

# Amazon

## Exam Questions DVA-C02

DVA-C02



### NEW QUESTION 1

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

**Answer: D**

#### Explanation:

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

Reference: Using dynamic references to specify template values

### NEW QUESTION 2

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
- B. Use unique paths in Parameter Store for each variable in each environment
- C. Store the credentials in AWS Secrets Manager in each environment.
- D. 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.
- E. Update the application to retrieve the variables from an encrypted file that is stored with the application
- F. Store the API URL and credentials in unique files for each environment.
- G. Update the application to retrieve the variables from each of the deployed environments
- H. 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.

References:

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

? [Parameter Store - AWS Systems Manager]

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

### NEW QUESTION 3

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
- B. Instruct the other developers to point the endpoints to the development stage.
- C. Define a new API Gateway API that points to the new API application code
- D. Instruct the other developers to point the endpoints to the new API.
- E. Implement a query parameter in the API application code that determines which code version to call.
- F. 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.

References:

- ? [What Is Amazon API Gateway? - Amazon API Gateway]
- ? [Set up a Stage in API Gateway - Amazon API Gateway]

**NEW QUESTION 4**

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:** A

**Explanation:**

Increasing the memory configuration of the Lambda function will also increase the CPU and network throughput available to the function. This can improve the performance of the video conversion process and reduce the execution time of the function. This solution does not require any code changes or additional resources. It is also recommended to follow the best practices for preventing Lambda function timeouts<sup>1</sup>. References:

- ? Troubleshoot Lambda function invocation timeout errors | AWS re:Post

**NEW QUESTION 5**

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 the 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]

**NEW QUESTION 6**

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.
- C. Store the access key ID and secret access key on the EC2 instances.
- D. Modify the application to use the S3 GeneratePresignedUrl API call.
- E. Modify the application to use the S3 GetObject API call and to return the object handle to the user.
- F. Modify the application to delegate requests to the S3 bucket.

**Answer:** AC

**Explanation:**

The most secure way to allow the EC2 instances to access the S3 bucket is to use an EC2 instance profile and role with an appropriate policy that grants the necessary permissions. This way, the EC2 instances can use temporary security credentials that are automatically rotated and do not need to store any access keys on the instances. To allow the users who are signed in to the application to download objects from the S3 bucket, the application can use the S3 GeneratePresignedUrl API call to create a pre-signed URL that grants temporary access to a specific object. The pre-signed URL can be returned to the user, who can then use it to download the object within a specified time period. References:

- ? Use Amazon S3 with Amazon EC2

- ? How to Access AWS S3 Bucket from EC2 Instance In a Secured Way
- ? Sharing an Object with Others

#### NEW QUESTION 7

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
- B. Perform a query operation on the GSI by using the begins\_with key condition expression with the email\_address property.  
Add a global secondary index (GSI) to the DynamoDB table with email\_address as the partition key and customer\_type as the sort key
- C. Perform a query operation on the GSI by using the begins\_with key condition expression with the email\_address property.
- D. Perform a query operation on the GSI by using the begins\_with key condition expression with the email\_address property.
- E. Address property.
- F. Add a local secondary index (LSI) to the DynamoDB table with customer\_type as the partition Key and email\_address as the sort Key
- G. Perform a query operation on the LSI by using the begins\_with Key condition expression with the email-address property.
- H. Add a local secondary index (LSI) to the DynamoDB table with job-title as the partition key and email\_address as the sort key
- I. 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

#### NEW QUESTION 8

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
- B. Add the Lambda function as the target of the EventBridge rule.
- C. 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.
- D. Create an AWS Step Functions state machine
- E. Configure the state machine to invoke the Lambda function execution role at a specified interval by using a Wait state
- F. Set the interval to 15 minutes.
- G. Provision a small Amazon EC2 instance
- H. 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.

#### NEW QUESTION 9

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

Which solution will meet these requirements?

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

**Answer:** A

#### Explanation:

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

References:

? [What Is Amazon ElastiCache? - Amazon ElastiCache]

? [Encryption in Transit - Amazon ElastiCache for Redis]

? [Encryption at Rest - Amazon ElastiCache for Redis]

**NEW QUESTION 10**

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. Mastered
- B. Not Mastered

**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.

References:

- ? [Amazon Machine Images (AMI) - Amazon Elastic Compute Cloud]
- ? [Encrypting an Amazon EBS Snapshot - Amazon Elastic Compute Cloud]
- ? [Copying an AMI - Amazon Elastic Compute Cloud]

**NEW QUESTION 10**

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

**NEW QUESTION 11**

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 the 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 de-couple and scale for applications<sup>1</sup>. Amazon SQS offers two types of queues: Standard and FIFO (First In First Out) queues<sup>1</sup>. The FIFO queue uses the messageDeduplicationId property to treat messages with the same value as duplicate<sup>2</sup>.

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.

**NEW QUESTION 16**

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

frequently return values in the UnprocessedKeys element.

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

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

**Answer:** BC

**Explanation:**

The UnprocessedKeys element indicates that the BatchGetItem operation did not process all of the requested items in the current response. This can happen if the response size limit is exceeded or if the table's provisioned throughput is exceeded. To handle this situation, the developer should retry the batch operation with exponential backoff and randomized delay to avoid throttling errors and reduce the load on the table. The developer should also use an AWS SDK to make the requests, as the SDKs automatically retry requests that return UnprocessedKeys.

References:

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

**NEW QUESTION 21**

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

**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

**NEW QUESTION 26**

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. Update the Amazon S3 bucket settings and enable static website hosting.
- B. Specify index.html as the Index document. Update the S3 bucket policy to enable access.
- C. Update the CloudFront distribution's origin to use the S3 website endpoint.
- D. Update the CloudFront distribution's origin to use the S3 website endpoint.
- E. 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.

F. 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:** A

**Explanation:**

The simplest and most efficient way to enable accessing directories without specifying a file name is to update the CloudFront distribution's settings to index.html as the default root object. This will instruct CloudFront to return the index.html object when a user requests the root URL or a directory URL for the distribution.

This solution does not require enabling static website hosting on the S3 bucket, creating a CloudFront function, or creating a custom error response. References

? Specifying a default root object

? cloudfront-default-root-object-configured

? How to setup CloudFront default root object?

? Ensure a default root object is configured for AWS Cloudfront ...

**NEW QUESTION 28**

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

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

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

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

K. Use the Lambda function to store the photos in Amazon S3. Store the object's S3 key as part of the photo details in the

L. Retrieve previously uploaded photos by querying DynamoDB for the S3 key.

M. Create a users table in DynamoD

N. Use the table to manage user account

O. Create a Lambda authorizer that validates user credentials against the users tabl

P. Integrate the Lambda authorizer with API Gateway to control access to the AP

Q. Use the Lambda function to store the photos in Amazon S3. Store the object's S3 key as par of the photo details in the DynamoDB tabl

R. Retrieve previously uploaded photos by querying DynamoDB for the S3 key.

DynamoDB

**Answer:** B

**Explanation:**

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

References:

? [Amazon Cognito User Pools]

? [Use Amazon Cognito User Pools - Amazon API Gateway]

? [Amazon Simple Storage Service (S3)]

? [Amazon DynamoDB]

**NEW QUESTION 31**

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

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

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

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

**Answer:** A

**Explanation:**

AWS Secrets Manager is a service that allows you to store and manage secrets, such as database credentials, API keys, and passwords, in a secure and centralized way. It also provides features such as automatic secret rotation, auditing, and 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 purposes<sup>2</sup>. To access your secrets from your application, you can use the ARN of the secret, which is a unique identifier that includes the Region name. This way, your application can use the appropriate secret based on the Region where it is deployed<sup>3</sup>.

References:

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

### NEW QUESTION 35

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

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

**Answer: D**

#### Explanation:

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

- ? AssumeRole
- ? Requesting Temporary Security Credentials

### NEW QUESTION 36

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

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

Which solution will meet these requirements?

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

**Answer: D**

#### Explanation:

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

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

### NEW QUESTION 41

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

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

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

**Answer: C**

#### Explanation:

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

because it will not provide customers with access to their reports within one hour, as Amazon SNS email delivery is not guaranteed. Option D is not optimal because it will require more operational overhead to manage an RDS database and a Lambda function for storing and deleting the reports.  
 References: Amazon S3 Presigned URLs, Amazon S3 Lifecycle

**NEW QUESTION 45**

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 mad
- B. Call Amazon S3 by using a GET request to access the object without PII.
- C. Set up an S3 event notification that invokes the removePii function when an S3 PUT request is mad
- D. Call Amazon S3 by using a PUT request to access the object without PII.
- E. Create an S3 Object Lambda access point from the S3 consol
- F. Select the removePii functio
- G. Use S3 Access Points to access the object without PII.
- H. Create an S3 access point from the S3 consol
- I. Use the access point name to call the GetObjectLegalHold S3 API functio
- J. 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

**NEW QUESTION 50**

A developer has observed an increase in bugs in the AWS Lambda functions that a development team has deployed in its Node.js application. To minimize these bugs, the developer wants to implement automated testing of Lambda functions in an environment that closely simulates the Lambda environment.  
 The developer needs to give other developers the ability to run the tests locally. The developer also needs to integrate the tests into the team's continuous integration and continuous delivery (CI/CD) pipeline before the AWS Cloud Development Kit (AWS CDK) deployment.  
 Which solution will meet these requirements?

- A. Create sample events based on the Lambda documentatio
- B. Create automated test scripts that use the cdk local invoke command to invoke the Lambda function
- C. Check the respons
- D. Document the test scripts for the other developers on the tea
- E. Update the CI/CD pipeline to run the test scripts.
- F. Install a unit testing framework that reproduces the Lambda execution environment.
- G. Invoke the handler function by using a unit testing framewor
- H. Check the respons
- I. Document how to run the unit testing framework for the other developers on the tea
- J. Update the CI/CD pipeline to run the unit testing framework.
- K. Install the AWS Serverless Application Model (AWS SAM) CLI too
- L. Use the sam local generate-event command to generate sample events for the automated test
- M. Create automated test scripts that use the sam local invoke command to invoke the Lambda function
- N. Check the respons
- O. Document the test scripts for the other developers on the tea
- P. Update the CI/CD pipeline to run the test scripts.
- Q. Create sample events based on the Lambda documentatio
- R. Create a Docker container from the Node.js base image to invoke the Lambda function
- S. Check the respons
- T. Document how to run the Docker container for the other developers on the tea
- . Update the CI/CD pipeline to run the Docker container.

Create sample events based on the Lambda

**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 applications<sup>3</sup>. The sam local generate-event command of AWS SAM CLI generates sample events for automated tests<sup>3</sup>. The sam local invoke command is used to invoke Lambda functions<sup>3</sup>. Therefore, option C is correct.

**NEW QUESTION 53**

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]

**NEW QUESTION 57**

A developer is creating a new REST API by using Amazon API Gateway and AWS Lambda. 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 applications<sup>2</sup>. The sam local start-api subcommand of AWS SAM CLI is used to simulate a REST API by starting a new local endpoint<sup>3</sup>. Therefore, option D is correct.

**NEW QUESTION 62**

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
- C. Set up a DynamoDB Accelerator (DAX) cluster.
- D. Configure the Amazon RDS instances to use Multi-AZ deployment with one standby instance.
- E. Offload read requests from the main database to the standby instance.
- F. 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 key-value store that can be used to cache frequently accessed data and improve application performance<sup>1</sup>. 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 cluster<sup>2</sup>. The developer can use any of the supported Memcached clients to interact with the cache cluster<sup>3</sup>. The developer can also use Auto Discovery to dynamically discover and connect to all cache nodes in a cluster<sup>4</sup>.

? Amazon ElastiCache for Memcached is compatible with the Memcached protocol, which means that the developer can use existing tools and libraries that work with

Memcached<sup>1</sup>. 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.

**NEW QUESTION 63**

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

**Answer: C**

**Explanation:**

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

References:

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

**NEW QUESTION 66**

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

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

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

- A. The command is incorrect; it should be rewritten to use put-item with a string argument
- B. The developer needs to log a ticket with AWS Support to enable access to the demoman-table
- C. Amazon 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.

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

**NEW QUESTION 69**

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

- A. Set the Deployment Preference Type to Canary10Percent10Minute  
AutoPublishAlias property to the Lambda alias.
- B. Set the Deployment Preference Type to Linear10PercentEvery10Minute
- D. Set AutoPublishAlias property to the Lambda alias.
- E. Set the Deployment Preference Type to Canary10Percent10Minute
- F. Set the PreTraffic and PostTraffic properties to the Lambda alias.
- G. Set the Deployment Preference Type to Linear10PercentEvery10Minute
- H. Set PreTraffic and Post Traffic properties to the Lambda alias.

**Answer: A**

**Explanation:**

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

**NEW QUESTION 72**

A company's developer has deployed an application in AWS by using AWS CloudFormation. The CloudFormation stack includes parameters in AWS Systems Manager Parameter Store that the application uses as configuration settings. The application can modify the parameter values. When the developer updated the stack to create additional resources with tags, the developer noted that the parameter values were reset and that the values ignored the latest changes made by the application. The developer needs to change the way the company deploys the CloudFormation stack. The developer also needs to avoid resetting the parameter values outside the stack. Which solution will meet these requirements with the LEAST development effort?

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

**Answer: D**

**Explanation:**

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

**NEW QUESTION 76**

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 applicatio
- C. Deploy the unit tests by using the amplify push command in the Amplify CLI.
- D. Add a test phase to the amplify.yml build settings for the application.
- E. 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

**NEW QUESTION 80**

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]

**NEW QUESTION 82**

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

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

**Answer: B**

**Explanation:**

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

Reference: Using AWS KMS keys to encrypt S3 objects

**NEW QUESTION 84**

A developer is working on an ecommerce website The developer wants to review server logs without logging in to each of the application servers individually. The website runs on multiple Amazon EC2 instances, is written in Python, and needs to be highly available How can the developer update the application to meet these requirements with MINIMUM changes?

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

**Answer: D**

**Explanation:**

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

? Using the CloudWatch Agent

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

#### NEW QUESTION 87

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:** BD

#### 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.

References:

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

? [Amazon API Gateway Metrics and Dimensions - Amazon CloudWatch]

? [Troubleshooting API Errors - Amazon API Gateway]

#### NEW QUESTION 91

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

**Answer:** B

#### Explanation:

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

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

References:

- ? [What Is AWS Lambda? - AWS Lambda]
- ? [AWS Lambda Layers - AWS Lambda]

**NEW QUESTION 92**

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 failure
- B. Call the POST API manually by using the requests from the log file.
- C. Create and inspect the Lambda dead-letter queue
- D. Troubleshoot the failed function
- E. Reprocess the events.
- F. Inspect the Lambda logs in Amazon CloudWatch for possible error
- G. Fix the errors.
- H. 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

**NEW QUESTION 97**

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

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

#### NEW QUESTION 99

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 applicatio
- B. Update the Auto Scaling group launch configuration to use the AMI.
- C. 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 applicatio
- D. Update the Auto Scaling group launch configuration to use the AMI.
- E. Set up AWS CodeDeploy to deploy the most recent version of the application at runtime.
- F. Set up AWS CodePipeline to deploy the most recent version of the application at runtime.
- G. 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.

References:

? [What Is AWS CloudFormation? - AWS CloudFormation]

? [What Is AWS CodeDeploy? - AWS CodeDeploy]

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

## NEW QUESTION 101

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 tabl
- B. Configure the table with a primary key that consists of the title as the partition key and the release year as the sort ke
- C. Create a global secondary index that uses the genre as the partition key and the title as the sort key.
- D. Create an Amazon DynamoDB tabl
- E. Configure the table with a primary key that consists of the genre as the partition key and the release year as the sort ke
- F. Create a global secondary index that uses the title as the partition key.
- G. On an Amazon RDS DB instance, create a table that contains columns for title, release year, and genr
- H. Configure the title as the primary key.
- I. 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.

References:

? [Amazon DynamoDB]

? [Working with Queries - Amazon DynamoDB]

? [Working with Global Secondary Indexes - Amazon DynamoDB]

#### NEW QUESTION 104

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 group
- B. Encrypt the parameter by using an AWS Key Management Service (AWS KMS) key. Turn on secret rotation.
- C. Use IAM policies and roles to grant AWS KMS permissions to access Amazon RDS.
- D. Migrate the credentials to AWS Systems Manager Parameter Store.
- E. Encrypt the parameter by using an AWS Key Management Service (AWS KMS) key. Turn on secret rotation.
- F. Turn on secret rotation.
- G. Use IAM policies and roles to grant Amazon ECS Fargate permissions to access to AWS Secrets Manager.
- H. Migrate the credentials to ECS Fargate environment variable.
- I. Encrypt the credentials by using an AWS Key Management Service (AWS KMS) key. Turn on secret rotation.
- J. Use IAM policies and roles to grant Amazon ECS Fargate permissions to access to AWS Secrets Manager.
- K. Migrate the credentials to AWS Secrets Manager.
- L. 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:**

AWS Secrets Manager is a service that helps you store, distribute, and rotate secrets securely. You can use Secrets Manager to migrate your credentials from your application code to a secure and encrypted storage. You can also enable automatic rotation of your secrets by using AWS Lambda functions or custom logic. You can use IAM policies and roles to grant your Amazon ECS Fargate tasks permissions to access your secrets from Secrets Manager. This solution minimizes the operational overhead of managing your credentials and enhances the security of your application. References

- ? AWS Secrets Manager: Store, Distribute, and Rotate Credentials Securely | AWS News Blog
- ? Why You Should Audit and Rotate Your AWS Credentials Periodically - Cloud Academy
- ? Top 5 AWS root account best practices - TheServerSide

#### NEW QUESTION 106

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.
- B. Create an Amazon CloudWatch alarm for the GetSecretValue API operation requests.
- C. Create a secrets manager-secret-unused AWS Config managed rule.
- D. Create an Amazon EventBridge rule to initiate notification when the AWS Config managed rule is met.
- E. Deactivate the applications secrets and monitor the applications error logs temporarily.
- F. Configure AWS X-Ray for the application.
- G. 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. References: [AWS Config Managed Rules], [Amazon EventBridge]

**NEW QUESTION 111**

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
- C. Set the value to 600.
- D. Create an Amazon EventBridge rule that runs on a regular schedule to invoke the Lambda function.
- E. 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

#### NEW QUESTION 112

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:** AB

#### 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]

#### NEW QUESTION 117

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 returned 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.

References: [AWS Encryption SDK], [AWS Key Management Service]

#### NEW QUESTION 121

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]

#### NEW QUESTION 122

A company wants to automate part of its deployment process. A developer needs to automate the process of checking for and deleting unused resources that supported previously deployed stacks but that are no longer used. The company has a central application that uses the AWS Cloud Development Kit (AWS CDK) to manage all deployment stacks. The stacks are spread out across multiple accounts. The developer's solution must integrate as seamlessly as possible within the current deployment process. Which solution will meet these requirements with the LEAST amount of configuration?

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

**Answer:** B

#### Explanation:

This solution meets the requirements with the least amount of configuration because it uses a feature of AWS CDK that allows custom logic to be executed during stack deployment or deletion. The AWS Cloud Development Kit (AWS CDK) is a software development framework that allows you to define cloud infrastructure as code and provision it through CloudFormation. An AWS CDK custom resource is a construct that enables you to create resources that are not natively supported by CloudFormation or perform tasks that are not supported by CloudFormation during stack deployment or deletion. The developer can write a handler function in the code that uses AWS SDK calls to check for and delete unused resources, and create an AWS CDK custom resource that attaches the function code to a

Lambda function and invokes it when the deployment stack runs. This way, the developer can automate the cleanup process without requiring additional configuration or integration. Creating a CloudFormation template from a JSON file will require additional configuration and integration with the central AWS CDK application. Creating an API in AWS Amplify will require additional configuration and integration with the central AWS CDK application and may not provide optimal performance or availability. Writing a handler function in the AWS Lambda console will require additional configuration and integration with the central AWS CDK application.

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

#### NEW QUESTION 127

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.

References: AWS Lambda Monitoring, [CloudWatch Logs Insights]

#### NEW QUESTION 131

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

#### NEW QUESTION 132

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 automation 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 "config"
- 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

#### NEW QUESTION 134

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

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

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

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

**Answer:** C

**Explanation:**

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

**NEW QUESTION 137**

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 policies by using each VPC endpoint ID that have the aws SourceVpce value in the StringNotEquals condition.
- B. Create a single S3 bucket policy that has the aws SourceVpc value and in the StingNotEquals condition to use VPC ID.
- C. Create a single S3 bucket policy that the multiple aws SourceVpce value and in the SringNotEquals condton to use vpce.
- D. Create a single S3 bucket policy that has multiple aws sourceVpce value in the StingNotEquale conditio
- E. Repeat for all the VPC endpoint IDs.

**Answer:** D

**Explanation:**

This solution will meet the requirements by creating a single S3 bucket policy that denies access to the S3 bucket unless the request comes from one of the specified VPC endpoints. The aws:SourceVpce condition key is used to match the ID of the VPC endpoint that is used to access the S3 bucket. The

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

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

**NEW QUESTION 141**

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-access-dynamodb.html](https://docs.aws.amazon.com/IAM/latest/UserGuide/reference_policies_examples_lambda-access-dynamodb.html)

**NEW QUESTION 144**

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

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

**Answer:** C

**Explanation:**

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

**NEW QUESTION 149**

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.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ReadOnlyAPIActions",
      "Effect": "Allow",
      "Action": [
        "dynamodb:GetItem",
        "dynamodb:BatchGetItem",
        "dynamodb:Scan",
        "dynamodb:Query",
        "dynamodb:ConditionCheckItem"
      ],
      "Resource": "arn:aws:dynamodb:us-west-2:account-id:table/Cars"
    }
  ]
}
```

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:dynamo\*: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
- D. Associate the role with the policy.
- E. Create a trust relationship between the role and dynamodb.amazonaws.com.

**Answer: C**

**Explanation:**

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

**NEW QUESTION 153**

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.
- C. Configure S3 event notifications with a destination of the SNS topic.
- D. Subscribe the Lambda function to the SNS topic.
- E. 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.
- F. 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.
- G. Create an Amazon Simple Queue Service (Amazon SQS) queue.
- H. Send S3 event notifications to Amazon EventBridge.
- I. 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:**

This solution will allow the developer to receive notifications for each image uploaded to the S3 bucket, and also create a thumbnail using the Lambda function. The SNS topic will serve as a trigger for both the Lambda function and the email notification subscription. When an image is uploaded, S3 will send a notification to the SNS topic, which will trigger the Lambda function to create the thumbnail and also send an email notification to the specified email address.

**NEW QUESTION 155**

.....

## **Thank You for Trying Our Product**

### **We offer two products:**

1st - We have Practice Tests Software with Actual Exam Questions

2nd - Questions and Answers in PDF Format

### **DVA-C02 Practice Exam Features:**

- \* DVA-C02 Questions and Answers Updated Frequently
- \* DVA-C02 Practice Questions Verified by Expert Senior Certified Staff
- \* DVA-C02 Most Realistic Questions that Guarantee you a Pass on Your FirstTry
- \* DVA-C02 Practice Test Questions in Multiple Choice Formats and Updatesfor 1 Year

**100% Actual & Verified — Instant Download, Please Click**  
**[Order The DVA-C02 Practice Test Here](#)**