



## Amazon

### Exam Questions DVA-C02

DVA-C02

## About ExamBible

### *Your Partner of IT Exam*

## Found in 1998

ExamBible is a company specialized on providing high quality IT exam practice study materials, especially Cisco CCNA, CCDA, CCNP, CCIE, Checkpoint CCSE, CompTIA A+, Network+ certification practice exams and so on. We guarantee that the candidates will not only pass any IT exam at the first attempt but also get profound understanding about the certificates they have got. There are so many alike companies in this industry, however, ExamBible has its unique advantages that other companies could not achieve.

## Our Advances

### \* 99.9% Uptime

All examinations will be up to date.

### \* 24/7 Quality Support

We will provide service round the clock.

### \* 100% Pass Rate

Our guarantee that you will pass the exam.

### \* Unique Gurantee

If you do not pass the exam at the first time, we will not only arrange FULL REFUND for you, but also provide you another exam of your claim, ABSOLUTELY FREE!

### NEW QUESTION 1

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

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

**Answer: B**

#### Explanation:

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

### NEW QUESTION 2

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

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

**Answer: A**

#### Explanation:

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

References: [AWS X-Ray SDKs]

### NEW QUESTION 3

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

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

**Answer: B**

#### Explanation:

AWS CloudFormation is a service that allows developers to model and provision AWS resources using templates. A CloudFormation template can define the load

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

#### NEW QUESTION 4

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

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

**Answer: C**

#### Explanation:

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

References:

? [What Is AWS CodeCommit? - AWS CodeCommit]

? [AWS CodePipeline - AWS CodeCommit]

#### NEW QUESTION 5

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

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

**Answer: D**

#### Explanation:

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

References:

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

? [What Is Amazon CloudFront? - Amazon CloudFront]

? [Custom Domain Names for APIs - Amazon API Gateway]

#### NEW QUESTION 6

A company is using Amazon OpenSearch Service to implement an audit monitoring system. A developer needs to create an AWS CloudFormation custom resource that is

associated with an AWS Lambda function to configure the OpenSearch Service domain. The Lambda function must access the OpenSearch Service domain by using Open Search Service internal master user credentials. What is the MOST secure way to pass these credentials to the Lambdas function?

- A. Use a CloudFormation parameter to pass the master user credentials at deployment to the OpenSearch Service domain's MasterUserOptions and the Lambda function's environment variable
- B. Set the No Echo attribute to true.
- C. Use a CloudFormation parameter to pass the master user credentials at deployment to the OpenSearch Service domain's MasterUserOptions and to create a parameter
- D. In AWS Systems Manager Parameter Store
- E. Set the No Echo attribute to true
- F. Create an IAM role that has the ssm:GetParameter permission
- G. Assign the role to the Lambda function
- H. Store the parameter name as the Lambda function's environment variable
- I. Resolve the parameter's value at runtime.
- J. Use a CloudFormation parameter to pass the master user credentials at deployment to the OpenSearch Service domain's MasterUserOptions and the Lambda function's environment variable. We encrypt the parameter's value by using the AWS Key Management Service (AWS KMS) encrypt command.
- K. Use CloudFormation to create an AWS Secrets Manager secret
- L. Use a CloudFormation dynamic reference to retrieve the secret's value for the OpenSearch Service domain's MasterUserOptions
- M. Create an IAM role that has the secrets:manage
- 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 7**

A developer is designing a serverless application for a game in which users register and log in through a web browser. The application makes requests on behalf of users to a set of AWS Lambda functions that run behind an Amazon API Gateway HTTP API.

The developer needs to implement a solution to register and log in users on the application's sign-in page. The solution must minimize operational overhead and must minimize ongoing management of user identities.

Which solution will meet these requirements?

- A. Create Amazon Cognito user pools for external social identity providers. Configure IAM roles for the identity pools.
- B. Program the sign-in page to create users' IAM groups with the IAM roles attached to the groups.
- C. Create an Amazon RDS for SQL Server DB instance to store the users and manage the permissions to the backend resources in AWS.
- D. Configure the sign-in page to register and store the users and their passwords in an Amazon DynamoDB table with an attached IAM policy.

**Answer:** A

**Explanation:**

<https://docs.aws.amazon.com/cognito/latest/developerguide/signing-up-users-in-your-app.html>

**NEW QUESTION 8**

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

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

The company wants the support team to receive notifications in near real time only when

the payment processing external API error rate exceeds 5% of the total number of transactions in an hour. Developers need to use an existing Amazon Simple Notification Service (Amazon SNS) topic that is configured to notify the support team.

Which solution will meet these requirements?

- A. Write the results of payment processing API calls to Amazon CloudWatch.
- B. Use Amazon CloudWatch Logs Insights to query the CloudWatch log.
- C. Schedule the Lambda function to check the CloudWatch logs and notify the existing SNS topic.
- D. Publish custom metrics to CloudWatch that record the failures of the external payment processing API call.
- E. Configure a CloudWatch alarm to notify the existing SNS topic when error rate exceeds the specified rate.
- F. Publish the results of the external payment processing API calls to a new Amazon SNS topic.
- G. Subscribe the support team members to the new SNS topic.
- H. Write the results of the external payment processing API calls to Amazon S3. Schedule an Amazon Athena query to run at regular intervals.
- I. Configure Athena to send notifications to the existing SNS topic when the error rate exceeds the specified rate.

**Answer:** B

**Explanation:**

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

References:

? [What Is Amazon CloudWatch? - Amazon CloudWatch]

? [Publishing Custom Metrics - Amazon CloudWatch]

? [Creating Amazon CloudWatch Alarms - Amazon CloudWatch]

**NEW QUESTION 9**

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 10

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

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

**Answer:** A

#### Explanation:

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

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

#### NEW QUESTION 10

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

The developer needs to secure the API credentials and enforce automatic credentials rotation on a quarterly basis. Which solution will meet these requirements MOST securely?

- A. Use AWS Key Management Service (AWS KMS) to encrypt the configuration file
- B. Decrypt the configuration file when users make API calls to the SaaS vendor
- C. Enable rotation.
- D. Retrieve temporary credentials from AWS Security Token Service (AWS STS) every 15 minutes
- E. Use the temporary credentials when users make API calls to the SaaS vendor.
- F. Store the credentials in AWS Secrets Manager and enable rotation
- G. Configure the API to have Secrets Manager access.
- H. Store the credentials in AWS Systems Manager Parameter Store and enable rotation
- I. Retrieve the credentials when users make API calls to the SaaS vendor.

**Answer:** C

#### Explanation:

Store the credentials in AWS Secrets Manager and enable rotation. Configure the API to have Secrets Manager access. This is correct. This solution will meet the requirements most securely, because it uses a service that is designed to store and manage secrets such as API credentials. AWS Secrets Manager helps you protect access to your applications, services, and IT resources by enabling you to rotate, manage, and retrieve secrets throughout their lifecycle<sup>1</sup>. You can store secrets such as passwords, database strings, API keys, and license codes as encrypted values<sup>2</sup>. You can also configure automatic rotation of your secrets on a schedule that you specify<sup>3</sup>. You can use the AWS SDK or CLI to retrieve secrets from Secrets Manager when you need them<sup>4</sup>. This way, you can avoid storing credentials in plaintext files or hardcoding them in your code.

#### NEW QUESTION 14

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

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

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

**Answer:** D

#### NEW QUESTION 18

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 replic
- H. Connect to the read replica by using SS
- I. Configure the read replica to store frequently accessed data.
- J. Create an Amazon DynamoDB table and a DynamoDB Accelerator (DAX) cluster for the tabl
- 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 20**

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 sam local start-api subcommand allows you to run your serverless application locally for quick development and testing<sup>1</sup>. It creates a local HTTP server that acts as a proxy for API Gateway and invokes your Lambda functions based on the AWS SAM template<sup>1</sup>. You can use the sam local start-api subcommand to test your REST API locally by sending HTTP requests to the local endpoint<sup>1</sup>.

**NEW QUESTION 22**

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 26**

A company is building a compute-intensive application that will run on a fleet of Amazon EC2 instances. The application uses attached Amazon Elastic Block Store (Amazon EBS) volumes for storing data. The Amazon EBS volumes will be created at time of initial deployment. The application will process sensitive information. All of the data must be encrypted. The solution should not impact the application's performance.

Which solution will meet these requirements?

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

**Answer:** A

**Explanation:**

Amazon Elastic Block Store (Amazon EBS) provides block level storage volumes for use with Amazon EC2 instances<sup>1</sup>. Amazon EBS encryption offers a straightforward encryption solution for your EBS resources associated with your EC2 instances<sup>1</sup>. When you create an encrypted EBS volume and attach it to a supported instance type, the following types of data are encrypted: Data at rest inside the volume, all data moving between the volume and the instance, all snapshots created from the volume, and all volumes created from those snapshots<sup>1</sup>. Therefore, option A is correct.

**NEW QUESTION 27**

A company uses a custom root certificate authority certificate chain (Root CA Cert) that is 10 KB in size generate SSL certificates for its on-premises HTTPS

endpoints. One of the company's cloud based applications has hundreds of AWS Lambda functions that pull data from these endpoints. A developer updated the trust store of the Lambda execution environment to use the Root CA Cert when the Lambda execution environment is initialized. The developer bundled the Root CA Cert as a text file in the Lambdas deployment bundle.

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

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

- A. Mastered
- B. Not Mastered

**Answer: A**

**Explanation:**

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

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

**NEW QUESTION 30**

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 32**

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 Region
- E. Update the application to use the Amazon Resource Name (ARN) based on the Region.

- F. Store credentials in a config file
- G. Upload the config file to an S3 bucket in the primary Region
- H. Enable Cross-Region Replication (CRR) to an S3 bucket in the secondary region
- I. Update the application to access the config file from the S3 bucket based on the Region.
  - Store credentials in a config file
- K. Upload the config file to an Amazon Elastic File System (Amazon EFS) file system
- L. Update the application to use the Amazon EFS file system Regional endpoints to access the config file in the primary and secondary Regions.

**Answer:** A

**Explanation:**

AWS Secrets Manager is a service that allows you to store and manage secrets, such as database credentials, API keys, and passwords, in a secure and centralized way. It also provides features such as automatic secret rotation, auditing, and monitoring<sup>1</sup>. 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 34**

A financial company must store original customer records for 10 years for legal reasons. A complete record contains personally identifiable information (PII). According to local regulations, PII is available to only certain people in the company and must not be shared with third parties. The company needs to make the records available to third-party organizations for statistical analysis without sharing the PII. A developer wants to store the original immutable record in Amazon S3. Depending on who accesses the S3 document, the document should be returned as is or with all the PII removed. The developer has written an AWS Lambda function to remove the PII from the document. The function is named `removePii`. What should the developer do so that the company can meet the PII requirements while maintaining only one copy of the document?

- A. Set up an S3 event notification that invokes the `removePii` function when an S3 GET request is made
- 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 made
- 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 console
- F. Select the `removePii` function
- G. Use S3 Access Points to access the object without PII.
- H. Create an S3 access point from the S3 console
- I. Use the access point name to call the `GetObjectLegalHold` S3 API function
- 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 37**

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

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

**Answer:** B

**Explanation:**

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

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

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

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

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

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

Reference: [AWS CodeDeploy lifecycle event hooks reference]

**NEW QUESTION 38**

A company has a web application that runs on Amazon EC2 instances with a custom Amazon Machine Image (AMI). The company uses AWS CloudFormation to provision the application. The application runs in the `us-east-1` Region, and the company needs to deploy the application to the `us-west-1` Region. An attempt to create the AWS CloudFormation stack in `us-west-1` fails. An error message states that the AMI ID does not exist. A developer must resolve this error with a solution that uses the least amount of operational overhead. Which solution meets these requirements?

- A. Change the AWS CloudFormation templates for `us-east-1` and `us-west-1` to use an AWS AMI

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

**Answer: B**

**Explanation:**

<https://aws.amazon.com/blogs/aws/ec2-ami-copy-between-regions/>

**NEW QUESTION 43**

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

Which solution will meet this requirement?

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

**Answer: B**

**Explanation:**

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

[Protecting data using server-side encryption with Amazon S3-managed encryption keys \(SSE-S3\)](#)

**NEW QUESTION 46**

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

- A. Create an Amazon Kinesis data stream, and attach it to the DynamoDB table
- 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 51**

A company built an online event platform For each event the company organizes quizzes and generates leaderboards that are based on the quiz scores. The company stores the leaderboard data in Amazon DynamoDB and retains the data for 30 days after an event is complete The company then uses a scheduled job to delete the old leaderboard data

The DynamoDB table is configured with a fixed write capacity. During the months when many events occur, the DynamoDB write API requests are throttled when the scheduled delete job runs.

A developer must create a long-term solution that deletes the old leaderboard data and optimizes write throughput

Which solution meets these requirements?

- A. Configure a TTL attribute for the leaderboard data
- B. Use DynamoDB Streams to schedule and delete the leaderboard data
- C. Use AWS Step Functions to schedule and delete the leaderboard data.
- D. Set a higher write capacity when the scheduled delete job runs

**Answer: A**

**Explanation:**

"deletes the item from your table without consuming any write throughput" <https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/TTL.html>

**NEW QUESTION 54**

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<sup>1</sup>. The DeploymentPreference property in AWS SAM allows you to specify the type of deployment that you want. The Canary10Percent10Minutes option means that 10 percent of your customer traffic is immediately shifted to your new version. After 10 minutes, all traffic is shifted to the new version<sup>1</sup>. The AutoPublishAlias property in AWS SAM allows AWS SAM to automatically create an alias that points to the updated version of the Lambda function<sup>1</sup>. Therefore, option A is correct.

**NEW QUESTION 59**

A developer is writing an application that will retrieve sensitive data from a third-party system. The application will format the data into a PDF file. The PDF file could be more than 1 MB. The application will encrypt the data to disk by using AWS Key Management Service (AWS KMS). The application will decrypt the file when a user requests to download it. The retrieval and formatting portions of the application are complete.  
 The developer needs to use the GenerateDataKey API to encrypt the PDF file so that the PDF file can be decrypted later. The developer needs to use an AWS KMS symmetric customer managed key for encryption.  
 Which solutions will meet these requirements?

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

**Answer: A**

**Explanation:**

? The GenerateDataKey API returns a data key that is encrypted under a symmetric encryption KMS key that you specify, and a plaintext copy of the same data key<sup>1</sup>. The data key is a random byte string that can be used with any standard encryption algorithm, such as AES or SM42. The plaintext data key can be used to encrypt or decrypt data outside of AWS KMS, while the encrypted data key can be stored with the encrypted data and later decrypted by AWS KMS<sup>1</sup>.  
 ? In this scenario, the developer needs to use the GenerateDataKey API to encrypt the PDF file so that it can be decrypted later. The developer also needs to use an AWS KMS symmetric customer managed key for encryption. To achieve this, the developer can follow these steps:

**NEW QUESTION 64**

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

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

**Answer: C**

**Explanation:**

This solution will meet the requirements by creating a Lambda function execution role, which is an IAM role that grants permissions to a Lambda function to access AWS resources such as Amazon S3 objects. The developer can attach a policy to the role that grants access to specific objects in the S3 bucket that are required by the application, following the principle of least privilege. Option A is not optimal because it will hardcode the credentials that are required to access S3 objects in the application code, which is insecure and difficult to maintain. Option B is not optimal because it will create a secret access key and access key ID with permission to access the S3 bucket, which will introduce additional security risks and complexity for storing and managing credentials. Option D is not optimal because it will store the secret access key and access key ID as environment variables in Lambda, which is also insecure and difficult to maintain. References: [AWS Lambda Execution Role], [Using AWS Lambda with Amazon S3]

**NEW QUESTION 68**

An organization is using Amazon CloudFront to ensure that its users experience low-latency access to its web application. The organization has identified a need to encrypt all traffic between users and CloudFront, and all traffic between CloudFront and the web application.  
 How can these requirements be met? (Select TWO)

- A. Use AWS KMS to encrypt traffic between CloudFront and the web application.
- B. Set the Origin Protocol Policy to "HTTPS Only".
- C. Set the Origin's HTTP Port to 443.

- D. Set the Viewer Protocol Policy to "HTTPS Only" or Redirect HTTP to HTTPS"
- E. Enable the CloudFront option Restrict Viewer Access.

**Answer:** BD

**Explanation:**

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

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

**NEW QUESTION 72**

A developer is deploying an AWS Lambda function. The developer wants the ability to return to older versions of the function quickly and seamlessly. How can the developer achieve this goal with the LEAST operational overhead?

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

**Answer:** B

**Explanation:**

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

**NEW QUESTION 77**

A developer has created an AWS Lambda function that makes queries to an Amazon Aurora MySQL DB instance. When the developer performs a test the DB instance shows an error for too many connections. Which solution will meet these requirements with the LEAST operational effort?

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

**Answer:** D

**Explanation:**

This solution will meet the requirements by using Amazon RDS Proxy, which is a fully managed, highly available database proxy for Amazon RDS that makes applications more scalable, more resilient to database failures, and more secure. The developer can create a proxy in Amazon RDS Proxy, which sits between the application

and the DB instance and handles connection management, pooling, and routing. The developer can query the proxy instead of the DB instance, which reduces the number of open connections to the DB instance and avoids errors for too many connections. Option A is not optimal because it will create a read replica for the DB instance, which may not solve the problem of too many connections as read replicas also have connection limits and may incur additional costs. Option B is not optimal because it will migrate the data to an Amazon DynamoDB database, which may introduce additional complexity and overhead for migrating and accessing data from a different database service. Option C is not optimal because it will configure the Amazon Aurora MySQL DB instance for Multi-AZ deployment, which may improve availability and durability of the DB instance but not reduce the number of connections.

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

**NEW QUESTION 78**

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 83

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 88

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

Which solution will meet these requirements MOST cost-effectively?

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

**Answer:** A

#### Explanation:

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

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

? AWS::StepFunctions::StateMachine - AWS CloudFormation

? Call API Gateway with Step Functions - AWS Step Functions

? amazon-web-services aws-api-gateway terraform aws-step-functions

### NEW QUESTION 90

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

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

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

**Answer:** B

#### Explanation:

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

References:

? [Amazon DynamoDB]

? [Amazon Simple Storage Service (S3)]

### NEW QUESTION 92

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

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

**Answer: B**

**Explanation:**

API Gateway stage variables are name-value pairs that can be defined as configuration attributes associated with a deployment stage of a REST API. They act like environment variables and can be used in the API setup and mapping templates. For example, the development team can define a stage variable named endpoint and assign it different values for each stage, such as dev.example.com for development, uat.example.com for user acceptance testing, and prod.example.com for production. Then, the team can use the stage variable value in the integration request URL, such as http://\$ { stageVariables.endpoint}/api. This way, the team can use the same API setup with different endpoints at each stage by resetting the stage variable value. The other solutions are either not feasible or not cost-effective. Lambda layers are used to package and load dependencies for Lambda functions, not for storing endpoints. Encoding the endpoints as query parameters would expose them to the public and make the request URL unnecessarily long. Storing the endpoints in AWS AppConfig would incur additional costs and complexity, and would require additional logic to retrieve the values from the configuration store. References

- ? Using Amazon API Gateway stage variables
- ? Setting up stage variables for a REST API deployment
- ? Setting stage variables using the Amazon API Gateway console

**NEW QUESTION 94**

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

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

**Answer: D**

**Explanation:**

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

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

**NEW QUESTION 99**

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 104**

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

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

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

**Answer:** C

**Explanation:**

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

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

**NEW QUESTION 107**

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

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

Which solution should the developer use to meet these requirements?

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

**Answer:** A

**Explanation:**

<https://aws.amazon.com/blogs/compute/using-amazon-efs-for-aws-lambda-in-your-serverless-applications/>

**NEW QUESTION 112**

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

How can a developer increase the speed of the Lambda function deployment?

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

**Answer:** B

**Explanation:**

Using Lambda layers is a way to reduce the size of the deployment package and speed up the deployment process. Lambda layers are reusable components that can contain libraries, custom runtimes, or other dependencies. By using layers, the developer can separate the core function logic from the dependencies, and avoid uploading them every time the function code changes. Layers can also be shared across multiple functions or accounts, which can improve consistency and maintainability. References

? Working with AWS Lambda layers

? AWS Lambda Layers Best Practices

? Best practices for working with AWS Lambda functions

**NEW QUESTION 113**

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

Which option will meet these requirements with the HIGHEST level of security?

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

**Answer:** D

**Explanation:**

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

**NEW QUESTION 114**

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

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

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

**Answer: B**

**Explanation:**

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

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

References:

? [What Is AWS Lambda? - AWS Lambda]

? [AWS Lambda Layers - AWS Lambda]

**NEW QUESTION 116**

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

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

**Answer: B**

**Explanation:**

Amazon API Gateway provides tools for creating and documenting web APIs that route HTTP requests to Lambda functions<sup>2</sup>. You can secure access to your API with authentication and authorization controls. Your APIs can serve traffic over the internet or can be accessible only within your VPC<sup>2</sup>. You can override the cache method in the selected stage of API Gateway<sup>2</sup>. Therefore, option B is correct.

**NEW QUESTION 119**

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

There has been an increase in application usage. The third-party API frequently returns an HTTP 429 Too Many Requests error message. The error message prevents a significant number of messages from being processed successfully.

How can the developer resolve this issue?

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

**Answer: D**

**Explanation:**

? Maximum concurrency for SQS as an event source allows customers to control the maximum concurrent invokes by the SQS event source<sup>1</sup>. When multiple SQS event sources are configured to a function, customers can control the maximum concurrent invokes of individual SQS event source<sup>1</sup>.

? In this scenario, the developer needs to resolve the issue of the third-party API frequently returning an HTTP 429 Too Many Requests error message, which prevents a significant number of messages from being processed successfully. To achieve this, the developer can follow these steps:

? By using this solution, the developer can reduce the frequency of HTTP 429 errors and improve the message processing success rate. The developer can also avoid throttling or blocking by the third-party API.

**NEW QUESTION 121**

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

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

**Answer: C**

**Explanation:**

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

Reference: How Application Load Balancer works with your applications

**NEW QUESTION 126**

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 127**

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

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

**Answer: AD**

**Explanation:**

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

- ? Using the Amazon SQS message deduplication ID
- ? Exactly-once processing - Amazon Simple Queue Service
- ? Amazon SQS duplicated messages in queue - Stack Overflow
- ? amazon web services - How long can duplicate SQS messages persist ...
- ? Standard SQS - Duplicate message | AWS re:Post - Amazon Web Services, Inc.

**NEW QUESTION 128**

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 133**

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

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

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

For a given title, get all details about all movies that have that title. For a given genre, get all details about all movies in that genre. Which data store configuration will meet these requirements?

- A. Create an Amazon DynamoDB table
- B. Configure the table with a primary key that consists of the title as the partition key and the release year as the sort key
- 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 table
- E. Configure the table with a primary key that consists of the genre as the partition key and the release year as the sort key

- 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 genre.
- 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 134**

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.
- 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 137**

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

What is the MOST cost-effective way to solve the deployment issue?

- A. Change the Auto Scaling group to six desired instances.
- B. Change the deployment policy to traffic splitting
- C. Specify an evaluation time of 1 hour.
- D. Change the deployment policy to rolling with additional batch
- E. Specify a batch size of 1.
- F. Change the deployment policy to rolling
- G. Specify a batch size of 2.

**Answer:** C

**Explanation:**

This solution will solve the deployment issue by deploying the new version of the application to one new EC2 instance at a time, while keeping the old version running on

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

References: AWS Elastic Beanstalk Deployment Policies

**NEW QUESTION 138**

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 143**

A company is preparing to migrate an application to the company's first AWS environment. Before this migration, a developer is creating a proof-of-concept application to validate a model for building and deploying container-based applications on AWS.

Which combination of steps should the developer take to deploy the containerized proof-of-concept application with the LEAST operational effort? (Select TWO.)

- A. Mastered
- B. Not Mastered

**Answer:** A

**Explanation:**

To deploy a containerized application on AWS with the least operational effort, the developer should package the application into a container image by using the Docker CLI and upload the image to Amazon ECR, which is a fully managed container registry service. Then, the developer should deploy the application to Amazon ECS on AWS Fargate, which is a serverless compute engine for containers that eliminates the need to provision and manage servers or clusters. Amazon ECS will automatically scale, load balance, and monitor the application. References

? How to Deploy Docker Containers | AWS

? Deploy a Web App Using AWS App Runner

? How to Deploy Containerized Apps on AWS Using ECR and Docker

**NEW QUESTION 144**

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

Which set of steps would be necessary to achieve this?

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

**Answer:** B

**Explanation:**

Amazon S3 is a service that provides highly scalable, durable, and secure object storage. Amazon DynamoDB is a fully managed NoSQL database service that

provides fast and consistent performance with seamless scalability. AWS Lambda is a service that lets developers run code without provisioning or managing servers. The developer can configure an S3 event to invoke a Lambda function that inserts records into DynamoDB whenever a new file is added to the S3 bucket. This solution will meet the requirement of inserting a record into DynamoDB as soon as a new file is added to S3. References:

? [Amazon Simple Storage Service (S3)]

? [Amazon DynamoDB]

? [What Is AWS Lambda? - AWS Lambda]

? [Using AWS Lambda with Amazon S3 - AWS Lambda]

#### NEW QUESTION 147

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

Which solution will meet these requirements'?

A. Mastered

B. Not Mastered

**Answer: A**

#### Explanation:

Creating an Amazon ElastiCache for Redis cluster is the best solution for improving the application's performance. Redis is an in-memory data store that can serve read requests with minimal latency and handle complex data types, such as lists, sets, hashes, and streams. By using a write-through caching strategy, the application can ensure that the data in Redis is always consistent with the data in RDS. The application can read the data from Redis instead of RDS, reducing the load on the database and improving the response time. The other solutions are either not feasible or not effective. Amazon DynamoDB Accelerator (DAX) is a caching service that works only with DynamoDB, not RDS. Amazon S3 Transfer Acceleration is a feature that speeds up data transfers between S3 and clients across the internet, not between RDS and the application. Amazon CloudFront is a content delivery network that can cache static content, such as images, videos, or HTML files, but not dynamic content, such as user posts and interactions. References:

? Amazon ElastiCache for Redis

? Caching Strategies and Best Practices - Amazon ElastiCache for Redis

? Using Amazon ElastiCache for Redis with Amazon RDS

? Amazon DynamoDB Accelerator (DAX)

? Amazon S3 Transfer Acceleration

? Amazon CloudFront

#### NEW QUESTION 149

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

What is the recommended solution?

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

**Answer:** D

**Explanation:**

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

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

**NEW QUESTION 153**

A developer is modifying an existing AWS Lambda function. While checking the code, the developer notices hardcoded parameter values for an Amazon RDS for SQL Server user name, password, database, host, and port. There also are hardcoded parameter values for an Amazon DynamoDB table, an Amazon S3 bucket, and an Amazon Simple Notification Service (Amazon SNS) topic.

The developer wants to securely store the parameter values outside the code in an encrypted format and wants to turn on rotation for the credentials. The developer also wants to be able to reuse the parameter values from other applications and to update the parameter values without modifying code.

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

- A. Create an RDS database secret in AWS Secrets Manager
- B. Set the user name, password, database, host, and port
- C. Turn on secret rotation
- D. Create encrypted Lambda environment variables for the DynamoDB table, S3 bucket, and SNS topic.
- E. Create an RDS database secret in AWS Secrets Manager
- F. Set the user name, password, database, host, and port

- G. Turn on secret rotation
- H. Create Secure String parameters in AWS Systems Manager Parameter Store for the DynamoDB table, S3 bucket and SNS topic.
- I. Create RDS database parameters in AWS Systems Manager Parameter Store for the user name password, database, host and port
- J. Store for the user name password, database, host and port
- K. Create encrypted Lambda environment variables for the DynamoDB table, S3 bucket, and SNS topic
- L. Create a Lambda function and set the logic for the credentials rotation task Schedule the credentials rotation task in Amazon EventBridge.
- M. Create RDS database parameters in AWS Systems Manager Parameter Store for the user name password database, host, and port
- N. Store for the user name password database, host, and port
- O. Store the DynamoDB table
- P. S3 bucket, and SNS topic in Amazon S3 Create a Lambda function and set the logic for the credentials rotation Invoke the Lambda function on a schedule.

**Answer: B**

**Explanation:**

This solution will meet the requirements by using AWS Secrets Manager and AWS Systems Manager Parameter Store to securely store the parameter values outside the code in an encrypted format. AWS Secrets Manager is a service that helps protect secrets such as database credentials by encrypting them with AWS Key Management Service (AWS KMS) and enabling automatic rotation of secrets. The developer can create an RDS database secret in AWS Secrets Manager and set the user name, password, database, host, and port for accessing the RDS database. The developer can also turn on secret rotation, which will change the database credentials periodically according to a specified schedule or event. AWS Systems Manager Parameter Store is a service that provides secure and scalable storage for configuration data and secrets. The developer can create Secure String parameters in AWS Systems Manager Parameter Store for the DynamoDB table, S3 bucket, and SNS topic, which will encrypt them with AWS KMS. The developer can also reuse the parameter values from other applications and update them without modifying code. Option A is not optimal because it will create encrypted Lambda

environment variables for the DynamoDB table, S3 bucket, and SNS topic, which may not be reusable or updatable without modifying code. Option C is not optimal because it will create RDS database parameters in AWS Systems Manager Parameter Store, which does not support automatic rotation of secrets. Option D is not optimal because it will store the DynamoDB table, S3 bucket, and SNS topic in Amazon S3, which may introduce additional costs and complexity for accessing configuration data. References: AWS Secrets Manager, [AWS Systems Manager Parameter Store]

**NEW QUESTION 155**

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

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

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

**Answer: C**

**Explanation:**

This step should be completed prior to deploying the application because it prepares the application artifacts for deployment. The AWS Serverless Application Model (AWS SAM) is a framework that simplifies building and deploying serverless applications on AWS. The AWS SAM CLI is a command-line tool that helps you create, test, and deploy serverless applications using AWS SAM templates. The `sam package` command bundles the application artifacts, such as Lambda function code and API definitions, and uploads them to an Amazon S3 bucket. The command also returns a CloudFormation template that is ready to be deployed with the `sam deploy` command. Compressing the application to a zip file and uploading it to AWS Lambda will not work because it does not use AWS SAM templates or CloudFormation. Testing the new Lambda function by first tracing it in AWS X-Ray will not prepare the application for deployment, but only monitor its performance and errors. Creating the application environment using the `eb create my-env` command will not work because it is a command for AWS Elastic Beanstalk, not AWS SAM.

**NEW QUESTION 160**

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

**Answer: B**

**Explanation:**

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

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

**NEW QUESTION 162**

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

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

Which solution will meet these requirements?

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

**Answer: D**

**Explanation:**

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

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

**NEW QUESTION 165**

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

Which solution will meet these requirements MOST efficiently?

- A. Create a Lambda environment variable to store the table name
- B. Use the standard method for the programming language to retrieve the variable.
- C. Store the table name in a file
- D. Store the file in the /tmp folder
- E. Use the SDK for the programming language to retrieve the table name.
- F. Create a file to store the table name
- G. Zip the file and upload the file to the Lambda layer
- H. Use the SDK for the programming language to retrieve the table name.
- I. Create a global variable that is outside the handler in the Lambda function to store the table name.

**Answer: A**

**Explanation:**

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

Reference: Using AWS Lambda environment variables

**NEW QUESTION 167**

A developer is using an AWS Lambda function to generate avatars for profile pictures that are uploaded to an Amazon S3 bucket. The Lambda function is

automatically invoked for profile pictures that are saved under the /original/ S3 prefix. The developer notices that some pictures cause the Lambda function to time out. The developer wants to implement a fallback mechanism by using another Lambda function that resizes the profile picture.

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

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

**Answer:** A

**Explanation:**

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

Reference: Using AWS Lambda destinations

**NEW QUESTION 168**

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 173**

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 177**

An application runs on multiple EC2 instances behind an ELB.

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

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

**Answer:** A

**Explanation:**

The solution that will meet the requirements is to write data to Amazon ElastiCache. This way, the application can write session data to a fast, scalable, and reliable in-memory data store that can be served reliably across multiple requests. The other options either involve writing data to persistent storage, which is slower and more expensive than in-memory storage, or writing data to the root filesystem, which is not shared among multiple EC2 instances.

Reference: Using ElastiCache for session management

**NEW QUESTION 182**

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

only by using these VPC endpoints.

Which solution will meet these requirements?

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

allowed.

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 185**

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

A developer needs to implement a solution to store the access token. The access token must be encrypted at rest and in transit. The access token must also be accessible from other AWS accounts.

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

- A. Use an AWS Systems Manager Parameter Store SecureString parameter that uses an AWS Key Management Service (AWS KMS) AWS managed key to store the access toke
- B. Add a resource-based policy to the parameter to allow access from other account
- C. Update the IAM role of the EC2 instances with permissions to access Parameter Stor
- D. Retrieve the token from Parameter Store with the decrypt flag enable
- E. Use the decrypted access token to send the message to the chat.
- F. Encrypt the access token by using an AWS Key Management Service (AWS KMS) customer managed ke
- G. Store the access token in an Amazon DynamoDB tabl
- H. Update the IAM role of the EC2 instances with permissions to access DynamoDB and AWS KM
- I. Retrieve the token from DynamoD
- J. Decrypt the token by using AWS KMS on the EC2 instance
- K. Use the decrypted access token to send the message to the chat.
- L. Use AWS Secrets Manager with an AWS Key Management Service (AWS KMS) customer managed key to store the access toke
- M. Add a resource-based policy to the secret to allow access from other account
- N. Update the IAM role of the EC2 instances with permissions to access Secrets Manage
- O. Retrieve the token from Secrets Manage
- P. Use the decrypted access token to send the message to the chat.
- Q. Encrypt the access token by using an AWS Key Management Service (AWS KMS) AWS managed ke
- R. Store the access token in an Amazon S3 bucke
- S. Add a bucket policy to the S3 bucket to allow access from other account
- T. Update the IAM role of the EC2 instances with permissions to access Amazon S3 and AWS KM
- . Retrieve the token from the S3 bucke
- . Decrypt the token by using AWS KMS on the EC2 instance
- . Use the decrypted access token to send the message to the chat.

**Answer: C**

**Explanation:**

<https://aws.amazon.com/premiumsupport/knowledge-center/secrets-manager-share-between-accounts/>  
[https://docs.aws.amazon.com/secretsmanager/latest/userguide/auth-and-access\\_examples\\_cross.html](https://docs.aws.amazon.com/secretsmanager/latest/userguide/auth-and-access_examples_cross.html)

**NEW QUESTION 189**

A company runs an application on AWS The application stores data in an Amazon DynamoDB table Some queries are taking a long time to run These slow queries involve an attribute that is not the table's partition key or sort key

The amount of data that the application stores in the DynamoDB table is expected to increase significantly. A developer must increase the performance of the queries.

Which solution will meet these requirements'?

- A. Increase the page size for each request by setting the Limit parameter to be higher than the default value Configure the application to retry any request that exceeds the provisioned throughput.
- B. Create a global secondary index (GSI). Set query attribute to be the partition key of the index
- C. Perform a parallel scan operation by issuing individual scan requests in the parameters specify the segment for the scan requests and the total number of segments for the parallel scan.
- D. Turn on read capacity auto scaling for the DynamoDB tabl
- E. Increase the maximum read capacity units (RCUs).

**Answer: B**

**Explanation:**

Creating a global secondary index (GSI) is the best solution to improve the performance of the queries that involve an attribute that is not the table's partition key or sort key. A GSI allows you to define an alternate key for your table and query the data using that key. This way, you can avoid scanning the entire table and reduce the latency and cost of your queries. You should also follow the best practices for designing and using GSIs in DynamoDB12. References

- ? Working with Global Secondary Indexes - Amazon DynamoDB
- ? DynamoDB Performance & Latency - Everything You Need To Know

**NEW QUESTION 194**

.....

## Relate Links

**100% Pass Your DVA-C02 Exam with Exam Bible Prep Materials**

<https://www.exambible.com/DVA-C02-exam/>

## Contact us

We are proud of our high-quality customer service, which serves you around the clock 24/7.

Viste - <https://www.exambible.com/>