Free Amazon DVA-C02 Practice Questions

Shared by Gardner on 12-08-2026

For More Free Questions and Preparation Resources

Check the Links on Last Page

Question 1

Question Type: MultipleChoice

A company has an Amazon S3 bucket that contains sensitive data. The data must be encrypted in transit and at rest. The company encrypts the data in the S3 bucket by using an AWS Key Management Service (AWS KMS) key. A developer needs to grant several other AWS accounts the permission to use the S3 GetObject operation to retrieve the data from the S3 bucket.

How can the developer enforce that all requests to retrieve the data provide encryption in transit?

Options:

A- Define a resource-based policy on the S3 bucket to deny access when a request meets the condition ''aws:SecureTransport'': ''false''.
B- Define a resource-based policy on the S3 bucket to allow access when a request meets the condition ''aws:SecureTransport'': ''false''.
C- Define a role-based policy on the other accounts' roles to deny access when a request meets the condition of ''aws:SecureTransport'': ''false''.
D- Define a resource-based policy on the KMS key to deny access when a request meets the condition of ''aws:SecureTransport'': ''false''.

Answer:

A

Explanation:

Amazon S3 supports resource-based policies, which are JSON documents that specify the permissions for accessing S3 resources. A resource-based policy can be used to enforce encryption in transit by denying access to requests that do not use HTTPS. The condition key aws:SecureTransport can be used to check if the request was sent using SSL. If the value of this key is false, the request is denied; otherwise, the request is allowed. Reference:How do I use an S3 bucket policy to require requests to use Secure Socket Layer (SSL)?

Question 2

Question Type: MultipleChoice

A company wants to push a new UI for its application to users in a staging environment before

the company gradually rolls out the UI to 10% of the company's production users in a single AWS Region. The company needs a solution that can target the company's audience based on user attributes. The solution must validate the configuration before deployment. The solution also must have an option for instant rollback without redeploying code. Which solution will meet these requirements?

Options:

A- Use environment variables and deploy new UI versions for each environment to change feature flags.
B- Use AWS AppConfig feature flags that have environment-scoped configurations, staged rollouts, and instant rollback.
C- Store environment flags in AWS Systems Manager Parameter Store. Configure the application to call for parameters at runtime.
D- Run a custom feature flag service on an Application Load Balancer (ALB). Configure the application to query the flag service to target the company's audience and perform percentage rollouts.

Answer:

B

Explanation:

The correct answer is B because AWS AppConfig is designed specifically for application configuration, feature flags, controlled rollouts, configuration validation, and safe deployment practices. The scenario requires rollout in staging first, then gradual exposure to 10% of production users, audience targeting by user attributes, validation before deployment, and the ability to perform instant rollback without redeploying code. AWS AppConfig feature flags provide all of these capabilities.

AWS documentation describes AppConfig as a service for creating, managing, and deploying dynamic application configurations. With feature flags, a company can separate release from deployment and safely enable or disable features for selected audiences. AppConfig supports environment-scoped configurations, so staging and production can have different flag states. It also supports deployment strategies for gradual rollouts and includes validators to check configuration before deployment. In addition, AppConfig supports immediate rollback if a problem occurs, which is a major advantage over approaches that depend on application redeployment.

Option A is incorrect because environment variables typically require a deployment change and do not provide audience targeting, staged rollout controls, or fast rollback. Option C is better suited for storing configuration values, but it does not provide the same rich feature flag targeting and rollout management capabilities as AppConfig. Option D is possible in theory, but it requires building and maintaining a custom service, which is more complex and less reliable than using the managed AWS service built for this purpose.

Therefore, AWS AppConfig feature flags are the best fit for the stated requirements.

Question 3

Question Type: MultipleChoice

A company runs a web application on Amazon EC2 instances behind an Application Load Balancer. The application uses Amazon DynamoDB as its database. The company wants to ensure high performance for reads and writes.

Which solution will meet these requirements MOST cost-effectively?

Options:

A- Configure auto-scaling for the DynamoDB table with a target utilization of 70%. Set the minimum and maximum capacity units based on the expected workload.
B- Use DynamoDB on-demand capacity mode for the table. Specify a maximum throughput higher than the expected peak read and write capacity units.
C- Use DynamoDB provisioned throughput mode for the table. Create an Amazon CloudWatch alarm on the ThrottledRequests metric. Invoke an AWS Lambda function to increase provisioned capacity.
D- Create an Amazon DynamoDB Accelerator (DAX) cluster. Configure the application to use the DAX endpoint.

Answer:

A

Explanation:

Why Option A is Correct:Auto-scaling with a target utilization ensures the DynamoDB table dynamically adjusts capacity based on workload, maintaining high performance while optimizing cost. Setting a reasonable target utilization minimizes overprovisioning and throttling risks.

Why Other Options are Incorrect:

Option B: On-demand capacity is costlier than provisioned throughput for predictable workloads.

Option C: Using manual CloudWatch alarms and Lambda for scaling is less efficient and adds operational overhead.

Option D: DAX accelerates read performance but does not improve write performance.

AWS Documentation Reference:

DynamoDB Auto Scaling

Question 4

Question Type: MultipleChoice

A developer works in an environment with multiple AWS accounts that have AWS Lambda functions processing the same 100 KB payloads. The developer wants to centralize the point of origin of the payloads to one account and have all the Lambda functions be invoked whenever the initiating event occurs in the parent account.

How can the developer design the workflow in the MOST efficient way, so all the multi-account Lambda functions get invoked when the event occurs?

Options:

A- Create a Lambda function in the parent account and use cross-account IAM roles with the AWS STS AssumeRole API call to make AWS Lambda invoke the API call to invoke all the cross-account Lambda functions.
B- Subscribe all the multi-account Lambda functions to an Amazon SNS topic and make an SNS Publish API call with the payload to the SNS topic.
C- Set up an Amazon SQS queue with the queue policy permitting the ReceiveMessage action for multi-account Lambda functions. Then send the payload to the SQS queue using the sqs:SendMessage permission and poll the queue using multi-account Lambda functions.
D- Use a worker on an Amazon EC2 instance to poll for the payload event. Invoke all Lambda functions using the Lambda Invoke API after using cross-account IAM roles with the AWS STS AssumeRole API call.

Answer:

B

Explanation:

Amazon SNS is the most efficient fanout service for this requirement. A single publish operation can distribute the same message payload to multiple subscribers, including Lambda functions. Because the payload is 100 KB, it is within the SNS message size limit, making direct fanout practical. This centralizes the event origin in the parent account and avoids custom orchestration code. Using STS AssumeRole and manually invoking every Lambda function creates unnecessary code, permissions management, and retry complexity. SQS is a queueing service where messages are consumed, not broadcast to all consumers; multiple Lambda functions polling the same queue would compete for messages rather than each receive every payload. An EC2 worker is the worst option because it adds server management and polling overhead. SNS topic subscriptions are the intended pub/sub design.

==========

Question 5

Question Type: MultipleChoice

A developer has an application that is composed of many different AWS Lambda functions. The Lambda functions all use some of the same dependencies. To avoid security issues the developer is constantly updating the dependencies of all of the Lambda functions. The result is duplicated effort to reach function.

How can the developer keep the dependencies of the Lambda functions up to date with the LEAST additional complexity?

Options:

A- Define a maintenance window for the Lambda functions to ensure that the functions get updated copies of the dependencies.
B- Upgrade the Lambda functions to the most recent runtime version.
C- Define a Lambda layer that contains all of the shared dependencies.
D- Use an AWS CodeCommit repository to host the dependencies in a centralized location.

Answer:

C

Explanation:

This solution allows the developer to keep the dependencies of the Lambda functions up to date with the least additional complexity because it eliminates the need to update each function individually. A Lambda layer is a ZIP archive that contains libraries, custom runtimes, or other dependencies. The developer can create a layer that contains all of the shared dependencies and attach it to multiple Lambda functions. When the developer updates the layer, all of the functions that use the layer will have access to the latest version of the dependencies.

Question 6

Question Type: MultipleChoice

A developer is integrating Amazon ElastiCache in an application. The cache will store data from a database. The cached data must populate real-time dashboards. Which caching strategy will

meet these requirements?

Options:

A- A read-through cache
B- A write-behind cache
C- A lazy-loading cache
D- A write-through cache

Answer:

D

Question 7

Question Type: MultipleChoice

A company introduced a new feature that should be accessible to only a specific group of premium customers. A developer needs the ability to turn the feature on and off in response to performance and feedback. The developer needs a solution to validate and deploy these configurations quickly without causing any disruptions.

What should the developer do to meet these requirements?

Options:

A- Use AWS AppConfig to manage the feature configuration and to validate and deploy changes. Use feature flags to turn the feature on and off.
B- Use AWS Secrets Manager to securely manage and validate the feature configurations. Enable lifecycle rules to turn the feature on and off.
C- Use AWS Config to manage the feature configuration and validation. Set up AWS Config rules to turn the feature on and off based on predefined conditions.
D- Use AWS Systems Manager Parameter Store to store and validate the configuration settings for the feature. Enable lifecycle rules to turn the feature on and off.

Answer:

A

Explanation:

This is a feature flag / dynamic configuration requirement: enable a new feature for a subset of

users (premium customers), toggle it on/off quickly based on feedback/performance, and deploy configuration changes safely with validation and without disruption. AWS AppConfig (part of AWS Systems Manager) is designed for exactly this use case.

AWS AppConfig helps create, manage, and deploy application configurations, including feature flags, in a controlled way. It supports configuration validators (such as JSON schema or Lambda validators) to catch errors before deployment. It also supports controlled rollouts with deployment strategies to reduce blast radius (for example, gradual deployments). This allows rapid changes without redeploying code and minimizes disruption.

Option A aligns perfectly: use AppConfig feature flags to target premium users and toggle the feature quickly.

Option B is incorrect because Secrets Manager is for sensitive secrets (passwords, API keys), not feature flag management and progressive configuration deployments.

Option C is incorrect because AWS Config evaluates resource compliance and records configuration changes of AWS resources; it does not serve runtime feature flag delivery.

Option D (Parameter Store) can store configuration values, but it does not provide the same feature-flag-focused capabilities, validation/deployment strategies, and safe rollout controls that AppConfig provides. Also, ''lifecycle rules'' is not how Parameter Store toggles features.

Question 8

Question Type: MultipleChoice

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'?

Options:

A- Create Amazon Cognito user pools for external social identity providers Configure 1AM roles for the identity pools.
B- Program the sign-in page to create users' 1AM groups with the 1AM 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:

Amazon Cognito User Pools:A managed user directory service, simplifying user registration and login.

Social Identity Providers:Cognito supports integration with external providers (e.g., Google, Facebook), reducing development effort.

IAM Roles for Authorization:Cognito-managed IAM roles grant fine-grained access to AWS resources (like Lambda functions).

Operational Overhead:Cognito minimizes the need to manage user identities and credentials independently.

Amazon Cognito Documentationhttps://docs.aws.amazon.com/cognito/

Cognito User Pools for Web Applications:https://docs.aws.amazon.com/cognito/latest/developerguide/cognito-user-pools-app-integration.html

Question 9

Question Type: MultipleChoice

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?

Options:

A- Export the existing API to an OpenAPI file. Create a new API. Import the OpenAPI file. Modify the new API to add request validation. Perform the tests. Modify the existing API to add request validation. Deploy the existing API to production.

B- Modify the existing API to add request validation. Deploy the updated API to a new API Gateway stage. Perform the tests. Deploy the updated API to the API Gateway production stage.
C- Create a new API. Add the necessary resources and methods, including new request validation. Perform the tests. Modify the existing API to add request validation. Deploy the existing API to production.
D- Clone the existing API. Modify the new API to add request validation. Perform the tests. Modify the existing API to add request validation. 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 services1.You can use API Gateway to perform basic validation of an API request before proceeding with the integration request1.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 Logs1.

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 stage1. 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 stage1.

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 API1.

Question 10

Question Type: MultipleChoice

A developer has created an AWS Lambda function that is written in Python. The Lambda function reads data from objects in Amazon S3 and writes data to an Amazon DynamoDB table.

The function is successfully invoked from an S3 event notification when an object is created. However, the function fails when it attempts to write to the DynamoDB table.

What is the MOST likely cause of this issue?

Options:

A- The Lambda function's concurrency limit has been exceeded.
B- The DynamoDB table requires a global secondary index (GSI) to support writes.
C- The Lambda function does not have IAM permissions to write to DynamoDB.
D- The DynamoDB table is not running in the same Availability Zone as the Lambda function.

Answer:

C

Explanation:

Because the Lambda function is successfully triggered by the S3 event notification, the invocation path (S3 Lambda) is working correctly. The failure occurs specifically when the function tries to write to DynamoDB, which strongly indicates an authorization problem rather than an invocation, scaling, or infrastructure issue.

In AWS, a Lambda function interacts with other services by using its execution role (an IAM role). AWS documentation explains that a Lambda function must have explicit IAM permissions to call downstream services such as DynamoDB. To write items, the role typically needs actions like dynamodb:PutItem (and sometimes dynamodb:UpdateItem, dynamodb:BatchWriteItem, depending on code behavior) on the target table resource ARN. If these permissions are missing or scoped incorrectly, DynamoDB returns an AccessDeniedException (or similar) and the function fails at the write step.

Option A is unlikely because exceeding concurrency would typically prevent invocation or cause throttling at the Lambda service level, not selectively fail only at DynamoDB write time after the function begins executing.

Option B is incorrect: DynamoDB does not require a GSI to support writes. GSIs are for alternate query access patterns, not mandatory for write operations.

Option D is incorrect because DynamoDB is a regional service, not tied to a single Availability Zone, and Lambda does not need to be ''in the same AZ'' to access it.

Therefore, the most likely cause is that the Lambda execution role lacks the necessary IAM permissions to perform DynamoDB write operations.

Question 11

Question Type: MultipleChoice

A developer at a company needs to create a small application that makes the same API call once each day at a designated time. The company does not have infrastructure in the AWS Cloud yet, but the company wants to implement this functionality on AWS.

Which solution meets these requirements in the MOST operationally efficient manner?

Options:

A- Use a Kubernetes cron job that runs on Amazon Elastic Kubernetes Service (Amazon EKS).
B- Use an Amazon Linux crontab scheduled job that runs on Amazon EC2.
C- Use an AWS Lambda function that is invoked by an Amazon EventBridge scheduled event.
D- Use an AWS Batch job that is submitted to an AWS Batch job queue.

Answer:

C

Explanation:

The correct answer is C. Use an AWS Lambda function that is invoked by an Amazon EventBridge scheduled event.

C . Use an AWS Lambda function that is invoked by an Amazon EventBridge scheduled event. This is correct. AWS Lambda is a serverless compute service that lets you run code without provisioning or managing servers. Lambda runs your code on a high-availability compute infrastructure and performs all of the administration of the compute resources, including server and operating system maintenance, capacity provisioning and automatic scaling, and logging1. Amazon EventBridge is a serverless event bus service that enables you to connect your applications with data from a variety of sources2. EventBridge can create rules that run on a schedule, either at regular intervals or at specific times and dates, and invoke targets such as Lambda functions3. This solution meets the requirements of creating a small application that makes the same API call once each day at a designated time, without requiring any infrastructure in the AWS Cloud or any operational overhead.

A . Amazon EKS is a fully managed Kubernetes service that allows you to run containerized applications on AWS4. Kubernetes cron jobs are tasks that run periodically on a given schedule5. This solution could meet the functional requirements of creating a small application that makes the same API call once each day at a designated time, but it would not be the most operationally efficient manner. The company would need to provision and manage an EKS cluster, which would incur additional costs and complexity.

B . Use an Amazon Linux crontab scheduled job that runs on Amazon EC2. This is incorrect. Amazon EC2 is a web service that provides secure, resizable compute capacity in the cloud6. Crontab is a Linux utility that allows you to schedule commands or scripts to run automatically at a specified time or date7. This solution could meet the functional requirements of creating a small application that makes the same API call once each day at a designated time, but it would not be the most operationally efficient manner. The company would need to provision and manage an EC2 instance, which would incur additional costs and complexity.

D . Use an AWS Batch job that is submitted to an AWS Batch job queue. This is incorrect. AWS Batch enables you to run batch computing workloads on the AWS Cloud8. Batch jobs are units of work that can be submitted to job queues, where they are executed in parallel or sequentially

on compute environments9. This solution could meet the functional requirements of creating a small application that makes the same API call once each day at a designated time, but it would not be the most operationally efficient manner. The company would need to configure and manage an AWS Batch environment, which would incur additional costs and complexity.

1: What is AWS Lambda? - AWS Lambda

2: What is Amazon EventBridge? - Amazon EventBridge

3: Creating an Amazon EventBridge rule that runs on a schedule - Amazon EventBridge

4: What is Amazon EKS? - Amazon EKS

5: CronJob - Kubernetes

6: What is Amazon EC2? - Amazon EC2

7: Crontab in Linux with 20 Useful Examples to Schedule Jobs - Tecmint

8: What is AWS Batch? - AWS Batch

9: Jobs - AWS Batch