Amazon DVA-C03 Dumps

Shared by Montgomery on 26-11-2026

For More Free Questions and Preparation Resources

Check the Links on Last Page

Question 1

Question Type: MultipleChoice

A stock trading company is developing a mobile app that provides near real-time market data and historical trading analytics to thousands of professional traders who make trades concurrently. The app needs to provide market data to traders in under 100 milliseconds. Historical analytics charts can take 2--3 seconds to generate. The app needs to be highly available and handle market events that cause sudden spikes in trading activity. Some trades involve messages that are over 1 MB in size.

The company must provide equal market access to all traders. The company must ensure that all traders experience the same performance when they use the app.

Which solution will meet these requirements?

Options:

A- Use Amazon ElastiCache Redis OSS in cluster mode and include read replicas. Implement write-through caching for market data updates and lazy loading with a TTL for analytics results.
B- Implement client-side caching by using browser local storage. Use Amazon CloudFront to cache static content.
C- Store all data in Amazon DynamoDB by using DynamoDB Accelerator (DAX). Enable DynamoDB global tables for cross-Region replication.
D- Deploy Amazon ElastiCache Redis OSS in single-node mode. Implement a cache-aside pattern with periodic bulk updates.

Answer:

A

Question 2

Question Type: MultipleChoice

A developer receives reports about slow performance in an application that uses Amazon DynamoDB. The developer reviews the application and finds a function that uses a scan operation on a large table that has more than 350,000 rows. The function returns only one or two items.

The developer needs to improve the application's performance.

Which solution will meet this requirement in the MOST operationally efficient way?

Options:

A- Increase the provisioned read capacity units (RCUs) for the DynamoDB table.
B- Implement Amazon ElastiCache between the application and DynamoDB.
C- Replace the scan operation with a query that uses the partition and sort keys.
D- Enable DynamoDB Streams to asynchronously process the data.

Answer:

C

Explanation:

The correct answer is C because DynamoDB Scan operations are inefficient on large tables. A Scan reads every item in the table or index and then filters results, which is exactly why performance is poor when only one or two records are needed from more than 350,000 rows. AWS recommends using Query when the access pattern can be satisfied by a partition key and, when needed, a sort key condition because Query provides efficient access to the physical locations where matching data is stored. Increasing RCUs only masks the bad access pattern and increases cost. ElastiCache adds operational complexity and does not fix the inefficient DynamoDB request. DynamoDB Streams captures item changes; it does not improve read-path lookup performance.

Question 3

Question Type: MultipleChoice

A company has a three-tier application that is deployed in Amazon ECS. The application is using an Amazon RDS for MySQL DB instance. The application performs more database reads than writes.

During times of peak usage, the application's performance degrades. When this performance degradation occurs, the DB instance's ReadLatency metric in Amazon CloudWatch increases suddenly.

How should a developer modify the application to improve performance?

Options:

A- Use Amazon ElastiCache to cache query results.
B- Scale the ECS cluster to contain more ECS instances.

C- Add read capacity units (RCUs) to the DB instance.
D- Modify the ECS task definition to increase the task memory.

Answer:

A

Explanation:

The bottleneck is the database read path, shown by the sudden increase in the RDS ReadLatency metric. Because the application performs more reads than writes, caching frequently requested query results with Amazon ElastiCache is the most direct way to reduce database load and improve response time. Scaling ECS tasks or instances might increase application capacity, but it can also increase read pressure on the database and worsen latency. RDS for MySQL does not use DynamoDB-style read capacity units, so option C is invalid. Increasing ECS task memory helps only if the application containers are memory constrained, which is not indicated. AWS performance guidance identifies caching as a way to improve read latency, read throughput, and efficiency for read-heavy workloads. (AWS Documentation)

==========

Question 4

Question Type: MultipleChoice

A company has a monolithic desktop-based application that processes images. A developer is converting the application into an AWS Lambda function by using Python. Currently, the desktop application runs every 5 minutes to process the latest image from an Amazon S3 bucket. The desktop application completes the image processing task within 1 minute.

During testing on AWS, the developer notices that the Lambda function runs at the specified 5-minute interval. However, the Lambda function takes more than 2 minutes to complete the image processing task. The developer needs a solution that will improve the Lambda function's performance.

Which solution will meet this requirement?

Options:

A- Update the instance type of the Lambda function to a compute optimized instance with at least eight virtual CPUs (vCPUs).
B- Update the configuration of the Lambda function to use the latest Python runtime.

C- Increase the memory that is allocated to the Lambda function.
D- Configure reserved concurrency on the Lambda function.

Answer:

C

Explanation:

Lambda does not allow the developer to select an EC2 instance type, so option A is invalid. For Lambda, CPU allocation scales proportionally with memory. If the image-processing workload is CPU-bound or memory-constrained, increasing the configured memory can also increase available CPU power and reduce execution duration. Updating the Python runtime may provide minor improvements but is not the primary Lambda performance control. Reserved concurrency controls how many concurrent executions are reserved for the function; it does not make one invocation process faster. The issue is that each run takes too long, not that the function lacks concurrency. AWS Lambda documentation confirms that increasing function memory also increases CPU allocation, and memory can be tuned to improve performance. (AWS Documentation)==========

Question 5

Question Type: MultipleChoice

A company wants to migrate its web application to AWS and leverage auto scaling to handle peak workloads. The solutions architect determined that the best metric for an auto scaling event is the number of concurrent users.

Based on this information, what should the developer use to auto scale based on concurrent users?

Options:

A- An Amazon SNS topic to be invoked when a concurrent user threshold is met
B- An Amazon CloudWatch NetworkIn metric
C- Amazon CloudFront to leverage AWS edge locations
D- A custom Amazon CloudWatch metric for concurrent users

Answer:

D

Explanation:

Concurrent users is an application-level metric, not a default infrastructure metric. CloudWatch provides standard metrics such as CPU utilization, network input, and request counts for many AWS services, but it will not automatically know how many active users the application currently has unless the application publishes that value. The correct design is to publish concurrent user count as a custom CloudWatch metric and use that metric in the scaling policy. SNS is a notification service, not a metric source. CloudFront can improve content delivery, but it does not directly define application auto scaling based on concurrent users. NetworkIn might correlate with load but is not the metric chosen by the architect. CloudWatch supports custom metrics and alarms for threshold-based actions. (AWS Documentation)

==========

Question 6

Question Type: MultipleChoice

A healthcare company develops a patient monitoring application that uses AWS AppSync and an AWS Lambda function. The application stores patient data in an Amazon DynamoDB database. The application processes real-time patient vital signs from thousands of IoT devices. The application must support queries from medical staff who access patient historical dat

a. A developer notices significant latency when medical staff query patient histories, specifically for frequently-accessed records of patients currently under observation. The developer observes that the Lambda functions are consuming high read capacity units (RCUs) from DynamoDB. The developer needs to optimize the application's performance and maintain data consistency. Which solution will meet these requirements?

Options:

A- Enable response caching in AppSync with a TTL value of 1 hour for all queries. Update the Lambda code to use in-memory caching.
B- Increase the Lambda function memory allocation to maximum. Configure provisioned concurrency for the function.
C- Create a caching Lambda function that aggregates patient data periodically and stores it in Amazon S3. Implement a read-through cache pattern.
D- Implement in-memory caching by using Amazon ElastiCache for frequently-accessed patient records. Configure the Lambda functions to check the cache before querying the database. Set up cache invalidation when patient data is updated.

Answer:

D

Explanation:

For frequently accessed data that causes high RCUs and latency in DynamoDB, ElastiCache (Redis or Memcached) is the recommended solution. It sits in front of DynamoDB as a high-speed, in-memory cache. Implementing a 'Cache-Aside' or 'Read-Through' pattern reduces the load on DynamoDB and provides sub-millisecond response times for queries. Invalidation ensures that when vital signs are updated, the cache is refreshed, maintaining data consistency.

Question 7

Question Type: MultipleChoice

A social media application stores millions of video clips in Amazon S3 and serves them to users across the world. The traffic to the application varies, but access often increases suddenly to more than 10,000 PUT requests and GET requests each second. As the application has grown in popularity, users report poor quality of video streaming. Which solution will provide the LARGEST improvement in performance?

Options:

A- Create an Amazon Route 53 geolocation routing policy.
B- Duplicate content in multiple AWS Regions by using S3 Cross-Region Replication.
C- Use S3 Intelligent-Tiering to move data based on access patterns.
D- Create an Amazon CloudFront distribution with Amazon S3 as an origin.

Answer:

D

Explanation:

The correct answer is D because Amazon CloudFront is the AWS content delivery network (CDN) designed to improve performance for users who access content globally. In this scenario, the application serves millions of video clips from Amazon S3 to users around the world, and the main problem is poor streaming quality as traffic scales. CloudFront addresses this by caching content at edge locations that are geographically closer to users, reducing latency and improving

throughput.

AWS documentation emphasizes that CloudFront works especially well with Amazon S3 as an origin for static and streaming content. Instead of every viewer retrieving video data directly from the S3 bucket's Region, CloudFront serves cached objects from nearby edge locations. This provides a major performance boost for global audiences and helps absorb large traffic spikes more efficiently.

Option A is incorrect because Route 53 geolocation routing directs DNS queries based on location, but it does not cache or accelerate video delivery. Option B can improve regional redundancy, but simply replicating objects across Regions does not automatically provide fast global delivery or edge caching. Option C is about storage cost optimization, not delivery performance.

CloudFront is also well suited for sudden spikes in traffic because AWS's edge network is built for high-scale distribution. For video streaming workloads, the reduction in latency and the ability to serve content from nearby cached locations usually provide the largest performance improvement compared to the other options.

Therefore, the best solution is to create an Amazon CloudFront distribution with Amazon S3 as the origin, making D the correct answer.

To Get Premium Files for DVA-C03 Visit
https://www.p2pexams.com/products/dva-c03

For More Free Questions Visit
https://www.p2pexams.com/amazon/pdf/dva-c03

