Free Amazon SOA-C03 Practice Questions

Shared by Mcdaniel 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 two AWS accounts connected by a transit gateway. Each account has one VPC in the same AWS Region. The company wants to simplify inbound and outbound rules in security groups by referencing security group IDs instead of IP CIDR blocks.

Which solution will meet this requirement?

Options:

A- Create VPC peering connections and remove the transit gateway.
B- Enable security group referencing support on the transit gateway.
C- Enable security group referencing support on each transit gateway attachment.
D- Deploy private NAT gateways in each VPC.

Answer:

C

Explanation:

AWS Transit Gateway supports security group referencing across VPCs, but this feature must be explicitly enabled on each transit gateway attachment. Once enabled, security groups in one VPC can reference security groups in another VPC attached to the same transit gateway, simplifying rule management and improving security posture.

Enabling the feature on the transit gateway itself is not sufficient; it must be enabled per attachment to allow traffic evaluation based on security group IDs. This approach avoids brittle CIDR-based rules and allows dynamic scaling without rule updates.

Option A removes the transit gateway, which contradicts the existing architecture. Option B is incomplete. Option D does not address security group referencing.

Thus, enabling security group referencing on each transit gateway attachment is the correct solution.

Question 2

Question Type: MultipleChoice

A CloudOps engineer needs to ensure that AWS resources across multiple AWS accounts are tagged consistently. The company uses an organization in AWS Organizations to centrally manage the accounts. The company wants to implement cost allocation tags to accurately track the costs that are allocated to each business unit.

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

Options:

A- Use Organizations tag policies to enforce mandatory tagging on all resources. Enable cost allocation tags in the AWS Billing and Cost Management console.
B- Configure AWS CloudTrail events to invoke an AWS Lambda function to detect untagged resources and to automatically assign tags based on predefined rules.
C- Use AWS Config to evaluate tagging compliance. Use AWS Budgets to apply tags for cost allocation.
D- Use AWS Service Catalog to provision only pre-tagged resources. Use AWS Trusted Advisor to enforce tagging across the organization.

Answer:

A

Explanation:

AWS Organizations Tag Policies provide a centralized, scalable governance mechanism to standardize tagging across accounts. Tag policies let an organization define tag keys, allowed values, and tagging expectations, helping teams apply consistent tagging conventions across many accounts without building custom logic. This matches the requirement for consistent tags ''across multiple accounts'' with minimal operational overhead, because the policy is managed centrally and applied at the organization/OUs level.

For cost tracking, user-defined tags must be activated as cost allocation tags in AWS Billing and Cost Management. Enabling cost allocation tags is the required step to make those tags usable in billing views (for example, Cost Explorer allocation and reporting). Combining Tag Policies (governance/consistency) with cost allocation tag activation (billing attribution) directly meets both parts of the requirement.

Option B (CloudTrail + Lambda auto-tagging) is higher operational overhead: it requires event processing, permissions, continuous maintenance, exception handling, and careful logic to avoid incorrect tag assignments. Option C is partially relevant for compliance detection, but AWS Budgets does not ''apply tags'' to resources; Budgets is for cost/usage alerts and budget tracking. Option D can enforce tagged provisioning paths, but it's not comprehensive for all resource creation mechanisms and Trusted Advisor is not a global ''tag enforcement'' engine.

Therefore, A is the most native and least-ops approach for consistent tags across an organization and enabling cost allocation tracking.

Question 3

Question Type: MultipleChoice

A company has a stateful web application that is hosted on Amazon EC2 instances in an Auto Scaling group. The instances run behind an Application Load Balancer (ALB) that has a single target group. The ALB is configured as the origin in an Amazon CloudFront distribution. Users are reporting random logouts from the web application.

Which combination of actions should a CloudOps engineer take to resolve this problem? (Select TWO.)

Options:

A- Change to the least outstanding requests algorithm on the ALB target group.
B- Configure cookie forwarding in the CloudFront distribution cache behavior.
C- Configure header forwarding in the CloudFront distribution cache behavior.
D- Enable group-level stickiness on the ALB listener rule.
E- Enable sticky sessions on the ALB target group.

Answer:

B, E

Explanation:

Stateful applications require session persistence to ensure that subsequent requests from the same user are routed to the same backend instance. When CloudFront is used in front of an ALB, session-related cookies must be forwarded correctly; otherwise, CloudFront can route requests to different targets, causing session loss and random logouts.

Configuring cookie forwarding in the CloudFront cache behavior ensures that session cookies (such as authentication tokens) are forwarded to the ALB and not stripped or cached incorrectly. Without this configuration, CloudFront may serve cached responses that do not align with the user's active session state, leading to authentication issues.

On the ALB side, sticky sessions (session affinity) must be enabled on the target group to ensure that requests with the same session cookie are consistently routed to the same EC2 instance. ALB stickiness uses application cookies to bind a user session to a specific target, which is critical for stateful applications that store session data in memory.

Option A affects load distribution efficiency but does not address session persistence. Option C (header forwarding) is unnecessary unless the application explicitly stores session state in headers, which is uncommon. Option D applies only when using multiple target groups and

listener rules, which is not the case here.

Together, enabling cookie forwarding in CloudFront and sticky sessions at the ALB target group resolves the logout issue by maintaining consistent session routing from the user through CloudFront to the same backend instance.

Question 4

Question Type: MultipleChoice

A company's architecture team must receive immediate email notifications whenever new Amazon EC2 instances are launched in the company's main AWS production account.

What should a CloudOps engineer do to meet this requirement?

Options:

A- Create a user data script that sends an email message through a smart host connector. Include the architecture team's email address in the user data script as the recipient. Ensure that all new EC2 instances include the user data script as part of a standardized build process.
B- Create an Amazon Simple Notification Service (Amazon SNS) topic and a subscription that uses the email protocol. Enter the architecture team's email address as the subscriber. Create an Amazon EventBridge rule that reacts when EC2 instances are launched. Specify the SNS topic as the rule's target.
C- Create an Amazon Simple Queue Service (Amazon SQS) queue and a subscription that uses the email protocol. Enter the architecture team's email address as the subscriber. Create an Amazon EventBridge rule that reacts when EC2 instances are launched. Specify the SQS queue as the rule's target.
D- Create an Amazon Simple Notification Service (Amazon SNS) topic. Configure AWS Systems Manager to publish EC2 events to the SNS topic. Create an AWS Lambda function to poll the SNS topic. Configure the Lambda function to send any messages to the architecture team's email address.

Answer:

B

Explanation:

As per the AWS Cloud Operations and Event Monitoring documentation, the most efficient method for event-driven notification is to use Amazon EventBridge to detect specific EC2 API events and trigger a Simple Notification Service (SNS) alert.

EventBridge continuously monitors AWS service events, including RunInstances, which signals the creation of new EC2 instances. When such an event occurs, EventBridge sends it to an SNS topic, which then immediately emails subscribed recipients --- in this case, the architecture team.

This combination provides real-time, serverless notifications with minimal management. SQS (Option C) is designed for queue-based processing, not direct user alerts. User data scripts (Option A) and custom polling with Lambda (Option D) introduce unnecessary operational complexity and latency.

Hence, Option B is the correct and AWS-recommended CloudOps design for immediate launch notifications.

Question 5

Question Type: MultipleChoice

A company uses default settings to create an AWS Lambda function. The function needs to access an Amazon RDS database that is in a private subnet of a VPC. The function has the correct IAM permissions to access the database. The private subnet has appropriate routing configurations and is accessible from within the VPC. However, the Lambda function is unable to connect to the RDS instance.

What is the likely reason the Lambda function cannot connect to the RDS instance?

Options:

A- The company did not set the RDS instance as the destination for the Lambda function in the function configuration.
B- The Lambda function configuration did not deploy the function in the same VPC that contains the RDS instance.
C- The VPC where the Lambda function is deployed is not peered with the VPC where the RDS instance is deployed.
D- The security group for the Lambda function does not allow outbound access to the RDS instance.

Answer:

B

Explanation:

By default, a newly created AWS Lambda function is not attached to a customer VPC. In its

default configuration, Lambda runs in an AWS-managed network environment and can reach public internet endpoints, but it does not automatically have network-level connectivity into private subnets within a VPC. Accessing an Amazon RDS instance that is placed in a private subnet requires the Lambda function to be configured for VPC access. This means selecting the target VPC, choosing subnets (typically private subnets with appropriate routing), and associating one or more security groups with the Lambda function's elastic network interfaces (ENIs). Once configured, Lambda creates ENIs in the selected subnets and uses them to communicate with resources such as RDS inside the VPC.

In this scenario, the RDS database is in a private subnet and is reachable from within the VPC, but the Lambda function cannot connect. The most likely cause is that the function was created with default settings and therefore was not deployed into the VPC that contains the database. IAM permissions are not sufficient for network connectivity; IAM controls authorization to call AWS APIs, while VPC attachment and security groups control the network path.

Option A is incorrect because Lambda does not require an ''RDS destination'' setting; connectivity is established through VPC networking and the database endpoint. Option C is not supported by the prompt; there is no indication that the database is in a different VPC, and the default issue is usually ''not in a VPC at all.'' Option D is less likely because Lambda security groups allow all outbound traffic by default unless explicitly restricted; the more common default failure is lack of VPC configuration.

Therefore, the likely reason is that the Lambda function was not configured to run in the same VPC as the RDS instance.

Question 6

Question Type: MultipleChoice

A company runs a worker process on three Amazon EC2 instances. The instances are in an Auto Scaling group that is configured to use a simple scaling policy. The instances process messages from an Amazon SQS queue. Random periods of increased messages are causing a decrease in the performance of the worker process. A CloudOps engineer must scale the instances to accommodate the increased number of messages.

Which solution will meet these requirements?

Options:

A- Use CloudWatch to create a metric math expression to calculate the approximate age of the oldest message in the SQS queue. Create a target tracking scaling policy for the metric math expression to modify the Auto Scaling group.
B- Use CloudWatch to create a metric math expression to calculate the approximate number of messages visible in the SQS queue for each instance. Create a target tracking scaling policy for the metric math expression to modify the Auto Scaling group.

C- Create an Application Load Balancer (ALB). Attach the ALB to the Auto Scaling group. Create a target tracking scaling policy for the ALBRequestCountPerTarget metric to modify the Auto Scaling group.
D- Create an Application Load Balancer (ALB). Attach the ALB to the Auto Scaling group. Create a scheduled scaling policy for the Auto Scaling group.

Answer:

B

Explanation:

For SQS-backed worker fleets, the most useful scaling signal is backlog per instance, calculated as the approximate number of visible messages divided by the number of running instances. This metric shows whether each worker instance has too much work to process and is therefore a strong target tracking metric. Target tracking scaling is better than simple scaling for this pattern because it continuously adjusts capacity to keep the selected metric near the desired value. Option A uses the age of the oldest message, which can indicate delay but does not directly express workload per instance. Options C and D are wrong because an ALB request metric applies to HTTP request routing, not SQS message processing. Scheduled scaling is also unsuitable because the spikes are random and unpredictable. Therefore, option B is the correct performance optimization solution.

Question 7

Question Type: MultipleChoice

A CloudOps engineer creates a new VPC that contains a private subnet, a security group that allows all outbound traffic, and an endpoint for Amazon EC2 Instance Connect in a private subnet. The CloudOps engineer associates the security group with EC2 Instance Connect.

The CloudOps engineer launches an EC2 instance from an Amazon Linux Amazon Machine Image (AMI) in the private subnet. The CloudOps engineer launches the EC2 instance without an SSH key pair.

The CloudOps engineer tries to connect to the instance by using the EC2 Instance Connect endpoint. However, the connection fails.

How can the CloudOps engineer connect to the instance?

Options:

A- Create an inbound rule in the security group to allow HTTPS traffic on port 443 from the

private subnet.
B- Create an inbound rule in the security group to allow SSH traffic on port 22 from the private subnet.
C- Create an IAM instance profile that allows AWS Systems Manager Session Manager to access the EC2 instance. Associate the instance profile with the instance.
D- Recreate the EC2 instance. Associate an SSH key pair with the instance.

Answer:

B

Explanation:

Amazon EC2 Instance Connect enables secure SSH access to EC2 instances without requiring a traditional SSH key pair. However, although authentication is handled through IAM and the Instance Connect endpoint, the underlying network requirements for SSH still apply.

For EC2 Instance Connect to function, the EC2 instance's security group must allow inbound traffic on TCP port 22 from the network where the Instance Connect endpoint resides. In this case, both the endpoint and the EC2 instance are in the private subnet, so the security group must explicitly allow SSH traffic from that subnet or from the security group associated with the endpoint.

Allowing HTTPS traffic on port 443 does not enable SSH access. Systems Manager Session Manager is a separate access mechanism and does not resolve an EC2 Instance Connect failure. Recreating the instance with an SSH key pair is unnecessary because EC2 Instance Connect does not rely on key pairs.

Therefore, enabling inbound SSH traffic on port 22 from the private subnet resolves the connection issue.

Question 8

Question Type: MultipleChoice

A company has an AWS Lambda function in Account A. The Lambda function needs to read the objects in an Amazon S3 bucket in Account B. A CloudOps engineer must create corresponding IAM roles in both accounts.

Which solution will meet these requirements?

Options:

A- In Account A, create a Lambda execution role to assume the role in Account B. In Account B, create a role that the function can assume to gain access to the S3 bucket.
B- In Account A, create a Lambda execution role that provides access to the S3 bucket. In Account B, create a role that the function can assume.
C- In Account A, create a role that the function can assume. In Account B, create a Lambda execution role that provides access to the S3 bucket.
D- In Account A, create a role that the function can assume to gain access to the S3 bucket. In Account B, create a Lambda execution role to assume the role in Account A.

Answer:

A

Explanation:

For cross-account access, the Lambda function in Account A needs an execution role that permits it to call sts:AssumeRole on a role in Account B. The role in Account B must trust the Lambda execution role from Account A and must have permissions to read the target S3 bucket objects. This follows the standard AWS cross-account access model: the resource-owning account creates a role with permissions to the resource, and the external workload assumes that role. Option B is incomplete because permissions granted only in Account A do not automatically grant access to resources in Account B. Options C and D reverse the trust relationship incorrectly. The secure CloudOps model is cross-account role assumption with least privilege: Lambda execution role in Account A assumes a read-only S3 access role in Account B.

Question 9

Question Type: MultipleChoice

A company hosts a static website on an Amazon S3 bucket behind an Amazon CloudFront distribution. When the company deploys a new version of the website, users sometimes do not see the new content until the next day.

A CloudOps engineer must implement a solution to display updates to the website as quickly as possible.

Which solution will meet this requirement?

Options:

A- Configure the CloudFront distribution to add a custom Cache-Control header to requests for content from the S3 bucket.
B- Modify the distribution settings to specify the protocol as HTTPS only.

C- Attach the CachingOptimized managed cache policy to the distribution.
D- Create a CloudFront invalidation.

Answer:

D

Explanation:

CloudFront caches content at edge locations to reduce origin requests and improve performance. When a static website is updated by overwriting existing object keys (for example, updating index.html in the same S3 path), CloudFront may continue to serve the cached versions of those objects until the cache's time-to-live (TTL) expires. This behavior commonly produces a delay where some viewers still receive the old site version, sometimes for many hours, depending on the configured TTLs and any cache headers that CloudFront is honoring.

A CloudFront invalidation (Option D) is designed to address exactly this situation: it removes cached objects from CloudFront edge caches so that the next request for those objects forces CloudFront to retrieve the latest versions from the S3 origin and cache them again. By invalidating key objects such as /index.html (and potentially additional paths if needed), the company can ensure users see the newest deployment quickly without waiting for TTL expiration.

Option A is not the best answer because ''adding a Cache-Control header to requests'' does not immediately clear content already cached at edge locations; it primarily influences caching behavior for future fetches and can still leave existing cached objects in place until they expire. Option B only enforces HTTPS and does not affect cache freshness or object replacement visibility. Option C (CachingOptimized) is intended to improve caching efficiency and performance for typical workloads; it does not provide immediate propagation of updated objects and can result in CloudFront serving cached responses until TTLs expire.

Therefore, the most direct solution to display the updated website as quickly as possible is to create a CloudFront invalidation.

Question 10

Question Type: MultipleChoice

A company has a VPC that contains a public subnet and a private subnet. The company deploys an Amazon EC2 instance that uses an Amazon Linux Amazon Machine Image (AMI) and has the AWS Systems Manager Agent (SSM Agent) installed in the private subnet. The EC2 instance is in a security group that allows only outbound traffic.

A CloudOps engineer needs to give a group of privileged administrators the ability to connect to the instance through SSH without exposing the instance to the internet.

Which solution will meet this requirement?

Options:

A- Create an EC2 Instance Connect endpoint in the private subnet. Update the security group to allow inbound SSH traffic. Create an IAM group for privileged administrators. Assign the PowerUserAccess managed policy to the IAM group.
B- Create a Systems Manager endpoint in the private subnet. Update the security group to allow SSH traffic from the private network where the Systems Manager endpoint is connected. Create an IAM group for privileged administrators. Assign the PowerUserAccess managed policy to the IAM group.
C- Create an EC2 Instance Connect endpoint in the public subnet. Update the security group to allow SSH traffic from the private network. Create an IAM group for privileged administrators. Assign the PowerUserAccess managed policy to the IAM group.
D- Create a Systems Manager endpoint in the public subnet. Create an IAM role that has the AmazonSSMManagedInstanceCore permission for the EC2 instance. Create an IAM group for privileged administrators. Assign the AmazonEC2ReadOnlyAccess IAM policy to the IAM group.

Answer:

A

Explanation:

Comprehensive and Detailed Explanation From Exact Extract of AWS CloudOps Doocuments:

EC2 Instance Connect Endpoint (EIC Endpoint) enables SSH to instances in private subnets without public IPs and without needing to traverse the public internet. CloudOps guidance explains that you deploy the endpoint in the same VPC/subnet as the targets, then allow inbound SSH on the instance security group from the endpoint's security group. Access is governed by IAM---administrators must have Instance Connect permissions; while the example uses a broad policy, the key mechanism is EIC in the private subnet plus SG rules scoped to the endpoint. Systems Manager Session Manager can provide shell access without SSH, but the requirement explicitly states ''connect through SSH,'' making EIC the purpose-built solution. Options B and D misuse Systems Manager for SSH and propose unnecessary SG changes or incorrect endpoint placement; Option C places the endpoint in a public subnet, which is not required for private SSH access. Therefore, creating an EC2 Instance Connect endpoint in the private subnet and updating SGs accordingly meets the requirement while keeping the instance non-internet-exposed.