DUMPSBOSS.

AWS Certified Security - Specialty

Amazon AWS SCS-C02

Version Demo

Total Demo Questions: 49

Total Premium Questions: 497

Buy Premium PDF

https://dumpsboss.co

support@dumpsboss.co

support@dumpsboss.co
dumpsboss.co

Topic Break Down

Topic	No. of Questions
Topic 1, Threat Detection and Incident Response	59
Topic 2, Security Logging and Monitoring	82
Topic 3, Infrastructure Security	100
Topic 4, Identity and Access Management	101
Topic 5, Data Protection	116
Topic 6, Management and Security Governance	36
Topic 7, Mix Questions	3
Total	497

QUESTION NO: 1

A security engineer is implementing a solution to allow users to seamlessly encrypt Amazon S3 objects without having to touch the keys directly. The solution must be highly scalable without requiring continual management. Additionally, the organization must be able to immediately delete the encryption keys.

Which solution meets these requirements?

A. Use AWS KMS with AWS managed keys and the ScheduleKeyDeletion API with a PendingWindowInDays set to 0 to remove the keys if necessary.

B. Use KMS with AWS imported key material and then use the DeleteImportedKeyMaterial API to remove the key material if necessary.

C. Use AWS CloudHSM to store the keys and then use the CloudHSM API or the PKCS11 library to delete the keys if necessary.

D. Use the Systems Manager Parameter Store to store the keys and then use the service API operations to delete the keys if necessary.

ANSWER: B

Explanation:

Use KMS with AWS imported key material and then use the DeleteImportedKeyMaterial API to remove the key material if necessary is the best fit because it provides seamless, scalable encryption for Amazon S3 via SSE-KMS while also enabling immediate “key destruction” by deleting the imported key material. With SSE-KMS, users don’t handle keys directly; S3 calls AWS KMS to generate and use data keys on demand, and KMS scales as a managed service without continual operational management. The critical requirement is the ability to immediately delete the encryption keys. Standard KMS key deletion is not immediate: scheduling deletion enforces a waiting period (7–30 days) and cannot be set to 0 days. In contrast, for KMS keys with imported key material, you can delete the key material right away using the DeleteImportedKeyMaterial API, rendering the KMS key unusable immediately and preventing decryption of objects encrypted under that key. This meets the “immediate delete” requirement while preserving the managed, integrated experience for S3 encryption. See the KMS key deletion waiting period and imported key material deletion behavior in the AWS docs:
https://docs.aws.amazon.com/kms/latest/developerguide/deleting-keys.html and
https://docs.aws.amazon.com/kms/latest/APIReference/API_DeleteImportedKeyMaterial.html.

QUESTION NO: 2

An international company has established a new business entity in South Korea. The company also has established a new AWS account to contain the workload for the South Korean region. The company has set up the workload in the new account in the ap-northeast-2 Region. The workload consists of three Auto Scaling groups of Amazon EC2 instances. All workloads that operate in this Region must keep system logs and application logs for 7 years.

A security engineer must implement a solution to ensure that no logging data is lost for each instance during scaling activities. The solution also must keep the logs for only the required period of 7 years.

Which combination of steps should the security engineer take to meet these requirements? (Choose three.)

A. Ensure that the Amazon CloudWatch agent is installed on all the EC2 instances that the Auto Scaling groups launch. Generate a CloudWatch agent configuration file to forward the required logs to Amazon CloudWatch Logs.

B. Set the log retention for desired log groups to 7 years.

C. Attach an IAM role to the launch configuration or launch template that the Auto Scaling groups use. Configure the role to provide the necessary permissions to forward logs to Amazon
CloudWatch Logs.

D. Attach an IAM role to the launch configuration or launch template that the Auto Scaling groups use. Configure the role to provide the necessary permissions to forward logs to Amazon S3.

E. Ensure that a log forwarding application is installed on all the EC2 instances that the Auto Scaling groups launch. Configure the log forwarding application to periodically bundle the logs and forward the logs to Amazon S3.

F. Configure an Amazon S3 Lifecycle policy on the target S3 bucket to expire objects after 7 years.

ANSWER: A B C

Explanation:

To prevent log loss during Auto Scaling events, logs must be shipped off-instance in near real time because instance storage is ephemeral and instances can terminate at any time. Installing and configuring the Amazon CloudWatch agent on every launched instance provides a managed, durable destination by streaming system and application logs into Amazon CloudWatch Logs as the logs are generated. This approach is designed for dynamic fleets because the agent runs on each instance and continuously forwards log events, so termination does not strand local log files. The instances also need AWS API access to create log streams and put log events; attaching an IAM role to the launch configuration or launch template is the standard way to grant the CloudWatch Logs permissions without embedding static credentials. Finally, to meet the “only 7 years” requirement, CloudWatch Logs retention should be set to 7 years (for example, 2555 days) on the relevant log groups so older log events are automatically deleted and the organization does not retain data longer than required. See CloudWatch agent and log collection and CloudWatch Logs retention settings.

QUESTION NO: 3

A company has an application that uses an Amazon RDS PostgreSQL database. The company is developing

an application feature that will store sensitive information for an individual in the database.

During a security review of the environment, the company discovers that the RDS DB instance is not encrypting data at rest. The company needs a solution that will provide encryption at rest for all the existing data and for any new data that is entered for an individual.

Which combination of options can the company use to meet these requirements? (Select TWO.)

A. Create a snapshot of the DB instance. Copy the snapshot to a new snapshot, and enable encryption for the copy process. Use the new snapshot to restore the DB instance.

B. Modify the configuration of the DB instance by enabling encryption. Create a snapshot of the DB instance. Use the snapshot to restore the DB instance.

C. Use IAM Key Management Service (IAM KMS) to create a new default IAM managed awa/rds key. Select this key as the encryption key for operations with Amazon RDS.

D. Use IAM Key Management Service (IAM KMS] to create a new CMK. Select this key as the encryption key for operations with Amazon RDS.

E. Create a snapshot of the DB instance. Enable encryption on the snapshoVUse the snapshot to restore the DB instance.

ANSWER: A D

Explanation:

To encrypt an existing unencrypted Amazon RDS for PostgreSQL DB instance at rest, you can’t simply “turn on” encryption in place. The supported approach is to take a snapshot of the unencrypted DB instance, copy that snapshot while enabling encryption (and choosing an AWS KMS key), and then restore a new DB instance from the encrypted snapshot. This results in a new encrypted DB instance where all existing data becomes encrypted at rest, and any new data written to the database is also encrypted at rest automatically because storage encryption is enabled for the instance.

In addition, selecting an AWS KMS customer managed key for Amazon RDS encryption is a valid way to control the encryption key used for the encrypted snapshot copy and the restored DB instance. Using a customer managed key provides key policy control, rotation options, and auditability via AWS CloudTrail. Together, copying the snapshot with encryption enabled and using a KMS customer managed key meet the requirement to encrypt both existing and future data at rest.

References: Amazon RDS encryption, AWS KMS concepts (customer managed keys).

QUESTION NO: 4

A company needs a forensic-logging solution for hundreds of applications running in Docker on Amazon EC2 The solution must perform real-time analytics on the togs must support the replay of messages and must persist the logs.

Which IAM services should be used to meet these requirements? (Select TWO)

A. Amazon Athena

B. Amazon Kinesis

C. Amazon SQS

D. Amazon Elasticsearch

E. Amazon EMR

ANSWER: B D

Explanation:

Amazon Kinesis is a strong fit for forensic logging pipelines because it is designed for real-time ingestion and processing of streaming data. With Kinesis Data Streams, log events can be collected from many producers (for example, containers on Amazon EC2) and consumed by multiple analytics applications with low latency. Kinesis also supports replay by allowing consumers to re-read data within the stream's retention window (configurable up to 365 days), which is a common requirement for investigations and reprocessing. For persistent storage and fast search/analytics over logs, Amazon Elasticsearch (Amazon OpenSearch Service) is commonly used to index and analyze log data in near real time, enabling interactive queries, dashboards, and forensic exploration. In typical architectures, Kinesis provides the durable, ordered stream and replay capability, while Elasticsearch/OpenSearch provides indexed persistence and analytics capabilities for operational and security investigations. This combination meets the requirements of real-time analytics, message replay, and persisted logs in a scalable way for hundreds of applications.

References: https://docs.aws.amazon.com/streams/latest/dev/introduction.html, https://docs.aws.amazon.com/opensearch-service/latest/developerguide/what-is.html

QUESTION NO: 5

Amazon GuardDuty has detected communications to a known command and control endpoint from a company's Amazon EC2 instance. The instance was found to be running a vulnerable version of a common web framework. The company's security operations team wants to quickly identity other compute resources with the specific version of that framework installed.

Which approach should the team take to accomplish this task?

A. Scan all the EC2 instances for noncompliance with IAM Config. Use Amazon Athena to query IAM CloudTrail logs for the framework installation

B. Scan all the EC2 instances with the Amazon Inspector Network Reachability rules package to identity instances running a web server with RecognizedPortWithListener findings

C. Scan all the EC2 instances with IAM Systems Manager to identify the vulnerable version of the web framework

D. Scan an the EC2 instances with IAM Resource Access Manager to identify the vulnerable version of the web framework

ANSWER: C

Explanation:

Scan all the EC2 instances with IAM Systems Manager to identify the vulnerable version of the web framework is the right approach because AWS Systems Manager provides inventory and fleet management capabilities that can quickly report installed software and application versions across managed instances. By enabling Systems Manager (SSM) Agent on EC2

instances and configuring Systems Manager Inventory, the security team can collect metadata about installed applications (including versions, depending on the platform/package manager) and then query that inventory across the fleet to find instances running the vulnerable framework version. This is designed for rapid operational visibility without needing to log in to each instance, and it scales across accounts/regions when combined with appropriate SSM setup. In practice, teams often use Inventory data with Resource Data Sync to centralize results (for example into Amazon S3) and then query at scale, or use Systems Manager tools to search/filter managed nodes based on inventory attributes. This directly addresses the requirement to identify other compute resources with a specific installed version.

References: AWS Systems Manager Inventory, Managed instances in AWS Systems Manager

QUESTION NO: 6

A company is designing a new application stack. The design includes web servers and backend servers that are hosted on Amazon EC2 instances. The design also includes an Amazon Aurora MySQL DB cluster.

The EC2 instances are in an Auto Scaling group that uses launch templates. The EC2 instances for the web layer and the backend layer are backed by Amazon Elastic Block Store (Amazon EBS)

volumes. No layers are encrypted at rest A security engineer needs to implement encryption at rest.

Which combination of steps will meet these requirements? (Choose two.)

A. Modify EBS default encryption settings in the target AWS Region to enable encryption. Use an Auto Scaling group instance refresh.

B. Modify the launch templates for the web layer and the backend layer to add AWS Certificate Manager (ACM) encryption for the attached EBS volumes. Use an Auto Scaling group instance refresh.

C. Create a new AWS Key Management Service (AWS KMS) encrypted DB cluster from a snapshot of the existing DB cluster.

D. Apply AWS Key Management Service (AWS KMS) encryption to the existing DB cluster.

E. Apply AWS Certificate Manager (ACM) encryption to the existing DB cluster.

ANSWER: A C

Explanation:

To implement encryption at rest for the EC2 layers that use Amazon EBS, enabling EBS encryption by default in the Region ensures that any newly created EBS volumes (including those created by Auto Scaling during replacements) are encrypted with AWS KMS. Because existing instances and their already-created volumes will not be retroactively encrypted, using an Auto Scaling group instance refresh is an appropriate operational step to replace instances so that new encrypted volumes are created under the updated default encryption setting.

For the Amazon Aurora MySQL DB cluster, Aurora encryption at rest is configured at cluster creation time and cannot be turned on for an existing unencrypted cluster in place. The standard approach is to take a snapshot of the existing cluster and restore it into a new DB cluster with encryption enabled using AWS KMS. This results in an encrypted Aurora cluster while preserving data from the original cluster snapshot.

References: Amazon EBS encryption, Aurora encryption

QUESTION NO: 7

A company has AWS accounts in an organization in AWS Organizations. The organization includes a

dedicated security account.

All AWS account activity across all member accounts must be logged and reported to the dedicated security account. The company must retain all the activity logs in a secure storage location within the dedicated security account for 2 years. No changes or deletions of the logs are allowed.

Which combination of steps will meet these requirements with the LEAST operational overhead? (Select TWO.)

A. In the dedicated security account, create an Amazon S3 bucket. Configure S3 Object Lock in compliance mode and a retention period of 2 years on the S3 bucket. Set the bucket policy to allow the organization's management account to write to the S3 bucket.

B. In the dedicated security account, create an Amazon S3 bucket. Configure S3 Object Lock in compliance mode and a retention period of 2 years on the S3 bucket. Set the bucket policy to allow the organization's member accounts to write to the S3 bucket.

C. In the dedicated security account, create an Amazon S3 bucket that has an S3 Lifecycle configuration that expires objects after 2 years. Set the bucket policy to allow the organization's member accounts to write to the S3 bucket.

D. Create an AWS Cloud Trail trail for the organization. Configure logs to be delivered to the logging Amazon S3 bucket in the dedicated security account.

E. Turn on AWS CloudTrail in each account. Configure logs to be delivered to an Amazon S3 bucket that is created in the organization's management account. Forward the logs to the S3 bucket in the dedicated security account by using AWS Lambda and Amazon Kinesis Data Firehose.

ANSWER: B D

Explanation:

Using an AWS CloudTrail organization trail is the lowest-overhead way to capture activity across all member accounts because it is centrally managed and automatically applies to accounts in the organization (including new accounts, when configured appropriately). Delivering the organization trail logs directly to an Amazon S3 bucket in the dedicated security account centralizes custody of the logs and supports the requirement to “report” activity to that account without deploying per-account trails or log-forwarding pipelines. To ensure the logs cannot be altered or deleted for 2 years, the S3 bucket should be configured with S3 Object Lock in compliance mode and a 2-year retention period. Compliance mode enforces write-once-read-many (WORM) retention such that even highly privileged users cannot shorten retention or delete protected objects before the retention period expires, which aligns with “No changes or deletions of the logs are allowed.” A bucket policy that permits organization member accounts (via CloudTrail delivery) to write to the bucket enables cross-account delivery while keeping storage and retention controls in the security account. This combination meets centralized logging, immutability, and retention requirements with minimal ongoing operations.

References: https://docs.aws.amazon.com/awscloudtrail/latest/userguide/creating-trail-organization.html, https://docs.aws.amazon.com/AmazonS3/latest/userguide/object-lock.html

QUESTION NO: 8

A company has a legacy application that runs on a single Amazon EC2 instance. A security audit shows that the application has been using an IAM access key within its code to access an Amazon S3 bucket that is named DOC-EXAMPLE-BUCKET1 in the same AWS account. This access key pair has the s3:GetObject permission to all objects in only this S3 bucket. The company takes the application offline because the application is not compliant with the company’s security policies for accessing other AWS resources from Amazon EC2.

A security engineer validates that AWS CloudTrail is turned on in all AWS Regions. CloudTrail is sending logs to an S3 bucket that is named DOC-EXAMPLE-BUCKET2. This S3 bucket is in the same AWS account as DOC-EXAMPLE-BUCKET1. However, CloudTrail has not been configured to send logs to Amazon CloudWatch Logs.

The company wants to know if any objects in DOC-EXAMPLE-BUCKET1 were accessed with the IAM access key in the past 60 days. If any objects were accessed, the company wants to know if any of the objects that are text files (.txt extension) contained personally identifiable information (PII).

Which combination of steps should the security engineer take to gather this information? (Choose two.)

A. Configure Amazon Macie to identify any objects in DOC-EXAMPLE-BUCKET1 that contain PII and that were available to the access key.

B. Use Amazon CloudWatch Logs Insights to identify any objects in DOC-EXAMPLE-BUCKET1 that contain PII and that were available to the access key.

C. Use Amazon OpenSearch Service (Amazon Elasticsearch Service) to query the CloudTrail logs in DOC-EXAMPLE-BUCKET2 for API calls that used the access key to access an object that contained PII.

D. Use Amazon Athena to query the CloudTrail logs in DOC-EXAMPLE-BUCKET2 for any API calls that used the access key to access an object that contained PII.

E. Use AWS Identity and Access Management Access Analyzer to identify any API calls that used the access key to access objects that contained PII in DOC-EXAMPLE-BUCKET1.

ANSWER: A D

Explanation:

To determine whether the IAM access key was used to read objects from the S3 bucket in the last 60 days, the most direct approach is to query the existing CloudTrail logs that are already being delivered to the dedicated logging bucket. CloudTrail data events for S3 object-level activity (such as GetObject) can be analyzed by running SQL queries over the log files in Amazon S3, and Amazon Athena is purpose-built for this “query in place” workflow. With Athena, the engineer can filter by the access key ID, eventName (for example, GetObject), eventTime (last 60 days), and the requested object key, which also allows easy filtering for the .txt extension.

To determine whether any accessed .txt objects contain PII, Amazon Macie is the AWS managed service designed to discover and classify sensitive data in S3, including many PII types. After identifying which .txt objects were accessed, Macie can be used to run sensitive data discovery on the relevant bucket (or scoped via job criteria) and report findings for PII within those objects.

References: Querying AWS CloudTrail logs with Amazon Athena, Amazon Macie sensitive data discovery and classification

QUESTION NO: 9

A-company uses a third-party identity provider and SAML-based SSO for its AWS accounts. After the third-party identity provider renewed an expired signing certificate, users saw the following message when trying to log in:

Error: Response Signature Invalid (Service: AWSSecurityTokenService; Status Code: 400; Error Code: InvalidIdentityToken)

A security engineer needs to provide a solution that corrects the error and minimizes operational overhead.

Which solution meets these requirements?

A. Upload the third-party signing certificate’s new private key to the AWS identity provider entity defined in AWS Identity and Access Management (IAM) by using the AWS Management Console.

B. Sign the identity provider's metadata file with the new public key. Upload the signature to the AWS identity provider entity defined in AWS Identity and Access Management (IAM) by using the AWS CLI.

C. Download the updated SAML metadata file from the identity service provider. Update the file in the AWS identity provider entity defined in AWS Identity and Access Management (IAM) by using the AWS CLI.

D. Configure the AWS identity provider entity defined in AWS Identity and Access Management (IAM) to synchronously fetch the new public key by using the AWS Management Console.

ANSWER: C

Explanation:

This error occurs when AWS STS cannot validate the XML signature on the SAML assertion because the certificate AWS has on record for the SAML identity provider no longer matches the certificate that the third-party identity provider is using to sign responses. In AWS, the trusted signing certificate is stored as part of the IAM SAML provider configuration, and AWS uses the certificate embedded in the SAML provider metadata document to validate incoming assertions. When the IdP rotates/renews its signing certificate, the operationally simplest and correct fix is to obtain the IdP’s updated SAML metadata (which includes the new X.509 signing certificate) and update the existing IAM SAML provider with that new metadata. This restores signature validation immediately without changing application logic, roles, or trust policies, and it keeps ongoing operations straightforward because future rotations follow the same metadata update process. AWS provides an API/CLI

operation to update a SAML provider's metadata document, which is the intended mechanism for certificate changes. See IAM SAML identity providers and aws iam update-saml-provider.

QUESTION NO: 10

A Security Engineer is building a Java application that is running on Amazon EC2. The application communicates with an Amazon RDS instance and authenticates with a user name and password.

Which combination of steps can the Engineer take to protect the credentials and minimize downtime when the credentials are rotated? (Choose two.)

A. Have a Database Administrator encrypt the credentials and store the ciphertext in Amazon S3. Grant permission to the instance role associated with the EC2 instance to read the object and decrypt the ciphertext.

B. Configure a scheduled job that updates the credential in AWS Systems Manager Parameter Store and notifies the Engineer that the application needs to be restarted.

C. Configure automatic rotation of credentials in AWS Secrets Manager.

D. Store the credential in an encrypted string parameter in AWS Systems Manager Parameter Store. Grant permission to the instance role associated with the EC2 instance to access the parameter and the AWS KMS key that is used to encrypt it.

E. Configure the Java application to catch a connection failure and make a call to AWS Secrets Manager to retrieve updated credentials when the password is rotated. Grant permission to the instance role associated with the EC2 instance to access Secrets Manager.

ANSWER: C E

Explanation:

Configuring automatic rotation of credentials in AWS Secrets Manager is a strong fit because it centralizes storage of the database username/password, encrypts the secret at rest with AWS KMS, and can automatically rotate supported Amazon RDS credentials using an AWS Lambda rotation function. This reduces operational risk from long-lived credentials and avoids manual rotation processes that often cause outages.

Configuring the Java application to catch a connection failure and make a call to AWS Secrets Manager to retrieve updated credentials when the password is rotated further minimizes downtime because the application can recover quickly when existing pooled connections fail after rotation. With the EC2 instance role permitted to call Secrets Manager (and decrypt via KMS as needed), the application can fetch the current secret value at runtime and re-establish database connectivity without requiring a restart or redeploy. This pattern aligns with AWS guidance to retrieve secrets programmatically and to design applications to handle rotation events gracefully.

References: https://docs.aws.amazon.com/secretsmanager/latest/userguide/rotating-secrets.html, https://docs.aws.amazon.com/secretsmanager/latest/userguide/retrieving-secrets.html

QUESTION NO: 11

A company has a single AWS account and uses an Amazon EC2 instance to test application code. The company recently discovered that the instance was compromised. The instance was serving up malware. The analysis of the instance showed that the instance was compromised 35 days ago.

A security engineer must implement a continuous monitoring solution that automatically notifies the company's security team about compromised instances through an email distribution list for high severity findings. The security engineer must implement the solution as soon as possible.

Which combination of steps should the security engineer take to meet these requirements? (Choose three.) A. Enable AWS Security Hub in the AWS account.

B. Enable Amazon GuardDuty in the AWS account.

C. Create an Amazon Simple Notification Service (Amazon SNS) topic. Subscribe the security team's email distribution list to the topic.

D. Create an Amazon Simple Queue Service (Amazon SQS) queue. Subscribe the security team's email distribution list to the queue.

E. Create an Amazon EventBridge (Amazon CloudWatch Events) rule for GuardDuty findings of high severity. Configure the rule to publish a message to the topic.

F. Create an Amazon EventBridge (Amazon CloudWatch Events) rule for Security Hub findings of high severity. Configure the rule to publish a message to the queue.

Answer: B C E

Explanation:

A. Enable Amazon GuardDuty in the AWS account.

B. Create an Amazon Simple Notification Service (Amazon SNS) topic. Subscribe the security team's email distribution list to the topic.

C. Create an Amazon Simple Queue Service (Amazon SQS) queue. Subscribe the security team's email distribution list to the queue.

D. Create an Amazon EventBridge (Amazon CloudWatch Events) rule for GuardDuty findings of high severity. Configure the rule to publish a message to the topic.

E. Create an Amazon EventBridge (Amazon CloudWatch Events) rule for Security Hub findings of high severity. Configure the rule to publish a message to the queue.

ANSWER: A B D

Explanation:

Enabling Amazon GuardDuty in the AWS account provides continuous threat detection for the account, including findings that can indicate compromised EC2 instances (for example, malware-related activity or command-and-control behavior). GuardDuty generates findings with a severity value, which makes it straightforward to filter for high severity events. To notify an email distribution list quickly and natively, creating an Amazon Simple Notification Service (Amazon SNS) topic and subscribing the security team's email distribution list to the topic is the standard approach; SNS supports email subscriptions out of the box and is designed for fan-out notifications. Finally, creating an Amazon EventBridge (Amazon CloudWatch Events) rule for GuardDuty findings of high severity and configuring the rule to publish a message to the topic connects detection to notification without custom code. EventBridge can match GuardDuty finding events and route only those with the desired severity to SNS, meeting the "automatic" and "as soon as possible" requirements using managed services. This pattern is a common AWS-recommended integration for alerting on GuardDuty findings.

References: https://docs.aws.amazon.com/guardduty/latest/ug/guardduty_findings.html, https://docs.aws.amazon.com/guardduty/latest/ug/guardduty_findings_cloudwatch.html

QUESTION NO: 12

A company deployed Amazon GuardDuty In the us-east-1 Region. The company wants all DNS logs that relate to the company's Amazon EC2 instances to be inspected. What should a security engineer do to ensure that the EC2 instances are logged?

A. Use IPv6 addresses that are configured for hostnames.

B. Configure external DNS resolvers as internal resolvers that are visible only to IAM.

C. Use IAM DNS resolvers for all EC2 instances.

D. Configure a third-party DNS resolver with logging for all EC2 instances.

ANSWER: C

Explanation:

To have Amazon GuardDuty inspect DNS activity for Amazon EC2 instances, the instances must use the VPC-provided DNS resolver (the “AmazonProvidedDNS” resolver). GuardDuty’s DNS findings are generated from VPC DNS query telemetry that is available when instances resolve names through the Amazon VPC DNS infrastructure; this is the default behavior for instances in a VPC when “enableDnsSupport” and “enableDnsHostnames” are enabled and the instance is using the VPC’s default resolver (typically the VPC base address plus 2). Ensuring that all EC2 instances use the VPC-provided resolver keeps DNS query data in the path GuardDuty can analyze, allowing GuardDuty to detect suspicious domain lookups (for example, known C2 domains) and produce DNS-related findings. This aligns with GuardDuty’s documented data sources and how it monitors DNS requests originating from resources in your VPC.

References: https://docs.aws.amazon.com/guardduty/latest/ug/guardduty_data-sources.html, https://docs.aws.amazon.com/vpc/latest/userguide/vpc-dns.html

QUESTION NO: 13

A company uses SAML federation to grant users access to AWS accounts. A company workload that is in an isolated AWS account runs on immutable infrastructure with no human access to Amazon EC2. The company requires a specialized user known as a break glass user to have access to the workload AWS account and instances in the case of SAML errors. A recent audit discovered that the company did not create the break glass user for the AWS account that contains the workload.

The company must create the break glass user. The company must log any activities of the break glass user and send the logs to a security team.

Which combination of solutions will meet these requirements? (Choose two.)

A. Create a local individual break glass IAM user for the security team. Create a trail in AWS CloudTrail that has Amazon CloudWatch Logs turned on. Use Amazon EventBridge to monitor local user activities.

B. Create a break glass EC2 key pair for the AWS account. Provide the key pair to the security team. Use AWS CloudTrail to monitor key pair activity. Send notifications to the security team by using Amazon Simple Notification Service (Amazon SNS).

C. Create a break glass IAM role for the account. Allow security team members to perform the AssumeRoleWithSAML operation. Create an AWS CloudTrail trail that has Amazon CloudWatch Logs turned on. Use Amazon EventBridge to monitor security team activities.

D. Create a local individual break glass IAM user on the operating system level of each workload instance. Configure unrestricted security groups on the instances to grant access to the break glass IAM users.

E. Configure AWS Systems Manager Session Manager for Amazon EC2. Configure an AWS CloudTrail filter based on Session Manager. Send the results to an Amazon Simple Notification Service (Amazon SNS) topic.

ANSWER: A E

Explanation:

A break-glass design for a federated environment should provide an authentication path that does not depend on SAML, plus strong auditing and alerting. Creating a local individual break glass IAM user for the security team satisfies the requirement to access the AWS account even if SAML federation is unavailable. Pairing that with an AWS CloudTrail trail that delivers events to CloudWatch Logs enables centralized logging of all API activity performed by that IAM user, and Amazon EventBridge rules can match CloudTrail events (for example, by userIdentity.arn or userIdentity.type) to route near-real-time notifications to the security team.

For access to immutable EC2 instances without opening inbound ports or relying on SSH keys, AWS Systems Manager Session Manager provides controlled interactive access using the SSM agent and IAM permissions. Session Manager activity can be logged (session logs to CloudWatch Logs/S3) and API calls are captured in CloudTrail, allowing you to filter and alert on Session Manager-related events and notify the security team through Amazon SNS. This approach aligns with AWS best practices by avoiding unrestricted security groups and by ensuring all break-glass actions are auditable and alertable.

References: AWS CloudTrail User Guide, AWS Systems Manager Session Manager

QUESTION NO: 14

A security engineer needs to run an AWS CloudFormation script. The CloudFormation script builds AWS infrastructure to support a stack that includes web servers and a MySQL database. The stack has been deployed in pre-production environments and is ready for production.

The production script must comply with the principle of least privilege. Additionally, separation of duties must exist between the security engineer's IAM account and CloudFormation.

Which solution will meet these requirements?

A. Use IAM Access Analyzer policy generation to generate a policy that allows the CloudFormation script to run and manage the stack. Attach the policy to a new
IAM role. Modify the security engineer's IAM permissions to be able to pass the new role to CloudFormation.
Use IAM Access Analyzer policy generation to generate a policy that allows
the CloudFormation script to run and manage the stack. Attach the policy to a new IAM role. Modify the security engineers IAM permissions to be able to pass the new role to CloudFormation.
According to the AWS documentation, IAM Access Analyzer is a service that helps you identify the resources in your organization and accounts, such as Amazon S3 buckets or IAM roles, that are shared with an external entity. You can also use IAM Access Analyzer to generate fine-grained policies that grant least privilege access based on access activity and access attempts.
To use IAM Access Analyzer policy generation, you need to enable IAM Access Analyzer in your account or organization. You can then use the IAM console or the AWS CLI to generate a policy for a resource based on its access activity or access attempts. You can review and edit the generated
policy before applying it to the resource.
To use IAM Access Analyzer policy generation with CloudFormation, you can follow these steps: Run the CloudFormation script in a pre-production environment and monitor its access activity or access attempts using IAM Access Analyzer.
Use IAM Access Analyzer policy generation to generate a policy that allows the CloudFormation
script to run and manage the stack. The policy will include only the permissions that are necessary for the script to function.
Attach the policy to a new IAM role that has a trust relationship with CloudFormation. This will allow CloudFormation to assume the role and execute the script.
Modify the security engineers IAM permissions to be able to pass the new role to CloudFormation. This will allow the security engineer to launch the stack using the role.
Run the CloudFormation script in the production environment using the new role.
This solution will meet the requirements of least privilege and separation of duties, as it will limit the permissions of both CloudFormation and the security engineer to only what is needed for running and managing the stack.
Option B is incorrect because creating an IAM policy that allows ec2:* and rds:* permissions is not following the principle of least privilege, as it will grant more permissions than necessary for running and managing the stack. Moreover, modifying the security engineers IAM permissions to be able to assume the new role is not ensuring separation of duties, as it will allow the security engineer to bypass CloudFormation and directly access the resources.
Option C is incorrect because modifying the security engineers IAM permissions to be able to run the CloudFormation script is not ensuring separation of duties, as it will allow the security engineer to execute the script without using CloudFormation.

Option D is incorrect because creating an IAM policy that allows ec2:* and rds:* permissions is not following the principle of least privilege, as it will grant more permissions than necessary for running and managing the stack. Using the IAM policy simulator to confirm that the policy allows the AWS API calls that are necessary to build the stack is not sufficient, as it will not generate a fine-grained policy based on access activity or access attempts.

B. Create an IAM policy that allows ec2:* and rds:* permissions. Attach the policy to a new IAM role.
Modify the security engineer's IAM permissions to be able to assume the new role.

C. Use IAM Access Analyzer policy generation to generate a policy that allows the CloudFormation script to run and manage the stack. Modify the security
engineer's IAM permissions to be able to run the CloudFormation script.

D. Create an IAM policy that allows ec2:* and rds:* permissions. Attach the policy to a new IAM role. Use the IAM policy simulator to confirm that the policy allows the AWS API calls that are necessary to build the stack. Modify the security engineer's IAM permissions to be able to pass the new role to CloudFormation.

ANSWER: A

Explanation:

Use IAM Access Analyzer policy generation to generate a policy that allows the CloudFormation script to run and manage the stack. Attach the policy to a new IAM role. Modify the security engineer's IAM permissions to be able to pass the new role to CloudFormation. is the right approach because it cleanly implements both least privilege and separation of duties using CloudFormation's service role pattern. With a dedicated IAM role that CloudFormation assumes, the permissions needed to create and update the stack are held by CloudFormation at execution time, not by the engineer's user. The engineer only needs permission to initiate the stack operation and to pass the specific role to CloudFormation via iam:PassRole, which prevents the engineer from directly using broad infrastructure permissions outside the CloudFormation workflow. IAM Access Analyzer policy generation helps achieve least privilege by producing a policy based on observed access activity, which is especially useful after validating the template in pre-production and then generating a tighter production policy for the CloudFormation execution role. This aligns with AWS best practices for delegating permissions to roles and minimizing human user privileges while still enabling controlled automation.

References: https://docs.aws.amazon.com/IAM/latest/UserGuide/access-analyzer-policy-generation.html, https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/using-iam-servicerole.html

QUESTION NO: 15

A security engineer is troubleshooting an AWS Lambda function that is named MyLambdaFunction. The function is encountering an error when the function attempts to read the objects in an Amazon S3 bucket that is named DOC-EXAMPLE-BUCKET. The S3 bucket has the following bucket policy:

```
{
    "Effect": "Allow",
    "Principal": {
        "Service": "lambda.amazonaws.com"
    },
    "Action": "s3:GetObject",
    "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET",
    "Condition": {
        "ArnLike": {
            "aws:SourceArn": "arn:aws:lambda:::function:MyLambdaFunction"
        }
    }
}
```

Which change should the security engineer make to the policy to ensure that the Lambda function can read the bucket objects?

A. Remove the Condition element. Change the Principal element to the following:

B. Change the Action element to the following:

C. Change the Resource element to "arn:aws:s3:::DOC-EXAMPLE- BUCKET/*".

D. Change the Resource element to "arn:aws:lambda:::function:MyLambdaFunction". Change the Principal element to the following:

ANSWER: C

Explanation:

Change the Resource element to "arn:aws:s3:::DOC-EXAMPLE- BUCKET/*". is the required fix because Amazon S3 permissions for reading objects (for example, s3:GetObject) must be granted against the object ARNs, not just the bucket ARN. A common cause of Lambda access failures to S3 is a bucket policy that allows an object-level action but specifies only the bucket resource (arn:aws:s3:::bucket-name) rather than the object resource pattern (arn:aws:s3:::bucket-name/*). When the Lambda function tries to read an object key, S3 evaluates the request against the object ARN, and the policy won't match unless the object resource is included. Updating the Resource to include the /* suffix ensures the statement applies to all objects in DOC-EXAMPLE-BUCKET, allowing the function's role (or other allowed principal in the policy) to successfully perform object reads. This aligns with AWS's distinction between bucket-level and object-level resources in IAM and bucket policies.

References: https://docs.aws.amazon.com/AmazonS3/latest/userguide/using-with-iam.html, https://docs.aws.amazon.com/AmazonS3/latest/userguide/example-bucket-policies.html

QUESTION NO: 16

While securing the connection between a company's VPC and its on-premises data center, a Security Engineer sent a ping command from an on-premises host

(IP address 203.0.113.12) to an Amazon EC2 instance (IP address 172.31.16.139). The ping command did not return a response. The flow log in the VPC showed the following:

2 123456789010 eni-1235b8ca 203.0.113.12 172.31.16.139 0 0 1 4 336 1432917027 1432917142 ACCEPT

OK

2 123456789010 eni-1235b8ca 172.31.16.139 203.0.113.12 0 0 1 4 336 1432917094 1432917142 REJECT

OK

What action should be performed to allow the ping to work?

A. In the security group of the EC2 instance, allow inbound ICMP traffic.

B. In the security group of the EC2 instance, allow outbound ICMP traffic.

C. In the VPC's NACL, allow inbound ICMP traffic.

D. In the VPC's NACL, allow outbound ICMP traffic.

ANSWER: D

Explanation:

The VPC Flow Logs show the inbound ICMP echo request from 203.0.113.12 to 172.31.16.139 as ACCEPT, but the return traffic from 172.31.16.139 back to 203.0.113.12 is REJECT. That pattern strongly indicates the instance received the ping request, but the echo reply is being blocked on the way out. Because network ACLs are stateless, they must explicitly allow both directions of traffic. Even if the inbound ICMP is allowed, the outbound ICMP (echo reply) must also be allowed by the subnet's network ACL rules; otherwise, the return packet is dropped and the ping appears to fail. Security groups, in contrast, are stateful and would automatically allow the response if the request was permitted, so the stateless behavior aligns with a network ACL outbound rule issue. Therefore, updating the VPC's NACL to allow outbound ICMP traffic enables the echo reply to leave the subnet and reach the on-premises host. See the AWS documentation on NACL stateless behavior and rule requirements: https://docs.aws.amazon.com/vpc/latest/userguide/vpc-network-acls.html and Flow Logs interpretation: https://docs.aws.amazon.com/vpc/latest/userguide/flow-logs.html.

QUESTION NO: 17

What are the MOST secure ways to protect the AWS account root user of a recently opened AWS account? (Choose two.)

A. Use the AWS account root user access keys instead of the AWS Management Console.

B. Enable multi-factor authentication for the AWS IAM users with the AdministratorAccess managed policy attached to them.

C. Use AWS KMS to encrypt all AWS account root user and AWS IAM access keys and set automatic rotation to 30 days.

D. Do not create access keys for the AWS account root user; instead, create AWS IAM users.

E. Enable multi-factor authentication for the AWS account root user.

ANSWER: D E

Explanation:

The most secure approach to protecting the AWS account root user is to minimize its use and harden the few authentication paths that remain. Enabling multi-factor authentication for the AWS account root user adds a strong second factor to the highest-privilege identity in the account, significantly reducing the risk of compromise from password theft or credential reuse. AWS explicitly recommends enabling MFA on the root user as a foundational security best practice.

In addition, avoiding creation of access keys for the AWS account root user and instead creating AWS IAM users (or roles) ensures day-to-day administrative and programmatic access is performed with least privilege and with identities that can be governed (permission boundaries, access key rotation, CloudTrail attribution, and easy revocation). Root access keys are long-lived, extremely high impact if leaked, and are not intended for routine operations. Using IAM identities for normal administration also improves auditability because actions are attributable to specific users/roles rather than the shared root identity.

These two measures together align with AWS guidance to lock down root, use it only for a small set of account-level tasks, and rely on IAM for operational access. References: AWS IAM User Guide – Root user, AWS IAM Best practices.

QUESTION NO: 18

A company's AWS CloudTrail logs are all centrally stored in an Amazon S3 bucket. The security team controls the company's AWS account. The security team must prevent unauthorized access and tampering of the CloudTrail logs.

Which combination of steps should the security team take? (Choose three.)

A. Configure server-side encryption with AWS KMS managed encryption keys (SSE-KMS)

B. Compress log file with secure gzip.

C. Create an Amazon EventBridge (Amazon CloudWatch Events) rule to notify the security team of any modifications on CloudTrail log files.

D. Implement least privilege access to the S3 bucket by configuring a bucket policy.

E. Configure CloudTrail log file integrity validation.

F. Configure Access Analyzer for S3.

ANSWER: A C D E

Explanation:

To prevent unauthorized access to centrally stored CloudTrail logs in Amazon S3, the security team should enforce strong access controls and encryption. Implementing least privilege access to the S3 bucket by configuring a bucket policy ensures only required principals (such as the CloudTrail service and designated security roles) can write/read logs, and it can explicitly deny risky actions (for example, deletes) unless performed under tightly controlled conditions. Configuring server-side encryption with AWS KMS managed encryption keys (SSE-KMS) protects log data at rest and enables key policies and auditability around key usage, which is important for sensitive audit trails. To detect tampering, configuring CloudTrail log file integrity validation provides cryptographic digest files so the team can validate that delivered log files have not been modified or deleted after delivery. Together, these measures address both confidentiality (unauthorized access) and integrity (tamper detection) for CloudTrail logs stored in S3.

References: CloudTrail log file integrity validation, Amazon S3 bucket policies

QUESTION NO: 19

A business requires a forensic logging solution for hundreds of Docker-based apps running on Amazon EC2. The solution must analyze logs in real time, provide message replay, and persist logs.

Which Amazon Web Offerings (IAM) services should be employed to satisfy these requirements? (Select two.)

A. Amazon Athena

B. Amazon Kinesis

C. Amazon SQS

D. Amazon Elasticsearch

E. Amazon EMR

ANSWER: B D

Explanation:

Amazon Kinesis is a strong fit for forensic logging pipelines that need real-time analysis and the ability to replay messages. With Kinesis Data Streams, log events can be ingested at high throughput from hundreds of containerized applications, processed in near real time by consumers, and retained for a configurable period so consumers can re-read (replay) data from a given shard/sequence number. This directly supports both real-time analytics and message replay requirements.

Amazon Elasticsearch (Amazon OpenSearch Service) complements this by providing persistent, searchable storage and fast interactive analysis of log data once it is indexed. A common pattern is to stream logs into Kinesis and then deliver/index them into OpenSearch for durable retention, querying, dashboards, and forensic investigations across large volumes of events. This combination is widely used for centralized logging architectures on AWS.

References: https://docs.aws.amazon.com/streams/latest/dev/introduction.html, https://docs.aws.amazon.com/opensearch-service/latest/developerguide/what-is.html

QUESTION NO: 20

A company is undergoing a layer 3 and layer 4 DDoS attack on its web servers running on AWS.

Which combination of AWS services and features will provide protection in this scenario? (Choose three.)

A. Amazon Route 53

B. AWS Certificate Manager (ACM)

C. Amazon S3

D. AWS Shield

E. Network Load Balancer

F. Amazon GuardDuty

ANSWER: A D E

Explanation:

For layer 3 and layer 4 DDoS attacks (network/transport floods such as SYN/UDP floods), AWS's primary protections come from edge/network DDoS mitigation and scalable, anycast front doors that can absorb and distribute traffic. AWS Shield provides always-on detection and inline mitigation for common L3/L4 DDoS vectors, and Shield Advanced adds enhanced protections and response options for larger events. Amazon Route 53 contributes by using a highly available, globally distributed authoritative DNS service with built-in DDoS resilience, helping keep name resolution available during attacks and enabling traffic steering patterns that improve survivability. A Network Load Balancer operates at layer 4 and is designed for very high throughput and sudden traffic spikes; placing targets behind an NLB helps absorb volumetric L4 floods and

reduces direct exposure of instances by presenting a managed, scalable entry point. Together, these services form a standard AWS-recommended baseline for L3/L4 DDoS resilience: edge mitigation (Shield), resilient DNS (Route 53), and a scalable L4 front end (Network Load Balancer). See AWS guidance on DDoS resilience and Shield protections at AWS Best Practices for DDoS Resiliency and AWS DDoS protection overview.

QUESTION NO: 21

A company hosts multiple externally facing applications, each isolated in its own IAM account The company'B Security team has enabled IAM WAF. IAM Config. and Amazon GuardDuty on all accounts. The company's Operations team has also joined all of the accounts to IAM Organizations and established centralized logging for CloudTrail. IAM Config, and GuardDuty. The company wants the Security team to take a reactive remediation in one account, and automate implementing this remediation as proactive prevention in all the other accounts.

How should the Security team accomplish this?

A. Update the IAM WAF rules in the affected account and use IAM Firewall Manager to push updated IAM WAF rules across all other accounts.

B. Use GuardDuty centralized logging and Amazon SNS to set up alerts to notify all application teams of security incidents.

C. Use GuardDuty alerts to write an IAM Lambda function that updates all accounts by adding additional NACLs on the Amazon EC2 instances to block known malicious IP addresses.

D. Use IAM Shield Advanced to identify threats in each individual account and then apply the account-based protections to all other accounts through Organizations.

ANSWER: A

Explanation:

Updating the IAM WAF rules in the affected account and using IAM Firewall Manager to push updated IAM WAF rules across all other accounts is the right approach because it directly supports a “fix once, deploy everywhere” model for web-facing protections in a multi-account AWS Organizations environment. AWS WAF lets the security team implement a reactive remediation (for example, adding an IP set, managed rule, or custom rule to block an observed attack pattern) in the impacted account. AWS Firewall Manager is purpose-built to centrally manage and automatically enforce AWS WAF protections across multiple accounts and resources (such as ALBs, API Gateway, and CloudFront) using Organizations. This enables proactive prevention by ensuring the same updated WAF policy is consistently applied to other accounts without requiring each application team to manually replicate changes. It also aligns with centralized security governance and reduces configuration drift, which is especially important when accounts are isolated per application. For details on centralized WAF policy management across accounts, see AWS WAF and AWS Firewall Manager and AWS Organizations documentation.

QUESTION NO: 22

An ecommerce company is developing new architecture for an application release. The company needs to implement TLS for incoming traffic to the application. Traffic for the application will originate from the internet TLS does not have to be implemented in an end-to-end configuration because the company is concerned about impacts on performance. The incoming traffic types will be HTTP and HTTPS The application uses ports 80 and 443.

What should a security engineer do to meet these requirements?

A. Create a public Application Load Balancer. Create two listeners one listener on port 80 and one listener on port 443. Create one target group. Create a rule to forward traffic from port 80 to the listener on port 443 Provision a public TLS certificate in AWS Certificate Manager (ACM). Attach the certificate to the listener on port 443.

B. Create a public Application Load Balancer. Create two listeners one listener on port 80 and one listener on port 443. Create one target group. Create a rule to forward traffic from port 80 to the listener on port 443 Provision a public TLS certificate in AWS Certificate Manager (ACM). Attach the certificate to the listener on port 80.

C. Create a public Network Load Balancer. Create two listeners one listener on port 80 and one listener on port 443. Create one target group. Create a rule to forward traffic from port 80 to the listener on port 443. Set the protocol for the listener on port 443 to TLS.

D. Create a public Network Load Balancer. Create a listener on port 443. Create one target group. Create a rule to forward traffic from port 443 to the target group. Set the protocol for the listener on port 443 to TLS.

ANSWER: A

Explanation:

Create a public Application Load Balancer with listeners on ports 80 (HTTP) and 443 (HTTPS) and attach an ACM-managed public certificate to the HTTPS listener to terminate TLS at the load balancer. This meets the requirement to implement TLS for incoming internet traffic without requiring end-to-end encryption to the targets, because the ALB can decrypt client connections and then forward traffic to the target group using HTTP. Using an HTTP listener on port 80 also allows you to enforce secure access by configuring a redirect action from HTTP to HTTPS, ensuring clients ultimately use TLS while keeping backend processing simpler and avoiding the overhead of encrypting traffic all the way to the application instances. ACM is the standard AWS service for provisioning and managing public TLS certificates and integrates directly with ALB HTTPS listeners, including automatic renewals for eligible public certificates. This approach is a common AWS best practice for internet-facing web applications that need TLS at the edge with minimal operational burden.

References: https://docs.aws.amazon.com/elasticloadbalancing/latest/application/create-https-listener.html, https://docs.aws.amazon.com/acm/latest/userguide/acm-overview.html

QUESTION NO: 23

A company is using AWS WAF to protect a customized public API service that is based on Amazon EC2 instances. The API uses an Application Load Balancer.

The AWS WAF web ACL is configured with an AWS Managed Rules rule group. After a software upgrade to the API and the client application, some types of requests are no longer working and are causing application stability issues. A security engineer discovers that AWS WAF logging is not turned on for the web ACL.

The security engineer needs to immediately return the application to service, resolve the issue, and ensure that logging is not turned off in the future. The security engineer turns on logging for the web ACL and specifies Amazon Cloud-Watch Logs as the destination.

Which additional set of steps should the security engineer take to meet the re-quirements?

A. Edit the rules in the web ACL to include rules with Count actions. Review the logs to determine which rule is blocking the request. Modify the IAM policy of all AWS WAF administrators so that they cannot remove the log-ging configuration for any AWS WAF web ACLs.

B. Edit the rules in the web ACL to include rules with Count actions. Review the logs to determine which rule is blocking the request. Modify the AWS WAF resource policy so that AWS WAF administrators cannot remove the log-ging configuration for any AWS WAF web ACLs.

C. Edit the rules in the web ACL to include rules with Count and Challenge actions. Review the logs to determine which rule is blocking the request. Modify the AWS WAF resource policy so that AWS WAF administrators cannot remove the logging configuration for any AWS WAF web ACLs.

D. Edit the rules in the web ACL to include rules with Count and Challenge actions. Review the logs to determine which rule is blocking the request. Modify the IAM policy of all AWS WAF administrators so that they cannot remove the logging configuration for any AWS WAF web ACLs.

ANSWER: A

Explanation:

Editing the rules in the web ACL to use Count actions is the fastest safe way to restore service while still collecting evidence about what AWS WAF would have blocked. With managed rule groups, a common operational pattern is to temporarily switch problematic rules (or the entire rule group) to Count so requests are no longer blocked, then use the resulting WAF logs to pinpoint which specific rule labels and match details correlate with the failing API calls. Once identified, the engineer can tune the configuration (for example, add an exception via scope-down statements or rule exclusions) and then re-enable blocking where appropriate.

To ensure logging cannot be turned off in the future, the correct control is to restrict the IAM permissions that allow changing WAF logging configuration (for example, denying actions such as wafv2:PutLoggingConfiguration and wafv2:DeleteLoggingConfiguration except for a break-glass role). AWS WAF does not use a “resource policy” on the web ACL to govern who can disable logging; access is controlled through IAM. This combination meets all requirements: immediate recovery, root-cause identification via logs, and preventative guardrails via IAM.

References: AWS WAF rule actions (Count), AWS WAF logging

QUESTION NO: 24

A company deployed IAM Organizations to help manage its increasing number of IAM accounts. A security engineer wants to ensure only principals in the Organization structure can access a specic Amazon S3 bucket. The solution must also minimize operational overhead

Which solution will meet these requirements?

A. 1 Put all users into an IAM group with an access policy granting access to the J bucket.

B. Have the account creation trigger an IAM Lambda function that manages the bucket policy, allowing access to accounts listed in the policy only.

C. Add an SCP to the Organizations master account, allowing all principals access to the bucket.

D. Specify the organization ID in the global key condition element of a bucket policy, allowing all principals access.

ANSWER: D

Explanation:

Using an S3 bucket policy condition that checks the AWS Organizations identifier is the most scalable, low-ops way to ensure that only principals that belong to the company’s organization can access the bucket. Amazon S3 supports the global condition key `aws:PrincipalOrgID`, which evaluates the organization ID associated with the calling principal’s AWS account. By adding a bucket policy statement that allows access only when `aws:PrincipalOrgID` equals the company’s organization ID, the bucket automatically permits access from any current or future account in the organization without needing to update the policy as accounts are added or removed. This directly meets the requirement to restrict access to principals in the organization structure while minimizing operational overhead, because it avoids per-account allow lists, automation to rewrite policies, or ongoing IAM group management across accounts. This pattern is an AWS-recommended approach for organization-wide access controls on resource policies such as S3 bucket policies.

References: Amazon S3 bucket policy examples, IAM JSON policy global condition keys

QUESTION NO: 25

A security engineer is checking an AWS CloudFormation template for vulnerabilities. The security engineer finds a parameter that has a default value that exposes an application's API key in plaintext. The parameter is referenced several times throughout the template. The security engineer must replace the parameter while maintaining the ability to reference the value in the template.

Which solution will meet these requirements in the MOST secure way?

A. Store the API key value as a SecureString parameter in AWS Systems Manager Parameter Store. In the template, replace all references to the value with {{resolve:ssm:MySSMParameterName:1}}.

B. Store the API key value in AWS Secrets Manager. In the template, replace all references to the value with {{resolve:secretsmanager:MySecretId:SecretString}}.

C. Store the API key value in Amazon DynamoDB. In the template, replace all references to the value with {{resolve:dynamodb:MyTableName:MyPrimaryKey}}.

D. Store the API key value in a new Amazon S3 bucket. In the template, replace all references to the value with {{resolve:s3:MyBucketName:MyObjectName}}.

ANSWER: B

Explanation:

Store the API key value in AWS Secrets Manager. In the template, replace all references to the value with {{resolve:secretsmanager:MySecretId:SecretString}} is the most secure approach because Secrets Manager is purpose-built for managing sensitive application secrets (like API keys) with strong access control, encryption at rest using AWS KMS, and native secret rotation workflows. By using a CloudFormation dynamic reference to Secrets Manager, the secret value is retrieved securely at deployment time and is not embedded in the template, parameter defaults, or source control. This preserves the ability to reference the value multiple times throughout the template while keeping the secret out of plaintext. Additionally, Secrets Manager integrates tightly with IAM for least-privilege access and supports auditing via CloudTrail, which is important for security governance. Dynamic references are the recommended mechanism for injecting secrets into CloudFormation-managed resources without exposing them in the template body.

References: https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/dynamic-references.html, https://docs.aws.amazon.com/secretsmanager/latest/userguide/intro.html

QUESTION NO: 26

A security engineer recently rotated the host keys for an Amazon EC2 instance. The security engineer is trying to access the EC2 instance by using the EC2 Instance. Connect feature. However, the security engineer receives an error (or failed host key validation. Before the rotation of the host keys EC2 Instance Connect worked correctly with this EC2 instance.

What should the security engineer do to resolve this error?

A. Import the key material into AWS Key Management Service (AWS KMS).

B. Manually upload the new host key to the AWS trusted host keys database.

C. Ensure that the AmazonSSMManagedInstanceCore policy is attached to the EC2 instance profile.

D. Create a new SSH key pair for the EC2 instance.

ANSWER: B

Explanation:

Manually upload the new host key to the AWS trusted host keys database is the correct action because EC2 Instance Connect can perform SSH host key validation against a trusted host keys store. When you rotate an instance's SSH host keys (for example, the OpenSSH host keys under /etc/ssh), the fingerprint that EC2 Instance Connect expects no longer matches what the instance presents during the SSH handshake. That mismatch results in a host key validation failure even though the instance is otherwise reachable and previously worked. Updating the trusted host keys database with the instance's new host key (or its new fingerprint) re-establishes trust so EC2 Instance Connect can validate the server identity and proceed with the connection. This aligns with the security best practice of pinning/validating host keys to prevent man-in-the-middle attacks rather than disabling validation. After uploading the new host key, subsequent EC2 Instance Connect sessions should succeed without changing user key pairs or instance IAM/SSM configuration. See the EC2 Instance Connect documentation on host key validation and trusted keys for the authoritative workflow: https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-instance-connect-methods.html and https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-instance-connect-set-up.html.

QUESTION NO: 27

A security engineer is building a Java application that is running on Amazon EC2. The application communicates with an Amazon RDS instance and authenticates with a user name and password.

Which combination of steps can the engineer take to protect the credentials and minimize downtime when the credentials are rotated? (Choose two.)

A. Have a database administrator encrypt the credentials and store the ciphertext in Amazon S3. Grant permission to the instance role associated with the EC2 instance to read the object and decrypt the ciphertext.

B. Configure a scheduled job that updates the credential in AWS Systems Manager Parameter Store and notifies the engineer that the application needs to be restarted.

C. Configure automatic rotation of credentials in AWS Secrets Manager.

D. Store the credential in an encrypted string parameter in AWS Systems Manager Parameter Store. Grant permission to the instance role associated with the EC2 instance to access the parameter and the AWS KMS key that is used to encrypt it.

E. Configure the Java application to catch a connection failure and make a call to AWS Secrets Manager to retrieve updated credentials when the password is rotated. Grant permission to the instance role associated with the EC2 instance to access Secrets Manager.

ANSWER: C E

Explanation:

Using AWS Secrets Manager to manage the Amazon RDS username/password is a best-practice approach for protecting credentials because Secrets Manager encrypts secrets at rest with AWS KMS and provides fine-grained IAM access control. More importantly for minimizing downtime during rotation, Secrets Manager supports built-in automatic rotation for supported databases (including Amazon RDS engines) through an AWS Lambda rotation function, allowing credentials to be rotated regularly without manual handling of passwords.

To minimize downtime when rotation occurs, the application should be designed to retrieve the latest secret value at runtime rather than relying on a static configuration that requires a restart. A common pattern is to handle authentication/connection failures (or proactively refresh on a timer) and then call Secrets Manager to fetch the current credentials, so the application can re-establish connections using the rotated password with minimal interruption. Granting the EC2 instance role permission to read the secret enables this without embedding long-term credentials in code or on disk.

These two steps together provide secure storage, automated rotation, and an application-side retrieval strategy that reduces operational impact when the password changes. See AWS Secrets Manager rotation and Secrets Manager best practices.

QUESTION NO: 28

A company is designing a multi-account structure for its development teams. The company is using AWS Organizations and AWS IAM Identity Center (AWS Single Sign-On). The company must

implement a solution so that the development teams can use only specific AWS Regions and so that each AWS account allows access to only specific AWS services.

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

A. Use IAM Identity Center to set up service-linked roles with IAM policy statements that include the Condition, Resource, and NotAction elements to allow access to only the Regions and services that are needed.

B. Deactivate AWS Security Token Service (AWS STS) in Regions that the developers are not allowed to use.

C. Create SCPs that include the Condition, Resource, and NotAction elements to allow access to only the Regions and services that are needed.

D. For each AWS account, create tailored identity-based policies for IAM Identity Center. Use statements that include the Condition, Resource, and NotAction elements to allow access to only the Regions and services that are needed.

ANSWER: C

Explanation:

Create SCPs that include the Condition, Resource, and NotAction elements to allow access to only the Regions and services that are needed is the right approach because service control policies (SCPs) in AWS Organizations provide centralized, preventative guardrails that apply across accounts (or OUs) regardless of which IAM principal is used (including roles provisioned via IAM Identity Center). This is the lowest operational overhead option because you can define and attach a small number of SCPs at the OU level to consistently restrict allowed AWS Regions (commonly via aws:RequestedRegion conditions) and to constrain which services can be used (commonly via allow-list patterns using NotAction). SCPs set the maximum available permissions in an account, so even if a user is granted broader permissions through IAM Identity Center permission sets or account-local IAM policies, actions outside the approved Regions/services are still blocked. This matches the requirement to enforce both Region and service restrictions per account in a scalable way across a multi-account environment.

References: https://docs.aws.amazon.com/organizations/latest/userguide/orgs_manage_policies_scps.html, https://docs.aws.amazon.com/organizations/latest/userguide/orgs_manage_policies_scps_examples.html

QUESTION NO: 29

A security engineer needs to set up an Amazon CloudFront distribution for an Amazon S3 bucket that hosts a static website. The security engineer must allow only specified IP addresses to access the website. The security engineer also must prevent users from accessing the website directly by using S3 URLs.

Which solution will meet these requirements?

A. Generate an S3 bucket policy. Specify cloudfront.amazonaws.com as the principal. Use the aws:SourceIp condition key to allow access only if the request comes from the specified IP addresses.

B. Create a CloudFront origin access control (OAC). Create the S3 bucket policy so that only the OAC has access. Create an AWS WAF web ACL, and add an IP set rule. Associate the web ACL with the CloudFront distribution.

C. Implement security groups to allow only the specified IP addresses access and to restrict S3 bucket access by using the CloudFront distribution.

D. Create an S3 bucket access point to allow access from only the CloudFront distribution. Create an AWS WAF web ACL and add an IP set rule. Associate the web ACL with the CloudFront distribution.

ANSWER: B

Explanation:

Create a CloudFront origin access control (OAC). Create the S3 bucket policy so that only the OAC has access. Create an AWS WAF web ACL, and add an IP set rule. Associate the web ACL with the CloudFront distribution. This meets both requirements by enforcing that all content access must flow through CloudFront and then restricting who can reach CloudFront. OAC is the recommended mechanism to securely connect CloudFront to an S3 origin by using SigV4 signing; with an OAC-restricted bucket policy, direct requests to S3 object URLs are denied because the bucket only allows requests that are authenticated as coming from the CloudFront distribution. Separately, AWS WAF attached to CloudFront can enforce an allow list of client IPs using an IP set rule, ensuring only the specified source IP addresses can access the website. This combination is a standard pattern: CloudFront provides the public edge endpoint, S3 is kept private behind OAC, and AWS WAF provides IP-based access control at the edge. See CloudFront documentation on restricting access to S3 origins and AWS WAF documentation on associating a web ACL with CloudFront.

QUESTION NO: 30

A company accidentally deleted the private key for an Amazon Elastic Block Store (Amazon EBS)-backed Amazon EC2 instance. A security engineer needs to regain access to the instance.

Which combination of steps will meet this requirement? (Choose two.)

A. Stop the instance. Detach the root volume. Generate a new key pair.

B. Keep the instance running. Detach the root volume. Generate a new key pair.

C. When the volume is detached from the original instance, attach the volume to another instance as a data volume. Modify the authorized_keys file with a new public key. Move the volume back to the original instance. Start the instance.

D. When the volume is detached from the original instance, attach the volume to another instance as a data volume. Modify the authorized_keys file with a new private key. Move the volume back to the original instance. Start the instance.

E. When the volume is detached from the original instance, attach the volume to another instance as a data volume. Modify the authorized_keys file with a new public key. Move the volume back to the original instance that is running.

ANSWER: A C

Explanation:

To regain access to an Amazon EBS-backed EC2 instance after the original SSH private key is lost, the supported recovery approach is to take the instance offline, manipulate the root volume out-of-band, and then boot back up with a new key. The instance must be stopped first so the root EBS volume can be safely detached. After detaching the root volume, you attach it to a separate “rescue” instance as a secondary (data) volume, mount the filesystem, and update the appropriate user’s `~/.ssh/authorized_keys` file by adding a newly generated public key. Once the public key is in place, you detach the volume from the rescue instance, reattach it as the root volume to the original instance, and then start the original instance. This restores SSH access using the new key pair without needing the old private key. This is the documented EC2 procedure for replacing a lost key pair on EBS-backed instances. See AWS EC2 User Guide: Replacing a lost key pair and the broader connectivity troubleshooting guidance at Troubleshoot connecting to your instance.

QUESTION NO: 31

A company's security engineer wants to receive an email alert whenever Amazon GuardDuty, AWS Identity and Access Management Access Analyzer, or Amazon Made generate a high-severity security finding. The company uses AWS Control Tower to govern all of its accounts. The company also uses AWS Security Hub with all of the AWS service integrations turned on.

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

A. Set up separate AWS Lambda functions for GuardDuty, 1AM Access Analyzer, and Macie to call each service's public API to retrieve high-severity findings. Use Amazon Simple Notification Service (Amazon SNS) to send the email alerts. Create an Amazon EventBridge rule to invoke the functions on a schedule.

B. Create an Amazon EventBridge rule with a pattern that matches Security Hub findings events with high severity. Configure the rule to send the findings to a target Amazon Simple Notification Service (Amazon SNS) topic. Subscribe the desired email addresses to the SNS topic.

C. Create an Amazon EventBridge rule with a pattern that matches AWS Control Tower events with high severity. Configure the rule to send the findings to a target Amazon Simple Notification Service (Amazon SNS) topic. Subscribe the desired email addresses to the SNS topic.

D. Host an application on Amazon EC2 to call the GuardDuty, 1AM Access Analyzer, and Macie APIs. Within the application, use the Amazon Simple Notification Service (Amazon SNS) API to retrieve high-severity findings and to send the findings to an SNS topic. Subscribe the desired email addresses to the SNS topic.

ANSWER: B

Explanation:

Creating an Amazon EventBridge rule that matches AWS Security Hub findings with high severity and routing those events to an Amazon SNS topic is the lowest-operations approach because it uses fully managed, event-driven integrations that already aggregate findings across services. With AWS Security Hub service integrations enabled, findings from Amazon GuardDuty, IAM Access Analyzer, and Amazon Macie are normalized into the AWS Security Finding Format (ASFF) and emitted as Security Hub events. EventBridge can filter these events (for example, by severity label such as HIGH/CRITICAL)

and deliver them directly to an SNS topic without any custom polling, servers, or scheduled jobs. Subscribing email endpoints to the SNS topic provides the required email notifications, and this pattern scales cleanly across a multi-account environment governed by AWS Control Tower because Security Hub can be administered centrally and events can be routed in the delegated administrator/administrator account. This minimizes operational overhead by avoiding custom code, EC2 management, and per-service API orchestration, while leveraging native integrations designed for security alerting workflows.

References: https://docs.aws.amazon.com/securityhub/latest/userguide/securityhub-cloudwatch-events.html, https://docs.aws.amazon.com/eventbridge/latest/userguide/eb-rules.html

QUESTION NO: 32

A company needs a solution to protect critical data from being permanently deleted. The data is stored in Amazon S3 buckets.

The company needs to replicate the S3 objects from the company's primary AWS Region to a secondary Region to meet disaster recovery requirements. The company must also ensure that users who have administrator access cannot permanently delete the data in the secondary Region.

Which solution will meet these requirements?

A. Configure AWS Backup to perform cross-Region S3 backups. Select a backup vault in the secondary Region. Enable AWS Backup Vault Lock in governance mode for the backups in the secondary Region

B. Implement S3 Object Lock in compliance mode in the primary Region. Configure S3 replication to replicate the objects to an S3 bucket in the secondary Region.

C. Configure S3 replication to replicate the objects to an S3 bucket in the secondary Region. Create an S3 bucket policy to deny the s3:ReplicateDelete action on the S3 bucket in the secondary Region

D. Configure S3 replication to replicate the objects to an S3 bucket in the secondary Region. Configure S3 object versioning on the S3 bucket in the secondary Region.

ANSWER: B

Explanation:

Implementing S3 Object Lock in compliance mode in the primary Region and configuring S3 replication to a secondary Region meets both the disaster recovery and anti-deletion requirements because Object Lock provides WORM (write once, read many) protection that cannot be overridden by any user, including administrators, during the retention period. In compliance mode, S3 enforces retention and legal holds so that protected object versions cannot be deleted or overwritten, and the retention settings cannot be shortened or removed by any IAM principal. When you replicate objects to a bucket in another Region using S3 Replication, you can preserve object lock protections so the destination copy remains immutable as well, which directly addresses the requirement that even administrators cannot permanently delete the data in the secondary Region. This approach is purpose-built for preventing permanent deletion while still enabling cross-Region resiliency for DR. See Amazon S3 Object Lock and Replicating objects (S3 Replication).

QUESTION NO: 33

The security engineer is managing a traditional three-tier web application that is running on Amazon EC2 instances. The application has become the target of increasing numbers of malicious attacks from the internet.

What steps should the security engineer take to check for known vulnerabilities and limit the attack surface? (Choose two.)

A. Use AWS Certificate Manager to encrypt all traffic between the client and application servers.

B. Review the application security groups to ensure that only the necessary ports are open.

C. Use Elastic Load Balancing to offload Secure Sockets Layer encryption.

D. Use Amazon Inspector to periodically scan the backend instances.

E. Use AWS Key Management Service (AWS KMS) to encrypt all the traffic between the client and application servers.

ANSWER: B D

Explanation:

To check for known vulnerabilities on Amazon EC2, the most direct AWS-native approach is to use Amazon Inspector to run recurring assessments against the instances. Inspector can identify software vulnerabilities (CVEs) and unintended network exposure, and it integrates with EC2 and Systems Manager to continuously evaluate findings over time. This directly addresses the requirement to “check for known vulnerabilities” in a scalable, automated way.

To limit the attack surface for a three-tier application exposed to the internet, reviewing and tightening security group rules is a foundational control. Ensuring that only required inbound ports are open (for example, restricting database tiers from any internet access and allowing only necessary inter-tier traffic) reduces reachable services and minimizes exposure to scanning and exploitation attempts. This is a key best practice for reducing the externally accessible footprint of the application.

References: Amazon Inspector User Guide, Amazon VPC Security Groups

QUESTION NO: 34

A security engineer is using AWS Organizations and wants to optimize SCPs. The security engineer needs to ensure that the SCPs conform to best practices.

Which approach should the security engineer take to meet this requirement?

A. Use AWS IAM Access Analyzer to analyze the polices. View the findings from policy validation checks.

B. Review AWS Trusted Advisor checks for all accounts in the organization.

C. Set up AWS Audit Manager. Run an assessment for all AWS Regions for all accounts.

D. Ensure that Amazon Inspector agents are installed on all Amazon EC2 instances in all accounts.

ANSWER: A

Explanation:

Use AWS IAM Access Analyzer to analyze the polices. View the findings from policy validation checks is the right approach because IAM Access Analyzer includes policy validation capabilities that check IAM policy JSON (including Organizations service control policies) against AWS best practices and common policy authoring issues. These checks can identify problems such as malformed policy elements, unsupported actions/condition keys, overly broad statements, and other findings that help a security engineer harden and standardize SCPs across an organization. This directly supports “optimizing” SCPs by improving correctness and maintainability and by reducing the risk of unintended permissions behavior caused by policy mistakes. In contrast to account-level operational tools, policy validation is purpose-built for evaluating policy documents and producing actionable findings that can be used to iteratively refine SCPs before broad rollout. This aligns with AWS guidance to validate policies and use automated analysis to detect issues early in the policy lifecycle.

References: https://docs.aws.amazon.com/IAM/latest/UserGuide/access-analyzer-policy-validation.html, https://docs.aws.amazon.com/organizations/latest/userguide/orgs_manage_policies_scps.html

QUESTION NO: 35

A company has several workloads running on AWS. Employees are required to authenticate using on-premises ADFS and SSO to access the AWS Management

Console. Developers migrated an existing legacy web application to an Amazon EC2 instance. Employees need to access this application from anywhere on the internet, but currently, there is no authentication system built into the application.

How should the Security Engineer implement employee-only access to this system without changing the application?

A. Place the application behind an Application Load Balancer (ALB). Use Amazon Cognito as authentication for the ALB. Define a SAML-based Amazon Cognito user pool and connect it to ADFS.

B. Implement AWS SSO in the master account and link it to ADFS as an identity provider. Define the EC2 instance as a managed resource, then apply an IAM policy on the resource.

C. Define an Amazon Cognito identity pool, then install the connector on the Active Directory server. Use the Amazon Cognito SDK on the application instance to authenticate the employees using their Active Directory user names and passwords.

D. Create an AWS Lambda custom authorizer as the authenticator for a reverse proxy on Amazon EC2. Ensure the security group on Amazon EC2 only allows access from the Lambda function.

ANSWER: A

Explanation:

Place the application behind an Application Load Balancer (ALB). Use Amazon Cognito as authentication for the ALB. Define a SAML-based Amazon Cognito user pool and connect it to ADFS is correct because it adds an authentication layer in front of the legacy application without requiring any code changes. ALB supports built-in authentication actions (authenticate with Cognito) at the listener rule level, so unauthenticated users are redirected to a hosted sign-in experience and only receive traffic forwarding to the target group after successful authentication. By configuring the Cognito user pool to federate with the existing on-premises ADFS using SAML 2.0, employees can continue using their corporate identities and SSO flow while the application remains unchanged on the EC2 instance. This pattern is a common “auth at the edge” approach for legacy apps and provides centralized control, session handling, and the ability to restrict access to authenticated users only, while still allowing internet reachability through the ALB. See ALB authentication with Cognito at https://docs.aws.amazon.com/elasticloadbalancing/latest/application/listener-authenticate-users.html and Cognito federation with SAML identity providers at https://docs.aws.amazon.com/cognito/latest/developerguide/cognito-user-pools-saml-idp.html.

QUESTION NO: 36

A company uses AWS Organizations and has production workloads across multiple AWS accounts. A security engineer needs to design a solution that will proactively monitor for suspicious behavior across all the accounts that contain production workloads.

The solution must automate remediation of incidents across the production accounts. The solution also must publish a notification to an Amazon Simple Notification Service (Amazon SNS) topic when a critical security finding is detected. In addition, the solution must send all security incident logs to a dedicated account.

Which solution will meet these requirements?

A. Activate Amazon GuardDuty in each production account. In a dedicated logging account, aggregate all GuardDuty logs from each production account. Remediate incidents by configuring GuardDuty to directly invoke an AWS Lambda function. Configure the Lambda function to also publish notifications to the SNS topic.

B. Activate AWS Security Hub in each production account. In a dedicated logging account, aggregate all Security Hub findings from each production account. Remediate incidents by using AWS Config and AWS Systems Manager. Configure Systems Manager to also publish notifications to the SNS topic.

C. Activate Amazon GuardDuty in each production account. In a dedicated logging account, aggregate all GuardDuty logs from each production account. Remediate incidents by using Amazon EventBridge to invoke a custom AWS Lambda function from the GuardDuty findings. Configure the Lambda function to also publish notifications to the SNS topic.

D. Activate AWS Security Hub in each production account. In a dedicated logging account, aggregate all Security Hub findings from each production account. Remediate incidents by using Amazon EventBridge to invoke a custom AWS Lambda function from the Security Hub findings. Configure the Lambda function to also publish notifications to the SNS topic.

ANSWER: D

Explanation:

Activate AWS Security Hub in each production account. In a dedicated logging account, aggregate all Security Hub findings from each production account. Remediate incidents by using Amazon EventBridge to invoke a custom AWS Lambda function from the Security Hub findings. Configure the Lambda function to also publish notifications to the SNS topic. This works well in an AWS Organizations multi-account setup because Security Hub is designed to centralize, normalize, and prioritize security findings from multiple sources (including GuardDuty, Inspector, IAM Access Analyzer, and partner products) and supports cross-account aggregation into an administrator account. Once findings are centralized, Amazon EventBridge can match on finding attributes such as severity/label (for example, CRITICAL) and route those events to automated responders like AWS Lambda. The Lambda function can then implement remediation steps (for example, isolating an instance, revoking credentials, updating security group rules) and publish a message to an SNS topic for notification workflows. Central aggregation into a dedicated account satisfies the requirement to send security incident logs/findings to a separate account for governance and retention. References: https://docs.aws.amazon.com/securityhub/latest/userguide/securityhub-accounts.html and https://docs.aws.amazon.com/securityhub/latest/userguide/securityhub-cloudwatch-events.html.

QUESTION NO: 37

A company is building a data processing application mat uses AWS Lambda functions. The application's

Lambda functions need to communicate with an Amazon RDS OB instance that is deployed within a VPC in the same AWS account

Which solution meets these requirements in the MOST secure way?

A. Configure the DB instance to allow public access Update the DB instance security group to allow access from the Lambda public address space for the AWS Region

B. Deploy the Lambda functions inside the VPC Attach a network ACL to the Lambda subnet Provide outbound rule access to the VPC CIDR range only Update the DB instance security group to allow traffic from 0.0.0.0/0

C. Deploy the Lambda functions inside the VPC Attach a security group to the Lambda functions Provide outbound rule access to the VPC CIDR range only Update the DB instance security group to allow traffic from the Lambda security group

D. Peer the Lambda default VPC with the VPC that hosts the DB instance to allow direct network access without the need for security groups

ANSWER: C

Explanation:

Deploy the Lambda functions inside the VPC Attach a security group to the Lambda functions Provide outbound rule access to the VPC CIDR range only Update the DB instance security group to allow traffic from the Lambda security group is the most secure approach because it keeps all database connectivity on private VPC networking and enforces least-privilege access using security groups. When a Lambda function is configured for VPC access, it uses elastic network interfaces in the specified subnets, allowing it to reach private resources such as an RDS DB instance without exposing the database to the public internet. Using a dedicated security group for the Lambda function and referencing that security group as the only allowed source in the DB instance security group tightly scopes inbound access to just that workload, rather than broad IP ranges. Restricting the Lambda security group egress to only the VPC CIDR (or even more narrowly to the DB port and DB subnet CIDRs/prefix lists where applicable) further reduces the blast radius if the function is compromised. This pattern aligns with AWS guidance for private RDS access from Lambda and security-group-based controls for stateful, identity-based network permissions.

References: https://docs.aws.amazon.com/lambda/latest/dg/configuration-vpc.html, https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-security-groups.html

QUESTION NO: 38

A company has a set of EC2 Instances hosted in IAM. The EC2 Instances have EBS volumes which is used to store critical information. There is a business continuity requirement to ensure high availability for the EBS volumes. How can you achieve this?

A. Use lifecycle policies for the EBS volumes

B. Use EBS Snapshots

C. Use EBS volume replication

D. Use EBS volume encryption

ANSWER: B

Explanation:

Use EBS Snapshots is the right approach because Amazon EBS volumes are designed for high availability within a single Availability Zone (AZ) by replicating data across multiple servers in that AZ, but they are not inherently highly available across AZs. For business continuity, you typically need a durable, restorable copy that can be used to recreate volumes in the same or a different AZ (and even in another Region if copied). EBS snapshots are incremental, stored in Amazon S3 (managed by AWS), and provide a standard mechanism to back up EBS data and restore it by creating new volumes from snapshots. This enables recovery from volume corruption, accidental deletion, or an AZ impairment by restoring volumes elsewhere and reattaching them to replacement instances. You can also automate snapshot creation and retention using Amazon Data Lifecycle Manager (DLM) policies, which is commonly used to meet RPO/RTO objectives for EBS-backed workloads. For business continuity planning, snapshots (and optionally copying them cross-Region) are the foundational AWS-native method for EBS data protection and recovery.

References: Amazon EBS snapshots (EC2 User Guide), Amazon Data Lifecycle Manager for EBS snapshots

QUESTION NO: 39

A company has several workloads running on AWS. Employees are required to authenticate using on-premises ADFS and SSO to access the AWS Management Console. Developers migrated an existing legacy web application to an Amazon EC2 instance. Employees need to access this application from anywhere on the internet, but currently, there is no authentication system built into the application.

How should the security engineer implement employee-only access to this system without changing the application?

A. Place the application behind an Application Load Balancer (ALB). Use Amazon Cognito as authentication for the ALB. Define a SAML-based Amazon Cognito user pool and connect it to ADFS.

B. Implement AWS IAM Identity Center (AWS Single Sign-On) in the management account and link it to ADFS as an identity provider. Define the EC2 instance as a managed resource, then apply an IAM policy on the resource.

C. Define an Amazon Cognito identity pool, then install the connector on the Active Directory server. Use the Amazon Cognito SDK on the application instance to authenticate the employees using their Active Directory user names and passwords.

D. Create an AWS Lambda custom authorizer as the authenticator for a reverse proxy on Amazon EC2. Ensure the security group on Amazon EC2 only allows access from the Lambda function.

ANSWER: A

Explanation:

Place the application behind an Application Load Balancer (ALB). Use Amazon Cognito as authentication for the ALB. Define a SAML-based Amazon Cognito user pool and connect it to ADFS is correct because ALB can enforce authentication at the edge before any request reaches the legacy application, which satisfies the requirement to avoid changing the application itself. ALB supports an “authenticate” action that integrates with an OIDC-compatible IdP, and Amazon Cognito user pools can federate enterprise identities via SAML 2.0 (such as on-premises ADFS). With this pattern, users who are not already authenticated are redirected through the Cognito hosted UI, Cognito delegates authentication to ADFS, and then

ALB forwards only authenticated requests to the EC2 target. This provides employee-only access from anywhere on the internet while centralizing authentication and session handling at the load balancer layer, and it aligns with AWS best practices for adding authentication to existing apps without modifying code. See ALB authentication actions and Cognito federation documentation: https://docs.aws.amazon.com/elasticloadbalancing/latest/application/listener-authenticate-users.html and https://docs.aws.amazon.com/cognito/latest/developerguide/cognito-user-pools-saml-idp.html.

QUESTION NO: 40

A Network Load Balancer (NLB) target instance is not entering the InService state. A security engineer determines that health checks are failing.

Which factors could cause the health check failures? (Select THREE.)

A. The target instance's security group does not allow traffic from the NLB.

B. The target instance's security group is not attached to the NLB.

C. The NLB's security group is not attached to the target instance.

D. The target instance's subnet network ACL does not allow traffic from the NLB.

E. The target instance's security group is not using IP addresses to allow traffic from the NLB.

F. The target network ACL is not attached to the NLB.

ANSWER: A D E

Explanation:

For Network Load Balancer health checks to succeed, the load balancer nodes must be able to reach the target on the configured health check protocol/port and receive a valid response. That requires the target instance’s security group to allow inbound traffic for the health check from the load balancer. If the security group blocks that traffic, the target will remain unhealthy. In addition, the subnet network ACLs on the target side must allow the inbound health check traffic and the return traffic (NACLs are stateless), otherwise the TCP connection (or HTTP/HTTPS request) will fail and the health check will time out. These are common causes when targets never transition to InService. Note that NLBs do not require (and historically did not support) security groups in the same way as ALBs; the key is permitting traffic to the targets and ensuring the subnet NACLs allow the flow. See the NLB health check behavior and target troubleshooting guidance in the Elastic Load Balancing documentation: https://docs.aws.amazon.com/elasticloadbalancing/latest/network/target-group-health-checks.html and https://docs.aws.amazon.com/elasticloadbalancing/latest/network/load-balancer-troubleshooting.html.

QUESTION NO: 41

A company has a group of Amazon EC2 instances in a single private subnet of a VPC with no internet gateway attached. A security engineer has installed the Amazon CloudWatch agent on all instances in that subnet to capture logs from a specific application. To ensure that the logs flow securely, the company's networking team has created VPC endpoints for CloudWatch monitoring and CloudWatch logs. The networking team has attached the endpoints to the VPC.

The application is generating logs. However, when the security engineer queries CloudWatch, the logs do not appear.

Which combination of steps should the security engineer take to troubleshoot this issue? (Choose three.)

A. Ensure that the EC2 instance profile that is attached to the EC2 instances has permissions to create log streams and write logs.

B. Create a metric filter on the logs so that they can be viewed in the AWS Management Console.

C. Check the CloudWatch agent configuration file on each EC2 instance to make sure that the CloudWatch agent is collecting the proper log files.

D. Check the VPC endpoint policies of both VPC endpoints to ensure that the EC2 instances have permissions to use them.

E. Create a NAT gateway in the subnet so that the EC2 instances can communicate with CloudWatch.

F. Ensure that the security groups allow all the EC2 instances to communicate with each other to aggregate logs before sending.

ANSWER: A C D

Explanation:

The most productive troubleshooting steps focus on the three prerequisites for CloudWatch Logs ingestion from private subnets: authorization, correct agent collection settings, and allowed access through the interface endpoints. First, the CloudWatch agent running on each instance uses the instance profile credentials, so the role must include CloudWatch Logs permissions such as creating log groups/streams and putting log events; without these, the agent will fail to deliver even if networking is correct. Second, the agent must be configured to actually read the intended application log paths and send them to the expected log group/stream; validating the CloudWatch agent configuration file (and that it matches the real file locations and permissions on disk) is essential. Third, when using interface VPC endpoints (AWS PrivateLink) for CloudWatch Logs and monitoring, the endpoint policy can explicitly allow/deny actions and principals; confirming the endpoint policies permit the instances' role to call CloudWatch Logs APIs through the endpoints is a key check in locked-down environments. Together, these steps validate the IAM, agent configuration, and endpoint authorization layers that commonly prevent logs from appearing even when the application is generating them. See https://docs.aws.amazon.com/AmazonCloudWatch/latest/logs/iam-access-control-overview-cwl.html and https://docs.aws.amazon.com/vpc/latest/privatelink/vpc-endpoints.html.

QUESTION NO: 42

A corporation is preparing to acquire several companies. A Security Engineer must design a solution to ensure that newly acquired IAM accounts follow the corporation's security best practices. The solution should monitor each Amazon S3 bucket for unrestricted public write access and use IAM managed services.

What should the Security Engineer do to meet these requirements?

A. Configure Amazon Macie to continuously check the configuration of all S3 buckets.

B. Enable IAM Config to check the configuration of each S3 bucket.

C. Set up IAM Systems Manager to monitor S3 bucket policies for public write access.

D. Configure an Amazon EC2 instance to have an IAM role and a cron job that checks the status of all S3 buckets.

ANSWER: B

Explanation:

Enable IAM Config to check the configuration of each S3 bucket is the right approach because AWS Config is the managed service designed to continuously assess, audit, and evaluate the configurations of AWS resources against desired security policies. With AWS Config, the Security Engineer can deploy managed rules (or custom rules) across newly acquired accounts—typically via AWS Organizations and conformance packs—to detect S3 buckets that allow public access, including risky bucket policies and ACL settings that could permit unrestricted public write. This provides continuous monitoring, configuration history, and automated compliance reporting without needing to run custom cron jobs or manage servers. In practice, you would enable AWS Config in each account/region, aggregate results centrally, and use relevant AWS managed rules such as checks for public read/write exposure (often implemented via rules that evaluate S3 public access settings and policies) to enforce the corporation's baseline. This aligns with the requirement to use AWS managed services and to monitor each bucket for public write exposure.

References: https://docs.aws.amazon.com/config/latest/developerguide/WhatIsConfig.html, https://docs.aws.amazon.com/config/latest/developerguide/managed-rules-by-aws-config.html

QUESTION NO: 43

A company has a batch-processing system that uses Amazon S3, Amazon EC2, and AWS Key Management Service (AWS KMS). The system uses two AWS accounts: Account A and Account B.

Account A hosts an S3 bucket that stores the objects that will be processed. The S3 bucket also stores the results of the processing. All the S3 bucket objects are encrypted by a KMS key that is managed in

Account A.

Account B hosts a VPC that has a fleet of EC2 instances that access the S3 buck-et in Account A by using statements in the bucket policy. The VPC was created with DNS hostnames enabled and DNS resolution enabled.

A security engineer needs to update the design of the system without changing any of the system's code. No AWS API calls from the batch-processing EC2 in-stances can travel over the internet.

Which combination of steps will meet these requirements? (Select TWO.)

A. In the Account B VPC, create a gateway VPC endpoint for Amazon S3. For the gateway VPC endpoint, create a resource policy that allows the s3:GetObject, s3:ListBucket, s3:PutObject, and s3:PutObjectAcl actions for the S3 bucket.

B. In the Account B VPC, create an interface VPC endpoint for Amazon S3. For the interface VPC endpoint, create a resource policy that allows the s3:GetObject, s3:ListBucket, s3:PutObject, and s3:PutObjectAcl actions for the S3 bucket.

C. In the Account B VPC, create an interface VPC endpoint for AWS KMS. For the interface VPC endpoint, create a resource policy that allows the kms:Encrypt, kms:Decrypt, and kms:GenerateDataKey actions for the KMS key. Ensure that private DNS is turned on for the endpoint.

D. In the Account B VPC, create an interface VPC endpoint for AWS KMS. For the interface VPC endpoint, create a resource policy that allows the kms:Encrypt, kms:Decrypt, and kms:GenerateDataKey actions for the KMS key. Ensure that private DNS is turned off for the endpoint.

E. In the Account B VPC, verify that the S3 bucket policy allows the s3:PutObjectAcl action for cross-account use. In the Account B VPC, create a gateway VPC endpoint for Amazon S3. For the gateway VPC endpoint, create a resource policy that allows the s3:GetObject, s3:ListBucket, and s3:PutObject actions for the S3 bucket.

ANSWER: A C

Explanation:

To ensure no AWS API calls from the EC2 instances traverse the public internet without changing application code, the design should use AWS PrivateLink/VPC endpoints so traffic stays on the AWS network. For Amazon S3 access from a VPC, the standard approach is to create a gateway VPC endpoint for Amazon S3 and associate it with the route tables used by the instances. This allows S3 API traffic to flow privately to S3 without requiring NAT gateways, internet gateways, or public IPs, and it works transparently for existing code that calls the normal S3 regional endpoints.

Because the S3 objects are encrypted with an AWS KMS key, the instances will also need private connectivity to AWS KMS to perform cryptographic operations (such as decrypting data keys) during S3 SSE-KMS workflows. Creating an interface VPC endpoint for AWS KMS and enabling private DNS ensures that calls to the standard KMS regional DNS name resolve to the endpoint’s private IPs, again avoiding code changes while keeping KMS API traffic off the internet.

References: https://docs.aws.amazon.com/vpc/latest/privatelink/vpc-endpoints.html, https://docs.aws.amazon.com/kms/latest/developerguide/kms-vpc-endpoint.html

QUESTION NO: 44

A company has an application that uses dozens of Amazon DynamoDB tables to store data. Auditors find that the tables do not comply with the company's data protection policy.

The company's retention policy states that all data must be backed up twice each month: once at midnight on the 15th day of the month and again at midnight on the 25th day of the month. The company must retain the backups for 3 months.

Which combination of steps should a security engineer take to meet these requirements? (Choose two.)

A. Use the DynamoDB on-demand backup capability to create a backup plan. Configure a lifecycle policy to expire backups after 3 months.

B. Use AWS DataSync to create a backup plan. Add a backup rule that includes a retention period of 3 months.

C. Use AWS Backup to create a backup plan. Add a backup rule that includes a retention period of 3 months.

D. Set the backup frequency by using a cron schedule expression. Assign each DynamoDB table to the backup plan.

E. Set the backup frequency by using a rate schedule expression. Assign each DynamoDB table to the backup plan.

ANSWER: C D

Explanation:

Using AWS Backup is the right approach because it provides centralized, policy-based backup management for supported AWS services, including Amazon DynamoDB. With AWS Backup, you can create a backup plan and define one or more backup rules that specify when backups run and how long recovery points are retained. The 3-month retention requirement is met by configuring the rule's lifecycle/retention settings so recovery points expire after the required period.

The required schedule is twice per month on specific calendar days at midnight (the 15th and 25th). This is best expressed with a cron schedule expression in the backup rule, because cron supports specifying exact days of the month and times. After defining the plan and rule, you then assign each DynamoDB table (or use tag-based assignments) to the backup plan so that all tables are covered consistently and auditors can verify compliance across dozens of tables.

References: AWS Backup Developer Guide – What is AWS Backup?, AWS Backup Developer Guide – Creating a backup plan

QUESTION NO: 45

A company has two IAM accounts within IAM Organizations. In Account-1. Amazon EC2 Auto Scaling is launched using a service-linked role. In Account-2. Amazon EBS volumes are encrypted with an IAM KMS key A Security Engineer needs to ensure that the service-linked role can launch instances with these encrypted volumes

Which combination of steps should the Security Engineer take in both accounts? (Select TWO.)

A. Allow Account-1 to access the KMS key in Account-2 using a key policy

B. Attach an IAM policy to the service-linked role in Account-1 that allows these actions CreateGrant. DescnbeKey, Encrypt, GenerateDataKey, Decrypt, and ReEncrypt

C. Create a KMS grant for the service-linked role with these actions CreateGrant, DescnbeKey Encrypt GenerateDataKey Decrypt, and ReEncrypt

D. Attach an IAM policy to the role attached to the EC2 instances with KMS actions and then allow Account-1 in the KMS key policy.

E. Attach an IAM policy to the user who is launching EC2 instances and allow the user to access the KMS key policy of Account-2.

ANSWER: A C

Explanation:

To launch EC2 instances that use EBS volumes encrypted with a customer managed KMS key in a different AWS account, you must authorize cross-account use of that KMS key and also ensure the principal that will use the key has permissions to call the required KMS APIs. The foundational step is to allow the external account (and/or specific principals in that account) in the KMS key policy in the account that owns the key, because KMS key policies are the primary resource-based control for cross-account access. Without that key policy trust, IAM permissions alone in the other account are insufficient.

In addition, the principal that will actually perform cryptographic operations for EBS must have KMS permissions. For EBS encryption, AWS commonly uses grants under the hood to allow services to use the key on your behalf, and explicitly

creating a KMS grant for the service-linked role is a valid way to delegate the needed KMS actions (such as Encrypt, Decrypt, ReEncrypt, GenerateDataKey, and DescribeKey) for the lifecycle of volume attachment and instance launch.

References: AWS KMS key policies, AWS KMS grants

QUESTION NO: 46

A company's public Application Load Balancer (ALB) recently experienced a DDoS attack. To mitigate this issue, the company deployed Amazon CloudFront in front of the ALB so that users would not directly access the Amazon EC2 instances behind the ALB.

The company discovers that some traffic is still coming directly into the ALB and is still being handled by the EC2 instances.

Which combination of steps should the company take to ensure that the EC2 instances will receive traffic only from CloudFront? (Choose two.)

A. Configure CloudFront to add a cache key policy to allow a custom HTTP header that CloudFront sends to the ALB.

B. Configure CloudFront to add a custom HTTP header to requests that CloudFront sends to the ALB.

C. Configure the ALB to forward only requests that contain the custom HTTP header.

D. Configure the ALB and CloudFront to use the X-Forwarded-For header to check client IP addresses.

E. Configure the ALB and CloudFront to use the same X.509 certificate that is generated by AWS Certificate Manager (ACM).

ANSWER: B C

Explanation:

To ensure the origin behind CloudFront only processes requests that actually traversed CloudFront, a common and effective pattern is to have CloudFront inject a shared “secret” header on origin requests and then enforce that header at the origin. CloudFront supports adding custom headers to origin requests, which lets you include a value that end users cannot reliably supply unless they bypass CloudFront and guess the secret. On the origin side, you then configure the load balancer path so that only requests containing that header are forwarded to targets (typically implemented with listener rules that route valid requests to the target group and send all other requests to a fixed response or a different target group). This combination prevents direct-to-ALB traffic from reaching the EC2 instances unless it includes the expected header value, thereby ensuring the instances effectively receive traffic only from CloudFront. This approach is documented as a way to restrict access to your origin by validating a custom header added by CloudFront. See https://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/add-origin-custom-headers.html and ALB listener rule behavior at https://docs.aws.amazon.com/elasticloadbalancing/latest/application/load-balancer-listeners.html.

QUESTION NO: 47

A development team is creating an open source toolset to manage a company's software as a service (SaaS) application. The company stores the code in a public repository so that anyone can view and download the toolset's code.

The company discovers that the code contains an IAM access key and secret key that provide access to internal resources in the company’s AWS environment

A security engineer must implement a solution to identify whether unauthorized usage of the exposed credentials has occurred. The solution also must prevent any additional usage of the exposed credentials.

Which combination of steps will meet these requirements? (Choose two.)

A. Use AWS Identity and Access Management Access Analyzer to determine which resources the exposed credentials accessed and who used them.

B. Deactivate the exposed IAM access key from the user’s IAM account.

C. Create a rule in Amazon GuardDuty to block the access key in the source code from being used.

D. Create a new IAM access key and secret key for the user whose credentials were exposed.

E. Generate an IAM credential report. Check the report to determine when the user that owns the access key last logged in.

ANSWER: B D

Explanation:

Deactivating the exposed IAM access key from the user's IAM account immediately prevents any additional API requests from succeeding with that compromised access key, which is the fastest containment action available for long-term access keys. After containment, creating a new IAM access key and secret key for the user whose credentials were exposed restores legitimate access for the application or developers while ensuring the leaked key can no longer be used. This aligns with AWS best practice for rotating compromised credentials: disable (or delete) the compromised key and issue a replacement, then update dependent systems to use the new key. To identify whether unauthorized usage occurred, you would typically review AWS CloudTrail event history/logs for activity performed with that access key (via the accessKeyId in events) and correlate with other detections, but the required "choose two" steps that both stop further use and enable continued authorized access are key deactivation and key replacement. AWS explicitly recommends rotating access keys and making old keys inactive during rotation or after exposure. See IAM guidance on managing and rotating access keys and the general AWS security best practices for credential handling.

References: https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_access-keys.html, https://docs.aws.amazon.com/IAM/latest/UserGuide/best-practices.html

QUESTION NO: 48

A security engineer needs to create an Amazon S3 bucket policy to grant least privilege read access to IAM user accounts that are named User1, User2, and User3. These IAM user accounts are members of the AuthorizedPeople IAM group. The security engineer drafts the following S3 bucket policy:

```
{
    "Version": "2012-10-17",
    "Id": "AuthorizedPeoplePolicy",
    "Statement": [
        {
            "Sid": "Actions-Authorized-People",
            "Effect": "Allow",
            "Action": [
                "s3:GetObject"
            ],
            "Resource": "arn:aws:s3:::authorized-people-bucket/*"
        }
    ]
}
```

When the security engineer tries to add the policy to the S3 bucket, the following error message appears: "Missing required field Principal."

The security engineer is adding a Principal element to the policy. The addition must provide read access to only User1, User2, and User3.

Which solution meets these requirements?

A)

```
"Principal": {
    "AWS": [
        "arn:aws:iam::1234567890:user/User1",
        "arn:aws:iam::1234567890:user/User2",
        "arn:aws:iam::1234567890:user/User3"
    ]
}
```

B)

```
"Principal": {
    "AWS": [
        "arn:aws:iam::1234567890:root"
    ]
}
```

C)

```
"Principal": {
    "AWS": [
        "*"
    ]
}
```

D)

```
"Principal": {    "AWS": "arn:aws:iam::1234567890:group/AuthorizedPeople"
}
```

A. Option A

B. Option B

C. Option C

D. Option D

ANSWER: A

Explanation:

To satisfy the “Missing required field Principal” error and still enforce least-privilege access to only three specific IAM users, the bucket policy must include an explicit `Principal` that names those users directly. In Amazon S3 bucket policies, the `Principal` element is mandatory for resource-based policies and identifies who the statement applies to. Because the requirement is to grant read access only to User1, User2, and User3 (not everyone in the group), the correct approach is to specify the three IAM user ARNs in the `AWS` principal list (for example, `"Principal": {"AWS": ["arn:aws:iam::ACCOUNT_ID:user/User1", ...]}`). This precisely scopes the statement to those identities and avoids relying on group membership, which cannot be referenced as a principal in an S3 bucket policy. This is consistent with AWS guidance that principals in resource-based policies are AWS accounts, IAM users, IAM roles, or services, and that you can list multiple principals by providing an array of ARNs.

References: https://docs.aws.amazon.com/AmazonS3/latest/userguide/example-bucket-policies.html, https://docs.aws.amazon.com/IAM/latest/UserGuide/reference_policies_elements_principal.html

QUESTION NO: 49

A security engineer is creating an AWS Lambda function. The Lambda function needs to use a role that is named LambdaAuditRole to assume a role that is named AcmeAuditFactoryRole in a different AWS account.

When the code is processed, the following error message appears: "An error oc-curred (AccessDenied) when calling the AssumeRole operation."

Which combination of steps should the security engineer take to resolve this er-ror? (Select TWO.)

A. Ensure that LambdaAuditRole has the sts:AssumeRole permission for Ac-meAuditFactoryRole.

B. Ensure that LambdaAuditRole has the AWSLambdaBasicExecutionRole managed policy attached.

C. Ensure that the trust policy for AcmeAuditFactoryRole allows the sts:AssumeRole action from LambdaAuditRole.

D. Ensure that the trust policy for LambdaAuditRole allows the sts:AssumeRole action from the lambda.amazonaws.com service.

E. Ensure that the sts:AssumeRole API call is being issued to the us-east-I Region endpoint.

ANSWER: A C

Explanation:

To successfully call STS AssumeRole across AWS accounts, two independent permission checks must pass: the caller's identity-based permissions and the target role's trust policy. First, the Lambda execution role (LambdaAuditRole) must be allowed to call `sts:AssumeRole` on the specific target role ARN (AcmeAuditFactoryRole). This is done with an IAM policy attached to LambdaAuditRole granting `sts:AssumeRole` for that role resource. Second, the target role (AcmeAuditFactoryRole) must explicitly trust the calling principal. In a cross-account scenario, that trust relationship is expressed in AcmeAuditFactoryRole's trust policy by allowing the principal (LambdaAuditRole, or the source account with appropriate conditions) to perform `sts:AssumeRole`. If either side is missing, STS returns `AccessDenied` for the AssumeRole call. These two steps directly address the error and align with AWS's documented cross-account role assumption model. See Switching to a role (IAM User Guide) and AssumeRole API (STS).