



Free Questions for DVA-C02 by actualtestdumps

Shared by Barry on 24-05-2024

For More Free Questions and Preparation Resources

Check the Links on Last Page

Question 1

Question Type: MultipleChoice

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

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

Options:

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

Answer:

A

Explanation:

Pseudo Parameters:CloudFormation provides pseudo parameters that reference runtime context, including the current AWS Region.

Operational Efficiency:TheAWS::Regionpseudo parameter offers the most direct and self-contained way to obtain the Region dynamically within the template.

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

Question 2

Question Type: MultipleChoice

A developer maintains a critical business application that uses Amazon DynamoDB as the primary data store The DynamoDB table contains millions of documents and receives 30-60 requests each minute The developer needs to perform processing in near-real time on the documents when they are added or updated in the DynamoDB table

How can the developer implement this feature with the LEAST amount of change to the existing application code?

Options:

A- Set up a cron job on an Amazon EC2 instance Run a script every hour to query the table for changes and process the documents

- B-** Enable a DynamoDB stream on the table Invoke an AWS Lambda function to process the documents.
- C-** Update the application to send a PutEvents request to Amazon EventBridge. Create an EventBridge rule to invoke an AWS Lambda function to process the documents.
- D-** Update the application to synchronously process the documents directly after the DynamoDB write

Answer:

B

Explanation:

DynamoDB Streams: Capture near real-time changes to DynamoDB tables, triggering downstream actions.

Lambda for Processing: Lambda functions provide a serverless way to execute code in response to events like DynamoDB Stream updates.

Minimal Code Changes: This solution requires the least modifications to the existing application.

DynamoDB Streams: <https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/Streams.html>

AWS Lambda: <https://aws.amazon.com/lambda/>

Question 3

Question Type: MultipleChoice

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'?

Options:

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

Answer:

B

Explanation:

Global Secondary Index (GSI):GSIs enable alternative query patterns on a DynamoDB table by using different partition and sort keys.

Addressing Query Bottleneck:By making the slow-query attribute the GSI's partition key, you optimize queries on that attribute.

Scalability:GSIs automatically scale to handle increasing data volumes.

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

Question 4

Question Type: MultipleChoice

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.)

Options:

- A-** Package the application into a zip file by using a command line tool Upload the package to Amazon S3
- B-** Package the application into a container image by using the Docker CLI. Upload the image to Amazon Elastic Container Registry (Amazon ECR)
- C-** Deploy the application to an Amazon EC2 instance by using AWS CodeDeploy.
- D-** Deploy the application to Amazon Elastic Kubernetes Service (Amazon EKS) on AWS Fargate
- E-** Deploy the application to Amazon Elastic Container Service (Amazon ECS) on AWS Fargate

Answer:

B, E

Explanation:

Containerization:Packaging the application as a container image promotes portability and standardization. Docker is the standard tool for containerization.

Amazon ECR:ECR is a managed container registry designed to work seamlessly with AWS container services.

Fargate:ECS Fargate provides serverless container orchestration, minimizing operational overhead for this proof-of-concept.

Docker:<https://www.docker.com/>

Amazon ECR:<https://aws.amazon.com/ecr/>

Question 5

Question Type: MultipleChoice

A company has an analytics application that uses an AWS Lambda function to process transaction data asynchronously. A developer notices that asynchronous invocations of the Lambda function sometimes fail. When failed Lambda function invocations occur, the developer wants to invoke a second Lambda function to handle errors and log details.

Which solution will meet these requirements?

Options:

- A-** Configure a Lambda function destination with a failure condition. Specify Lambda function as the destination type. Specify the error-handling Lambda function's Amazon Resource Name (ARN) as the resource.
- B-** Enable AWS X-Ray active tracing on the initial Lambda function. Configure X-Ray to capture stack traces of the failed invocations. Invoke the error-handling Lambda function by including the stack traces in the event object.
- C-** Configure a Lambda function trigger with a failure condition. Specify Lambda function as the destination type. Specify the error-handling Lambda function's Amazon Resource Name (ARN) as the resource.
- D-** Create a status check alarm on the initial Lambda function. Configure the alarm to invoke the error-handling Lambda function when the alarm is initiated. Ensure that the alarm passes the stack trace in the event object.

Answer:

A

Explanation:

Lambda Destinations on Failure: Allow routing asynchronous function invocations to specified resources (like another Lambda function) upon failure.

Error Handling: The error-handling Lambda receives details about the failure, enabling logging and custom actions.

Direct Integration: This solution leverages native Lambda functionality for a simpler implementation.

Question 6

Question Type: MultipleChoice

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

Options:

- A- AssumeRoleWithWebidentity
- B- GetFederationToken
- C- AssumeRoleWithSAML
- D- AssumeRole

Answer:

D

Explanation:

AWS STS AssumeRole: The central operation for assuming temporary security credentials, commonly used for cross-account access.

MFA Integration: The AssumeRole call can include MFA information to enforce multi-factor authentication.

Credentials for S3 Access: The returned temporary credentials would provide the necessary permissions to access the S3 bucket in the other account.

[AWS STS AssumeRole Documentation: https://docs.aws.amazon.com/STS/latest/APIReference/API_AssumeRole.html](https://docs.aws.amazon.com/STS/latest/APIReference/API_AssumeRole.html)

Question 7

Question Type: MultipleChoice

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?

Options:

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

Answer:

A

Explanation:

DynamoDB TTL (Time-to-Live): A native feature that automatically deletes items after a specified expiration time.

Efficiency: Eliminates the need for scheduled deletion jobs, optimizing write throughput by avoiding potential throttling conflicts.

Seamless Integration: TTL works directly within DynamoDB, requiring minimal development overhead.

DynamoDB TTL Documentation: <https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/TTL.html>

Question 8

Question Type: MultipleChoice

A developer is creating an AWS Lambda function. The Lambda function needs an external library to connect to a third-party solution. The external library is a collection of files with a total size of 100 MB. The developer needs to make the external library available to the Lambda execution environment and reduce the Lambda package space.

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

Options:

- A-** Create a Lambda layer to store the external library Configure the Lambda function to use the layer
- B-** Create an Amazon S3 bucket Upload the external library into the S3 bucket. Mount the S3 bucket folder in the Lambda function Import the library by using the proper folder in the mount point.
- C-** Load the external library to the Lambda function's /tmp directory during deployment of the Lambda package. Import the library from the /tmp directory.
- D-** Create an Amazon Elastic File System (Amazon EFS) volume. Upload the external library to the EFS volume Mount the EFS volume in the Lambda function. Import the library by using the proper folder in the mount point.

Answer:

A

Explanation:

Lambda Layers:These are designed to package dependencies that you can share across functions.

How to Use:

Create a layer, upload your 100MB library as a zip.

Attach the layer to your function.

In your function code, import the library from the standard layer path.

[Lambda Layers:https://docs.aws.amazon.com/lambda/latest/dg/configuration-layers.html](https://docs.aws.amazon.com/lambda/latest/dg/configuration-layers.html)

To Get Premium Files for DVA-C02 Visit

<https://www.p2pexams.com/products/dva-c02>

For More Free Questions Visit

<https://www.p2pexams.com/amazon/pdf/dva-c02>

