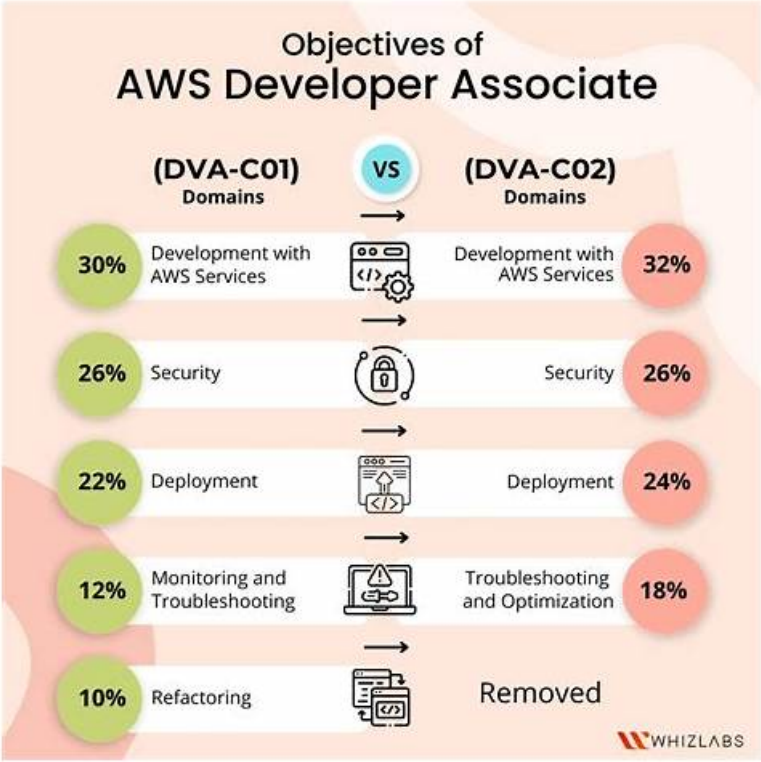
100% Pass 2026 The Best Amazon DVA-C02: AWS Certified Developer - Associate Reliable Exam Questions



P.S. Free & New DVA-C02 dumps are available on Google Drive shared by PrepAwayExam: https://drive.google.com/open?id=1I5ZjG7u7BKuBlM7JTKwh0mguSMkY9BQG

The punishment received by laziness is not only its own failure, but also the success of others. No one wants to be inferior to others. So, it's time to change yourself and make yourself better! Our DVA-C02 study materials want to give you some help on your dream journey. Believe me, the help you get is definitely what you need. On one hand, you can easily pass the DVA-C02 Exam and get the according DVA-C02 certification. On the other hand, you will be definitely encouraged to make better progress from now on.

To prepare for the Amazon DVA-C02 exam, candidates are recommended to have at least one year of experience in developing and maintaining applications on AWS. They can also take advantage of the training and resources provided by AWS, such as online courses, practice exams, and study guides. Passing the Amazon DVA-C02 Exam requires a score of at least 720 out of 1000, and the certification is valid for two years. Overall, the Amazon DVA-C02 certification is a valuable credential for developers who want to advance their careers in cloud computing and demonstrate their expertise in AWS.

>> DVA-C02 Reliable Exam Questions <<

Get Amazon DVA-C02 Practice Test To Gain Brilliant Result [2026]

AWS Certified Developer - Associate DVA-C02 study guide are high quality, since we have a professional team to collect the information for the exam, and we can ensure you that DVA-C02 study guide you receive are the latest information we have. In order to strengthen your confidence for Amazon DVA-C02 Exam Dumps, we are pass guarantee and money back guarantee.

Amazon AWS Certified Developer - Associate Sample Questions (Q246-Q251):

NEW QUESTION # 246
A developer is building an application that uses an Amazon RDS for PostgreSQL database. To meet security requirements, the developer needs to ensure that data is encrypted at rest. The developer must be able to rotate the encryption keys on demand.

- A. Create a 256-bit AES-GCM encryption key. Store the key in AWS Secrets Manager, and enable managed rotation. Use the key to encrypt the database.
- B. Create a 256-bit AES-GCM encryption key. Store the key in AWS Secrets Manager. Configure an AWS Lambda function to perform key rotation. Use the key to encrypt the database.
- C. Create a symmetric customer managed AWS KMS key. Use the key to encrypt the database.
- D. Use an AWS KMS managed encryption key to encrypt the database.

Answer: C

Explanation:
Comprehensive Detailed Explanation with all AWS References
* Why Option B is Correct:A customer-managed AWS Key Management Service (KMS) key allows for encryption at rest and provides the ability to rotate the key on demand. This ensures compliance with security requirements for key management and database encryption.
* RDS integrates natively with AWS KMS, allowing the use of a customer-managed key for encrypting data at rest.
* Key rotation can be managed directly in AWS KMS without needing custom solutions.
* Why Other Options are Incorrect:
* Option A: AWS KMS managed encryption keys (AWS-owned keys) do not support key rotation on demand.
* Option C & D: Storing keys in AWS Secrets Manager with custom rotation is not a recommended approach for database encryption. AWS KMS is designed specifically for secure key management and encryption.
* AWS Documentation References:
* Encrypting Amazon RDS Resources
* AWS Key Management Service (KMS)

NEW QUESTION # 247
A company is building a new application that runs on AWS and uses Amazon API Gateway to expose APIs Teams of developers are working on separate components of the application in parallel The company wants to publish an API without an integrated backend so that teams that depend on the application backend can continue the development work before the API backend development is complete.
Which solution will meet these requirements?

- A. Create an EC2 application that returns mocked HTTP responses Create API Gateway resources and set the integration type value to AWS Create an API Gateway stage and deploy the API.
- B. Create an AWS Lambda function that returns mocked responses and various HTTP status codes. Create API Gateway resources and set the integration type value to AWS_PROXY Deploy the API.
- C. Create API Gateway resources and set the integration type value to MOCK Configure the method integration request and integration response to associate a response with an HTTP status code Create an API Gateway stage and deploy the API.
- D. Create API Gateway resources and set the integration type value set to HTTP_PROXY. Add mapping templates and deploy the API. Create an AWS Lambda layer that returns various HTTP status codes Associate the Lambda layer with the API deployment

Answer: C

Explanation:
API Gateway Mocking: This feature is built for decoupling development dependencies. Here's the process:
Create resources and methods in your API Gateway.
Set the integration type to 'MOCK'.
Define Integration Responses, mapping HTTP status codes to desired mocked responses (JSON, etc.).
Deployment and Use:
Create a deployment stage for the API.
Frontend teams can call this API and get the mocked responses without a real backend.
Reference:
Mocking API Gateway APIs: https://docs.aws.amazon.com/apigateway/latest/developerguide/how-to-mock-integration.html

NEW QUESTION # 248
A company is building an application for stock trading. The application needs sub-millisecond latency for processing trade requests. The company uses Amazon DynamoDB to store all the trading data that is used to process each trading request A development team performs load testing on the application and finds that the data retrieval time is higher than expected. The development team needs a solution that reduces the data retrieval time with the least possible effort.

Which solution meets these requirements'?

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

Answer: B

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

NEW QUESTION # 249
A mobile app stores blog posts in an Amazon DynacnoDB table Millions of posts are added every day and each post represents a single item in the table. The mobile app requires only recent posts. Any post that is older than 48 hours can be removed.
What is the MOST cost-effective way to delete posts that are older man 48 hours?

- A. For each item add a new attribute of type. String that has a timestamp that its set to the blog post creation time. Create a script to find old posts with a table scan and remove posts that are Oder than 48 hours by using the Batch Write item API operating. Place the script in a container image. Schedule an Amazon Elastic Container Service (Amazon ECS) task on AWS Far gate that invokes the container every 5 minutes.
- B. For each item, add a new attribute of type Date that has a timestamp that is set to 48 hours after the blog post creation time. Create a global secondary index (GSI) that uses the new attribute as a sort key. Create an AWS Lambda function that references the GSI and removes expired items by using the Batch Write item API operation Schedule me function with an Amazon CloudWatch event every minute.
- C. For each item add a new attribute of type. Number that has timestamp that is set to 48 hours after the blog post. creation time Configure the DynamoDB table with a TTL that references the new attribute.
- D. For each item add a new attribute of type String that has a timestamp that is set to the blog post creation time. Create a script to find old posts with a table scan and remove posts that are order than 48 hours by using the Balch Write ltem API operation. Schedule a cron job on an Amazon EC2 instance once an hour to start the script.

Answer: C

Explanation:
This solution will meet the requirements by using the Time to Live (TTL) feature of DynamoDB, which enables automatically deleting items from a table after a certain time period. The developer can add a new attribute of type Number that has a timestamp that is set to 48 hours after the blog post creation time, which represents the expiration time of the item. The developer can configure the DynamoDB table with a TTL that references the new attribute, which instructs DynamoDB to delete the item when the current time is greater than or equal to the expiration time. This solution is also cost-effective as it does not incur any additional charges for deleting expired items. Option A is not optimal because it will create a script to find and remove old posts with a table scan and a batch write item API operation, which may consume more read and write capacity units and incur more costs. Option B is not optimal because it will use Amazon Elastic Container Service (Amazon ECS) and AWS Fargate to run the script, which may introduce additional costs and complexity for managing and scaling containers. Option C is not optimal because it will create a global secondary index (GSI) that uses the expiration time as a sort key, which may consume more storage space and incur more costs.

NEW QUESTION # 250
A Developer has been asked to make changes to the source code of an AWS Lambda function.
The function is managed using an AWS CloudFormation template. The template is configured to load the source code from an Amazon S3 bucket. The Developer manually created a .ZIP file deployment package containing the changes and put the file into the

correct location on Amazon S3. When the function is invoked, the code changes have not been applied.
What step is required to update the function with the changes?

- A. Update the CloudFormation stack with the correct values for the function code properties S3Bucket, S3Key, or S3ObjectVersion.
- B. Ensure that the function source code is base64-encoded before uploading the deployment package to S3.
- C. Delete the .ZIP file on S3, and re-upload by using a different object key name.
- D. Modify the execution role of the Lambda function to allow S3 access permission to the deployment package .ZIP file.

Answer: A

Explanation:
Changes to a deployment package in Amazon S3 are not detected automatically during stack updates. To update the function code, change the object key or version in the template.
https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/aws-properties-lambda- function-code.html

NEW QUESTION # 251
.....