DynamoDB. The application starts behaving unexpectedly, and the developer wants to examine the logs of the Lambda function code for errors.

Based on this system configuration, where would the developer find the logs?

- A. Amazon S3
- B. AWS CloudTrail

- C. Amazon CloudWatch
- D. Amazon DynamoDB

Answer: C

Explanation:

Amazon CloudWatch is the service that collects and stores logs from AWS Lambda functions. The developer can use CloudWatch Logs Insights to query and analyze the logs for errors and metrics. Option A is not correct because Amazon S3 is a storage service that does not store Lambda function logs. Option B is not correct because AWS CloudTrail is a service that records API calls and events for AWS services, not Lambda function logs. Option D is not correct because Amazon DynamoDB is a database service that does not store Lambda function logs.

Reference: AWS Lambda Monitoring, [CloudWatch Logs Insights]

QUESTION 56

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.)

- A. Increase the number of shards of the Kinesis data stream.
- B. Decrease the timeout of the Lambda function.
- C. Increase the memory that is allocated to the Lambda function.
- D. Decrease the number of shards of the Kinesis data stream.
- E. Increase the timeout of the Lambda function.

Answer: A, C

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.

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

QUESTION 57

A company needs to harden its container images before the images are in a running state. The company's application uses Amazon Elastic Container Registry (Amazon ECR) as an image registry. Amazon Elastic Kubernetes Service (Amazon EKS) for compute, and an AWS CodePipeline pipeline that orchestrates a continuous integration and continuous delivery (CI/CD) workflow. Dynamic application security testing occurs in the final stage of the pipeline after a new image is deployed to a development namespace in the EKS cluster. A developer needs to place an analysis stage before this deployment to analyze the container image earlier in the CI/CD pipeline. Which solution will meet these requirements with the MOST operational efficiency?

- A. Build the container image and run the docker scan command locally. Mitigate any findings before pushing changes to the source code repository. Write a pre-commit hook that enforces the use of this workflow before commit.
- B. Create a new CodePipeline stage that occurs after the container image is built. Configure ECR basic image scanning to scan on image push. Use an AWS Lambda function as the action provider. Configure the Lambda function to check the scan results and to fail the pipeline if there are findings.
- C. Create a new CodePipeline stage that occurs after source code has been retrieved from its repository. Run a security scanner on the latest revision of the source code. Fail the pipeline if there are findings.
- D. Add an action to the deployment stage of the pipeline so that the action occurs before the deployment to the EKS cluster. Configure ECR basic image scanning to scan on image push. Use an AWS Lambda function as the action provider. Configure the Lambda function to check the scan results and to fail the pipeline if there are findings.

Answer: B

Explanation:

The solution that will meet the requirements with the most operational efficiency is to create a new CodePipeline stage that occurs after the container image is built. Configure ECR basic image scanning to scan on image push. Use an AWS Lambda function as the action provider. Configure the Lambda function to check the scan results and to fail the pipeline if there are findings. This way, the container image is analyzed earlier in the CI/CD pipeline and any vulnerabilities are detected and reported before deploying to the EKS cluster. The other options either delay the analysis until after deployment, which increases the risk of exposing insecure images, or perform analysis on the source code instead of the container image, which may not capture all the dependencies and configurations that affect the security posture of the image.

Reference: Amazon ECR image scanning

QUESTION 58

A developer is testing a new file storage application that uses an Amazon CloudFront distribution to serve content from an Amazon S3 bucket. The distribution accesses the S3 bucket by using an origin access identity (OAI). The S3 bucket's permissions explicitly deny access to all other users.

The application prompts users to authenticate on a login page and then uses signed cookies to allow users to access their personal storage directories. The developer has configured the distribution to use its default cache behavior with restricted viewer access and has set the origin to point to the S3 bucket. However, when the developer tries to navigate to the login page, the developer receives a 403 Forbidden error.

The developer needs to implement a solution to allow unauthenticated access to the login page. The solution also must keep all private content secure.

Which solution will meet these requirements?

- A. Add a second cache behavior to the distribution with the same origin as the default cache behavior. Set the path pattern for the second cache behavior to the path of the login page, and make viewer access unrestricted. Keep the default cache behavior's settings unchanged.
- B. Add a second cache behavior to the distribution with the same origin as the default cache behavior. Set the path pattern for the second cache behavior to *, and make viewer access restricted. Change the default cache behavior's path pattern to the path of the login page, and make viewer access unrestricted.
- C. Add a second origin as a failover origin to the default cache behavior. Point the failover origin to the S3 bucket. Set the path pattern for the primary origin to *, and make viewer access restricted. Set the path pattern for the failover origin to the path of the login page, and make viewer access unrestricted.
- D. Add a bucket policy to the S3 bucket to allow read access. Set the resource on the policy to the Amazon Resource Name (ARN) of the login page object in the S3 bucket. Add a CloudFront function to the default cache behavior to redirect unauthorized requests to the login page's S3 URL.

Answer: A

Explanation:

The solution that will meet the requirements is to add a second cache behavior to the distribution with the same origin as the default cache behavior. Set the path pattern for the second cache behavior to the path of the login page, and make viewer access unrestricted. Keep the default cache behaviors settings unchanged. This way, the login page can be accessed without authentication, while all other content remains secure and requires signed cookies. The other options either do not allow unauthenticated access to the login page, or expose private content to unauthorized users.

Reference: Restricting Access to Amazon S3 Content by Using an Origin Access Identity

QUESTION 59

A developer is using AWS Amplify Hosting to build and deploy an application. The developer is receiving an increased number of bug reports from users. The developer wants to add end-to-end testing to the application to eliminate as many bugs as possible before the bugs reach production. Which solution should the developer implement to meet these requirements?

- A. Run the amplify add test command in the Amplify CLI.

- B. Create unit tests in the application. Deploy the unit tests by using the amplify push command in the Amplify CLI.
- C. Add a test phase to the amplify.yml build settings for the application.
- D. Add a test phase to the aws-exports.js file for the application.

Answer: C

Explanation:

The solution that will meet the requirements is to add a test phase to the amplify.yml build settings for the application. This way, the developer can run end-to-end tests on every code commit and catch any bugs before deploying to production. The other options either do not support end-to-end testing, or do not run tests automatically.

Reference: End-to-end testing

QUESTION 60

An ecommerce company is using an AWS Lambda function behind Amazon API Gateway as its application tier. To process orders during checkout, the application calls a POST API from the frontend. The POST API invokes the Lambda function asynchronously. In rare situations, the application has not processed orders. The Lambda application logs show no errors or failures. What should a developer do to solve this problem?

- A. Inspect the frontend logs for API failures. Call the POST API manually by using the requests from the log file.
- B. Create and inspect the Lambda dead-letter queue. Troubleshoot the failed functions. Reprocess the events.
- C. Inspect the Lambda logs in Amazon CloudWatch for possible errors. Fix the errors.
- D. Make sure that caching is disabled for the POST API in API Gateway.

Answer: B

Explanation:

The solution that will solve this problem is to create and inspect the Lambda dead-letter queue. Troubleshoot the failed functions. Reprocess the events. This way, the developer can identify and fix any issues that caused the Lambda function to fail when invoked asynchronously by API Gateway. The developer can also reprocess any orders that were not processed due to failures. The other options either do not address the root cause of the problem, or do not help recover from failures.

Reference: Asynchronous invocation

QUESTION 61

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. Generate a URL that retrieves the reports from DynamoDB. Provide the URL to customers through the web application.
- B. Generate the reports and then store the reports in an Amazon S3 bucket that uses server-side encryption. Attach the reports to an Amazon Simple Notification Service (Amazon SNS) message. Subscribe the customer to email notifications from Amazon SNS.
- C. Generate the reports and then store the reports in an Amazon S3 bucket that uses server-side encryption. Generate a presigned URL that contains an expiration date. Provide the URL to customers through the web application. Add S3 Lifecycle configuration rules to the S3 bucket to delete old reports.
- D. Generate the reports and then store the reports in an Amazon RDS database with a date stamp. Generate a URL that retrieves the reports from the RDS database. Provide the URL to customers through the web application. 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.

Reference: Amazon S3 Presigned URLs, Amazon S3 Lifecycle

QUESTION 62

A company has deployed an application on AWS Elastic Beanstalk. The company has configured the Auto Scaling group that is associated with the Elastic Beanstalk environment to have five Amazon EC2 instances. If the capacity is fewer than four EC2 instances during the deployment, application performance degrades. The company is using the all-at-once deployment policy. What is the MOST cost-effective way to solve the deployment issue?

- A. Change the Auto Scaling group to six desired instances.
- B. Change the deployment policy to traffic splitting. Specify an evaluation time of 1 hour.

- C. Change the deployment policy to rolling with additional batch. Specify a batch size of 1.
- D. Change the deployment policy to rolling. Specify a batch size of 2.

Answer: C

Explanation:

This solution will solve the deployment issue by deploying the new version of the application to one new EC2 instance at a time, while keeping the old version running on the existing instances. This way, there will always be at least four instances serving traffic during the deployment, and no downtime or performance degradation will occur. Option A is not optimal because it will increase the cost of running the Elastic Beanstalk environment without solving the deployment issue. Option B is not optimal because it will split the traffic between two versions of the application, which may cause inconsistency and confusion for the customers. Option D is not optimal because it will deploy the new version of the application to two existing instances at a time, which may reduce the capacity below four instances during the deployment.

Reference: AWS Elastic Beanstalk Deployment Policies

QUESTION 63

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. Call the X-Ray API from AWS Lambda.
- D. 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 XRay 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.

Reference: [AWS X-Ray SDKs]

QUESTION 64

A developer is migrating some features from a legacy monolithic application to use AWS Lambda functions instead. The application currently stores data in an Amazon Aurora DB cluster that runs in private subnets in a VPC. The AWS account has one VPC deployed. The Lambda functions and the DB cluster are deployed in the same AWS Region in the same AWS account.

The developer needs to ensure that the Lambda functions can securely access the DB cluster without crossing the public internet.

Which solution will meet these requirements?

- A. Configure the DB cluster's public access setting to Yes.
- B. Configure an Amazon RDS database proxy for the Lambda functions.
- C. Configure a NAT gateway and a security group for the Lambda functions.
- D. Configure the VPC, subnets, and a security group for the Lambda functions.

Answer: D

Explanation:

This solution will meet the requirements by allowing the Lambda functions to access the DB cluster securely within the same VPC without crossing the public internet. The developer can configure a VPC endpoint for RDS in a private subnet and assign it to the Lambda functions. The developer can also configure a security group for the Lambda functions that allows inbound traffic from the DB cluster on port 3306 (MySQL). Option A is not optimal because it will expose the DB cluster to public access, which may compromise its security and data integrity. Option B is not optimal because it will introduce additional latency and complexity to use an RDS database proxy for accessing the DB cluster from Lambda functions within the same VPC. Option C is not optimal because it will require additional costs and configuration to use a NAT gateway for accessing resources in private subnets from Lambda functions.

Reference: [Configuring a Lambda Function to Access Resources in a VPC]

QUESTION 65

A developer is building a new application on AWS. The application uses an AWS Lambda function that retrieves information from an Amazon DynamoDB table. The developer hard coded the DynamoDB table name into the Lambda function code. The table name might change over time. The developer does not want to modify the Lambda code if the table name changes.

Which solution will meet these requirements MOST efficiently?

- A. Create a Lambda environment variable to store the table name. Use the standard method for the programming language to retrieve the variable.
- B. Store the table name in a file. Store the file in the /tmp folder. Use the SDK for the programming language to retrieve the table name.
- C. Create a file to store the table name. Zip the file and upload the file to the Lambda layer. Use the

SDK for the programming language to retrieve the table name.

D. Create a global variable that is outside the handler in the Lambda function to store the table name.

Answer: A

Explanation:

The solution that will meet the requirements most efficiently is to create a Lambda environment variable to store the table name. Use the standard method for the programming language to retrieve the variable. This way, the developer can avoid hard-coding the table name in the Lambda function code and easily change the table name by updating the environment variable. The other options either involve storing the table name in a file, which is less efficient and secure than using an environment variable, or creating a global variable, which is not recommended as it can cause concurrency issues.

Reference: Using AWS Lambda environment variables

QUESTION 66

A company has installed smart meters in all its customer locations. The smart meters measure power usage at 1-minute intervals and send the usage readings to a remote endpoint for collection. The company needs to create an endpoint that will receive the smart meter readings and store the readings in a database. The company wants to store the location ID and timestamp information. The company wants to give its customers low-latency access to their current usage and historical usage on demand. The company expects demand to increase significantly. The solution must not impact performance or include downtime while writing.

Which solution will meet these requirements MOST cost-effectively?

- A. Store the smart meter readings in an Amazon RDS database. Create an index on the location ID and timestamp columns. Use the columns to filter on the customers' data.
- B. Store the smart meter readings in an Amazon DynamoDB table. Create a composite key by using the location ID and timestamp columns. Use the columns to filter on the customers' data.
- C. Store the smart meter readings in Amazon ElastiCache for Redis. Create a Sorted Set key by using the location ID and timestamp columns. Use the columns to filter on the customers' data.
- D. Store the smart meter readings in Amazon S3. Partition the data by using the location ID and timestamp columns. Use Amazon Athena to filter on the customers' data.

Answer: B

Explanation:

The solution that will meet the requirements most cost-effectively is to store the smart meter readings in an Amazon DynamoDB table. Create a composite key by using the location ID and timestamp columns. Use the columns to filter on the customers' data. This way, the company can leverage the scalability, performance, and low latency of DynamoDB to store and retrieve the smart

meter readings. The company can also use the composite key to query the data by location ID and timestamp efficiently. The other options either involve more expensive or less scalable services, or do not provide low-latency access to the current usage.

Reference: Working with Queries in DynamoDB

QUESTION 67

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 a 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

QUESTION 68

An online food company provides an Amazon API Gateway HTTP API to receive orders for partners. The API is integrated with an AWS Lambda function. The Lambda function stores the orders in an Amazon DynamoDB table.

The company expects to onboard additional partners. Some of the partners require additional Lambda functions to receive orders. The company has created an Amazon S3 bucket. The company needs to store all orders and updates in the S3 bucket for future analysis.

How can the developer ensure that all orders and updates are stored to Amazon S3 with the LEAST development effort?

- A. Create a new Lambda function and a new API Gateway API endpoint. Configure the new Lambda function to write to the S3 bucket. Modify the original Lambda function to post updates to the new API endpoint.

- B. Use Amazon Kinesis Data Streams to create a new data stream. Modify the Lambda function to publish orders to the data stream. Configure the data stream to write to the S3 bucket.
- C. Enable DynamoDB Streams on the DynamoDB table. Create a new Lambda function. Associate the stream's Amazon Resource Name (ARN) with the Lambda function. Configure the Lambda function to write to the S3 bucket as records appear in the table's stream.
- D. Modify the Lambda function to publish to a new Amazon SNS topic. Create a new Lambda function. Associate the stream's Amazon Resource Name (ARN) with the Lambda function. Configure the Lambda function to write to the S3 bucket as updates come through the topic.

Answer: C

Explanation:

This solution will ensure that all orders and updates are stored to Amazon S3 with the least development effort because it uses DynamoDB Streams to capture changes in the DynamoDB table and trigger a Lambda function to write those changes to the S3 bucket. This way, the original Lambda function and API Gateway API endpoint do not need to be modified, and no additional services are required. Option A is not optimal because it will require more development effort to create a new Lambda function and a new API Gateway API endpoint, and to modify the original Lambda function to post updates to the new API endpoint. Option B is not optimal because it will introduce additional costs and complexity to use Amazon Kinesis Data Streams to create a new data stream, and to modify the Lambda function to publish orders to the data stream. Option D is not optimal because it will require more development effort to modify the Lambda function to publish to a new Amazon SNS topic, and to create and subscribe a new Lambda function to the topic.

Reference: Using DynamoDB Streams, Using AWS Lambda with Amazon S3

QUESTION 69

A company has an Amazon S3 bucket containing premier content that it intends to make available to only paid subscribers of its website. The S3 bucket currently has default permissions of all objects being private to prevent inadvertent exposure of the premier content to non-paying website visitors. How can the company limit the ability to download a premier content file in the S3 Bucket to paid subscribers only?

- A. Apply a bucket policy that allows anonymous users to download the content from the S3 bucket.
- B. Generate a pre-signed object URL for the premier content file when a paid subscriber requests a download.
- C. Add a Docket policy that requires multi-factor authentication for request to access the S3 bucket objects.
- D. Enable server-side encryption on the S3 bucket for data protection against the non-paying website visitors.

Answer: B

Explanation:

This solution will limit the ability to download a premier content file in the S3 bucket to paid subscribers only because it uses a pre-signed object URL that grants temporary access to an S3 object for a specified duration. The pre-signed object URL can be generated by the companys website when a paid subscriber requests a download, and can be verified by Amazon S3 using the signature in the URL. Option A is not optimal because it will allow anyone to download the content from the S3 bucket without verifying their subscription status. Option C is not optimal because it will require additional steps and costs to configure multi-factor authentication for accessing the S3 bucket objects, which may not be feasible or user-friendly for paid subscribers. Option D is not optimal because it will not prevent non-paying website visitors from accessing the S3 bucket objects, but only encrypt them at rest.

Reference: Share an Object with Others, [Using Amazon S3 Pre-Signed URLs]

QUESTION 70

A developer is creating an AWS Lambda function that searches for Items from an Amazon DynamoDB table that contains customer contact information. The DynamoDB table items have the customers as the partition and additional properties such as customer_type, name, and job_title.

The Lambda function runs whenever a user types a new character into the customer_type text Input.

The developer wants to search to return partial matches of all the email_address property of a particular customer type. The developer does not want to recreate the DynamoDB table.

What should the developer do to meet these requirements?

- A. Add a global secondary index (GSI) to the DynamoDB table with customer-type input, as the partition key and email_address as the sort key. Perform a query operation on the GSI by using the begins_with key condition expression with the email_address property.
- B. Add a global secondary index (GSI) to the DynamoDB table with email_address as the partition key and customer_type as the sort key. Perform a query operation on the GSI by using the begins_with key condition expression with the email_address property.
- C. Add a local secondary index (LSI) to the DynamoDB table with customer_type as the partition Key and email_address as the sort Key. Perform a query operation on the LSI by using the begins_with Key condition expression with the email-address property.
- D. Add a local secondary index (LSI) to the DynamoDB table with job-title as the partition key and email_address as the sort key. Perform a query operation on the LSI by using the begins_with key condition expression with the email_address property.

Answer: A

Explanation:

The solution that will meet the requirements is to add a global secondary index (GSI) to the DynamoDB table with customer_type as the partition key and email_address as the sort key. Perform a query operation on the GSI by using the begins_with key condition expression with the email_address property. This way, the developer can search for partial matches of the email_address

property of a particular customer type without recreating the DynamoDB table. The other options either involve using a local secondary index (LSI), which requires recreating the table, or using a different partition key, which does not allow filtering by customer_type.

Reference: Using Global Secondary Indexes in DynamoDB

QUESTION 71

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 of 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

QUESTION 72

A developer is building an application that gives users the ability to view bank account from multiple sources in a single dashboard. The developer has automated the process to retrieve API credentials for these sources. The process invokes an AWS Lambda function that is associated with an AWS CloudFormation cotton resource.

The developer wants a solution that will store the API credentials with minimal operational overhead.

When solution will meet these requirements?

- A. Add an AWS Secrets Manager GenerateSecretString resource to the CloudFormation template. Set the value to reference new credentials to the Cloudformation resource.
- B. Use the AWS SDK `ssm PutParameter` operation in the Lambda function from the existing, custom resource to store the credentials as a parameter. Set the parameter value to reference the new

credentials. Set the parameter type to SecureString.

C. Add an AWS Systems Manager Parameter Store resource to the CloudFormation template. Set the CloudFormation resource value to reference the new credentials. Set the resource NoEcho attribute to true.

D. Use the AWS SDK ssm PutParameter operation in the Lambda function from the existing custom resources to store the credentials as a parameter. Set the parameter value to reference the new credentials. Set the parameter NoEcho attribute to true.

Answer: B

Explanation:

The solution that will meet the requirements is to use the AWS SDK ssm PutParameter operation in the Lambda function from the existing custom resource to store the credentials as a parameter. Set the parameter value to reference the new credentials. Set the parameter type to SecureString. This way, the developer can store the API credentials with minimal operational overhead, as AWS Systems Manager Parameter Store provides secure and scalable storage for configuration data. The SecureString parameter type encrypts the parameter value with AWS Key Management Service (AWS KMS). The other options either involve adding additional resources to the CloudFormation template, which increases complexity and cost, or do not encrypt the parameter value, which reduces security.

Reference: Creating Systems Manager parameters

QUESTION 73

A developer is configuring an applications deployment environment in AWS CodePipeline. The application code is stored in a GitHub repository. The developer wants to ensure that the repository package's unit tests run in the new deployment environment. The deployment has already set the pipeline's source provider to GitHub and has specified the repository and branch to use in the deployment.

When combination of steps should the developer take next to meet these requirements with the least the LEAST overhead' (Select TWO).

A. Create an AWS CodeCommit project. Add the repository package's build and test commands to the project's buildspec

B. Create an AWS CodeBuild project. Add the repository package's build and test commands to the project's buildspec

C. Create an AWS CodeDeploy protect. Add the repository package's build and test commands to the project's buildspec

D. Add an action to the source stage. Specify the newly created project as the action provider. Specify the build artifact as the action's input artifact.

E. Add a new stage to the pipeline after the source stage. Add an action to the new stage. Specify the newly created project as the action provider. Specify the source artifact as the action's input artifact.

Answer: B, E

Explanation:

This solution will ensure that the repository packages unit tests run in the new deployment environment with the least overhead because it uses AWS CodeBuild to build and test the code in a fully managed service, and AWS CodePipeline to orchestrate the deployment stages and actions. Option A is not optimal because it will use AWS CodeCommit instead of AWS CodeBuild, which is a source control service, not a build and test service. Option C is not optimal because it will use AWS CodeDeploy instead of AWS CodeBuild, which is a deployment service, not a build and test service. Option D is not optimal because it will add an action to the source stage instead of creating a new stage, which will not follow the best practice of separating different deployment phases. Reference: AWS CodeBuild, AWS CodePipeline

QUESTION 74

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": {"M": "1234"}}'
```

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 DynamoDB cannot be accessed from the AWS CLI and needs to be 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. Reference: AWS CLI for DynamoDB, [IAM Policies for DynamoDB]

QUESTION 75

An organization is using Amazon CloudFront to ensure that its users experience low-latency access to

its web application. The organization has identified a need to encrypt all traffic between users and CloudFront, and all traffic between CloudFront and the web application.

How can these requirements be met? (Select TWO)

- A. Use AWS KMS to encrypt traffic between CloudFront and the web application.
- B. Set the Origin Protocol Policy to "HTTPS Only".
- C. Set the Origins HTTP Port to 443.
- D. Set the Viewer Protocol Policy to "HTTPS Only" or Redirect HTTP to HTTPS
- E. Enable the CloudFront option Restrict Viewer Access.

Answer: B, D

Explanation:

This solution will meet the requirements by ensuring that all traffic between users and CloudFront, and all traffic between CloudFront and the web application, are encrypted using HTTPS protocol. The Origin Protocol Policy determines how CloudFront communicates with the origin server (the web application), and setting it to "HTTPS Only" will force CloudFront to use HTTPS for every request to the origin server. The Viewer Protocol Policy determines how CloudFront responds to HTTP or HTTPS requests from users, and setting it to "HTTPS Only" or "Redirect HTTP to HTTPS" will force CloudFront to use HTTPS for every response to users. Option A is not optimal because it will use AWS KMS to encrypt traffic between CloudFront and the web application, which is not necessary or supported by CloudFront. Option C is not optimal because it will set the origins HTTP port to 443, which is incorrect as port 443 is used for HTTPS protocol, not HTTP protocol. Option E is not optimal because it will enable the CloudFront option Restrict Viewer Access, which is used for controlling access to private content using signed URLs or signed cookies, not for encrypting traffic.

Reference: [Using HTTPS with CloudFront], [Restricting Access to Amazon S3 Content by Using an Origin Access Identity]

QUESTION 76

A company is developing an ecommerce application that uses Amazon API Gateway APIs. The application uses AWS Lambda as a backend. The company needs to test the code in a dedicated, monitored test environment before the company releases the code to the production environment. When solution will meet these requirements?

- A. Use a single stage in API Gateway. Create a Lambda function for each environment. Configure API clients to send a query parameter that indicates the environment and the specific lambda function.
- B. Use multiple stages in API Gateway. Create a single Lambda function for all environments. Add different code blocks for different environments in the Lambda function based on Lambda environment variables.
- C. Use multiple stages in API Gateway. Create a Lambda function for each environment. Configure API Gateway stage variables to route traffic to the Lambda function in different environments.
- D. Use a single stage in API Gateway. Configure a API client to send a query parameter that indicated

the environment. Add different code blocks for different environments in the Lambda function to match the value of the query parameter.

Answer: C

Explanation:

The solution that will meet the requirements is to use multiple stages in API Gateway. Create a Lambda function for each environment. Configure API Gateway stage variables to route traffic to the Lambda function in different environments. This way, the company can test the code in a dedicated, monitored test environment before releasing it to the production environment. The company can also use stage variables to specify the Lambda function version or alias for each stage, and avoid hard-coding the Lambda function name in the API Gateway integration. The other options either involve using a single stage in API Gateway, which does not allow testing in different environments, or adding different code blocks for different environments in the Lambda function, which increases complexity and maintenance.

Reference: Set up stage variables for a REST API in API Gateway

QUESTION 77

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

QUESTION 78

A team of developed is using an AWS CodePipeline pipeline as a continuous integration and

continuous delivery (CI/CD) mechanism for a web application. A developer has written unit tests to programmatically test the functionality of the application code. The unit tests produce a test report that shows the results of each individual check. The developer now wants to run these tests automatically during the CI/CD process.

- A. Write a Git pre-commit hook that runs the test before every commit. Ensure that each developer who is working on the project has the pre-commit hook instated locally. Review the test report and resolve any issues before pushing changes to AWS CodeCommit.
- B. Add a new stage to the pipeline. Use AWS CodeBuild as the provider. Add the new stage after the stage that deploys code revisions to the test environment. Write a buildspec that fails the CodeBuild stage if any test does not pass. Use the test reports feature of Codebuild to integrate the report with the CodoBuild console. View the test results in CodeBuild Resolve any issues.
- C. Add a new stage to the pipeline. Use AWS CodeBuild at the provider. Add the new stage before the stage that deploys code revisions to the test environment. Write a buildspec that fails the CodeBuild stage if any test does not pass. Use the test reports feature of CodeBuild to integrate the report with the CodeBuild console. View the test results in codeBuild Resolve any issues.
- D. Add a new stage to the pipeline. Use Jenkins as the provider. Configure CodePipeline to use Jenkins to run the unit tests. Write a Jenkinsfile that fails the stage if any test does not pass. Use the test report plugin for Jenkins to integrate the repot with the Jenkins dashboard. View the test results in Jenkins. Resolve any issues.

Answer: C

Explanation:

The solution that will meet the requirements is to add a new stage to the pipeline. Use AWS CodeBuild as the provider. Add the new stage before the stage that deploys code revisions to the test environment. Write a buildspec that fails the CodeBuild stage if any test does not pass. Use the test reports feature of CodeBuild to integrate the report with the CodeBuild console. View the test results in CodeBuild. Resolve any issues. This way, the developer can run the unit tests automatically during the CI/CD process and catch any bugs before deploying to the test environment. The developer can also use the test reports feature of CodeBuild to view and analyze the test results in a graphical interface. The other options either involve running the tests manually, running them after deployment, or using a different provider that requires additional configuration and integration.

Reference: Test reports for CodeBuild

QUESTION 79

A company has multiple Amazon VPC endpoints in the same VPC. A developer needs 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 polices 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 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 StringNotEquals condition. 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.

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

QUESTION 80

A company uses a custom root certificate authority certificate chain (Root CA Cert) that is 10 KB in size generate SSL certificates for its on-premises HTTPS endpoints. One of the company's cloud based applications has hundreds of AWS Lambda functions that pull data from these endpoints. A developer updated the trust store of the Lambda execution environment to use the Root CA Cert when the Lambda execution environment is initialized. The developer bundled the Root CA Cert as a text file in the Lambda's deployment bundle.

After 3 months of development the root CA Cert is no longer valid and must be updated. The developer needs a more efficient solution to update the Root CA Cert for all deployed Lambda functions. The solution must not include rebuilding or updating all Lambda functions that use the Root CA Cert. The solution must also work for all development, testing and production environment. Each environment is managed in a separate AWS account.

When combination of steps Would the developer take to meet these environments MOST costeffectively? (Select TWO)

- A. Store the Root CA Cert as a secret in AWS Secrets Manager. Create a resource-based policy. Add IAM users to allow access to the secret
- B. Store the Root CA Cert as a Secure String parameter in aws Systems Manager Parameter Store Create a resource-based policy. Add IAM users to allow access to the policy.
- C. Store the Root CA Cert in an Amazon S3 bucket. Create a resource- based policy to allow access to

the bucket.

D. Refactor the Lambda code to load the Root CA Cert from the Root CA Certs location. Modify the runtime trust store inside the Lambda function handler.

E. Refactor the Lambda code to load the Root CA Cert from the Root CA Cert's location. Modify the runtime trust store outside the Lambda function handler.

Answer: BE

Explanation:

This solution will meet the requirements by storing the Root CA Cert as a Secure String parameter in AWS Systems Manager Parameter Store, which is a secure and scalable service for storing and managing configuration data and secrets. The resource-based policy will allow IAM users in different AWS accounts and environments to access the parameter without requiring cross-account roles or permissions. 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 use AWS Secrets Manager instead of AWS Systems Manager Parameter Store, which will incur additional costs and complexity for storing and managing a non-secret configuration data such as Root CA Cert. Option C is not optimal because it will deactivate the application secrets and monitor the application error logs temporarily, which will cause application downtime and potential data loss. Option D is not optimal because it will modify the runtime trust store inside the Lambda function handler, which will degrade performance and increase latency by repeating unnecessary operations for each invocation of the Lambda function.

Reference: AWS Systems Manager Parameter Store, [Using SSL/TLS to Encrypt a Connection to a DB Instance]

QUESTION 81

A developer maintains applications that store several secrets in AWS Secrets Manager. The applications use secrets that have changed over time. The developer needs to identify required secrets that are still in use. The developer does not want to cause any application downtime. What should the developer do to meet these requirements?

- A. Configure an AWS CloudTrail log file delivery to an Amazon S3 bucket. Create an Amazon CloudWatch alarm for the GetSecretValue. Secrets Manager API operation requests
- B. Create a secrets manager-secret-unused AWS Config managed rule. Create an Amazon EventBridge rule to Initiate notification when the AWS Config managed rule is met.
- C. Deactivate the applications secrets and monitor the applications error logs temporarily.
- D. Configure AWS X-Ray for the applications. Create a sampling rule to match the GetSecretValue Secrets Manager API operation requests.

Answer: B

Explanation:

This solution will meet the requirements by using AWS Config to monitor and evaluate whether Secrets Manager secrets are unused or have been deleted, based on specified time periods. The secrets manager-secret-unused managed rule is a predefined rule that checks whether Secrets Manager secrets have been rotated within a specified number of days or have been deleted within a specified number of days after last accessed date. The Amazon EventBridge rule will trigger a notification when the AWS Config managed rule is met, alerting the developer about unused secrets that can be removed without causing application downtime. Option A is not optimal because it will use AWS CloudTrail log file delivery to an Amazon S3 bucket, which will incur additional costs and complexity for storing and analyzing log files that may not contain relevant information about secret usage. Option C is not optimal because it will deactivate the application secrets and monitor the application error logs temporarily, which will cause application downtime and potential data loss. Option D is not optimal because it will use AWS X-Ray to trace secret usage, which will introduce additional overhead and latency for instrumenting and sampling requests that may not be related to secret usage.

Reference: [AWS Config Managed Rules], [Amazon EventBridge]

QUESTION 82

A developer is writing a serverless application that requires an AWS Lambda function to be invoked every 10 minutes.

What is an automated and serverless way to invoke the function?

- A. Deploy an Amazon EC2 instance based on Linux, and edit its `/etc/crontab` file by adding a command to periodically invoke the lambda function
- B. Configure an environment variable named `PERIOD` for the Lambda function. Set the value to 600.
- C. Create an Amazon EventBridge rule that runs on a regular schedule to invoke the Lambda function.
- D. Create an Amazon Simple Notification Service (Amazon SNS) topic that has a subscription to the Lambda function with a 600-second timer.

Answer: C

Explanation:

The solution that will meet the requirements is to create an Amazon EventBridge rule that runs on a regular schedule to invoke the Lambda function. This way, the developer can use an automated and serverless way to invoke the function every 10 minutes. The developer can also use a cron expression or a rate expression to specify the schedule for the rule. The other options either involve using an Amazon EC2 instance, which is not serverless, or using environment variables or query parameters, which do not trigger the function.

Reference: Schedule AWS Lambda functions using EventBridge

QUESTION 83

Users are reporting errors in an application. The application consists of several micro services that are deployed on Amazon Elastic Container Services (Amazon ECS) with AWS Fargate.

When combination of steps should a developer take to fix the errors? (Select TWO)

- A. Deploy AWS X-Ray as a sidecar container to the micro services. Update the task role policy to allow access to the X-Ray API.
- B. Deploy AWS X-Ray as a daemon set to the Fargate cluster. Update the service role policy to allow access to the X-Ray API.
- C. Instrument the application by using the AWS X-Ray SDK. Update the application to use the Put-Trace API call to communicate with the X-Ray API.
- D. Instrument the application by using the AWS X-Ray SDK. Update the application to communicate with the X-Ray daemon.
- E. Instrument the ECS task to send the stdout and stderr output to Amazon CloudWatch Logs. Update the task role policy to allow the cloudwatch PutLogs action.

Answer: A, E

Explanation:

The combination of steps that the developer should take to fix the errors is to deploy AWS X-Ray as a sidecar container to the microservices and instrument the ECS task to send the stdout and stderr output to Amazon CloudWatch Logs. This way, the developer can use AWS X-Ray to analyze and debug the performance of the microservices and identify any issues or bottlenecks. The developer can also use CloudWatch Logs to monitor and troubleshoot the logs from the ECS task and detect any errors or exceptions. The other options either involve using AWS X-Ray as a daemon set, which is not supported by Fargate, or using the PutTraceSegments API call, which is not necessary when using a sidecar container.

Reference: Using AWS X-Ray with Amazon ECS

QUESTION 84

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 Lambda 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. Set the NoEcho attribute to true.
- B. Use a CloudFormation parameter to pass the master user credentials at deployment to the OpenSearch Service domain's MasterUserOptions and to create a parameter. In AWS Systems Manager Parameter Store. Set the NoEcho attribute to true. Create an IAM role that has the ssm

GetParameter permission. Assign me role to the Lambda function. Store me parameter name as the Lambda function's environment variable. Resolve the parameter's value at runtime.

C. 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. Encrypt the parameter's value by using the AWS Key Management Service (AWS KMS) encrypt command.

D. 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.

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

QUESTION 85

An application runs on multiple EC2 instances behind an ELB.

Where is the session data best written so that it can be served reliably across multiple requests?

- A. Write data to Amazon ElastiCache
- B. Write data to Amazon Elastic Block Store
- C. Write data to Amazon EC2 instance Store
- D. Write data to the root filesystem

Answer: A

Explanation:

The solution that will meet the requirements is to write data to Amazon ElastiCache. This way, the application can write session data to a fast, scalable, and reliable in-memory data store that can be served reliably across multiple requests. The other options either involve writing data to persistent

storage, which is slower and more expensive than in-memory storage, or writing data to the root filesystem, which is not shared among multiple EC2 instances.

Reference: Using ElastiCache for session management

QUESTION 86

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

QUESTION 87

A developer migrated a legacy application to an AWS Lambda function. The function uses a thirdparty service to pull data with a series of API calls at the end of each month. The function then processes the data to generate the monthly reports. The function has been working with no issues so far.

The third-party service recently issued a restriction to allow a fixed number of API calls each minute and each day. If the API calls exceed the limit for each minute or each day, then the service will produce errors. The API also provides the minute limit and daily limit in the response header. This restriction might extend the overall process to multiple days because the process is consuming more API calls than the available limit.

What is the MOST operationally efficient way to refactor the serverless application to accommodate this change?

- A. Use an AWS Step Functions State machine to monitor API failures. Use the Wait state to delay calling the Lambda function.
- B. Use an Amazon Simple Queue Service (Amazon SQS) queue to hold the API calls. Configure the Lambda function to poll the queue within the API threshold limits.

- C. Use an Amazon CloudWatch Logs metric to count the number of API calls. Configure an Amazon CloudWatch alarm that stops the currently running instance of the Lambda function when the metric exceeds the API threshold limits.
- D. Use Amazon Kinesis Data Firehose to batch the API calls and deliver them to an Amazon S3 bucket with an event notification to invoke the Lambda function.

Answer: A

Explanation:

The solution that will meet the requirements is to use an AWS Step Functions state machine to monitor API failures. Use the Wait state to delay calling the Lambda function. This way, the developer can refactor the serverless application to accommodate the change in a way that is automated and scalable. The developer can use Step Functions to orchestrate the Lambda function and handle any errors or retries. The developer can also use the Wait state to pause the execution for a specified duration or until a specified timestamp, which can help avoid exceeding the API limits. The other options either involve using additional services that are not necessary or appropriate for this scenario, or do not address the issue of API failures.

Reference: AWS Step Functions Wait state

QUESTION 88

A developer must analyze performance issues with production-distributed applications written as AWS Lambda functions. These distributed Lambda applications invoke other components that make up the applications. How should the developer identify and troubleshoot the root cause of the performance issues in production?

- A. Add logging statements to the Lambda functions. then use Amazon CloudWatch to view the logs.
- B. Use AWS CloudTrail and then examine the logs.
- C. Use AWS X-Ray. then examine the segments and errors.
- D. Run Amazon Inspector agents and then analyze performance.

Answer: C

Explanation:

This solution will meet the requirements by using AWS X-Ray to analyze and debug the performance issues with the distributed Lambda applications. AWS X-Ray is a service that collects data about requests that the applications serve, and provides tools to view, filter, and gain insights into that data. The developer can use AWS X-Ray to identify the root cause of the performance issues by examining the segments and errors that show the details of each request and the components that make up the applications. Option A is not optimal because it will use logging statements and Amazon CloudWatch, which may not provide enough information or visibility into the distributed applications. Option B is not optimal because it will use AWS CloudTrail, which is a service that records API calls and events for AWS services, not application performance data. Option D is not

optimal because it will use Amazon Inspector, which is a service that helps improve the security and compliance of applications on Amazon EC2 instances, not Lambda functions.

Reference: AWS X-Ray, Using AWS X-Ray with AWS Lambda

QUESTION 89

A developer wants to deploy a new version of an AWS Elastic Beanstalk application. During deployment the application must maintain full capacity and avoid service interruption. Additionally, the developer must minimize the cost of additional resources that support the deployment.

Which deployment method should the developer use to meet these requirements?

- A. All at once
- B. Rolling with additional batch
- C. Bluegreen
- D. Immutable

Answer: B

Explanation:

This solution will meet the requirements by using a rolling with additional batch deployment method, which deploys the new version of the application to a separate group of instances and then shifts traffic to those instances in batches. This way, the application maintains full capacity and avoids service interruption during deployment, as well as minimizes the cost of additional resources that support the deployment. Option A is not optimal because it will use an all at once deployment method, which deploys the new version of the application to all instances simultaneously, which may cause service interruption or downtime during deployment. Option C is not optimal because it will use a blue/green deployment method, which deploys the new version of the application to a separate environment and then swaps URLs with the original environment, which may incur more costs for additional resources that support the deployment. Option D is not optimal because it will use an immutable deployment method, which deploys the new version of the application to a fresh group of instances and then redirects traffic to those instances, which may also incur more costs for additional resources that support the deployment.

Reference: AWS Elastic Beanstalk Deployment Policies

QUESTION 90

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 documentation. Create automated test scripts that use the `cdk local invoke` command to invoke the Lambda functions. Check the response Document the test scripts for the other developers on the team Update the CI/CD pipeline to run the test scripts.
- B. Install a unit testing framework that reproduces the Lambda execution environment. Create sample events based on the Lambda Documentation Invoke the handler function by using a unit testing framework. Check the response Document how to run the unit testing framework for the other developers on the team. Update the OCD pipeline to run the unit testing framework.
- C. Install the AWS Serverless Application Model (AWS SAW) CLI tool Use the `Sam local generateevent` command to generate sample events for me automated tests. Create automated test scripts that use the `Sam local invoke` command to invoke the Lambda functions. Check the response Document the test scripts tor the other developers on the team Update the CI/CD pipeline to run the test scripts.
- D. Create sample events based on the Lambda documentation. Create a Docker container from the Node is base image to invoke the Lambda functions. 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.

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

QUESTION 91

A developer is troubleshooting an application mat uses Amazon DynamoDB in the uswest-2 Region. The application is deployed to an Amazon EC2 instance. The application requires read-only

permissions to a table that is named Cars The EC2 instance has an attached IAM role that contains the following IAM policy.

```
1 {
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Sid": "ReadOnlyAPIActions",
6       "Effect": "Allow",
7       "Action": [
8         "dynamodb:GetItem",
9         "dynamodb:BatchGetItem",
10        "dynamodb:Scan",
11        "dynamodb:Query",
12        "dynamodb:ConditionCheckItem"
13      ],
14      "Resource": "arn:aws:dynamodb:us-west-2:account-id:table/Cars"
15    }
16  ]
17 }
```

When the application tries to read from the Cars table, an Access Denied error occurs. How can the developer resolve this error?

- A. Modify the IAM policy resource to be "arn:aws:dynamodb:us-west-2:account-id:table/*"
- B. Modify the IAM policy to include the dynamodb * action
- C. Create a trust policy that specifies the EC2 service principal. Associate the role with the policy.
- D. Create a trust relationship between the role and dynamodb:Amazon.com.

Answer: C

Explanation:

<https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/access-control-overview.html#access-control-resource-ownership>

QUESTION 92

A developer needs to store configuration variables for an application. The developer needs to set an expiration date and time for the 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 Expiration 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 cron 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.

Reference: AWS Systems Manager Parameter Store, [Using SSL/TLS to Encrypt a Connection to a DB Instance]

QUESTION 93

When using the AWS Encryption SDK how does the developer keep track of the data encryption keys used to encrypt data?

- A. The developer must manually keep track of the data encryption keys used for each data object.
- B. The SDK encrypts the data encryption key and stores it (encrypted) as part of the ciphertext.
- C. The SDK stores the data encryption keys automatically in Amazon S3.
- D. The data encryption key is stored in the user data for the EC2 instance.

Answer: B

Explanation:

This solution will meet the requirements by using AWS Encryption SDK, which is a client-side encryption library that enables developers to encrypt and decrypt data using data encryption keys that are protected by AWS Key Management Service (AWS KMS). The SDK encrypts the data

encryption key with a customer master key (CMK) that is managed by AWS KMS, and stores it (encrypted) as part of the returned ciphertext. The developer does not need to keep track of the data encryption keys used to encrypt data, as they are stored with the encrypted data and can be retrieved and decrypted by using AWS KMS when needed. Option A is not optimal because it will require manual tracking of the data encryption keys used for each data object, which is error-prone and inefficient. Option C is not optimal because it will store the data encryption keys automatically in Amazon S3, which is unnecessary and insecure as Amazon S3 is not designed for storing encryption keys. Option D is not optimal because it will store the data encryption key in the user data for the EC2 instance, which is also unnecessary and insecure as user data is not encrypted by default. Reference: [AWS Encryption SDK], [AWS Key Management Service]

QUESTION 94

An application that runs on AWS Lambda requires access to specific highly confidential objects in an Amazon S3 bucket. In accordance with the principle of least privilege a company grants access to the S3 bucket by using only temporary credentials.

How can a developer configure access to the S3 bucket in the MOST secure way?

- A. Hardcode the credentials that are required to access the S3 objects in the application code. Use the credentials to access the required S3 objects.
- B. Create a secret access key and access key ID with permission to access the S3 bucket. Store the key and key ID in AWS Secrets Manager. Configure the application to retrieve the Secrets Manager secret and use the credentials to access the S3 objects.
- C. Create a Lambda function execution role. Attach a policy to the role that grants access to specific objects in the S3 bucket.
- D. Create a secret access key and access key ID with permission to access the S3 bucket. Store the key and key ID as environment variables in Lambda. Use the environment variables to access the required S3 objects.

Answer: C

Explanation:

This solution will meet the requirements by creating a Lambda function execution role, which is an IAM role that grants permissions to a Lambda function to access AWS resources such as Amazon S3 objects. The developer can attach a policy to the role that grants access to specific objects in the S3 bucket that are required by the application, following the principle of least privilege. Option A is not optimal because it will hardcode the credentials that are required to access S3 objects in the application code, which is insecure and difficult to maintain. 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 store the secret access key and access key ID as environment variables in Lambda, which is also insecure and difficult to maintain.

Reference: [AWS Lambda Execution Role], [Using AWS Lambda with Amazon S3]

QUESTION 95

A developer has code that is stored in an Amazon S3 bucket. The code must be deployed as an AWS Lambda function across multiple accounts in the same AWS Region as the S3 bucket an AWS CloudFormation template that runs for each account will deploy the Lambda function. What is the MOST secure way to allow CloudFormation to access the Lambda Code in the S3 bucket?

- A. Grant the CloudFormation service role the S3 ListBucket and GetObject permissions. Add a bucket policy to Amazon S3 with the principal of "AWS" (account numbers)
- B. Grant the CloudFormation service role the S3 GetObject permission. Add a Bucket policy to Amazon S3 with the principal of ""
- C. Use a service-based link to grant the Lambda function the S3 ListBucket and GetObject permissions by explicitly adding the S3 bucket's account number in the resource.
- D. Use a service-based link to grant the Lambda function the S3 GetObject permission Add a resource of "*" to allow access to the S3 bucket.

Answer: B

Explanation:

This solution allows the CloudFormation service role to access the S3 bucket from any account, as long as it has the S3 GetObject permission. The bucket policy grants access to any principal with the GetObject permission, which is the least privilege needed to deploy the Lambda code. This is more secure than granting ListBucket permission, which is not required for deploying Lambda code, or using a service-based link, which is not supported for Lambda functions.

Reference: AWS CloudFormation Service Role, Using AWS Lambda with Amazon S3

QUESTION 96

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. Create a new API Import the OpenAPI file Modify the new API to add request validation. Perform the tests Modify the existing API to add request validation. Deploy the existing API to production.
- B. Modify the existing API to add request validation. Deploy the updated API to a new API Gateway stage Perform the tests Deploy the updated API to the API Gateway production stage.
- C. Create a new API Add the necessary resources and methods including new request validation. Perform the tests Modify the existing API to add request validation. Deploy the existing API to production.
- D. Clone the existing API Modify the new API to add request validation. Perform the tests Modify the existing API to add request validation Deploy the existing API to production.

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]

QUESTION 97

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:

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]

QUESTION 98

A developer is building a serverless application that is based on AWS Lambda

a. The developer initializes the AWS software development kit (SDK) outside of the Lambda handler function.

What is the PRIMARY benefit of this action?

- A. Improves legibility and stylistic convention
- B. Takes advantage of runtime environment reuse
- C. Provides better error handling
- D. Creates a new SDK instance for each invocation

Answer: B

Explanation:

This benefit occurs when initializing the AWS SDK outside of the Lambda handler function because it allows the SDK instance to be reused across multiple invocations of the same function. This can improve performance and reduce latency by avoiding unnecessary initialization overhead. If the SDK is initialized inside the handler function, it will create a new SDK instance for each invocation, which can increase memory usage and execution time.

Reference: [AWS Lambda execution environment], [Best Practices for Working with AWS Lambda Functions]

QUESTION 99

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]

QUESTION 100

A developer at a company recently created a serverless application to process and show data from business reports. The application's user interface (UI) allows users to select and start processing the files. The UI displays a message when the result is available to view. The application uses AWS Step Functions with AWS Lambda functions to process the files. The developer used Amazon API Gateway and Lambda functions to create an API to support the UI.

The company's UI team reports that the request to process a file is often returning timeout errors because of the size or complexity of the files. The UI team wants the API to provide an immediate response so that the UI can display a message while the files are being processed. The backend process that is invoked by the API needs to send an email message when the report processing is complete.

What should the developer do to configure the API to meet these requirements?

- A. Change the API Gateway route to add an X-Amz-Invocation-Type header with a static value of 'Event' in the integration request. Deploy the API Gateway stage to apply the changes.
- B. Change the configuration of the Lambda function that implements the request to process a file. Configure the maximum age of the event so that the Lambda function will run asynchronously.
- C. Change the API Gateway timeout value to match the Lambda function timeout value. Deploy the API Gateway stage to apply the changes.
- D. Change the API Gateway route to add an X-Amz-Target header with a static value of 'A sync' in the integration request. Deploy the API Gateway stage to apply the changes.

Answer: A

Explanation:

This solution allows the API to invoke the Lambda function asynchronously, which means that the API will return an immediate response without waiting for the function to complete. The X-Amz-Invocation-Type header specifies the invocation type of the Lambda function, and setting it to Event means that the function will be invoked asynchronously. The function can then use Amazon Simple Email Service (SES) to send an email message when the report processing is complete.

Reference: [Asynchronous invocation], [Set up Lambda proxy integrations in API Gateway]

QUESTION 101

A developer has an application that is composed of many different AWS Lambda functions. The Lambda functions all use some of the same dependencies. To avoid security issues the developer is constantly updating the dependencies of all of the Lambda functions. The result is duplicated effort to reach function.

How can the developer keep the dependencies of the Lambda functions up to date with the LEAST additional complexity?

- A. Define a maintenance window for the Lambda functions to ensure that the functions get updated copies of the dependencies.
- B. Upgrade the Lambda functions to the most recent runtime version.
- C. Define a Lambda layer that contains all of the shared dependencies.
- D. Use an AWS CodeCommit repository to host the dependencies in a centralized location.

Answer: C

Explanation:

This solution allows the developer to keep the dependencies of the Lambda functions up to date with the least additional complexity because it eliminates the need to update each function individually. A Lambda layer is a ZIP archive that contains libraries, custom runtimes, or other dependencies. The developer can create a layer that contains all of the shared dependencies and attach it to multiple Lambda functions. When the developer updates the layer, all of the functions that use the layer will have access to the latest version of the dependencies.

Reference: [AWS Lambda layers]

QUESTION 102

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. 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. Schedule a cron job on an Amazon EC2 instance once an hour to start the script.
- B. For each item add a new attribute of type String that has a timestamp that is set to the blog post creation time. 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. Place the script in a container image. Schedule an Amazon Elastic Container Service (Amazon ECS) task on AWS Fargate that invokes the container every 5 minutes.
- C. 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. Create a global secondary index (GSI) that uses the new attribute as a sort key. 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.
- D. 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. Configure the DynamoDB table with a TTL that references the new attribute.

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.

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

QUESTION 103

A developer is modifying an existing AWS Lambda function. While checking the code the developer notices hardcoded parameter values for an Amazon RDS for SQL Server user name password database host and port. There also are hardcoded parameter values for an Amazon DynamoDB table, an Amazon S3 bucket, and an Amazon Simple Notification Service (Amazon SNS) topic. The developer wants to securely store the parameter values outside the code in an encrypted format and wants to turn on rotation for the credentials. The developer also wants to be able to reuse the parameter values from other applications and to update the parameter values without modifying code.

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

- A. Create an RDS database secret in AWS Secrets Manager. Set the user name password, database, host and port. Turn on secret rotation. Create encrypted Lambda environment variables for the DynamoDB table, S3 bucket and SNS topic.
- B. Create an RDS database secret in AWS Secrets Manager. Set the user name password, database, host and port. Turn on secret rotation. Create Secure String parameters in AWS Systems Manager Parameter Store for the DynamoDB table, S3 bucket and SNS topic.
- C. Create RDS database parameters in AWS Systems Manager Parameter Store. Store for the user name password, database, host and port. Create encrypted Lambda environment variables for the DynamoDB table, S3 bucket, and SNS topic. Create a Lambda function and set the logic for the credentials rotation task. Schedule the credentials rotation task in Amazon EventBridge.
- D. Create RDS database parameters in AWS Systems Manager Parameter Store. Store for the user name password database, host, and port. Store the DynamoDB table, S3 bucket, and SNS topic in Amazon S3. Create a Lambda function and set the logic for the credentials rotation. Invoke the Lambda function on a schedule.

Answer: B

Explanation:

This solution will meet the requirements by using AWS Secrets Manager and AWS Systems Manager Parameter Store to securely store the parameter values outside the code in an encrypted format.

AWS Secrets Manager 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 RDS database secret in AWS Secrets Manager and set the user name, password, database, host, and port for accessing the RDS database. The developer can also turn on secret rotation, which will change the database credentials periodically according to a specified schedule or event. AWS Systems Manager Parameter Store is a service that provides secure and scalable storage for configuration data and secrets. The developer can create Secure String parameters in AWS Systems Manager Parameter Store for the DynamoDB table, S3 bucket, and SNS topic, which will encrypt them with AWS KMS. The developer can also reuse the parameter values from other applications and update them without modifying code. Option A is not optimal because it will create encrypted Lambda environment variables for the DynamoDB table, S3 bucket, and SNS topic, which may not be reusable or updatable without modifying code. Option C is not optimal because it will create RDS database parameters in AWS Systems Manager Parameter Store, which does not support automatic rotation of secrets. Option D is not optimal because it will store the DynamoDB table, S3 bucket, and SNS topic in Amazon S3, which may introduce additional costs and complexity for accessing configuration data.

Reference: AWS Secrets Manager, [AWS Systems Manager Parameter Store]

QUESTION 104

A developer accesses AWS CodeCommit over SSH. The SSH keys configured to access AWS CodeCommit are tied to a user with the following permissions:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "codecommit:BatchGetRepositories",
        "codecommit:Get*",
        "codecommit:List*",
        "codecommit:GitPull"
      ],
      "Resource": "*"
    }
  ]
}
```

The developer needs to create/delete branches

Which specific IAM permissions need to be added based on the principle of least privilege?

A. "codecommit:CreateBranch"
"codecommit>DeleteBranch"

B. "codecommit:Put*"

C. "codecommit:Update*"

D. "codecommit:*"

- A. Option A
- B. Option B
- C. Option C
- D. Option D

Answer: A

Explanation:

This solution allows the developer to create and delete branches in AWS CodeCommit by granting the codecommit:CreateBranch and codecommit>DeleteBranch permissions. These are the minimum permissions required for this task, following the principle of least privilege. Option B grants too many permissions, such as codecommit:Put*, which allows the developer to create, update, or delete any resource in CodeCommit. Option C grants too few permissions, such as codecommit:Update*, which does not allow the developer to create or delete branches. Option D grants all permissions, such as codecommit:*, which is not secure or recommended.

Reference: [AWS CodeCommit Permissions Reference], [Create a Branch (AWS CLI)]

QUESTION 105

An application that is deployed to Amazon EC2 is using Amazon DynamoDB. The application calls the DynamoDB REST API. Periodically the application receives a ProvisionedThroughputExceededException error when the application writes to a DynamoDB table. Which solutions will mitigate this error MOST cost-effectively? (Select TWO)

- A. Modify the application code to perform exponential back off when the error is received.
- B. Modify the application to use the AWS SDKs for DynamoDB.
- C. Increase the read and write throughput of the DynamoDB table.
- D. Create a DynamoDB Accelerator (DAX) cluster for the DynamoDB table.
- E. Create a second DynamoDB table. Distribute the reads and writes between the two tables.

Answer: A, B

Explanation:

These solutions will mitigate the error most cost-effectively because they do not require increasing the provisioned throughput of the DynamoDB table or creating additional resources. Exponential backoff is a retry strategy that increases the waiting time between retries to reduce the number of requests sent to DynamoDB. The AWS SDKs for DynamoDB implement exponential backoff by default and also provide other features such as automatic pagination and encryption. Increasing the read and write throughput of the DynamoDB table, creating a DynamoDB Accelerator (DAX) cluster, or creating a second DynamoDB table will incur additional costs and complexity.

Reference: [Error Retries and Exponential Backoff in AWS], [Using the AWS SDKs with DynamoDB]

QUESTION 106

When a developer tries to run an AWS Code Build project, it raises an error because the length of all environment variables exceeds the limit for the combined maximum of characters.

What is the recommended solution?

- A. Add the export LC- _ALL" on _ US, tuft" command to the pre _ build section to ensure POSIX Localization.
- B. Use Amazon Cognate to store key-value pairs for large numbers of environment variables
- C. Update the settings for the build project to use an Amazon S3 bucket for large numbers of environment variables
- D. Use AWS Systems Manager Parameter Store to store large numbers of environment variables

Answer: D

Explanation:

This solution allows the developer to overcome the limit for the combined maximum of characters for environment variables in AWS CodeBuild. AWS Systems Manager Parameter Store provides secure, hierarchical storage for configuration data management and secrets management. The developer can store large numbers of environment variables as parameters in Parameter Store and reference them in the buildspec file using parameter references. Adding export LC- _ALL=œen _US.utf8 command to the pre _ build section will not affect the environment variables limit. Using Amazon Cognito or an Amazon S3 bucket to store key-value pairs for environment variables will require additional configuration and integration.

Reference: [Build Specification Reference for AWS CodeBuild], [What Is AWS Systems Manager Parameter Store?]

QUESTION 107

A company is expanding the compatibility of its photo-snaring mobile app to hundreds of additional devices with unique screen dimensions and resolutions. Photos are stored in Amazon S3 in their original format and resolution. The company uses an Amazon CloudFront distribution to serve the photos. The app includes the dimension and resolution of the display as GET parameters with every request.

A developer needs to implement a solution that optimizes the photos that are served to each device to reduce load time and increase photo quality.

Which solution will meet these requirements MOST cost-effective?

- A. Use S3 Batch Operations to invoke an AWS Lambda function to create new variants of the photos with the required dimensions and resolutions. Create a dynamic CloudFront origin that automatically maps the request of each device to the corresponding photo variant.
- B. Use S3 Batch Operations to invoke an AWS Lambda function to create new variants of the photos with the required dimensions and resolutions. Create a Lambda@Edge function to route requests to the corresponding photo variant by using request headers.
- C. Create a Lambda@Edge function that optimizes the photos upon request and returns the photos as a response. Change the CloudFront TTL cache policy to the maximum value possible.
- D. Create a Lambda@Edge function that optimizes the photos upon request and returns the photos as a response. In the same function store a copy of the processed photos on Amazon S3 for subsequent requests.

Answer: D

Explanation:

This solution meets the requirements most cost-effectively because it optimizes the photos on demand and caches them for future requests. Lambda@Edge allows the developer to run Lambda functions at AWS locations closer to viewers, which can reduce latency and improve photo quality. The developer can create a Lambda@Edge function that uses the GET parameters from each request to optimize the photos with the required dimensions and resolutions and returns them as a response. The function can also store a copy of the processed photos on Amazon S3 for subsequent requests, which can reduce processing time and costs. Using S3 Batch Operations to create new variants of the photos will incur additional storage costs and may not cover all possible dimensions and resolutions. Creating a dynamic CloudFront origin or a Lambda@Edge function to route requests to corresponding photo variants will require maintaining a mapping of device types and photo variants, which can be complex and error-prone.

Reference: [Lambda@Edge Overview], [Resizing Images with Amazon CloudFront & Lambda@Edge]

QUESTION 108

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.

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

QUESTION 109

A developer is working on a Python application that runs on Amazon EC2 instances. The developer wants to enable tracing of application requests to debug performance issues in the code.

Which combination of actions should the developer take to achieve this goal? (Select TWO)

- A. Install the Amazon CloudWatch agent on the EC2 instances.
- B. Install the AWS X-Ray daemon on the EC2 instances.
- C. Configure the application to write JSON-formatted logs to `/var/log/cloudwatch`.
- D. Configure the application to write trace data to `/var/log/xray`.
- E. Install and configure the AWS X-Ray SDK for Python in the application.

Answer: BE

Explanation:

This solution will meet the requirements by using AWS X-Ray to enable tracing of application requests to debug performance issues in the code. AWS X-Ray is a service that collects data about requests that the applications serve, and provides tools to view, filter, and gain insights into that data. The developer can install the AWS X-Ray daemon on the EC2 instances, which is a software that listens for traffic on UDP port 2000, gathers raw segment data, and relays it to the X-Ray API. The developer can also install and configure the AWS X-Ray SDK for Python in the application, which is a library that enables instrumenting Python code to generate and send trace data to the X-Ray

daemon. Option A is not optimal because it will install the Amazon CloudWatch agent on the EC2 instances, which is a software that collects metrics and logs from EC2 instances and on-premises servers, not application performance data. Option C is not optimal because it will configure the application to write JSON-formatted logs to /var/log/cloudwatch, which is not a valid path or destination for CloudWatch logs. Option D is not optimal because it will configure the application to write trace data to /var/log/xray, which is also not a valid path or destination for X-Ray trace data. Reference: [AWS X-Ray], [Running the X-Ray Daemon on Amazon EC2]

QUESTION 110

A company has an application that runs as a series of AWS Lambda functions. Each Lambda function receives data from an Amazon Simple Notification Service (Amazon SNS) topic and writes the data to an Amazon Aurora DB instance.

To comply with an information security policy, the company must ensure that the Lambda functions all use a single securely encrypted database connection string to access Aurora.

Which solution will meet these requirements'?

- A. Use IAM database authentication for Aurora to enable secure database connections for all the Lambda functions.
- B. Store the credentials and read the credentials from an encrypted Amazon RDS DB instance.
- C. Store the credentials in AWS Systems Manager Parameter Store as a secure string parameter.
- D. Use Lambda environment variables with a shared AWS Key Management Service (AWS KMS) key for encryption.

Answer: A

Explanation:

This solution will meet the requirements by using IAM database authentication for Aurora, which enables using IAM roles or users to authenticate with Aurora databases instead of using passwords or other secrets. The developer can use IAM database authentication for Aurora to enable secure database connections for all the Lambda functions that access Aurora DB instance. The developer can create an IAM role with permission to connect to Aurora DB instance and attach it to each Lambda function. The developer can also configure Aurora DB instance to use IAM database authentication and enable encryption in transit using SSL certificates. This way, the Lambda functions can use a single securely encrypted database connection string to access Aurora without needing any secrets or passwords. Option B is not optimal because it will store the credentials and read them from an encrypted Amazon RDS DB instance, which may introduce additional costs and complexity for managing and accessing another RDS DB instance. Option C is not optimal because it will store the credentials in AWS Systems Manager Parameter Store as a secure string parameter, which may require additional steps or permissions to retrieve and decrypt the credentials from Parameter Store. Option D is not optimal because it will use Lambda environment variables with a shared AWS Key Management Service (AWS KMS) key for encryption, which may not be secure or scalable as environment variables are stored as plain text unless encrypted with AWS KMS.

Reference: [IAM Database Authentication for MySQL and PostgreSQL], [Using SSL/TLS to Encrypt a Connection to a DB Instance]

QUESTION 111

A developer is troubleshooting an Amazon API Gateway API Clients are receiving HTTP 400 response errors when the clients try to access an endpoint of the API.
How can the developer determine the cause of these errors?

- A. Create an Amazon Kinesis Data Firehose delivery stream to receive API call logs from API Gateway. Configure Amazon CloudWatch Logs as the delivery stream's destination.
- B. Turn on AWS CloudTrail Insights and create a trail Specify the Amazon Resource Name (ARN) of the trail for the stage of the API.
- C. Turn on AWS X-Ray for the API stage Create an Amazon CloudWatch Logs log group Specify the Amazon Resource Name (ARN) of the log group for the API stage.
- D. Turn on execution logging and access logging in Amazon CloudWatch Logs for the API stage. Create a CloudWatch Logs log group. Specify the Amazon Resource Name (ARN) of the log group for the API stage.

Answer: D

Explanation:

This solution will meet the requirements by using Amazon CloudWatch Logs to capture and analyze the logs from API Gateway. Amazon CloudWatch Logs is a service that monitors, stores, and accesses log files from AWS resources. The developer can turn on execution logging and access logging in Amazon CloudWatch Logs for the API stage, which enables logging information about API execution and client access to the API. The developer can create a CloudWatch Logs log group, which is a collection of log streams that share the same retention, monitoring, and access control settings. The developer can specify the Amazon Resource Name (ARN) of the log group for the API stage, which instructs API Gateway to send the logs to the specified log group. The developer can then examine the logs to determine the cause of the HTTP 400 response errors. Option A is not optimal because it will create an Amazon Kinesis Data Firehose delivery stream to receive API call logs from API Gateway, which may introduce additional costs and complexity for delivering and processing streaming data. Option B is not optimal because it will turn on AWS CloudTrail Insights and create a trail, which is a feature that helps identify and troubleshoot unusual API activity or operational issues, not HTTP response errors. Option C is not optimal because it will turn on AWS X-Ray for the API stage, which is a service that helps analyze and debug distributed applications, not HTTP response errors.

Reference: [Setting Up CloudWatch Logging for a REST API], [CloudWatch Logs Concepts]

QUESTION 112

A company developed an API application on AWS by using Amazon CloudFront. Amazon API Gateway, and AWS Lambda

a. The API has a minimum of four requests every second A developer notices that many API users run the same query by using the POST method. The developer wants to cache the POST request to optimize the API resources.

Which solution will meet these requirements'?

A. Configure the CloudFront cache Update the application to return cached content based upon the default request headers.

B. Override the cache method in me selected stage of API Gateway Select the POST method.

C. Save the latest request response in Lambda /tmp directory Update the Lambda function to check the /tmp directory

D. Save the latest request m AWS Systems Manager Parameter Store Modify the Lambda function to take the latest request response from Parameter Store

Answer: A

Explanation:

This solution will meet the requirements by using Amazon CloudFront, which is a content delivery network (CDN) service that speeds up the delivery of web content and APIs to end users. The developer can configure the CloudFront cache, which is a set of edge locations that store copies of popular or recently accessed content close to the viewers. The developer can also update the application to return cached content based upon the default request headers, which are a set of HTTP headers that CloudFront automatically forwards to the origin server and uses to determine whether an object in an edge location is still valid. By caching the POST requests, the developer can optimize the API resources and reduce the latency for repeated queries. Option B is not optimal because it will override the cache method in the selected stage of API Gateway, which is not possible or effective as API Gateway does not support caching for POST methods by default. Option C is not optimal because it will save the latest request response in Lambda /tmp directory, which is a local storage space that is available for each Lambda function invocation, not a cache that can be shared across multiple invocations or requests. Option D is not optimal because it will save the latest request in AWS Systems Manager Parameter Store, which is a service that provides secure and scalable storage for configuration data and secrets, not a cache for API responses.

Reference: [Amazon CloudFront], [Caching Content Based on Request Headers]

QUESTION 113

A company is building a micro services application that consists of many AWS Lambda functions. The development team wants to use AWS Serverless Application Model (AWS SAM) templates to automatically test the Lambda functions. The development team plans to test a small percentage of traffic that is directed to new updates before the team commits to a full deployment of the application.

Which combination of steps will meet these requirements in the MOST operationally efficient way?

(Select TWO.)

- A. Use AWS SAM CLI commands in AWS CodeDeploy to invoke the Lambda functions to test the deployment
- B. Declare the EventInvokeConfig on the Lambda functions in the AWS SAM templates with OnSuccess and OnFailure configurations.
- C. Enable gradual deployments through AWS SAM templates.
- D. Set the deployment preference type to Canary10Percent30Minutes Use hooks to test the deployment.
- E. Set the deployment preference type to Linear10PercentEvery10Minutes Use hooks to test the deployment.

Answer: C, D

Explanation:

This solution will meet the requirements by using AWS Serverless Application Model (AWS SAM) templates and gradual deployments to automatically test the Lambda functions. AWS SAM templates are configuration files that define serverless applications and resources such as Lambda functions. Gradual deployments are a feature of AWS SAM that enable deploying new versions of Lambda functions incrementally, shifting traffic gradually, and performing validation tests during deployment. The developer can enable gradual deployments through AWS SAM templates by adding a DeploymentPreference property to each Lambda function resource in the template. The developer can set the deployment preference type to Canary10Percent30Minutes, which means that 10 percent of traffic will be shifted to the new version of the Lambda function for 30 minutes before shifting 100 percent of traffic. The developer can also use hooks to test the deployment, which are custom Lambda functions that run before or after traffic shifting and perform validation tests or rollback actions.

Reference: [AWS Serverless Application Model (AWS SAM)], [Gradual Code Deployment]

QUESTION 114

A company is using AWS CloudFormation to deploy a two-tier application. The application will use Amazon RDS as its backend database. The company wants a solution that will randomly generate the database password during deployment. The solution also must automatically rotate the database password without requiring changes to the application.

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

- A. Use an AWS Lambda function as a CloudFormation custom resource to generate and rotate the password.
- B. Use an AWS Systems Manager Parameter Store resource with the SecureString data type to generate and rotate the password.
- C. Use a cron daemon on the application's host to generate and rotate the password.
- D. Use an AWS Secrets Manager resource to generate and rotate the password.

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 use an AWS Secrets Manager resource in AWS CloudFormation template, which enables creating and managing secrets as part of a CloudFormation stack. The developer can use an `AWS::SecretsManager::Secret` resource type to generate and rotate the password for accessing RDS database during deployment. The developer can also specify a `RotationSchedule` property for the secret resource, which defines how often to rotate the secret and which Lambda function to use for rotation logic. Option A is not optimal because it will use an AWS Lambda function as a CloudFormation custom resource, which may introduce additional complexity and overhead for creating and managing a custom resource and implementing rotation logic. Option B is not optimal because it will use an AWS Systems Manager Parameter Store resource with the `SecureString` data type, which does not support automatic rotation of secrets. Option C is not optimal because it will use a cron daemon on the applications host to generate and rotate the password, which may incur more costs and require more maintenance for running and securing a host.

Reference: [AWS Secrets Manager], [AWS::SecretsManager::Secret]

QUESTION 115

A developer has been asked to create an AWS Lambda function that is invoked any time updates are made to items in an Amazon DynamoDB table. The function has been created and appropriate permissions have been added to the Lambda execution role Amazon DynamoDB streams have been enabled for the table, but the function is still not being invoked.

Which option would enable DynamoDB table updates to invoke the Lambda function?

- A. Change the `StreamViewType` parameter value to `NEW_AND_OLD_IMAGES` for the DynamoDB table.
- B. Configure event source mapping for the Lambda function.
- C. Map an Amazon Simple Notification Service (Amazon SNS) topic to the DynamoDB streams.
- D. Increase the maximum runtime (timeout) setting of the Lambda function.

Answer: B

Explanation:

This solution allows the Lambda function to be invoked by the DynamoDB stream whenever updates are made to items in the DynamoDB table. Event source mapping is a feature of Lambda that enables a function to be triggered by an event source, such as a DynamoDB stream, an Amazon Kinesis stream, or an Amazon Simple Queue Service (SQS) queue. The developer can configure event source mapping for the Lambda function using the AWS Management Console, the AWS CLI, or the AWS SDKs. Changing the `StreamViewType` parameter value to `NEW_AND_OLD_IMAGES` for the DynamoDB table will not affect the invocation of the Lambda function, but only change the

information that is written to the stream record. Mapping an Amazon Simple Notification Service (Amazon SNS) topic to the DynamoDB stream will not invoke the Lambda function directly, but require an additional subscription from the Lambda function to the SNS topic. Increasing the maximum runtime (timeout) setting of the Lambda function will not affect the invocation of the Lambda function, but only change how long the function can run before it is terminated.
Reference: [Using AWS Lambda with Amazon DynamoDB], [Using AWS Lambda with Amazon SNS]

QUESTION 116

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]

QUESTION 117

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]

QUESTION 118

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 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 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]

QUESTION 119

A company needs to distribute firmware updates to its customers around the world.

Which service will allow easy and secure control of the access to the downloads at the lowest cost?

- A. Use Amazon CloudFront with signed URLs for Amazon S3.
- B. Create a dedicated Amazon CloudFront Distribution for each customer.
- C. Use Amazon CloudFront with AWS Lambda@Edge.
- D. Use Amazon API Gateway and AWS Lambda to control access to an S3 bucket.

Answer: A

Explanation:

This solution allows easy and secure control of access to the downloads at the lowest cost because it uses a content delivery network (CDN) that can cache and distribute firmware updates to customers around the world, and uses a mechanism that can restrict access to specific files or versions. Amazon CloudFront is a CDN that can improve performance, availability, and security of web applications by delivering content from edge locations closer to customers. Amazon S3 is a storage service that can store firmware updates in buckets and objects. Signed URLs are URLs that include additional information, such as an expiration date and time, that give users temporary access to specific objects in S3 buckets. The developer can use CloudFront to serve firmware updates from S3 buckets and use signed URLs to control who can download them and for how long. Creating a dedicated CloudFront distribution for each customer will incur unnecessary costs and complexity. Using Amazon

CloudFront with AWS Lambda@Edge will require additional programming overhead to implement custom logic at the edge locations. Using Amazon API Gateway and AWS Lambda to control access to an S3 bucket will also require additional programming overhead and may not provide optimal performance or availability.

Reference: [Serving Private Content through CloudFront], [Using CloudFront with Amazon S3]

QUESTION 120

A developer is testing an application that invokes an AWS Lambda function asynchronously. During the testing phase the Lambda function fails to process after two retries.

How can the developer troubleshoot the failure?

- A. Configure AWS CloudTrail logging to investigate the invocation failures.
- B. Configure Dead Letter Queues by sending events to Amazon SQS for investigation.
- C. Configure Amazon Simple Workflow Service to process any direct unprocessed events.
- D. Configure AWS Config to process any direct unprocessed events.

Answer: B

Explanation:

This solution allows the developer to troubleshoot the failure by capturing unprocessed events in a queue for further analysis. Dead Letter Queues (DLQs) are queues that store messages that could not be processed by a service, such as Lambda, for various reasons, such as configuration errors, throttling limits, or permissions issues. The developer can configure DLQs for Lambda functions by sending events to either an Amazon Simple Queue Service (SQS) queue or an Amazon Simple Notification Service (SNS) topic. The developer can then inspect the messages in the queue or topic to identify and fix the root cause of the failure. Configuring AWS CloudTrail logging will not capture invocation failures for asynchronous Lambda invocations, but only record API calls made by or on behalf of Lambda. Configuring Amazon Simple Workflow Service (SWF) or AWS Config will not process any direct unprocessed events, but require additional integration and configuration.

Reference: [Using AWS Lambda with DLQs], [Asynchronous invocation]

QUESTION 121

A company is migrating its PostgreSQL database into the AWS Cloud. The company wants to use a database that will secure and regularly rotate database credentials. The company wants a solution that does not require additional programming overhead.

Which solution will meet these requirements?

- A. Use Amazon Aurora PostgreSQL for the database. Store the database credentials in AWS Systems Manager Parameter Store. Turn on rotation.
- B. Use Amazon Aurora PostgreSQL for the database. Store the database credentials in AWS Secrets Manager. Turn on rotation.
- C. Use Amazon DynamoDB for the database. Store the database credentials in AWS Systems Manager

Parameter Store Turn on rotation.

D. Use Amazon DynamoDB for the database. Store the database credentials in AWS Secrets Manager Turn on rotation.

Answer: B

Explanation:

This solution meets the requirements because it uses a PostgreSQL-compatible database that can secure and regularly rotate database credentials without requiring additional programming overhead. Amazon Aurora PostgreSQL is a relational database service that is compatible with PostgreSQL and offers high performance, availability, and scalability. AWS Secrets Manager is a service that helps you protect secrets needed to access your applications, services, and IT resources. You can store database credentials in AWS Secrets Manager and use them to access your Aurora PostgreSQL database. You can also enable automatic rotation of your secrets according to a schedule or an event. AWS Secrets Manager handles the complexity of rotating secrets for you, such as generating new passwords and updating your database with the new credentials. Using Amazon DynamoDB for the database will not meet the requirements because it is a NoSQL database that is not compatible with PostgreSQL. Using AWS Systems Manager Parameter Store for storing and rotating database credentials will require additional programming overhead to integrate with your database.

Reference: [What Is Amazon Aurora?], [What Is AWS Secrets Manager?]

QUESTION 122

A developer is creating a mobile application that will not require users to log in.

What is the MOST efficient method to grant users access to AWS resources'?

- A. Use an identity provider to securely authenticate with the application.
- B. Create an AWS Lambda function to create an IAM user when a user accesses the application.
- C. Create credentials using AWS KMS and apply these credentials to users when using the application.
- D. Use Amazon Cognito to associate unauthenticated users with an IAM role that has limited access to resources.

Answer: D

Explanation:

This solution is the most efficient method to grant users access to AWS resources without requiring them to log in. Amazon Cognito is a service that provides user sign-up, sign-in, and access control for web and mobile applications. Amazon Cognito identity pools support both authenticated and unauthenticated users. Unauthenticated users receive access to your AWS resources even if they aren't logged in with any of your identity providers (IdPs). You can use Amazon Cognito to associate unauthenticated users with an IAM role that has limited access to resources, such as Amazon S3

buckets or DynamoDB tables. This degree of access is useful to display content to users before they log in or to allow them to perform certain actions without signing up. Using an identity provider to securely authenticate with the application will require users to log in, which does not meet the requirement. Creating an AWS Lambda function to create an IAM user when a user accesses the application will incur unnecessary costs and complexity, and may pose security risks if not implemented properly. Creating credentials using AWS KMS and applying them to users when using the application will also incur unnecessary costs and complexity, and may not provide fine-grained access control for resources.

Reference: Switching unauthenticated users to authenticated users (identity pools), Allow user access to your API without authentication (Anonymous user access)

QUESTION 123

A company has developed a new serverless application using AWS Lambda functions that will be deployed using the AWS Serverless Application Model (AWS SAM) CLI.

Which step should the developer complete prior to deploying the application?

- A. Compress the application to a zip file and upload it into AWS Lambda.
- B. Test the new AWS Lambda function by first tracing it in AWS X-Ray.
- C. Bundle the serverless application using a SAM package.
- D. Create the application environment using the `eb create my-env` command.

Answer: C

Explanation:

This step should be completed prior to deploying the application because it prepares the application artifacts for deployment. The AWS Serverless Application Model (AWS SAM) is a framework that simplifies building and deploying serverless applications on AWS. The AWS SAM CLI is a commandline tool that helps you create, test, and deploy serverless applications using AWS SAM templates.

The `sam package` command bundles the application artifacts, such as Lambda function code and API definitions, and uploads them to an Amazon S3 bucket. The command also returns a CloudFormation template that is ready to be deployed with the `sam deploy` command. Compressing the application to a zip file and uploading it to AWS Lambda will not work because it does not use AWS SAM templates or CloudFormation. Testing the new Lambda function by first tracing it in AWS X-Ray will not prepare the application for deployment, but only monitor its performance and errors. Creating the application environment using the `eb create my-env` command will not work because it is a command for AWS Elastic Beanstalk, not AWS SAM.

QUESTION 124

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 developers 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 resources. Create an AWS CloudFormation template from a JSON file. Use the template to attach the function code to an AWS Lambda function and to invoke the Lambda function when the deployment stack runs.
- B. In the central AWS CDK application, write a handler function in the code that uses AWS SDK calls to check for and delete unused resources. 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.
- C. In the central AWS CDK, write a handler function in the code that uses AWS SDK calls to check for and delete unused resources. 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.
- D. In the AWS Lambda console write a handler function in the code that uses AWS SDK calls to check for and delete unused resources. Create an AWS CDK custom resource. 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]

QUESTION 125

A company built a new application in the AWS Cloud. The company automated the bootstrapping of

new resources with an Auto Scaling group by using AWS CloudFormation templates. The bootstrap scripts contain sensitive data.

The company needs a solution that is integrated with CloudFormation to manage the sensitive data in the bootstrap scripts.

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

- A. Put the sensitive data into a CloudFormation parameter. Encrypt the CloudFormation templates by using an AWS Key Management Service (AWS KMS) key.
- B. Put the sensitive data into an Amazon S3 bucket. Update the CloudFormation templates to download the object from Amazon S3 during bootstrap.
- C. Put the sensitive data into AWS Systems Manager Parameter Store as a secure string parameter. Update the CloudFormation templates to use dynamic references to specify template values.
- D. Put the sensitive data into Amazon Elastic File System (Amazon EFS). Enforce EFS encryption after file system creation. Update the CloudFormation templates to retrieve data from Amazon EFS.

Answer: C

Explanation:

This solution meets the requirements in the most secure way because it uses a service that is integrated with CloudFormation to manage sensitive data in encrypted form. AWS Systems Manager Parameter Store provides secure, hierarchical storage for configuration data management and secrets management. You can store sensitive data as secure string parameters, which are encrypted using an AWS Key Management Service (AWS KMS) key of your choice. You can also use dynamic references in your CloudFormation templates to specify template values that are stored in Parameter Store or Secrets Manager without having to include them in your templates. Dynamic references are resolved only during stack creation or update operations, which reduces exposure risks for sensitive data. Putting sensitive data into a CloudFormation parameter will not encrypt them or protect them from unauthorized access. Putting sensitive data into an Amazon S3 bucket or Amazon Elastic File System (Amazon EFS) will require additional configuration and integration with CloudFormation and may not provide fine-grained access control or encryption for sensitive data.

Reference: [What Is AWS Systems Manager Parameter Store?], [Using Dynamic Reference to Specify Template Values]

QUESTION 126

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. Generate user tokens to provide centralized access to RDS DB instances, Amazon DocumentDB clusters, and Aurora DB instances.

- B. Create parameters for the database credentials in AWS Systems Manager Parameter Store Set the Type parameter to Secure String. Set up automatic rotation on the parameters.
- C. Store the database access credentials as an encrypted Amazon S3 object in an S3 bucket Block all public access on the S3 bucket. Use S3 server-side encryption to set up automatic rotation on the encryption key.
- D. Create an AWS Lambda function by using the SecretsManagerRotationTemplate template in the AWS Secrets Manager console. 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.

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

QUESTION 127

A developer has created an AWS Lambda function that makes queries to an Amazon Aurora MySQL DB instance. When the developer performs a test the DB instance shows an error for too many connections.

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

- A. Create a read replica for the DB instance Query the replica DB instance instead of the primary DB instance.
- B. Migrate the data to an Amazon DynamoDB database.
- C. Configure the Amazon Aurora MySQL DB instance for Multi-AZ deployment.
- D. Create a proxy in Amazon RDS Proxy Query the proxy instead of the DB instance.

Answer: D

Explanation:

This solution will meet the requirements by using Amazon RDS Proxy, which is a fully managed, highly available database proxy for Amazon RDS that makes applications more scalable, more resilient to database failures, and more secure. The developer can create a proxy in Amazon RDS Proxy, which sits between the application and the DB instance and handles connection management, pooling, and routing. The developer can query the proxy instead of the DB instance, which reduces the number of open connections to the DB instance and avoids errors for too many connections. Option A is not optimal because it will create a read replica for the DB instance, which may not solve the problem of too many connections as read replicas also have connection limits and may incur additional costs. Option B is not optimal because it will migrate the data to an Amazon DynamoDB database, which may introduce additional complexity and overhead for migrating and accessing data from a different database service. Option C is not optimal because it will configure the Amazon Aurora MySQL DB instance for Multi-AZ deployment, which may improve availability and durability of the DB instance but not reduce the number of connections.

Reference: [Amazon RDS Proxy], [Working with Amazon RDS Proxy]

QUESTION 128

A company uses Amazon API Gateway to expose a set of APIs to customers. The APIs have caching enabled in API Gateway. Customers need a way to invalidate the cache for each API when they test the API.

What should a developer do to give customers the ability to invalidate the API cache?

- A. Ask the customers to use AWS credentials to call the InvalidateCache API operation.
- B. Attach an InvalidateCache policy to the IAM execution role that the customers use to invoke the API. Ask the customers to send a request that contains the HTTP header when they make an API call.
- C. Ask the customers to use the AWS SDK API Gateway class to invoke the InvalidateCache API operation.
- D. Attach an InvalidateCache policy to the IAM execution role that the customers use to invoke the API. Ask the customers to add the INVALIDATE_CACHE query string parameter when they make an API call.

Answer: D

Explanation:

QUESTION 129

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 Canaryl OPercent10Minutes. Set the AutoPublishAlias property to the Lambda alias.
- B. Set the Deployment Preference Type to Linearl OPercentEveryIOMinutes. Set AutoPubl-ishAlias property to the Lambda alias.
- C. Set the Deployment Preference Type to Canaryl OPercentIOMinutes. Set the PreTraffic and PostTraffic properties to the Lambda alias.
- D. Set the Deployment Preference Type to Linearl OPercentEvery10Minutes. Set PreTraffic and PostTraffic properties to the Lambda alias.

Answer: A

Explanation:

The Deployment Preference Type property specifies how traffic should be shifted between versions of a Lambda function¹. The Canary10Percent10Minutes option means that 10% of the traffic is immediately shifted to the new version, and after 10 minutes, the remaining 90% of the traffic is shifted¹. This matches the requirement of shifting 10% of the traffic for the first 10 minutes, and then switching all traffic to the new version.

The AutoPublishAlias property enables AWS SAM to automatically create and update a Lambda alias that points to the latest version of the function¹. This is required to use the Deployment Preference Type property¹. The alias name can be specified by the developer, and it can be used to invoke the function with the latest code.

QUESTION 130

A developer is preparing to begin development of a new version of an application. The previous version of the application is deployed in a production environment. The developer needs to deploy fixes and updates to the current version during the development of the new version of the application. The code for the new version of the application is stored in AWS CodeCommit. Which solution will meet these requirements?

- A. From the main branch, create a feature branch for production bug fixes. Create a second feature branch from the main branch for development of the new version.
- B. Create a Git tag of the code that is currently deployed in production. Create a Git tag for the development of the new version. Push the two tags to the CodeCommit repository.
- C. From the main branch, create a branch of the code that is currently deployed in production. Apply an IAM policy that ensures no other other users can push or merge to the branch.
- D. Create a new CodeCommit repository for development of the new version of the application. Create a Git tag for the development of the new version.

Answer: A

Explanation:

A feature branch is a branch that is created from the main branch to work on a specific feature or task¹. Feature branches allow developers to isolate their work from the main branch and avoid conflicts with other changes¹. Feature branches can be merged back to the main branch when the feature or task is completed and tested¹.

In this scenario, the developer needs to maintain two parallel streams of work: one for fixing and updating the current version of the application that is deployed in production, and another for developing the new version of the application. The developer can use feature branches to achieve this goal.

The developer can create a feature branch from the main branch for production bug fixes. This branch will contain the code that is currently deployed in production, and any fixes or updates that need to be applied to it. The developer can push this branch to the CodeCommit repository and use it to deploy changes to the production environment.

The developer can also create a second feature branch from the main branch for development of the new version of the application. This branch will contain the code that is under development for the new version, and any changes or enhancements that are part of it. The developer can push this branch to the CodeCommit repository and use it to test and deploy the new version of the application in a separate environment.

By using feature branches, the developer can keep the main branch stable and clean, and avoid mixing code from different versions of the application. The developer can also easily switch between branches and merge them when needed.

QUESTION 131

A developer is creating a new REST API by using Amazon API Gateway and AWS Lambda

a. The development team tests the API and validates responses for the known use cases before deploying the API to the production environment.

The developer wants to make the REST API available for testing by using API Gateway locally.

Which AWS Serverless Application Model Command Line Interface (AWS SAM CLI) subcommand will meet these requirements?

- A. Sam local invoke
- B. Sam local generate-event
- C. Sam local start-lambda
- D. Sam local start-api

Answer: D

Explanation:

The sam local start-api subcommand allows you to run your serverless application locally for quick development and testing¹. It creates a local HTTP server that acts as a proxy for API Gateway and

invokes your Lambda functions based on the AWS SAM template¹. You can use the `sam local startapi` subcommand to test your REST API locally by sending HTTP requests to the local endpoint¹.

QUESTION 132

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. Use the plaintext key from the `GenerateDataKey` API and a symmetric encryption algorithm to encrypt the file.
- B. Write the plain text key from the `GenerateDataKey` API to disk for later use. Use the encrypted key from the `GenerateDataKey` API and a symmetric encryption algorithm to encrypt the file.
- C. Write the encrypted key from the `GenerateDataKey` API to disk for later use. Use the plaintext key from the `GenerateDataKey` API to encrypt the file by using the KMS `Encrypt` API
- D. Write the plain text key from the `GenerateDataKey` API to disk for later use. 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 key¹. 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 KMS¹.

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:

Call the `GenerateDataKey` API with the symmetric customer managed key ID and the desired length or specification of the data key. The API will return an encrypted data key and a plaintext data key. Write the encrypted data key to disk for later use. This will allow the developer to decrypt the data key and the PDF file later by using AWS KMS.

Use the plaintext data key and a symmetric encryption algorithm to encrypt the PDF file. The developer can use any standard encryption library or tool to perform this operation, such as OpenSSL

or AWS Encryption SDK.

Discard the plaintext data key from memory as soon as possible after using it. This will prevent unauthorized access or leakage of the data key.

QUESTION 133

A developer is optimizing an AWS Lambda function and wants to test the changes in production on a small percentage of all traffic. The Lambda function serves requests to a REST API in Amazon API Gateway. The developer needs to deploy their changes and perform a test in production without changing the API Gateway URL.

Which solution will meet these requirements?

- A. Define a function version for the currently deployed production Lambda function. Update the API Gateway endpoint to reference the new Lambda function version. Upload and publish the optimized Lambda function code. On the production API Gateway stage, define a canary release and set the percentage of traffic to direct to the canary release. Update the API Gateway endpoint to use the \$LATEST version of the Lambda function. Publish the API to the canary stage.
- B. Define a function version for the currently deployed production Lambda function. Update the API Gateway endpoint to reference the new Lambda function version. Upload and publish the optimized Lambda function code. Update the API Gateway endpoint to use the \$LATEST version of the Lambda function.
Deploy a new API Gateway stage.
- C. Define an alias on the \$LATEST version of the Lambda function. Update the API Gateway endpoint to reference the new Lambda function alias. Upload and
publish the optimized Lambda function code. On the production API Gateway stage, define a canary release and set the percentage of traffic to direct to the canary release. Update the API Gateway endpoint to use the \$LATEST version of the Lambda function. Publish to the canary stage.
- D. Define a function version for the currently deployed production Lambda function. Update the API Gateway endpoint to reference the new Lambda function
version. Upload and publish the optimized Lambda function code. Update the API Gateway endpoint to use the \$LATEST version of the Lambda function.
Deploy the API to the production API Gateway stage.

Answer: C

Explanation:

A Lambda alias is a pointer to a specific Lambda function version or another alias¹. A Lambda alias allows you to invoke different versions of a function using the same name¹. You can also split traffic between two aliases by assigning weights to them¹.

In this scenario, the developer needs to test their changes in production on a small percentage of all traffic without changing the API Gateway URL. To achieve this, the developer can follow these steps: Define an alias on the \$LATEST version of the Lambda function. This will create a new alias that points to the latest code of the function.

Update the API Gateway endpoint to reference the new Lambda function alias. This will make the API Gateway invoke the alias instead of a specific version of the function.

Upload and publish the optimized Lambda function code. This will update the \$LATEST version of the function with the new code.

On the production API Gateway stage, define a canary release and set the percentage of traffic to direct to the canary release. This will enable API Gateway to perform a canary deployment on a new API2. A canary deployment is a software development strategy in which a new version of an API is deployed for testing purposes, and the base version remains deployed as a production release for normal operations on the same stage2. The canary release receives a small percentage of API traffic and the production release takes up the rest2.

Update the API Gateway endpoint to use the \$LATEST version of the Lambda function. This will make the canary release invoke the latest code of the function, which contains the optimized changes.

Publish to the canary stage. This will deploy the changes to a subset of users for testing.

By using this solution, the developer can test their changes in production on a small percentage of all traffic without changing the API Gateway URL. The developer can also monitor and compare metrics between the canary and production releases, and promote or disable the canary as needed2.

QUESTION 134

A company has an application that stores data in Amazon RDS instances. The application periodically experiences surges of high traffic that cause performance problems.

During periods of peak traffic, a developer notices a reduction in query speed in all database queries.

The team's technical lead determines that a multi-threaded and scalable caching solution should be used to offload the heavy read traffic. The solution needs to improve performance.

Which solution will meet these requirements with the LEAST complexity?

- A. Use Amazon ElastiCache for Memcached to offload read requests from the main database.
- B. Replicate the data to Amazon DynamoDB. Set up a DynamoDB Accelerator (DAX) cluster.
- C. Configure the Amazon RDS instances to use Multi-AZ deployment with one standby instance. Offload read requests from the main database to the standby instance.
- D. Use Amazon ElastiCache for Redis to offload read requests from the main database.

Answer: A

Explanation:

Amazon ElastiCache for Memcached is a fully managed, multithreaded, and scalable in-memory keyvalue store that can be used to cache frequently accessed data and improve application performance1. By using Amazon ElastiCache for Memcached, the developer can reduce the load on the main database and handle high traffic surges more efficiently.

To use Amazon ElastiCache for Memcached, the developer needs to create a cache cluster with one or more nodes, and configure the application to store and retrieve data from the cache cluster2. The developer can use any of the supported Memcached clients to interact with the cache cluster3. The

developer can also use Auto Discovery to dynamically discover and connect to all cache nodes in a cluster⁴.

Amazon ElastiCache for Memcached is compatible with the Memcached protocol, which means that the developer can use existing tools and libraries that work with Memcached¹. Amazon ElastiCache for Memcached also supports data partitioning, which allows the developer to distribute data among multiple nodes and scale out the cache cluster as needed.

Using Amazon ElastiCache for Memcached is a simple and effective solution that meets the requirements with the least complexity. The developer does not need to change the database schema, migrate data to a different service, or use a different caching model. The developer can leverage the existing Memcached ecosystem and easily integrate it with the application.

QUESTION 135

An application that runs on AWS receives messages from an Amazon Simple Queue Service (Amazon SQS) queue and processes the messages in batches. The application sends the data to another SQS queue to be consumed by another legacy application. The legacy system can take up to 5 minutes to process some transaction data.

a. A developer wants to ensure that there are no out-of-order updates in the legacy system. The developer cannot alter the behavior of the legacy system.

Which solution will meet these requirements?

- A. Use an SQS FIFO queue. Configure the visibility timeout value.
- B. Use an SQS standard queue with a `SendMessageBatchRequestEntry` data type. Configure the `DelaySeconds` values.
- C. Use an SQS standard queue with a `SendMessageBatchRequestEntry` data type. Configure the visibility timeout value.
- D. Use an SQS FIFO queue. Configure the `DelaySeconds` value.

Answer: A

Explanation:

An SQS FIFO queue is a type of queue that preserves the order of messages and ensures that each message is delivered and processed only once¹. This is suitable for the scenario where the developer wants to ensure that there are no out-of-order updates in the legacy system.

The visibility timeout value is the amount of time that a message is invisible in the queue after a consumer receives it². This prevents other consumers from processing the same message simultaneously. If the consumer does not delete the message before the visibility timeout expires, the message becomes visible again and another consumer can receive it².

In this scenario, the developer needs to configure the visibility timeout value to be longer than the maximum processing time of the legacy system, which is 5 minutes. This will ensure that the message remains invisible in the queue until the legacy system finishes processing it and deletes it. This will prevent duplicate or out-of-order processing of messages by the legacy system.

QUESTION 136

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 source¹. When multiple SQS event sources are configured to a function, customers can control the maximum concurrent invokes of individual SQS event source¹. 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:

Find out the documented rate limits of the third-party API, which specify how many requests can be made in a given time period.

Configure maximum concurrency on the SQS event source based on the rate limits of the third-party API. This will limit the number of concurrent invokes by the SQS event source and prevent exceeding the rate limits of the third-party API.

Test and monitor the application performance and adjust the maximum concurrency value as needed.

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.

QUESTION 137

An online sales company is developing a serverless application that runs on AWS. The application uses an AWS Lambda function that calculates order success rates and stores the data in an Amazon

DynamoDB table. A developer wants an efficient way to invoke the Lambda function every 15 minutes.

Which solution will meet this requirement with the LEAST development effort?

- A. Create an Amazon EventBridge rule that has a rate expression that will run the rule every 15 minutes. Add the Lambda function as the target of the EventBridge rule.
- B. Create an AWS Systems Manager document that has a script that will invoke the Lambda function on Amazon EC2. Use a Systems Manager Run Command task to run the shell script every 15 minutes.
- C. Create an AWS Step Functions state machine. Configure the state machine to invoke the Lambda function execution role at a specified interval by using a Wait state. Set the interval to 15 minutes.
- D. Provision a small Amazon EC2 instance. Set up a cron job that invokes the Lambda function every 15 minutes.

Answer: A

Explanation:

The best solution for this requirement is option

A) Creating an Amazon EventBridge rule that has a rate expression that will run the rule every 15 minutes and adding the Lambda function as the target of the EventBridge rule is the most efficient way to invoke the Lambda function periodically. This solution does not require any additional resources or development effort, and it leverages the built-in scheduling capabilities of EventBridge.

QUESTION 138

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 applications 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.

Reference:

- 1: Amazon EKS User Guide, æDeploying applications to your Amazon EKS cluster ,
<https://docs.aws.amazon.com/eks/latest/userguide/deploying-applications.html>
- 2: Amazon ECR User Guide, æPushing an image ,
<https://docs.aws.amazon.com/AmazonECR/latest/userguide/docker-push-ecr-image.html>
- 3: Amazon EKS User Guide, æUpdating an Amazon EKS cluster ,
<https://docs.aws.amazon.com/eks/latest/userguide/update-cluster.html>

QUESTION 139

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 logging¹. Amazon EventBridge is a serverless event bus service that enables you to connect your applications with data from a variety of sources². 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 functions³. 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 AWS⁴. Kubernetes cron jobs are tasks that run periodically on a given schedule⁵. 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 cloud⁶. Crontab is a Linux utility that allows you to schedule commands or scripts to run automatically at a specified time or date⁷. 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 Cloud⁸. Batch jobs are units of work that can be submitted to job queues, where they are executed in parallel or sequentially on compute environments⁹. 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.

Reference:

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
-

QUESTION 140

An developer is building a serverless application by using the AWS Serverless Application Model (AWS SAM). The developer is currently testing the application in a development environment. When the application is nearly finished, the developer will need to set up additional testing and staging environments for a quality assurance team.

The developer wants to use a feature of the AWS SAM to set up deployments to multiple environments.

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

- A. Add a configuration file in TOML format to group configuration entries to every environment. Add a table for each testing and staging environment. Deploy updates to the environments by using the `sam deploy` command and the `--config-env` flag that corresponds to the each environment.
- B. Create additional AWS SAM templates for each testing and staging environment. Write a custom shell script that uses the `sam deploy` command and the `--template-file` flag to deploy updates to the environments.
- C. Create one AWS SAM configuration file that has default parameters. Perform updates to the testing and staging environments by using the `--parameter-overrides` flag in the AWS SAM CLI and the parameters that the updates will override.
- D. Use the existing AWS SAM template. Add additional parameters to configure specific attributes for the serverless function and database table resources that are in each environment. Deploy updates to the testing and staging environments by using the `sam deploy` command.

Answer: A

Explanation:

The correct answer is

A. Add a configuration file in TOML format to group configuration entries to every environment. Add a table for each testing and staging environment. Deploy updates to the environments by using the `sam deploy` command and the `--config-env` flag that corresponds to the each environment.

A) Add a configuration file in TOML format to group configuration entries to every environment. Add a table for each testing and staging environment. Deploy updates to the environments by using the `sam deploy` command and the `--config-env` flag that corresponds to the each environment. This is correct. This solution will meet the requirements with the least development effort, because it uses a feature of the AWS SAM CLI that supports a project-level configuration file that can be used to configure AWS SAM CLI command parameter values¹. The configuration file can have multiple environments, each with its own set of parameter values, such as stack name, region, capabilities,

and more². The developer can use the `--config-env` option to specify which environment to use when deploying the application³. This way, the developer can avoid creating multiple templates or scripts, or manually overriding parameters for each environment.

B) Create additional AWS SAM templates for each testing and staging environment. Write a custom shell script that uses the `sam deploy` command and the `--template-file` flag to deploy updates to the environments. This is incorrect. This solution will not meet the requirements with the least development effort, because it requires creating and maintaining multiple templates and scripts for each environment. This can introduce duplication, inconsistency, and complexity in the deployment process.

C) Create one AWS SAM configuration file that has default parameters. Perform updates to the testing and staging environments by using the `--parameter-overrides` flag in the AWS SAM CLI and the parameters that the updates will override. This is incorrect. This solution will not meet the requirements with the least development effort, because it requires manually specifying and overriding parameters for each environment every time the developer deploys the application. This can be error-prone, tedious, and inefficient.

D) Use the existing AWS SAM template. Add additional parameters to configure specific attributes for the serverless function and database table resources that are in each environment. Deploy updates to the testing and staging environments by using the `sam deploy` command. This is incorrect. This solution will not meet the requirements with the least development effort, because it requires modifying the existing template and adding complexity to the resource definitions for each environment. This can also make it difficult to manage and track changes across different environments.

Reference:

1: AWS SAM CLI configuration file - AWS Serverless Application Model

2: Configuration file basics - AWS Serverless Application Model

3: Specify a configuration file - AWS Serverless Application Model

QUESTION 141

A company notices that credentials that the company uses to connect to an external software as a service (SaaS) vendor are stored in a configuration file as plaintext.

The developer needs to secure the API credentials and enforce automatic credentials rotation on a quarterly basis.

Which solution will meet these requirements MOST securely?

A. Use AWS Key Management Service (AWS KMS) to encrypt the configuration file. Decrypt the configuration file when users make API calls to the SaaS vendor. Enable rotation.

B. Retrieve temporary credentials from AWS Security Token Service (AWS STS) every 15 minutes. Use the temporary credentials when users make API calls to the SaaS vendor.

C. Store the credentials in AWS Secrets Manager and enable rotation. Configure the API to have Secrets Manager access.

D. Store the credentials in AWS Systems Manager Parameter Store and enable rotation. Retrieve the credentials when users make API calls to the SaaS vendor.

Answer: C

Explanation:

Store the credentials in AWS Secrets Manager and enable rotation. Configure the API to have Secrets Manager access. This is correct. This solution will meet the requirements most securely, because it uses a service that is designed to store and manage secrets such as API credentials. AWS Secrets Manager helps you protect access to your applications, services, and IT resources by enabling you to rotate, manage, and retrieve secrets throughout their lifecycle¹. You can store secrets such as passwords, database strings, API keys, and license codes as encrypted values². You can also configure automatic rotation of your secrets on a schedule that you specify³. You can use the AWS SDK or CLI to retrieve secrets from Secrets Manager when you need them⁴. This way, you can avoid storing credentials in plaintext files or hardcoding them in your code.

QUESTION 142

A developer wants to deploy a new version of an AWS Elastic Beanstalk application. During deployment, the application must maintain full capacity and avoid service interruption. Additionally, the developer must minimize the cost of additional resources that support the deployment. Which deployment method should the developer use to meet these requirements?

- A. All at once
- B. Rolling with additional batch
- C. Blue/green
- D. Immutable

Answer: D

Explanation:

The immutable deployment method is the best option for this scenario, because it meets the requirements of maintaining full capacity, avoiding service interruption, and minimizing the cost of additional resources.

The immutable deployment method creates a new set of instances in a separate Auto Scaling group and deploys the new version of the application to them. Then, it swaps the new instances with the old ones and terminates the old instances. This way, the application maintains full capacity during the deployment and avoids any downtime. The cost of additional resources is also minimized, because the new instances are only created for a short time and then replaced by the old ones.

The other deployment methods do not meet all the requirements:

The all at once method deploys the new version to all instances simultaneously, which causes a short period of downtime and reduced capacity.

The rolling with additional batch method deploys the new version in batches, but for the first batch it creates new instances instead of using the existing ones. This increases the cost of additional resources and reduces the capacity of the original environment.

The blue/green method creates a new environment with a new set of instances and deploys the new version to them. Then, it swaps the URLs between the old and new environments. This method maintains full capacity and avoids service interruption, but it also increases the cost of additional resources significantly, because it duplicates the entire environment.

QUESTION 143

A developer is building a serverless application by using AWS Serverless Application Model (AWS SAM) on multiple AWS Lambda functions.

When the application is deployed, the developer wants to shift 10% of the traffic to the new deployment of the application for the first 10 minutes after deployment. If there are no issues, all traffic must switch over to the new version.

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

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

Answer: A

Explanation:

The AWS Serverless Application Model (AWS SAM) comes built-in with CodeDeploy to provide gradual AWS Lambda deployments¹. 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 version¹. The `AutoPublishAlias` property in AWS SAM allows AWS SAM to automatically create an alias that points to the updated version of the Lambda function¹. Therefore, option A is correct.

QUESTION 144

A company developed an API application on AWS by using Amazon CloudFront, Amazon API Gateway, and AWS Lambda.

a. The API has a

minimum of four requests every second. A developer notices that many API users run the same query by using the POST method. The developer wants to cache the POST request to optimize the API resources.

Which solution will meet these requirements?

- A. Configure the CloudFront cache. Update the application to return cached content based upon the default request headers.
- B. Override the cache method in the selected stage of API Gateway. Select the POST method.
- C. Save the latest request response in Lambda /tmp directory. Update the Lambda function to check the /tmp directory.
- D. Save the latest request in AWS Systems Manager Parameter Store. Modify the Lambda function to take the latest request response from Parameter Store.

Answer: B

Explanation:

Amazon API Gateway provides tools for creating and documenting web APIs that route HTTP requests to Lambda functions². You can secure access to your API with authentication and authorization controls. Your APIs can serve traffic over the internet or can be accessible only within your VPC². You can override the cache method in the selected stage of API Gateway². Therefore, option B is correct.

QUESTION 145

A company is building a compute-intensive application that will run on a fleet of Amazon EC2 instances. The application uses attached Amazon Elastic Block Store (Amazon EBS) volumes for storing data.

a. The Amazon EBS volumes will be created at time of initial deployment. The application will process sensitive information. All of the data must be encrypted. The solution should not impact the application's performance.

Which solution will meet these requirements?

- A. Configure the fleet of EC2 instances to use encrypted EBS volumes to store data.
- B. Configure the application to write all data to an encrypted Amazon S3 bucket.
- C. Configure a custom encryption algorithm for the application that will encrypt and decrypt all data.
- D. Configure an Amazon Machine Image (AMI) that has an encrypted root volume and store the data to ephemeral disks.

Answer: A

Explanation:

Amazon Elastic Block Store (Amazon EBS) provides block level storage volumes for use with Amazon EC2 instances¹. Amazon EBS encryption offers a straight-forward encryption solution for your EBS resources associated with your EC2 instances¹. When you create an encrypted EBS volume and attach it to a supported instance type, the following types of data are encrypted: Data at rest inside the volume, all data moving between the volume and the instance, all snapshots created from the volume, and all volumes created from those snapshots¹. Therefore, option A is correct.

QUESTION 146

A developer is creating a new REST API by using Amazon API Gateway and AWS Lambda

a. The development team tests the API and validates responses for the known use cases before deploying the API to the production environment.

The developer wants to make the REST API available for testing by using API Gateway locally.

Which AWS Serverless Application Model Command Line Interface (AWS SAM CLI) subcommand will meet these requirements?

- A. Sam local invoke
- B. Sam local generate-event
- C. Sam local start-lambda
- D. Sam local start-api

Answer: D

Explanation:

The AWS Serverless Application Model Command Line Interface (AWS SAM CLI) is a command-line tool for local development and testing of Serverless applications². The `sam local startapi` subcommand of AWS SAM CLI is used to simulate a REST API by starting a new local endpoint³. Therefore, option D is correct.

QUESTION 147

A developer is creating an AWS Lambda function that consumes messages from an Amazon Simple Queue Service (Amazon SQS) standard queue. The developer notices that the Lambda function processes some messages multiple times.

How should developer resolve this issue MOST cost-effectively?

- A. Change the Amazon SQS standard queue to an Amazon SQS FIFO queue by using the Amazon SQS message deduplication ID.
- B. Set up a dead-letter queue.
- C. Set the maximum concurrency limit of the AWS Lambda function to 1
- D. Change the message processing to use Amazon Kinesis Data Streams instead of Amazon SQS.

Answer: A

Explanation:

Amazon Simple Queue Service (Amazon SQS) is a fully managed queue service that allows you to decouple and scale for applications¹. Amazon SQS offers two types of queues: Standard and FIFO (First In First Out) queues¹. The FIFO queue uses the `messageDeduplicationId` property to treat messages with the same value as duplicate². Therefore, changing the Amazon SQS standard queue to an Amazon SQS FIFO queue using the Amazon SQS message deduplication ID can help resolve the issue of the Lambda function processing some messages multiple times. Therefore, option A is correct.

QUESTION 148

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 documentation. Create automated test scripts that use the `cdk local invoke` command to invoke the Lambda functions. Check the response. Document the test scripts for the other developers on the team. Update the CI/CD pipeline to run the test scripts.

B. Install a unit testing framework that reproduces the Lambda execution environment. Create sample events based on the Lambda documentation. Invoke the handler function by using a unit testing framework. Check the response. Document how to run the unit testing framework for the other developers on the team. Update the CI/CD pipeline to run the unit testing framework.

C. Install the AWS Serverless Application Model (AWS SAM) CLI tool. Use the `sam local generateevent` command to generate sample events for the automated tests. Create automated test scripts that use the `sam local invoke` command to invoke the Lambda functions. 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. Create sample events based on the Lambda documentation. Create a Docker container from the Node.js base image to invoke the Lambda functions. Check the response. Document how to run the Docker container for the other developers on the team. Update the CI/CD pipeline to run the Docker container.

Answer: C

Explanation:

The AWS Serverless Application Model Command Line Interface (AWS SAM CLI) is a command-line tool for local development and testing of Serverless applications³. The `sam local generateevent` command of AWS SAM CLI generates sample events for automated tests³. The `sam local invoke` command is used to invoke Lambda functions³. Therefore, option C is correct.

QUESTION 149

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. Create a new API. Import the OpenAPI file. Modify the new API to add request validation. Perform the tests. Modify the existing API to add request validation. Deploy the existing API to production.
- B. Modify the existing API to add request validation. Deploy the updated API to a new API Gateway stage. Perform the tests. Deploy the updated API to the API Gateway production stage.
- C. Create a new API. Add the necessary resources and methods, including new request validation. Perform the tests. Modify the existing API to add request validation. Deploy the existing API to production.
- D. Clone the existing API. Modify the new API to add request validation. Perform the tests. Modify the existing API to add request validation. 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 services¹. You can use API Gateway to perform basic validation of an API request before proceeding with the integration request¹. 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 Logs¹.

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 stage¹. 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 stage¹.

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 API¹.

QUESTION 150

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 activepassive failover configuration to meet companys 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. Enable secret replication to the secondary Region Update the application to use the Amazon Resource Name (ARN) based on the Region.
- B. Store credentials in AWS Systems Manager Parameter Store in the primary Region. Enable parameter replication to the secondary Region. Update the application to use the Amazon Resource

Name (ARN) based on the Region.

C. Store credentials in a config file. Upload the config file to an S3 bucket in the primary Region.

Enable Cross-Region Replication (CRR) to an S3 bucket in the secondary region. Update the application to access the config file from the S3 bucket based on the Region.

D. Store credentials in a config file. Upload the config file to an Amazon Elastic File System (Amazon EFS) file system. 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 monitoring¹. 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 purposes². 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 deployed³.

Reference:

AWS Secrets Manager

Replicating and sharing secrets

Using your own encryption keys

QUESTION 151

A developer is creating an AWS Lambda function that searches for items from an Amazon DynamoDB table that contains customer contact information- The DynamoDB table items have the customer's email_address as the partition key and additional properties such as customer_type, name, and job_title.

The Lambda function runs whenever a user types a new character into the customer_type text input

The developer wants the search to return partial matches of all the email_address property of a particular customer_type The developer does not want to recreate the DynamoDB table.

What should the developer do to meet these requirements?

A. Add a global secondary index (GSI) to the DynamoDB table with customer_type as the partition key and email_address as the sort key Perform a query operation on the GSI by using the begins_with key condition expression With the email_address property

B. Add a global secondary index (GSI) to the DynamoDB table With email_address as the partition key and customer_type as the sort key Perform a query operation on the GSI by using the begins_with key condition expression With the email_address property.

C. Add a local secondary index (LSI) to the DynamoDB table With customer_type as the partition key and email_address as the sort key Perform a query operation on the LSI by using the begins_with key

condition expression With the email_address property

D. Add a local secondary Index (LSI) to the DynamoDB table With job_title as the partition key and email_address as the sort key Perform a query operation on the LSI by using the begins_with key condition expression With the email_address property

Answer: A

Explanation:

Understand the Problem: The existing DynamoDB table has email_address as the partition key. Searching by customer_type requires a different data access pattern. We need an efficient way to query for partial matches on email_address based on customer_type.

Why Global Secondary Index (GSI):

GSIs allow you to define a different partition key and sort key from the main table, enabling new query patterns.

In this case, having customer_type as the GSI's partition key lets you group all emails with the same customer type together.

Using email_address as the sort key allows ordering within each customer type, facilitating the partial matching.

Querying the GSI:

You'll perform a query operation on the GSI, not the original table.

Use the begins_with key condition expression on the GSI's sort key (email_address) to find partial matches as the user types in the customer_type field.

Reference:

DynamoDB Global Secondary

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

DynamoDB Query Operation: [invalid URL removed]

Key Condition Expressions: [invalid URL removed]

QUESTION 152

A developer is deploying a company's application to Amazon EC2 instances The application generates gigabytes of data files each day The files are rarely accessed but the files must be available to the application's users within minutes of a request during the first year of storage The company must retain the files for 7 years.

How can the developer implement the application to meet these requirements MOST costeffectively?

- A. Store the files in an Amazon S3 bucket Use the S3 Glacier Instant Retrieval storage class Create an S3 Lifecycle policy to transition the files to the S3 Glacier Deep Archive storage class after 1 year
- B. Store the files in an Amazon S3 bucket. Use the S3 Standard storage class. Create an S3 Lifecycle policy to transition the files to the S3 Glacier Flexible Retrieval storage class after 1 year.
- C. Store the files on an Amazon Elastic Block Store (Amazon EBS) volume Use Amazon Data Lifecycle Manager (Amazon DLM) to create snapshots of the EBS volumes and to store those snapshots in Amazon S3

D. Store the files on an Amazon Elastic File System (Amazon EFS) mount. Configure EFS lifecycle management to transition the files to the EFS Standard-Infrequent Access (Standard-IA) storage class after 1 year.

Answer: A

Explanation:

Amazon S3 Glacier Instant Retrieval is an archive storage class that delivers the lowest-cost storage for long-lived data that is rarely accessed and requires retrieval in milliseconds. With S3 Glacier Instant Retrieval, you can save up to 68% on storage costs compared to using the S3 Standard-Infrequent Access (S3 Standard-IA) storage class, when your data is accessed once per quarter.

<https://aws.amazon.com/s3/storage-classes/glacier/instant-retrieval/>

Understanding Storage Requirements:

Files are large and infrequently accessed, but need to be available within minutes when requested in the first year.

Long-term (7-year) retention is required.

Cost-effectiveness is a top priority.

Why S3 Glacier Instant Retrieval:

Matches the retrieval requirements (access within minutes).

More cost-effective than S3 Standard for infrequently accessed data.

Simpler to use than traditional Glacier where retrievals take hours.

Why S3 Glacier Deep Archive:

Most cost-effective S3 storage class for long term archival.

Meets the 7-year retention requirement.

S3 Lifecycle Policy:

Automate the transition from Glacier Instant Retrieval to Glacier Deep Archive after one year.

Optimize costs by matching storage classes to access patterns.

Reference:

Amazon S3 Storage Classes: <https://aws.amazon.com/s3/storage-classes/>

S3 Glacier Instant Retrieval: [invalid URL removed]

S3 Glacier Deep Archive: [invalid URL removed]

S3 Lifecycle Policies: <https://docs.aws.amazon.com/AmazonS3/latest/userguide/object-lifecyclemgmt.html>

QUESTION 153

A developer is creating a serverless application that uses an AWS Lambda function. The developer will use AWS CloudFormation to deploy the application. The application will write logs to Amazon CloudWatch Logs. The developer has created a log group in a CloudFormation template for the application to use. The developer needs to modify the CloudFormation template to make the name of the log group available to the application at runtime.

Which solution will meet this requirement?

- A. Use the AWS::Include transform in CloudFormation to provide the log group's name to the application
- B. Pass the log group's name to the application in the user data section of the CloudFormation template.
- C. Use the CloudFormation template's Mappings section to specify the log group's name for the application.
- D. Pass the log group's Amazon Resource Name (ARN) as an environment variable to the Lambda function

Answer: D

Explanation:

CloudFormation and Lambda Environment Variables:

CloudFormation is an excellent tool to manage infrastructure as code, including the log group resource.

Lambda functions can access environment variables at runtime, making them a suitable way to pass configuration information like the log group ARN.

CloudFormation Template Modification:

In your CloudFormation template, define the log group resource.

In the Lambda function resource, add an Environment section:

YAML

Environment:

Variables:

LOG_GROUP_ARN: !Ref LogGroupName

Use code with caution.

content_copy

The !Ref intrinsic function retrieves the log group's ARN, which CloudFormation generates during stack creation.

Using the ARN in Your Lambda Function:

Within your Lambda code, access the LOG_GROUP_ARN environment variable.

Configure your logging library (e.g., Python's logging module) to send logs to the specified log group.

Reference:

AWS Lambda Environment

Variables: <https://docs.aws.amazon.com/lambda/latest/dg/configuration-envvars.html>

CloudFormation !Ref Intrinsic

Function: <https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/intrinsic-functionreference-ref.html>

QUESTION 154

A company has a web application that runs on Amazon EC2 instances with a custom Amazon Machine Image (AMI) The company uses AWS CloudFormation to provision the application The application runs in the us-east-1 Region, and the company needs to deploy the application to the uswest-

1 Region

An attempt to create the AWS CloudFormation stack in us-west-1 fails. An error message states that the AMI ID does not exist. A developer must resolve this error with a solution that uses the least amount of operational overhead

Which solution meets these requirements?

- A. Change the AWS CloudFormation templates for us-east-1 and us-west-1 to use an AWS AMI. Relaunch the stack for both Regions.
- B. Copy the custom AMI from us-east-1 to us-west-1. Update the AWS CloudFormation template for us-west-1 to refer to AMI ID for the copied AMI Relaunch the stack
- C. Build the custom AMI in us-west-1 Create a new AWS CloudFormation template to launch the stack in us-west-1 with the new AMI ID
- D. Manually deploy the application outside AWS CloudFormation in us-west-1.

Answer: B

Explanation:

Problem: CloudFormation can't find the custom AMI in the target region (us-west-1) because AMIs are region-specific.

Copying AMIs:

AMIs can be copied across regions, maintaining their configuration.

This approach minimizes operational overhead as the existing CloudFormation template can be reused with a minor update.

Updating the Template:

Modify the CloudFormation template in us-west-1 to reference the newly copied AMI's ID in that region.

Reference:

Copying AMIs: <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/CopyingAMIs.html>

CloudFormation Templates and AMIs: [invalid URL removed]

QUESTION 155

A developer is working on a web application that uses Amazon DynamoDB as its data store The application has two DynamoDB tables one table that is named artists and one table that is named songs The artists table has artistName as the partition key. The songs table has songName as the partition key and artistName as the sort key

The table usage patterns include the retrieval of multiple songs and artists in a single database operation from the webpage. The developer needs a way to retrieve this information with minimal network traffic and optimal application performance.

Which solution will meet these requirements'?

- A. Perform a BatchGetItem operation that returns items from the two tables. Use the list of songName artistName keys for the songs table and the list of artistName key for the artists table.

- B. Create a local secondary index (LSI) on the songs table that uses artistName as the partition key
Perform a query operation for each artistName on the songs table that filters by the list of songName
Perform a query operation for each artistName on the artists table
- C. Perform a BatchGetItem operation on the songs table that uses the songName/artistName keys.
Perform a BatchGetItem operation on the artists table that uses artistName as the key.
- D. Perform a Scan operation on each table that filters by the list of songName/artistName for the songs table and the list of artistName in the artists table.

Answer: A

Explanation:

Scenario: Application needs to fetch songs and artists efficiently in a single operation.

BatchGetItem: This DynamoDB operation retrieves multiple items across different tables based on their primary keys in a single request.

Optimized for Request Batching: This approach reduces network traffic compared to performing multiple queries individually.

Data Modeling: The songs table is designed appropriately for this access pattern using artistName as the sort key.

Reference:

Amazon DynamoDB

BatchGetItem: https://docs.aws.amazon.com/amazondynamodb/latest/APIReference/API_BatchGetItem.html

QUESTION 156

A data visualization company wants to strengthen the security of its core applications. The applications are deployed on AWS across its development staging, pre-production, and production environments. The company needs to encrypt all of its stored sensitive credentials. The sensitive credentials need to be automatically rotated. A version of the sensitive credentials need to be stored for each environment.

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

- A. Configure AWS Secrets Manager versions to store different copies of the same credentials across multiple environments.
- B. Create a new parameter version in AWS Systems Manager Parameter Store for each environment. Store the environment-specific credentials in the parameter version.
- C. Configure the environment variables in the application code. Use different names for each environment type.
- D. Configure AWS Secrets Manager to create a new secret for each environment type. Store the environment-specific credentials in the secret.

Answer: D

Explanation:

Secrets Management: AWS Secrets Manager is designed specifically for storing and managing sensitive credentials.

Environment Isolation: Creating separate secrets for each environment (development, staging, etc.) ensures clear separation and prevents accidental leaks.

Automatic Rotation: Secrets Manager provides built-in rotation capabilities, enhancing security posture.

Versioning: Tracking changes to secrets is essential for auditing and compliance.

Reference:

AWS Secrets Manager: <https://aws.amazon.com/secrets-manager/>

Secrets Manager

Rotation: <https://docs.aws.amazon.com/secretsmanager/latest/userguide/rotating-secrets.html>

QUESTION 157

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. Create a table in the database for parameter configuration. Migrate the parameters that the application is modifying from Parameter Store to the configuration table.
- D. Modify the CloudFormation stack policy to deny updates on Parameter Store parameters.

Answer: A

Explanation:

Problem: CloudFormation updates reset Parameter Store parameters, disrupting application behavior.

Deletion Policy: CloudFormation has a deletion policy that controls resource behavior when a stack is deleted or updated. The 'Retain' policy instructs CloudFormation to preserve a resource's current state.

Least Development Effort: This solution involves a simple CloudFormation template modification,

requiring minimal code changes.

Reference:

CloudFormation Deletion

Policies: <https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/aws-attributedeletionpolicy.html>

QUESTION 158

A company has built an AWS Lambda function to convert large image files into output files that can be used in a third-party viewer application. The company recently added a new module to the function to improve the output of the generated files. However, the new module has increased the bundle size and has increased the time that is needed to deploy changes to the function code. How can a developer increase the speed of the Lambda function deployment?

- A. Use AWS CodeDeploy to deploy the function code
- B. Use Lambda layers to package and load dependencies.
- C. Increase the memory size of the function.
- D. Use Amazon S3 to host the function dependencies

Answer: B

Explanation:

Problem: Large bundle size increases Lambda deployment time.

Lambda Layers: Layers let you package dependencies separately from your function code. This optimizes the deployment package, making updates faster.

Modularization: Breaking down dependencies into layers improves code organization and reusability.

Reference:

AWS Lambda Layers: <https://docs.aws.amazon.com/lambda/latest/dg/configuration-layers.html>

QUESTION 159

A developer creates a static website for their department. The developer deploys the static assets for the website to an Amazon S3 bucket and serves the assets with Amazon CloudFront. The developer uses origin access control (OAC) on the CloudFront distribution to access the S3 bucket. The developer notices users can access the root URL and specific pages but cannot access directories without specifying a file name. For example, `/products/index.html` works, but `/products` returns an error. The developer needs to enable accessing directories without specifying a file name without exposing the S3 bucket publicly. Which solution will meet these requirements?

- A. Update the CloudFront distribution's settings to `index.html` as the default root object is set
- B. Update the Amazon S3 bucket settings and enable static website hosting. Specify `index.html` as the Index document. Update the S3 bucket policy to enable access. Update the CloudFront distribution's origin to use the S3 website endpoint

- C. Create a CloudFront function that examines the request URL and appends index.html when directories are being accessed Add the function as a viewer request CloudFront function to the CloudFront distribution's behavior.
- D. Create a custom error response on the CloudFront distribution with the HTTP error code set to the HTTP 404 Not Found response code and the response page path to /index.html Set the HTTP response code to the HTTP 200 OK response code

Answer: B

Explanation:

Problem: Directory access without file names fails.

S3 Static Website Hosting:

Configuring S3 as a static website enables automatic serving of index.html for directory requests.

Bucket policies ensure correct access permissions.

Updating the CloudFront origin simplifies routing.

Avoiding Public Exposure: The S3 website endpoint allows CloudFront to access content without making the bucket public.

Reference:

S3 Static Website

Hosting: <https://docs.aws.amazon.com/AmazonS3/latest/userguide/WebsiteHosting.html>

QUESTION 160

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 topic. Create an Amazon EventBridge schedule to invoke the Lambda function every 15 minutes
- B. 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
- C. Launch an Amazon EC2 instance that includes a script to filter events from CloudTrail if a role was created without CloudFormation. Configure the script to publish to the SNS topic. Create a cron job to run the script on the EC2 instance every 15 minutes.
- D. 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:

EventBridge (formerly CloudWatch Events) is the ideal service for real-time event monitoring.

CloudTrail logs IAM role creation.

EventBridge rules can filter CloudTrail events and trigger SNS notifications instantly.

QUESTION 161

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: A

Explanation:

SQS Delivery Behavior: Standard SQS queues guarantee at-least-once delivery, meaning messages may be processed more than once. This can lead to duplicate emails in this scenario.

Visibility Timeout: If the visibility timeout on the SQS queue is too short, a message might become visible for another consumer before the first Lambda function finishes processing it. This can also lead to duplicates.

Reference:

Amazon SQS Delivery Semantics: [invalid URL removed]

Amazon SQS Visibility

Timeout: <https://docs.aws.amazon.com/AWSSimpleQueueService/latest/SQSDeveloperGuide/sqsvisibility-timeout.html>

QUESTION 162

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:

CloudFormation and Dynamic Reference: The DefinitionSubstitutions property in CloudFormation allows you to pass values into Step Functions state machines at runtime.

Cost-Effectiveness: This solution is cost-effective as it leverages CloudFormation's built-in capabilities, avoiding the need for additional services like Secrets Manager or AppConfig.

Reference:

AWS Step Functions State

Machine: <https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/aws-resource-stepfunctions-statemachine.html>

CloudFormation DefinitionSubstitutions: <https://github.com/aws-cloudformation/awscloudformation-resource-providers-stepfunctions/issues>

QUESTION 163

A developer created an AWS Lambda function that performs a series of operations that involve multiple AWS services. The function's duration time is higher than normal. To determine the cause of the issue, the developer must investigate traffic between the services without changing the function code.

Which solution will meet these requirements?

- A. Enable AWS X-Ray active tracing in the Lambda function. Review the logs in X-Ray.
- B. Configure AWS CloudTrail. View the trail logs that are associated with the Lambda function.
- C. Review the AWS Config logs in Amazon CloudWatch.
- D. Review the Amazon CloudWatch logs that are associated with the Lambda function.

Answer: A

Explanation:

Tracing Distributed Systems: AWS X-Ray is designed to trace requests across services, helping identify bottlenecks in distributed applications like this one.

No Code Changes: Enabling X-Ray tracing often requires minimal code changes, meeting the requirement.

Identifying Bottlenecks: Analyzing X-Ray traces and logs will reveal latency in communications between different AWS services, leading to the high duration time.

Reference:

AWS X-Ray: <https://aws.amazon.com/xray/>

X-Ray and Lambda: <https://docs.aws.amazon.com/xray/latest/devguide/xray-services-lambda.html>

QUESTION 164

A developer designed an application on an Amazon EC2 instance. The application makes API requests to objects in an Amazon S3 bucket.

Which combination of steps will ensure that the application makes the API requests in the MOST secure manner? (Select TWO.)

- A. Create an IAM user that has permissions to the S3 bucket. Add the user to an IAM group.
- B. Create an IAM role that has permissions to the S3 bucket.
- C. Add the IAM role to an instance profile. Attach the instance profile to the EC2 instance.
- D. Create an IAM role that has permissions to the S3 bucket. Assign the role to an IAM group.
- E. Store the credentials of the IAM user in the environment variables on the EC2 instance.

Answer: BC

Explanation:

IAM Roles for EC2: IAM roles are the recommended way to provide AWS credentials to applications running on EC2 instances. Here's how this works:

You create an IAM role with the necessary permissions to access the target S3 bucket.

You create an instance profile and associate the IAM role with this profile.

When launching the EC2 instance, you attach this instance profile.

Temporary Security Credentials: When the application on the EC2 instance needs to access S3, it doesn't directly use access keys. Instead, the AWS SDK running on the instance retrieves temporary security credentials associated with the role. These are rotated automatically by AWS.

Reference:

IAM Roles for Amazon

EC2: https://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles_use_switch-role-ec2.html

Temporary Security

Credentials: https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_temp.html

QUESTION 165

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:

Centralized Logging Benefits: Centralized logging is essential for operational visibility in scalable systems, especially those using multiple EC2 instances like our e-commerce website. CloudWatch provides this capability, along with other monitoring features.

CloudWatch Agent: This is the best way to send custom application logs from EC2 instances to CloudWatch. Here's the process:

Install the CloudWatch agent on each EC2 instance.

Configure the agent with a configuration file, specifying:

Which log files to collect.

The format in which to send logs to CloudWatch (e.g., JSON).

The specific CloudWatch Logs log group and log stream for these logs.

Viewing and Analyzing Logs: Once the agent is pushing logs, use the CloudWatch Logs console or API:

View and search the logs across all instances.

Set up alarms based on log events.

Use CloudWatch Logs Insights for sophisticated queries and analysis.

Reference:

Amazon CloudWatch

Logs: <https://docs.aws.amazon.com/AmazonCloudWatch/latest/logs/WhatIsCloudWatchLogs.html>

Unified CloudWatch

Agent: <https://docs.aws.amazon.com/AmazonCloudWatch/latest/logs/AgentReference.html>

CloudWatch Logs

Insights: <https://docs.aws.amazon.com/AmazonCloudWatch/latest/logs/AnalyzingLogData.html>

QUESTION 166

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

Which solution will meet this requirement?

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

Answer: B

Explanation:

API Gateway Stage Variables: These are designed for configuring dynamic values for your APIs in different deployment stages (dev, test, prod). Here's how to use them for third-party endpoints:

In the API Gateway console, access the "Stages" section of your API.

For each stage, create a stage variable named something like `thirdPartyEndpoint`.

Set the value of this variable to the actual endpoint URL for that specific environment.

When configuring API requests within your API Gateway method, reference this endpoint using `${stageVariables.thirdPartyEndpoint}`.

Why Stage Variables Excel Here:

Environment Isolation: This approach keeps the endpoint configuration specific to each deployment stage, ensuring the right endpoints are used during development, testing, and production cycles.

Ease of Management: You manage the endpoints directly through the API Gateway console without additional infrastructure.

Reference:

Amazon API Gateway Stage

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

QUESTION 167

A company is creating an application that processes csv files from Amazon S3 A developer has created an S3 bucket The developer has also created an AWS Lambda function to process the csv files from the S3 bucket

Which combination of steps will invoke the Lambda function when a csv file is uploaded to Amazon S3? (Select TWO.)

- A. Create an Amazon EventBridge rule Configure the rule with a pattern to match the S3 object created event
- B. Schedule an Amazon EventBridge rule to run a new Lambda function to scan the S3 bucket.
- C. Add a trigger to the existing Lambda function. Set the trigger type to EventBridge Select the Amazon EventBridge rule.
- D. Create a new Lambda function to scan the S3 bucket for recently added S3 objects
- E. Add S3 Lifecycle rules to invoke the existing Lambda function

Answer: AE

Explanation:

Amazon EventBridge: A service that reacts to events from various AWS sources, including S3. Rules define which events trigger actions (like invoking Lambda functions).

S3 Object Created Events: EventBridge can detect these, providing seamless integration for automated CSV processing.

S3 Lifecycle Rules: Allow for actions based on object age or prefixes. These can directly trigger Lambda functions for file processing.

Reference:

Amazon EventBridge Documentation: <https://docs.aws.amazon.com/eventbridge/>

Working with S3 Event

Notifications: <https://docs.aws.amazon.com/AmazonS3/latest/userguide/EventNotifications.html>

S3 Lifecycle Configuration: <https://docs.aws.amazon.com/AmazonS3/latest/userguide/objectlifecycle-mgmt.html>

QUESTION 168

A developer is creating an AWS Lambda function in VPC mode. An Amazon S3 event will invoke the Lambda function when an object is uploaded into an S3 bucket. The Lambda function will process the object and produce some analytic results that will be recorded into a file. Each processed object will also generate a log entry that will be recorded into a file.

Other Lambda functions, AWS services, and on-premises resources must have access to the result files and log file. Each log entry must also be appended to the same shared log file. The developer needs a solution that can share files and append results into an existing file.

Which solution should the developer use to meet these requirements?

- A. Create an Amazon Elastic File System (Amazon EFS) file system. Mount the EFS file system in Lambda. Store the result files and log file in the mount point. Append the log entries to the log file.
- B. Create an Amazon Elastic Block Store (Amazon EBS) Multi-Attach enabled volume. Attach the EBS volume to all Lambda functions. Update the Lambda function code to download the log file, append the log entries, and upload the modified log file to Amazon EBS.
- C. Create a reference to the /tmp local directory. Store the result files and log file by using the directory reference. Append the log entry to the log file.
- D. Create a reference to the /opt storage directory. Store the result files and log file by using the directory reference. Append the log entry to the log file.

Answer: A

Explanation:

Amazon EFS: A network file system (NFS) providing shared, scalable storage across multiple Lambda functions and other AWS resources.

Lambda Mounting: EFS file systems can be mounted within Lambda functions to access a shared storage space.

Log Appending: EFS supports appending data to existing files, making it ideal for the log file scenario.

Reference:

Amazon EFS Documentation: <https://docs.aws.amazon.com/efs/>

Using Amazon EFS with AWS Lambda: <https://docs.aws.amazon.com/lambda/latest/dg/serviceefs.html>

QUESTION 169

A company hosts its application on AWS. The application runs on an Amazon Elastic Container Service (Amazon ECS) cluster that uses AWS Fargate. The cluster runs behind an Application Load Balancer. The application stores data in an Amazon Aurora database. A developer encrypts and manages database credentials inside the application.

The company wants to use a more secure credential storage method and implement periodic credential rotation.

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

- A. Migrate the secret credentials to Amazon RDS parameter groups. Encrypt the parameter by using an AWS Key Management Service (AWS KMS) key. Turn on secret rotation. Use IAM policies and roles to grant AWS KMS permissions to access Amazon RDS.
- B. Migrate the credentials to AWS Systems Manager Parameter Store. Encrypt the parameter by using an AWS Key Management Service (AWS KMS) key. Turn on secret rotation. Use IAM policies and roles to grant Amazon ECS Fargate permissions to access to AWS Secrets Manager.
- C. Migrate the credentials to ECS Fargate environment variables. Encrypt the credentials by using an AWS Key Management Service (AWS KMS) key. Turn on secret rotation. Use IAM policies and roles to grant Amazon ECS Fargate permissions to access to AWS Secrets Manager.
- D. Migrate the credentials to AWS Secrets Manager. Encrypt the credentials by using an AWS Key Management Service (AWS KMS) key. Turn on secret rotation. Use IAM policies and roles to grant Amazon ECS Fargate permissions to access to AWS Secrets Manager by using keys.

Answer: D

Explanation:

Secrets Management: AWS Secrets Manager is designed specifically for storing and managing sensitive credentials.

Built-in Rotation: Secrets Manager provides automatic secret rotation functionality, enhancing security posture significantly.

IAM Integration: IAM policies and roles grant fine-grained access to ECS Fargate, ensuring the principle of least privilege.

Reduced Overhead: This solution centralizes secrets management and automates rotation, reducing operational overhead compared to the other options.

Reference:

AWS Secrets Manager: <https://aws.amazon.com/secrets-manager/>
Secrets Manager

Rotation: <https://docs.aws.amazon.com/secretsmanager/latest/userguide/rotating-secrets.html>

QUESTION 170

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:

HTTP Status Codes: Each HTTP status code has a specific meaning in RESTful APIs.

HTTP 405 (Method Not Allowed): Indicates that the request method (e.g., POST) is not supported for the specified resource.

HTTP 401 (Unauthorized): Represents a failure to authenticate, which is the appropriate response for invalid login credentials.

Reference:

HTTP Status Codes: <https://developer.mozilla.org/en-US/docs/Web/HTTP/Status>

QUESTION 171

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:

Lambda Concurrency: The 'maximum concurrency' setting in event source mappings controls the maximum number of simultaneous invocations Lambda allows for that specific source.

Prioritizing Queues: Setting a lower maximum concurrency for the 'high priority queue' ensures it's processed first while allowing more concurrent invocations from the 'low priority queue'.

Batching: Batch size settings affect the number of messages Lambda retrieves from a queue per invocation, which is less relevant to the prioritization requirement.

Reference:

Lambda Event Source Mappings: <https://docs.aws.amazon.com/lambda/latest/dg/invokeeventsourcemapping.html>

Lambda Concurrency: <https://docs.aws.amazon.com/lambda/latest/dg/configurationconcurrency.html>

QUESTION 172

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/metadata/ami/`.
- D. Check the hosts file of the operating system.

Answer: A

Explanation:

Instance Metadata Service: EC2 instances have access to an internal metadata service. It provides instance-specific information like instance ID, security groups, and public IP address.

Accessing Metadata:

Make an HTTP GET request to the base URL: `http://169.254.169.254/latest/meta-data/`

You'll get a list of available categories. The public IPv4 address is under `public-ipv4`.

Reference:

Instance Metadata and User

Data: <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/instancedata-data-retrieval.html>

QUESTION 173

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. Configure the filter to send an SNS notification when the number of errors exceeds the defined threshold within a 5-minute period.
- C. 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
- D. Create a CloudWatch metric filter to match the application error pattern in the log data. Set up a CloudWatch alarm based on the new custom metric. 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:

CloudWatch for Log Analysis: CloudWatch is the best fit here because logs are already centralized.

Here's the process:

Metric Filter: Create a metric filter on the CloudWatch Logs log group. Design a pattern to specifically identify application error messages.

Custom Metric: This filter generates a new custom CloudWatch metric (e.g., ApplicationErrors). This metric tracks the error count.

CloudWatch Alarm: Create an alarm on the ApplicationErrors metric. Configure the alarm with your desired threshold and a 5-minute evaluation period.

SNS Action: Set the alarm to trigger an SNS notification when it enters the alarm state.

Reference:

CloudWatch Metric

Filters: <https://docs.aws.amazon.com/AmazonCloudWatch/latest/logs/MonitoringLogData.html>

CloudWatch

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

QUESTION 174

A developer is creating a service that uses an Amazon S3 bucket for image uploads. The service will use an AWS Lambda function to create a thumbnail of each image Each time an image is uploaded the service needs to send an email notification and create the thumbnail The developer needs to configure the image processing and email notifications setup.

Which solution will meet these requirements?

- A. Create an Amazon Simple Notification Service (Amazon SNS) topic Configure S3 event notifications with a destination of the SNS topic Subscribe the Lambda function to the SNS topic Create an email notification subscription to the SNS topic
- B. Create an Amazon Simple Notification Service (Amazon SNS) topic. Configure S3 event notifications with a destination of the SNS topic. Subscribe the Lambda function to the SNS topic. Create an Amazon Simple Queue Service (Amazon SQS) queue Subscribe the SQS queue to the SNS topic Create an email notification subscription to the SQS queue.
- C. Create an Amazon Simple Queue Service (Amazon SQS) queue Configure S3 event notifications with a destination of the SQS queue Subscribe the Lambda function to the SQS queue Create an email notification subscription to the SQS queue.
- D. Create an Amazon Simple Queue Service (Amazon SQS) queue. Send S3 event notifications to Amazon EventBridge. Create an EventBridge rule that runs the Lambda function when images are uploaded to the S3 bucket Create an EventBridge rule that sends notifications to the SQS queue Create an email notification subscription to the SQS queue

Answer: A

Explanation:

SNS as a Fan-out Mechanism: SNS is perfect for triggering multiple actions from a single event (here, the image upload).

Workflow:

SNS Topic: Create an SNS topic that will be the central notification point.

S3 Event Notification: Configure the S3 bucket to send 'Object Created' event notifications to the SNS topic.

Lambda Subscription: Subscribe your thumbnail-creating Lambda function to the SNS topic.

Email Subscription: Subscribe an email address to the SNS topic to trigger notifications.

Reference:

S3 Event

Notifications: <https://docs.aws.amazon.com/AmazonS3/latest/userguide/EventNotifications.html>

SNS Subscriptions: <https://docs.aws.amazon.com/sns/latest/dg/SNSMobilePush.html>

QUESTION 175

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. Install the X-Ray SDK for all the services that the application uses

- C. Enable X-Ray data aggregation in Amazon CloudWatch Logs for all the services that the application uses
- D. Increase the X-Ray service map timeout value in the X-Ray console

Answer: B

Explanation:

AWS X-Ray SDK: The primary way to enable X-Ray tracing within applications. The SDK sends data about requests and subsegments to the X-Ray daemon for service map generation.

Instrumenting All Services: To visualize a complete microservice architecture on the service map, each relevant service must include the X-Ray SDK.

Reference:

AWS X-Ray Documentation: <https://docs.aws.amazon.com/xray/>

X-Ray SDK for Python: <https://docs.aws.amazon.com/xray/latest/devguide/xray-sdk-python.html>

QUESTION 176

A company has a social media application that receives large amounts of traffic User posts and interactions are continuously updated in an Amazon RDS database The data changes frequently, and the data types can be complex The application must serve read requests with minimal latency The application's current architecture struggles to deliver these rapid data updates efficiently The company needs a solution to improve the application's performance. Which solution will meet these requirements'?

- A. Use Amazon DynamoDB Accelerator (DAX) in front of the RDS database to provide a caching layer for the high volume of rapidly changing data
- B. Set up Amazon S3 Transfer Acceleration on the RDS database to enhance the speed of data transfer from the databases to the application.
- C. Add an Amazon CloudFront distribution in front of the RDS database to provide a caching layer for the high volume of rapidly changing data
- D. Create an Amazon ElastiCache for Redis cluster. Update the application code to use a writethrough caching strategy and read the data from Redis.

Answer: D

Explanation:

Amazon ElastiCache for Redis: An in-memory data store known for extremely low latency, ideal for caching frequently accessed, complex data.

Write-Through Caching: Ensures that data is always consistent between the cache and the database.

Writes go to both Redis and RDS.

Performance Gains: Redis handles reads with minimal latency, offloading the RDS database and improving the application's responsiveness.

Reference:

QUESTION 177

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 properties. Retrieve the secrets during the application startup. Reference the secrets in the application code. Write a script to rotate the secrets saved as environment variables.
- D. 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: Built for managing secrets, providing encryption, automatic rotation, and access control.

Customer Master Key (CMK): Provides an extra layer of control over encryption through AWS KMS.

Automatic Rotation: Enhances security by regularly changing the secret.

User Data Script: Allows secrets retrieval at instance startup and sets them as environment variables for seamless use within the application.

Reference:

AWS Secrets Manager Documentation: <https://docs.aws.amazon.com/secretsmanager/>

AWS KMS Documentation: <https://docs.aws.amazon.com/kms/>

User Data for EC2 Instances: <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/userdata.html>

QUESTION 178

A company is using Amazon API Gateway to invoke a new AWS Lambda function. The company has

Lambda function versions in its PROD and DEV environments. In each environment, there is a Lambda function alias pointing to the corresponding Lambda function version API Gateway has one stage that is configured to point at the PROD alias
The company wants to configure API Gateway to enable the PROD and DEV Lambda function versions to be simultaneously and distinctly available
Which solution will meet these requirements?

- A. Enable a Lambda authorizer for the Lambda function alias in API Gateway Republish PROD and create a new stage for DEV Create API Gateway stage variables for the PROD and DEV stages. Point each stage variable to the PROD Lambda authorizer to the DEV Lambda authorizer.
- B. Set up a gateway response in API Gateway for the Lambda function alias. Republish PROD and create a new stage for DEV. Create gateway responses in API Gateway for PROD and DEV Lambda aliases
- C. Use an environment variable for the Lambda function alias in API Gateway. Republish PROD and create a new stage for development. Create API gateway environment variables for PROD and DEV stages. Point each stage variable to the PROD Lambda function alias to the DEV Lambda function alias.
- D. Use an API Gateway stage variable to configure the Lambda function alias Republish PROD and create a new stage for development Create API Gateway stage variables for PROD and DEV stages Point each stage variable to the PROD Lambda function alias and to the DEV Lambda function alias

Answer: D

Explanation:

API Gateway Stages: Stages in API Gateway represent distinct environments (like PROD and DEV) allowing different configurations.

Stage Variables: Stage variables store environment-specific information, including Lambda function aliases.

Ease of Management: This solution offers a straightforward way to manage different Lambda function versions across environments.

Reference:

API Gateway Stages: <https://docs.aws.amazon.com/apigateway/latest/developerguide/set-upstages.html>

API Gateway Stage

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

QUESTION 179

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

Answer: D

Explanation:

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

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

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

Reference:

Amazon API Gateway Mock

Integrations: <https://docs.aws.amazon.com/apigateway/latest/developerguide/how-to-mockintegration.html>

QUESTION 180

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:

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

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

Reference:

AWS::Lambda::Function

Resource <https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/aws-resource-lambda-function.html>

QUESTION 181

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

Which CodeDeploy predefined configuration will meet these requirements?

- A. CodeDeployDefault ECSCanary10Percent15Minutes
- B. CodeDeployDefault LambdaCanary10Percent5Minutes
- C. CodeDeployDefault LambdaCanary10Percent15Minutes
- D. CodeDeployDefault ECSLinear10PercentEvery1 Minutes

Answer: A

Explanation:

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

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

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

Reference:

AWS CodeDeploy Deployment

Configurations: <https://docs.aws.amazon.com/codedeploy/latest/userguide/deploymentconfigurations-create.html>

QUESTION 182

A developer is using AWS Step Functions to automate a workflow The workflow defines each step as

an AWS Lambda function task The developer notices that runs of the Step Functions state machine fail in the GetResource task with either an UlegalArgumentException error or a TooManyRequestsException error

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

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

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

Answer: C

Explanation:

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

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

Configure the retrier:

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

IntervalSeconds: Set to 10 for the desired retry delay.

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

Error Handling:

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

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

'IllegalArgumentException' causes error propagation as intended.

Reference:

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

QUESTION 183

An Amazon Simple Queue Service (Amazon SQS) queue serves as an event source for an AWS Lambda function. In the SQS queue, each item corresponds to a video file that the Lambda function must convert to a smaller resolution. The Lambda function is timing out on longer video files, but the Lambda function's timeout is already configured to its maximum value. What should a developer do to avoid the timeouts without additional code changes?

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

Answer: B

Explanation:

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

How It Helps:

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

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

Reference:

SQS Visibility

Timeout: <https://docs.aws.amazon.com/AWSSimpleQueueService/latest/SQSDeveloperGuide/sqsvisibility-timeout.html>

QUESTION 184

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

Answer: D

Explanation:

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

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

Reference:

DynamoDB Server-Side

Encryption: <https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/Encryption>

QUESTION 185

A developer is creating an AWS Lambda function. The Lambda function needs an external library to connect to a third-party solution. The external library is a collection of files with a total size of 100 MB. The developer needs to make the external library available to the Lambda execution environment and reduce the Lambda package space. Which solution will meet these requirements with the LEAST operational overhead?

- A. Create a Lambda layer to store the external library. Configure the Lambda function to use the layer.
- B. Create an Amazon S3 bucket. Upload the external library into the S3 bucket. Mount the S3 bucket folder in the Lambda function. Import the library by using the proper folder in the mount point.
- C. Load the external library to the Lambda function's /tmp directory during deployment of the Lambda package. Import the library from the /tmp directory.
- D. Create an Amazon Elastic File System (Amazon EFS) volume. Upload the external library to the EFS volume. Mount the EFS volume in the Lambda function. Import the library by using the proper folder in the mount point.

Answer: A

Explanation:

Lambda Layers: These are designed to package dependencies that you can share across functions.

How to Use:

Create a layer, upload your 100MB library as a zip.

Attach the layer to your function.

In your function code, import the library from the standard layer path.

Reference:

Lambda Layers: <https://docs.aws.amazon.com/lambda/latest/dg/configuration-layers.html>

QUESTION 186

A company built an online event platform. For each event, the company organizes quizzes and generates leaderboards that are based on the quiz scores. The company stores the leaderboard data in Amazon DynamoDB and retains the data for 30 days after an event is complete. The company then uses a scheduled job to delete the old leaderboard data.

The DynamoDB table is configured with a fixed write capacity. During the months when many events occur, the DynamoDB write API requests are throttled when the scheduled delete job runs. A developer must create a long-term solution that deletes the old leaderboard data and optimizes write throughput. Which solution meets these requirements?

- A. Configure a TTL attribute for the leaderboard data
- B. Use DynamoDB Streams to schedule and delete the leaderboard data
- C. Use AWS Step Functions to schedule and delete the leaderboard data.
- D. Set a higher write capacity when the scheduled delete job runs

Answer: A

Explanation:

DynamoDB TTL (Time-to-Live): A native feature that automatically deletes items after a specified expiration time.

Efficiency: Eliminates the need for scheduled deletion jobs, optimizing write throughput by avoiding potential throttling conflicts.

Seamless Integration: TTL works directly within DynamoDB, requiring minimal development overhead.

Reference:

DynamoDB TTL

Documentation: <https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/TTL.html>

QUESTION 187

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:

AWS STS AssumeRole: The central operation for assuming temporary security credentials, commonly used for cross-account access.

MFA Integration: The AssumeRole call can include MFA information to enforce multi-factor authentication.

Credentials for S3 Access: The returned temporary credentials would provide the necessary

permissions to access the S3 bucket in the other account.

Reference:

AWS STS AssumeRole

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

QUESTION 188

A company has an analytics application that uses an AWS Lambda function to process transaction data asynchronously. A developer notices that asynchronous invocations of the Lambda function sometimes fail. When failed Lambda function invocations occur, the developer wants to invoke a second Lambda function to handle errors and log details.

Which solution will meet these requirements?

- A. Configure a Lambda function destination with a failure condition. Specify Lambda function as the destination type. Specify the error-handling Lambda function's Amazon Resource Name (ARN) as the resource.
- B. Enable AWS X-Ray active tracing on the initial Lambda function. Configure X-Ray to capture stack traces of the failed invocations. Invoke the error-handling Lambda function by including the stack traces in the event object.
- C. Configure a Lambda function trigger with a failure condition. Specify Lambda function as the destination type. Specify the error-handling Lambda function's Amazon Resource Name (ARN) as the resource.
- D. Create a status check alarm on the initial Lambda function. Configure the alarm to invoke the error-handling Lambda function when the alarm is initiated. Ensure that the alarm passes the stack trace in the event object.

Answer: A

Explanation:

Lambda Destinations on Failure: Allow routing asynchronous function invocations to specified resources (like another Lambda function) upon failure.

Error Handling: The error-handling Lambda receives details about the failure, enabling logging and custom actions.

Direct Integration: This solution leverages native Lambda functionality for a simpler implementation.

QUESTION 189

A company is preparing to migrate an application to the company's first AWS environment. Before this migration, a developer is creating a proof-of-concept application to validate a model for building and deploying container-based applications on AWS.

Which combination of steps should the developer take to deploy the containerized proof-of-concept application with the LEAST operational effort? (Select TWO.)

- A. Package the application into a zip file by using a command line tool. Upload the package to

Amazon S3

- B. Package the application into a container image by using the Docker CLI. Upload the image to Amazon Elastic Container Registry (Amazon ECR)
- C. Deploy the application to an Amazon EC2 instance by using AWS CodeDeploy.
- D. Deploy the application to Amazon Elastic Kubernetes Service (Amazon EKS) on AWS Fargate
- E. Deploy the application to Amazon Elastic Container Service (Amazon ECS) on AWS Fargate

Answer: BE

Explanation:

Containerization: Packaging the application as a container image promotes portability and standardization. Docker is the standard tool for containerization.

Amazon ECR: ECR is a managed container registry designed to work seamlessly with AWS container services.

Fargate: ECS Fargate provides serverless container orchestration, minimizing operational overhead for this proof-of-concept.

Reference:

Docker: <https://www.docker.com/>

Amazon ECR: <https://aws.amazon.com/ecr/>

QUESTION 190

A company runs an application on AWS. The application stores data in an Amazon DynamoDB table. Some queries are taking a long time to run. These slow queries involve an attribute that is not the table's partition key or sort key.

The amount of data that the application stores in the DynamoDB table is expected to increase significantly. A developer must increase the performance of the queries.

Which solution will meet these requirements'?

- A. Increase the page size for each request by setting the Limit parameter to be higher than the default value. Configure the application to retry any request that exceeds the provisioned throughput.
- B. Create a global secondary index (GSI). Set query attribute to be the partition key of the index.
- C. Perform a parallel scan operation by issuing individual scan requests in the parameters specify the segment for the scan requests and the total number of segments for the parallel scan.
- D. Turn on read capacity auto scaling for the DynamoDB table. Increase the maximum read capacity units (RCUs).

Answer: B

Explanation:

Global Secondary Index (GSI): GSIs enable alternative query patterns on a DynamoDB table by using different partition and sort keys.

Addressing Query Bottleneck: By making the slow-query attribute the GSI's partition key, you

optimize queries on that attribute.

Scalability: GSIs automatically scale to handle increasing data volumes.

Reference:

Amazon DynamoDB Global Secondary

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

QUESTION 191

A developer maintains a critical business application that uses Amazon DynamoDB as the primary data store. The DynamoDB table contains millions of documents and receives 30-60 requests each minute. The developer needs to perform processing in near-real time on the documents when they are added or updated in the DynamoDB table.

How can the developer implement this feature with the LEAST amount of change to the existing application code?

- A. Set up a cron job on an Amazon EC2 instance. Run a script every hour to query the table for changes and process the documents.
- B. Enable a DynamoDB stream on the table. Invoke an AWS Lambda function to process the documents.
- C. Update the application to send a PutEvents request to Amazon EventBridge. Create an EventBridge rule to invoke an AWS Lambda function to process the documents.
- D. Update the application to synchronously process the documents directly after the DynamoDB write.

Answer: B

Explanation:

DynamoDB Streams: Capture near real-time changes to DynamoDB tables, triggering downstream actions.

Lambda for Processing: Lambda functions provide a serverless way to execute code in response to events like DynamoDB Stream updates.

Minimal Code Changes: This solution requires the least modifications to the existing application.

Reference:

DynamoDB

Streams: <https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/Streams.html>

AWS Lambda: <https://aws.amazon.com/lambda/>

QUESTION 192

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:

Pseudo Parameters: CloudFormation provides pseudo parameters that reference runtime context, including the current AWS Region.

Operational Efficiency: The AWS::Region pseudo parameter offers the most direct and self-contained way to obtain the Region dynamically within the template.

Reference:

CloudFormation Pseudo

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

QUESTION 193

A company has an application that runs across multiple AWS Regions. The application is experiencing performance issues at irregular intervals. A developer must use AWS X-Ray to implement distributed tracing for the application to troubleshoot the root cause of the performance issues. What should the developer do to meet this requirement?

- A. Use the X-Ray console to add annotations for AWS services and user-defined services
- B. Use Region annotation that X-Ray adds automatically for AWS services Add Region annotation for user-defined services
- C. Use the X-Ray daemon to add annotations for AWS services and user-defined services
- D. Use Region annotation that X-Ray adds automatically for user-defined services Configure X-Ray to add Region annotation for AWS services

Answer: B

Explanation:

Distributed Tracing with X-Ray: X-Ray helps visualize request paths and identify bottlenecks in applications distributed across Regions.

Region Annotations (Automatic for AWS Services): X-Ray automatically adds a Region annotation to segments representing calls to AWS services. This aids in tracing cross-Region traffic.

Region Annotations (Manual for User-Defined): For segments representing calls to user-defined services in different Regions, the developer needs to add the Region annotation manually to enable comprehensive tracing.

Reference:

AWS X-Ray: <https://aws.amazon.com/xray/>

QUESTION 194

A company is building a new application that runs on AWS and uses Amazon API Gateway to expose APIs. Teams of developers are working on separate components of the application in parallel. The company wants to publish an API without an integrated backend so that teams that depend on the application backend can continue the development work before the API backend development is complete.

Which solution will meet these requirements?

- A. Create API Gateway resources and set the integration type value to MOCK. Configure the method integration request and integration response to associate a response with an HTTP status code. Create an API Gateway stage and deploy the API.
- B. Create an AWS Lambda function that returns mocked responses and various HTTP status codes. Create API Gateway resources and set the integration type value to AWS_PROXY. Deploy the API.
- C. Create an EC2 application that returns mocked HTTP responses. Create API Gateway resources and set the integration type value to AWS. Create an API Gateway stage and deploy the API.
- D. Create API Gateway resources and set the integration type value set to HTTP_PROXY. Add mapping templates and deploy the API. Create an AWS Lambda layer that returns various HTTP status codes. Associate the Lambda layer with the API deployment.

Answer: A

Explanation:

API Gateway Mocking: This feature is built for decoupling development dependencies. Here's the process:

Create resources and methods in your API Gateway.

Set the integration type to 'MOCK'.

Define Integration Responses, mapping HTTP status codes to desired mocked responses (JSON, etc.).

Deployment and Use:

Create a deployment stage for the API.

Frontend teams can call this API and get the mocked responses without a real backend.

Reference:

Mocking API Gateway APIs: <https://docs.aws.amazon.com/apigateway/latest/developerguide/howto-mock-integration.html>

QUESTION 195

A company has an application that is hosted on Amazon EC2 instances. The application stores objects in an Amazon S3 bucket and allows users to download objects from the S3 bucket. A developer turns on S3 Block Public Access for the S3 bucket. After this change, users report errors when they attempt to download objects. The developer needs to implement a solution so that only users who are signed in to the application can access objects in the S3 bucket.

Which combination of steps will meet these requirements in the MOST secure way? (Select TWO.)

- A. Create an EC2 instance profile and role with an appropriate policy Associate the role with the EC2 instances
- B. Create an IAM user with an appropriate policy. Store the access key ID and secret access key on the EC2 instances
- C. Modify the application to use the S3 GeneratePresignedUrl API call
- D. Modify the application to use the S3 GetObject API call and to return the object handle to the user
- E. Modify the application to delegate requests to the S3 bucket.

Answer: AC

Explanation:

IAM Roles for EC2 (A): The most secure way to provide AWS permissions from EC2.

Create a role with a policy allowing s3:GetObject on the specific bucket.

Attach the role to an instance profile and associate that profile with your instances.

Pre-signed URLs (C): Temporary, authenticated URLs for specific S3 actions.

Modify the app to use the AWS SDK to call GeneratePresignedUrl.

Embed these URLs when a user is properly logged in, allowing download access.

Reference:

IAM Roles for EC2: https://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles_use_switch-role-ec2.html

Generating Presigned

URLs: <https://docs.aws.amazon.com/AmazonS3/latest/userguide/ShareObjectPreSignedURL.htm>

QUESTION 196

An AWS Lambda function requires read access to an Amazon S3 bucket and requires read/write access to an Amazon DynamoDB table The correct IAM policy already exists

What is the MOST secure way to grant the Lambda function access to the S3 bucket and the DynamoDB table?

- A. Attach the existing IAM policy to the Lambda function.
- B. Create an IAM role for the Lambda function Attach the existing IAM policy to the role Attach the role to the Lambda function
- C. Create an IAM user with programmatic access Attach the existing IAM policy to the user. Add the user access key ID and secret access key as environment variables in the Lambda function.
- D. Add the AWS account root user access key ID and secret access key as encrypted environment variables in the Lambda function

Answer: B

Explanation:

Principle of Least Privilege: Granting specific permissions through an IAM role is more secure than

directly attaching policies to a function or using root user credentials.

IAM Roles for Lambda: Designed to provide temporary credentials to Lambda functions, enhancing security.

Reusability: The existing IAM policy ensures the correct S3 and DynamoDB access is granted.

Reference:

IAM Roles for Lambda Documentation: <https://docs.aws.amazon.com/lambda/latest/dg/lambdaintro-execution-role.html>

IAM Best Practices: <https://docs.aws.amazon.com/IAM/latest/UserGuide/best-practices.html>

QUESTION 197

A developer is designing a serverless application for a game in which users register and log in through a web browser. The application makes requests on behalf of users to a set of AWS Lambda functions that run behind an Amazon API Gateway HTTP API.

The developer needs to implement a solution to register and log in users on the application's sign-in page. The solution must minimize operational overhead and must minimize ongoing management of user identities. Which solution will meet these requirements'?

- A. Create Amazon Cognito user pools for external social identity providers. Configure IAM roles for the identity pools.
- B. Program the sign-in page to create users' IAM groups with the IAM roles attached to the groups.
- C. Create an Amazon RDS for SQL Server DB instance to store the users and manage the permissions to the backend resources in AWS.
- D. Configure the sign-in page to register and store the users and their passwords in an Amazon DynamoDB table with an attached IAM policy.

Answer: A

Explanation:

Amazon Cognito User Pools: A managed user directory service, simplifying user registration and login.

Social Identity Providers: Cognito supports integration with external providers (e.g., Google, Facebook), reducing development effort.

IAM Roles for Authorization: Cognito-managed IAM roles grant fine-grained access to AWS resources (like Lambda functions).

Operational Overhead: Cognito minimizes the need to manage user identities and credentials independently.

Reference:

Amazon Cognito Documentation <https://docs.aws.amazon.com/cognito/>

Cognito User Pools for Web

Applications: <https://docs.aws.amazon.com/cognito/latest/developerguide/cognito-user-pools-appintegration.html>

QUESTION 198

A developer supports an application that accesses data in an Amazon DynamoDB table. One of the item attributes is expirationDate in the timestamp format. The application uses this attribute to find items, archive them, and remove them from the table based on the timestamp value

The application will be decommissioned soon, and the developer must find another way to implement this functionality. The developer needs a solution that will require the least amount of code to write.

Which solution will meet these requirements?

- A. Enable TTL on the expirationDate attribute in the table. Create a DynamoDB stream. Create an AWS Lambda function to process the deleted items. Create a DynamoDB trigger for the Lambda function.
- B. Create two AWS Lambda functions one to delete the items and one to process the items Create a DynamoDB stream Use the DeleteItem API operation to delete the items based on the expirationDate attribute Use the GetRecords API operation to get the items from the DynamoDB stream and process them
- C. Create two AWS Lambda functions, one to delete the items and one to process the items. Create an Amazon EventBridge scheduled rule to invoke the Lambda Functions Use the DeleteItem API operation to delete the items based on the expirationDate attribute. Use the GetRecords API operation to get the items from the DynamoDB table and process them.
- D. Enable TTL on the expirationDate attribute in the table Specify an Amazon Simple Queue Service (Amazon SQS) dead-letter queue as the target to delete the items Create an AWS Lambda function to process the items

Answer: A

Explanation:

TTL for Automatic Deletion: DynamoDB's Time-to-Live effortlessly deletes expired items without manual intervention.

DynamoDB Stream: Captures changes to the table, including deletions of expired items, triggering downstream actions.

Lambda for Processing: A Lambda function connected to the stream provides custom logic for handling the deleted items.

Code Efficiency: This solution leverages native DynamoDB features and stream-based processing, minimizing the need for custom code.

Reference:

DynamoDB TTL

Documentation: <https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/TTL.html>

DynamoDB Streams

Documentation: <https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/Streams.html>