Free Questions for SCS-C02 by certsdeals

Shared by Pruitt on 12-12-2023

For More Free Questions and Preparation Resources

Check the Links on Last Page

Question 1

Question Type: MultipleChoice

A company uses infrastructure as code (IaC) to create AWS infrastructure. The company writes the code as AWS CloudFormation templates to deploy the infrastructure. The company has an existing CI/CD pipeline that the company can use to deploy these templates.

After a recent security audit, the company decides to adopt a policy-as-code approach to improve the company's security posture on AWS. The company must prevent the deployment of any infrastructure that would violate a security policy, such as an unencrypted Amazon Elastic Block Store (Amazon EBS) volume.

Which solution will meet these requirements?

Options:

A- Turn on AWS Trusted Advisor. Configure security notifications as webhooks in the preferences section of the CI/CD pipeline.

B- Turn on AWS Config. Use the prebuilt rules or customized rules. Subscribe the CI/CD pipeline to an Amazon Simple Notification Service (Amazon SNS) topic that receives notifications from AWS Config.

C- Create rule sets in AWS CloudFormation Guard. Run validation checks for CloudFormation templates as a phase of the CI/CD process.

D- Create rule sets as SCPs. Integrate the SCPs as a part of validation control in a phase of the CI/CD process.

Answer:

C

Explanation:

The correct answer is C. Create rule sets in AWS CloudFormation Guard. Run validation checks for CloudFormation templates as a phase of the CI/CD process.

This answer is correct because AWS CloudFormation Guard is a tool that helps you implement policy-as-code for your CloudFormation templates. You can use Guard to write rules that define your security policies, such as requiring encryption for EBS volumes, and then validate your templates against those rules before deploying them. You can integrate Guard into your CI/CD pipeline as a step that runs the validation checks and prevents the deployment of any non-compliant templates12.

The other options are incorrect because:

A) Turning on AWS Trusted Advisor and configuring security notifications as webhooks in the preferences section of the CI/CD pipeline is not a solution, because AWS Trusted Advisor is not a policy-as-code tool, but a service that provides recommendations to help you follow AWS best practices. Trusted Advisor does not allow you to define your own security policies or validate your CloudFormation templates against them3.

B) Turning on AWS Config and using the prebuilt or customized rules is not a solution, because AWS Config is not a policy-as-code tool, but a service that monitors and records the configuration changes of your AWS resources. AWS Config does not allow you to validate your CloudFormation templates before deploying them, but only evaluates the compliance of your resources after they are created4.

D) Creating rule sets as SCPs and integrating them as a part of validation control in a phase of the CI/CD process is not a solution, because SCPs are not policy-as-code tools, but policies that you can use to manage permissions in your AWS Organizations. SCPs do not allow you to validate your CloudFormation templates, but only restrict the actions that users and roles can perform in your accounts5.

1: What is AWS CloudFormation Guard? 2: Introducing AWS CloudFormation Guard 2.0 3: AWS Trusted Advisor 4: What Is AWS Config? 5: Service control policies - AWS Organizations

Question 2

Question Type: **MultipleChoice**

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? (Select TWO.)

Options:

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 Cloud Trail 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 Cloud Trail filter based on Session Manager. Send the results to an Amazon Simple Notification Service (Amazon SNS) topic.

Answer:

A, E

Explanation:

The combination of solutions that will meet the requirements are:

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. This is a valid solution because it allows the security team to access the workload AWS account and instances using a local IAM user that does not depend on SAML federation. It also enables logging and monitoring of the break glass user activities using AWS CloudTrail, Amazon CloudWatch Logs, and Amazon EventBridge123.

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. This is a valid solution because it allows the security team to access the workload instances without opening any inbound ports or managing SSH keys or bastion hosts. It also enables logging and notification of the break glass user activities using AWS CloudTrail, Session Manager, and Amazon SNS456.

The other options are incorrect because:

B) Creating a break glass EC2 key pair for the AWS account and providing it to the security team is not a valid solution, because it requires opening inbound ports on the instances and managing SSH keys, which increases the security risk and complexity7.

C) Creating a break glass IAM role for the account and allowing security team members to perform the AssumeRoleWithSAML operation is not a valid solution, because it still depends on SAML federation, which might not work in case of SAML errors8.

D) Creating a local individual break glass IAM user on the operating system level of each workload instance and configuring unrestricted security groups on the instances to grant access to the break glass IAM users is not a valid solution, because it requires opening inbound ports on the instances and managing multiple local users, which increases the security risk and complexity9.

1: Creating an IAM User in Your AWS Account 2: Creating a Trail - AWS CloudTrail 3: Using Amazon EventBridge with AWS CloudTrail 4: Setting up Session Manager - AWS Systems Manager 5: Logging Session Manager sessions - AWS Systems Manager 6: Amazon Simple Notification Service 7: Connecting to your Linux instance using SSH - Amazon Elastic Compute Cloud 8: AssumeRoleWithSAML - AWS Security Token Service 9: IAM Users - AWS Identity and Access Management

Question 3

Question Type: **MultipleChoice**

A Security Engineer is working with a Product team building a web application on AWS. The application uses Amazon S3 to host the static content, Amazon API

Gateway to provide RESTful services; and Amazon DynamoDB as the backend data store. The users already exist in a directory that is exposed through a SAML identity provider.

Which combination of the following actions should the Engineer take to enable users to be authenticated into the web application and call APIs? (Choose three.)

Options:

A- Create a custom authorization service using AWS Lambda.

B- Configure a SAML identity provider in Amazon Cognito to map attributes to the Amazon Cognito user pool attributes.

C- Configure the SAML identity provider to add the Amazon Cognito user pool as a relying party.

D- Configure an Amazon Cognito identity pool to integrate with social login providers.

E- Update DynamoDB to store the user email addresses and passwords.

F- Update API Gateway to use a COGNITO_USER_POOLS authorizer.

Answer:

B, C, F

Explanation:

The combination of the following actions should the Engineer take to enable users to be authenticated into the web application and call APIs are:

B) Configure a SAML identity provider in Amazon Cognito to map attributes to the Amazon Cognito user pool attributes. This is a necessary step to federate the existing users from the SAML identity provider to the Amazon Cognito user pool, which will be used for authentication and authorization1.

C) Configure the SAML identity provider to add the Amazon Cognito user pool as a relying party. This is a necessary step to establish a trust relationship between the SAML identity provider and the Amazon Cognito user pool, which will allow the users to sign in using their existing credentials2.

F) Update API Gateway to use a COGNITO_USER_POOLS authorizer. This is a necessary step to enable API Gateway to use the Amazon Cognito user pool as an authorizer for the RESTful services, which will validate the identity or access tokens that are issued by Amazon Cognito when a user signs in successfully3.

The other options are incorrect because:

A) Creating a custom authorization service using AWS Lambda is not a necessary step, because Amazon Cognito user pools can provide built-in authorization features, such as scopes and groups, that can be used to control access to API resources4.

D) Configuring an Amazon Cognito identity pool to integrate with social login providers is not a necessary step, because the users already exist in a directory that is exposed through a SAML identity provider, and there is no requirement to support social login providers5.

E) Updating DynamoDB to store the user email addresses and passwords is not a necessary step, because the user credentials are already stored in the SAML identity provider, and there is no need to duplicate them in DynamoDB6.

1: Using Tokens with User Pools 2: Adding SAML Identity Providers to a User Pool 3: Control Access to a REST API Using Amazon Cognito User Pools as Authorizer 4: API Authorization with Resource Servers and OAuth 2.0 Scopes 5: Using Identity Pools (Federated Identities) 6: Amazon DynamoDB

Question 4

Question Type: MultipleChoice

A company needs complete encryption of the traffic between external users and an application. The company hosts the application on a fleet of Amazon EC2 instances that run in an Auto Scaling group behind an Application Load Balancer (ALB).

How can a security engineer meet these requirements?

Options:

A- Create a new Amazon-issued certificate in AWS Secrets Manager. Export the certificate from Secrets Manager. Import the certificate into the ALB and the EC2 instances.

B- Create a new Amazon-issued certificate in AWS Certificate Manager (ACM). Associate the certificate with the ALB. Export the certificate from ACM. Install the certificate on the EC2 instances.

C- Import a new third-party certificate into AWS Identity and Access Management (IAM). Export the certificate from IAM. Associate the certificate with the ALB and the EC2 instances.

D- Import a new third-party certificate into AWS Certificate Manager (ACM). Associate the certificate with the ALB. Install the certificate on the EC2 instances.

Answer:

D

Explanation:

The correct answer is D) Import a new third-party certificate into AWS Certificate Manager (ACM). Associate the certificate with the ALB. Install the certificate on the EC2 instances.

This answer is correct because it meets the requirements of complete encryption of the traffic between external users and the application. By importing a third-party certificate into ACM, the security engineer can use it to secure the communication between the ALB and the clients. By installing the same certificate on the EC2 instances, the security engineer can also secure the communication between the ALB and the instances. This way, both the front-end and back-end connections are encrypted with SSL/TLS1.

The other options are incorrect because:

A) Creating a new Amazon-issued certificate in AWS Secrets Manager is not a solution, because AWS Secrets Manager is not a service for issuing certificates, but for storing and managing secrets such as database credentials and API keys2. AWS Secrets Manager does not integrate with ALB or EC2 for certificate deployment.

B) Creating a new Amazon-issued certificate in AWS Certificate Manager (ACM) and exporting it from ACM is not a solution, because ACM does not allow exporting Amazon-issued certificates3. ACM only allows exporting private certificates that are issued by an AWS Private Certificate Authority (CA)4.

C) Importing a new third-party certificate into AWS Identity and Access Management (IAM) is not a solution, because IAM is not a service for managing certificates, but for controlling access to AWS resources5. IAM does not integrate with ALB or EC2 for certificate deployment.

1: How SSL/TLS works 2: What is AWS Secrets Manager? 3: Exporting an ACM Certificate 4: Exporting Private Certificates from ACM 5: What is IAM?

Question 5

Question Type: **MultipleChoice**

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.)

Options:

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 Services to encrypt all the traffic between the client and application servers.

Answer:

B, D

Explanation:

The steps that the Security Engineer should take to check for known vulnerabilities and limit the attack surface are:

B) Review the application security groups to ensure that only the necessary ports are open. This is a good practice to reduce the exposure of the EC2 instances to potential attacks from the Internet. Application security groups are a feature of Azure that allow you to group virtual machines and define network security policies based on those groups1.

D) Use Amazon Inspector to periodically scan the backend instances. This is a service that helps you to identify vulnerabilities and exposures in your EC2 instances and applications. Amazon Inspector can perform automated security assessments based on predefined or custom rules packages2.

Question 6

Question Type: **MultipleChoice**

A company is using AWS Organizations to implement a multi-account strategy. The company does not have on-premises infrastructure. All workloads run on AWS. The company currently has eight member accounts. The company anticipates that it will have no more than 20 AWS accounts total at any time.

The company issues a new security policy that contains the following requirements:

* No AWS account should use a VPC within the AWS account for workloads.

* The company should use a centrally managed VPC that all AWS accounts can access to launch workloads in subnets.

* No AWS account should be able to modify another AWS account's application resources within the centrally managed VPC.

* The centrally managed VPC should reside in an existing AWS account that is named Account-A within an organization.

The company uses an AWS CloudFormation template to create a VPC that contains multiple subnets in Account-A. This template exports the subnet IDs through the CloudFormation Outputs section.

Which solution will complete the security setup to meet these requirements?

Options:

A- Use a CloudFormation template in the member accounts to launch workloads. Configure the template to use the Fn::ImportValue function to obtain the subnet ID values.

B- Use a transit gateway in the VPC within Account-A. Configure the member accounts to use the transit gateway to access the subnets in Account-A to launch workloads.

C- Use AWS Resource Access Manager (AWS RAM) to share Account-A's VPC subnets with the remaining member accounts. Configure the member accounts to use the shared subnets to launch workloads.

D- Create a peering connection between Account-A and the remaining member accounts. Configure the member accounts to use the subnets in Account-A through the VPC peering connection to launch workloads.

Answer:

C

Explanation:

The correct answer is C. Use AWS Resource Access Manager (AWS RAM) to share Account-A's VPC subnets with the remaining member accounts. Configure the member accounts to use the shared subnets to launch workloads.

This answer is correct because AWS RAM is a service that helps you securely share your AWS resources across AWS accounts, within your organization or organizational units (OUs), and with IAM roles and users for supported resource types1. One of the supported resource types is VPC subnets2, which means you can share the subnets in Account-A's VPC with the other member accounts using AWS RAM. This way, you can meet the requirements of using a centrally managed VPC, avoiding duplicate VPCs in each account, and launching workloads in shared subnets. You can also control the access to the shared subnets by using IAM policies and resource-based policies3, which can prevent one account from modifying another account's resources.

The other options are incorrect because:

A) Using a CloudFormation template in the member accounts to launch workloads and using the Fn::ImportValue function to obtain the subnet ID values is not a solution, because Fn::ImportValue can only import values that have been exported by another stack within the same region4. This means that you cannot use Fn::ImportValue to reference the subnet IDs that are exported by Account-A's CloudFormation template, unless all the member accounts are in the same region as Account-A. This option also does not avoid creating duplicate VPCs in each account, which is one of the requirements.

B) Using a transit gateway in the VPC within Account-A and configuring the member accounts to use the transit gateway to access the subnets in Account-A to launch workloads is not a solution, because a transit gateway does not allow you to launch workloads in another account's subnets. A transit gateway is a network transit hub that enables you to route traffic between your VPCs and on-premises networks5, but it does not enable you to share subnets across accounts.

D) Creating a peering connection between Account-A and the remaining member accounts and configuring the member accounts to use the subnets in Account-A through the VPC peering connection to launch workloads is not a solution, because a VPC peering connection does not allow you to launch workloads in another account's subnets. A VPC peering connection is a networking connection between two VPCs that enables you to route traffic between them privately6, but it does not enable you to share subnets across accounts.

1: What is AWS Resource Access Manager? 2: Shareable AWS resources 3: Managing permissions for shared resources 4: Fn::ImportValue 5: What is a transit gateway? 6: What is VPC peering?

Question 7

Question Type: **MultipleChoice**

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?

Options:

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:I}}.

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:

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

This answer is correct because AWS Secrets Manager is a service that helps you protect secrets that are needed to access your applications, services, and IT resources. You can store and manage secrets such as database credentials, API keys, and other sensitive data in Secrets Manager. You can also use Secrets Manager to rotate, manage, and retrieve your secrets throughout their lifecycle1. Secrets Manager integrates with AWS CloudFormation, which allows you to reference secrets from your templates using the {{resolve:secretsmanager:...}} syntax2. This way, you can avoid exposing your secrets in plaintext and still use them in your resources.

The other options are incorrect because:

A) Storing the API key value as a SecureString parameter in AWS Systems Manager Parameter Store is not a solution, because AWS CloudFormation does not support references to SecureString parameters. This means that you cannot use the {{resolve:ssm:...}} syntax to retrieve encrypted parameter values from Parameter Store3. You would have to use a custom resource or a Lambda function to decrypt the parameter value, which adds complexity and overhead to your template.

C) Storing the API key value in Amazon DynamoDB is not a solution, because AWS CloudFormation does not support references to DynamoDB items. This means that you cannot use the {{resolve:dynamodb:...}} syntax to retrieve item values from DynamoDB tables4. You would have to use a custom resource or a Lambda function to query the DynamoDB table, which adds complexity and overhead to your template.

D) Storing the API key value in a new Amazon S3 bucket is not a solution, because AWS CloudFormation does not support references to S3 objects. This means that you cannot use the {{resolve:s3:...}} syntax to retrieve object values from S3 buckets5. You would have to use a custom resource or a Lambda function to download the object from S3, which adds complexity and overhead to your template.

1: What is AWS Secrets Manager? 2: Referencing AWS Secrets Manager secrets from Parameter Store parameters 3: Using dynamic references to specify template values 4: Amazon DynamoDB 5: Amazon Simple Storage Service (S3)

Question 8

Question Type: **MultipleChoice**

A company plans to use AWS Key Management Service (AWS KMS) to implement an encryption strategy to protect data at rest. The company requires client-side encryption for company projects. The company is currently conducting multiple projects to test the company's use of AWS KMS. These tests have led to a sudden increase in the company's AWS resource consumption. The test projects include applications that issue multiple requests each second to KMS endpoints for encryption activities.

The company needs to develop a solution that does not throttle the company's ability to use AWS KMS. The solution must improve key usage for client-side

encryption and must be cost optimized.

Which solution will meet these requirements?

Options:

A- Use keyrings with the AWS Encryption SDK. Use each keyring individually or combine keyrings into a multi-keyring. Decrypt the data by using a keyring that has the primary key in the multi-keyring.

B- Use data key caching. Use the local cache that the AWS Encryption SDK provides with a caching cryptographic materials manager.

C- Use KMS key rotation. Use a local cache in the AWS Encryption SDK with a caching cryptographic materials manager.

D- Use keyrings with the AWS Encryption SDK. Use each keyring individually or combine keyrings into a multi-keyring. Use any of the wrapping keys in the multi-keyring to decrypt the data.

Answer:

B

Explanation:

The correct answer is B. Use data key caching. Use the local cache that the AWS Encryption SDK provides with a caching cryptographic materials manager.

This answer is correct because data key caching can improve performance, reduce cost, and help the company stay within the service limits of AWS KMS. Data key caching stores data keys and related cryptographic material in a cache, and reuses them for encryption and decryption operations. This reduces the number of requests to AWS KMS endpoints and avoids throttling. The AWS Encryption SDK provides a local cache and a caching cryptographic materials manager (caching CMM) that interacts with the cache and enforces security thresholds that the company can set1.

The other options are incorrect because:

A) Using keyrings with the AWS Encryption SDK does not address the problem of throttling or cost optimization. Keyrings are used to generate, encrypt, and decrypt data keys, but they do not cache or reuse them. Using each keyring individually or combining them into a multi-keyring does not reduce the number of requests to AWS KMS endpoints2.

C) Using KMS key rotation does not address the problem of throttling or cost optimization. Key rotation is a security practice that creates new cryptographic material for a KMS key every year, but it does not affect the data that the KMS key protects. Key rotation does not reduce the number of requests to AWS KMS endpoints, and it might incur additional costs for storing multiple versions of key material3.

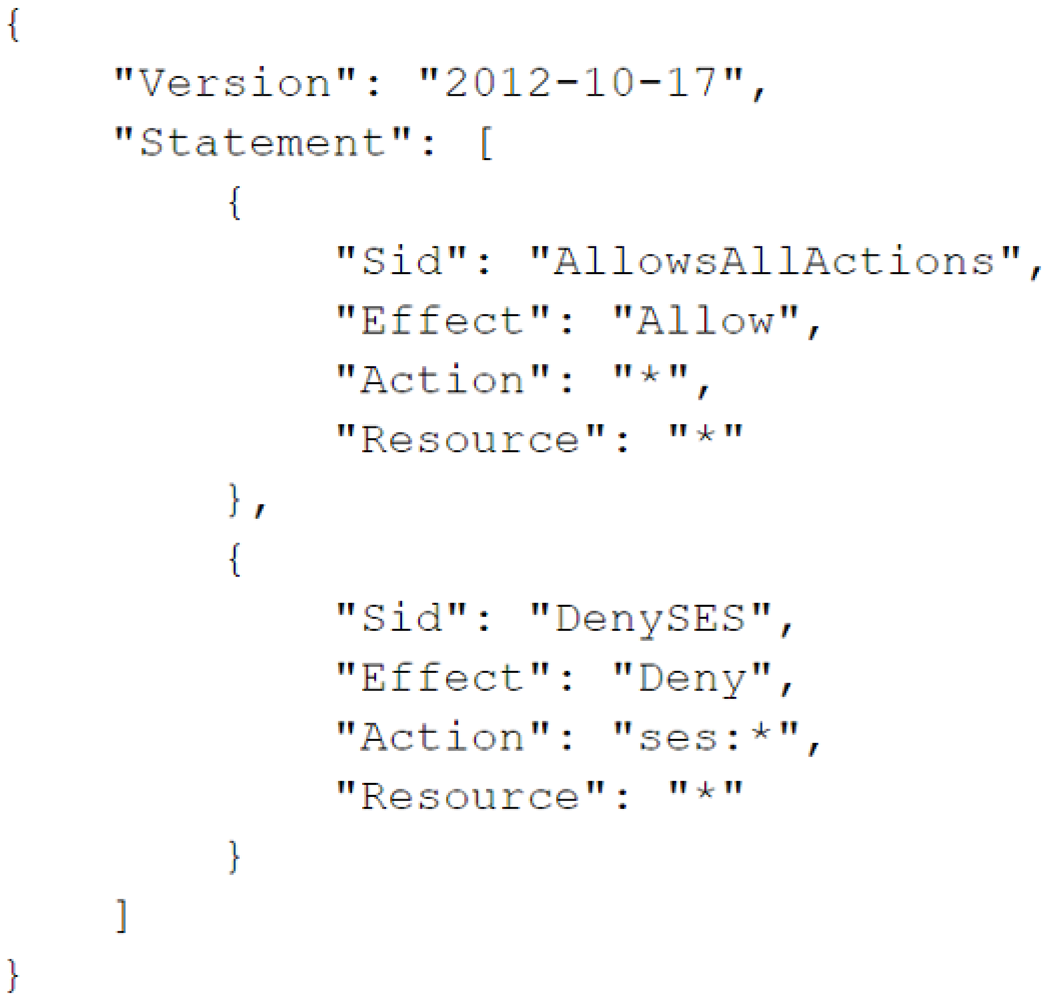
D) Using keyrings with the AWS Encryption SDK does not address the problem of throttling or cost optimization, as explained in option A. Moreover, using any of the wrapping keys in the multi-keyring to decrypt the data is not a valid option, because only one of the wrapping keys can decrypt a given data key. The wrapping key that encrypts a data key is stored in the encrypted data key structure, and only that wrapping key can decrypt it4.

1: Data key caching - AWS Encryption SDK 2: Using keyrings - AWS Encryption SDK 3: Rotating AWS KMS keys - AWS Key Management Service 4: How keyrings work - AWS Encryption SDK

Question 9

Question Type: MultipleChoice

A company has an organization with SCPs in AWS Organizations. The root SCP for the organization is as follows:



```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowsAllActions",
            "Effect": "Allow",
            "Action": "*",
            "Resource": "*"
        },
        {
            "Sid": "DenySES",
            "Effect": "Deny",
            "Action": "ses:*",
            "Resource": "*"
        }
    ]
}
```

The company's developers are members of a group that has an IAM policy that allows access to Amazon Simple Email Service (Amazon SES) by allowing ses:* actions. The account is a child to an OU that has an SCP that allows Amazon SES. The developers are receiving a not-authorized error when they try to access Amazon SES through the AWS Management Console.

Which change must a security engineer implement so that the developers can access Amazon SES?

Options:

A- Add a resource policy that allows each member of the group to access Amazon SES.

B- Add a resource policy that allows 'Principal': {'AWS': 'arn:aws:iam::account-number:group/Dev'}.

C- Remove the AWS Control Tower control (guardrail) that restricts access to Amazon SES.

D- Remove Amazon SES from the root SCP.

Answer:

D

Explanation:

The correct answer is D. Remove Amazon SES from the root SCP.

This answer is correct because the root SCP is the most restrictive policy that applies to all accounts in the organization. The root SCP explicitly denies access to Amazon SES by using the NotAction element, which means that any action that is not listed in the element is denied. Therefore, removing Amazon SES from the root SCP will allow the developers to access it, as long as there are no other SCPs or IAM policies that deny it.

The other options are incorrect because:

A) Adding a resource policy that allows each member of the group to access Amazon SES is not a solution, because resource policies are not supported by Amazon SES1. Resource policies are policies that are attached to AWS resources, such as S3 buckets or SNS topics, to control access to those resources2. Amazon SES does not have any resources that can have resource policies attached to them.

B) Adding a resource policy that allows "Principal": {"AWS": "arn:aws:iam::account-number:group/Dev"} is not a solution, because resource policies do not support IAM groups as principals3. Principals are entities that can perform actions on AWS resources, such as IAM users, roles, or AWS accounts4. IAM groups are not principals, but collections of IAM users that share the same permissions5.

C) Removing the AWS Control Tower control (guardrail) that restricts access to Amazon SES is not a solution, because AWS Control Tower does not have any guardrails that restrict access to Amazon SES6. Guardrails are high-level rules that govern the overall behavior of an organization's accounts7. AWS Control Tower provides a set of predefined guardrails that cover security, compliance, and operations domains8.

1: Amazon Simple Email Service endpoints and quotas 2: Resource-based policies and IAM policies 3: Specifying a principal in a policy 4: Policy elements: Principal 5: IAM groups 6: AWS Control Tower guardrails reference 7: AWS Control Tower concepts 8: AWS Control Tower guardrails

Question 10

Question Type: **MultipleChoice**

A company is using Amazon Elastic Container Service (Amazon ECS) to run its container-based application on AWS. The company needs to ensure that the container images contain no severe vulnerabilities. The company also must ensure that only specific IAM roles and specific AWS accounts can access the container images.

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

Options:

A- Pull images from the public container registry. Publish the images to Amazon Elastic Container Registry (Amazon ECR) repositories with scan on push configured in a centralized AWS account. Use a CI/CD pipeline to deploy the images to different AWS accounts. Use identity-based policies to restrict access to which IAM principals can access the images.

B- Pull images from the public container registry. Publish the images to a private container registry that is hosted on Amazon EC2 instances in a centralized AWS account. Deploy host-based container scanning tools to EC2 instances that run Amazon ECS. Restrict access to the container images by using basic authentication over HTTPS.

C- Pull images from the public container registry. Publish the images to Amazon Elastic Container Registry (Amazon ECR) repositories with scan on push configured in a centralized AWS account. Use a CI/CD pipeline to deploy the images to different AWS accounts. Use repository policies and identity-based policies to restrict access to which IAM principals and accounts can access the images.

D- Pull images from the public container registry. Publish the images to AWS CodeArtifact repositories in a centralized AWS account. Use a CI/CD pipeline to deploy the images to different AWS accounts. Use repository policies and identity-based policies to restrict access to which IAM principals and accounts can access the images.

Answer:

C

Explanation:

The correct answer is C. Pull images from the public container registry. Publish the images to Amazon Elastic Container Registry (Amazon ECR) repositories with scan on push configured in a centralized AWS account. Use a CI/CD pipeline to deploy the images to different AWS accounts. Use repository policies and identity-based policies to restrict access to which IAM principals and accounts can access the images.

This solution meets the requirements because:

Amazon ECR is a fully managed container registry service that supports Docker and OCI images and artifacts1. It integrates with Amazon ECS and other AWS services to simplify the development and deployment of container-based applications.

Amazon ECR provides image scanning on push, which uses the Common Vulnerabilities and Exposures (CVEs) database from the open-source Clair project to detect software vulnerabilities in container images2. The scan results are available in the AWS Management Console, AWS CLI, or AWS SDKs2.

Amazon ECR supports cross-account access to repositories, which allows sharing images across multiple AWS accounts3. This can be achieved by using repository policies, which are resource-based policies that specify which IAM principals and accounts can access the repositories and what actions they can perform4. Additionally, identity-based policies can be used to control which IAM roles in each account can access the repositories5.

The other options are incorrect because:

A) This option does not use repository policies to restrict cross-account access to the images, which is a requirement. Identity-based policies alone are not sufficient to control access to Amazon ECR repositories5.

B) This option does not use Amazon ECR, which is a fully managed service that provides image scanning and cross-account access features. Hosting a private container registry on EC2 instances would require more management overhead and additional security measures.

D) This option uses AWS CodeArtifact, which is a fully managed artifact repository service that supports Maven, npm, NuGet, PyPI, and generic package formats6. However, AWS CodeArtifact does not support Docker or OCI container images, which are required for Amazon ECS applications.

Question 11

Question Type: **MultipleChoice**

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.)

Options:

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 possible steps to troubleshoot this issue are:

A) Ensure that the EC2 instance profile that is attached to the EC2 instances has permissions to create log streams and write logs. This is a necessary step because the CloudWatch agent uses the credentials from the instance profile to communicate with CloudWatch1.

C) Check the CloudWatch agent configuration file on each EC2 instance to make sure that the CloudWatch agent is collecting the proper log files. This is a necessary step because the CloudWatch agent needs to know which log files to monitor and send to CloudWatch2.

D) Check the VPC endpoint policies of both VPC endpoints to ensure that the EC2 instances have permissions to use them. This is a necessary step because the VPC endpoint policies control which principals can access the AWS services through the endpoints3.

The other options are incorrect because:

B) Creating a metric filter on the logs is not a troubleshooting step, but a way to extract metric data from the logs. Metric filters do not affect the visibility of the logs in the AWS Management Console.

E) Creating a NAT gateway in the subnet is not a solution, because the EC2 instances do not need internet access to communicate with CloudWatch through the VPC endpoints. A NAT gateway would also incur additional costs.

F) Ensuring that the security groups allow all the EC2 instances to communicate with each other is not a necessary step, because the CloudWatch agent does not require log aggregation before sending. Each EC2 instance can send its own logs independently to CloudWatch.

1: IAM Roles for Amazon EC2 2: CloudWatch Agent Configuration File: Logs Section 3: Using Amazon VPC Endpoints : Metric Filters : NAT Gateways : CloudWatch Agent Reference: Log Aggregation

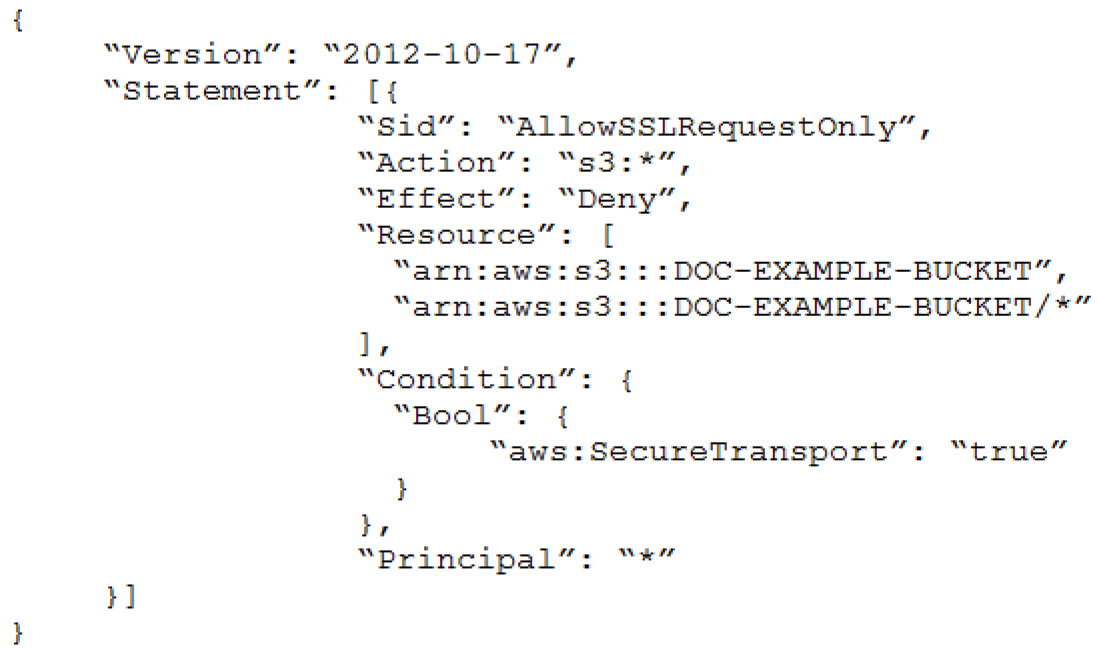
Question 12

Question Type: **MultipleChoice**

A company has a guideline that mandates the encryption of all Amazon S3 bucket data in transit. A security engineer must implement an S3 bucket policy that denies any S3 operations if data is not encrypted.

Which S3 bucket policy will meet this requirement?

A.

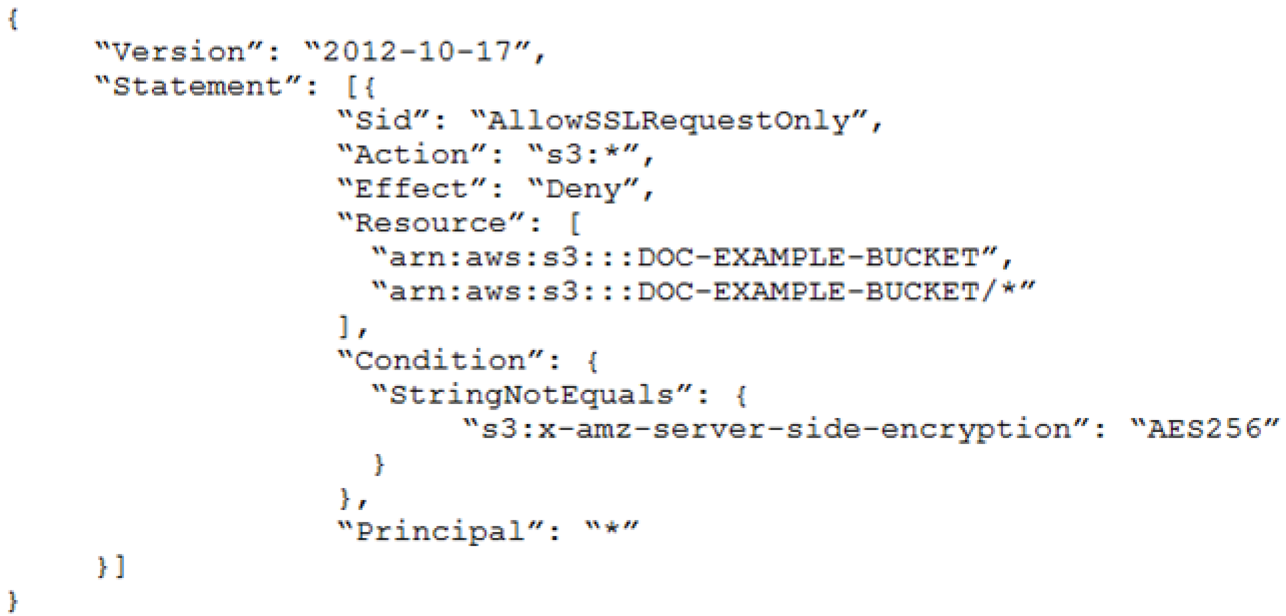


```
{
     “Version”: “2012-10-17”,
     “Statement”: [{
                  “Sid”: “AllowSSLRequestOnly”,
                  “Action”: “s3:*”,
                  “Effect”: “Deny”,
                  “Resource”: [
                    “arn:aws:s3:::DOC-EXAMPLE-BUCKET”,
                    “arn:aws:s3:::DOC-EXAMPLE-BUCKET/*”
                  ],
                  “Condition”: {
                    “Bool”: {
                         “aws:SecureTransport”: “true”
                    }
                  },
                  “Principal”: “*”
     }]
}
```

B.

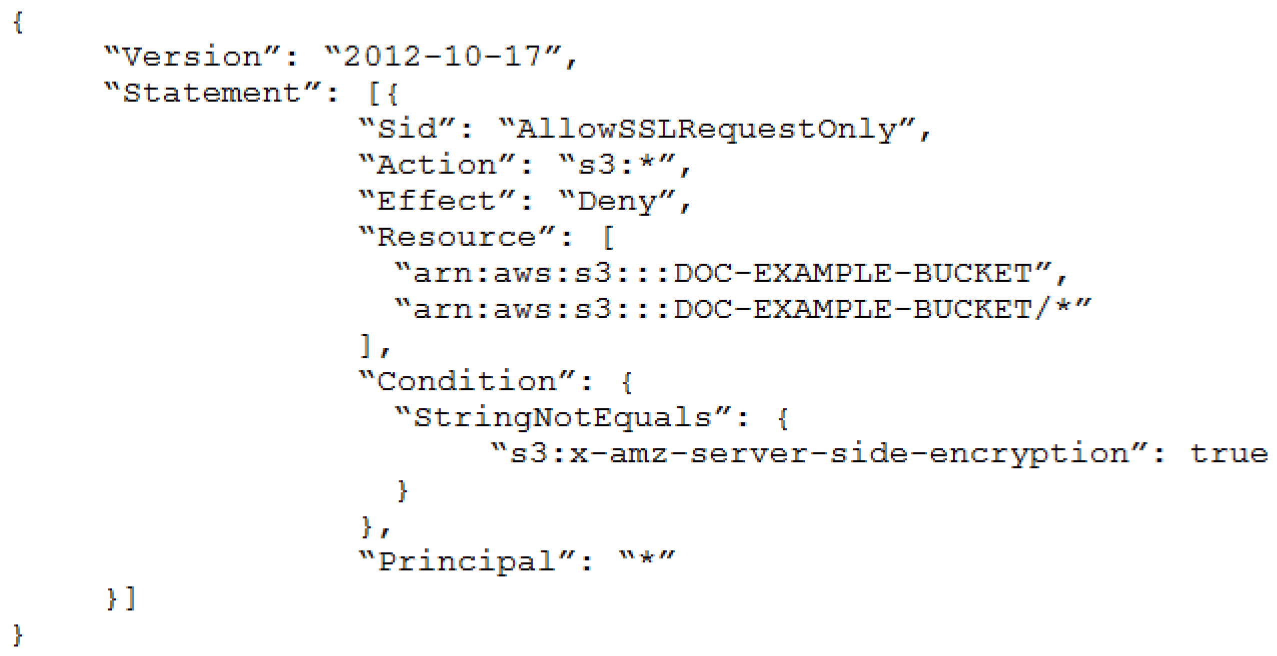
```
{
     “Version”: “2012-10-17”,
     “Statement”: [{
                  “Sid”: “AllowSSLRequestOnly”,
                  “Action”: “s3:*”,
                  “Effect”: “Deny”,
                  “Resource”: [
                    “arn:aws:s3:::DOC-EXAMPLE-BUCKET”,
                    “arn:aws:s3:::DOC-EXAMPLE-BUCKET/*”
                  ],
                  “Condition”: {
                    “Bool”: {
                         “aws:SecureTransport”: “false”
                    }
                  },
                  “Principal”: “*”
     }]
}
```

C.



```
{
     “Version”: “2012-10-17”,
     “Statement”: [{
                  “Sid”: “AllowSSLRequestOnly”,
                  “Action”: “s3:*”,
                  “Effect”: “Deny”,
                  “Resource”: [
                    “arn:aws:s3:::DOC-EXAMPLE-BUCKET”,
                    “arn:aws:s3:::DOC-EXAMPLE-BUCKET/*”
                  ],
                  “Condition”: {
                    “StringNotEquals”: {
                         “s3:x-amz-server-side-encryption”: “AES256”
                    }
                  },
                  “Principal”: “*”
     }]
}
```

D.



```
{
     “Version”: “2012-10-17”,
     “Statement”: [{
                   “Sid”: “AllowSSLRequestOnly”,
                   “Action”: “s3:*”,
                   “Effect”: “Deny”,
                   “Resource”: [
                     “arn:aws:s3:::DOC-EXAMPLE-BUCKET”,
                     “arn:aws:s3:::DOC-EXAMPLE-BUCKET/*”
                   ],
                   “Condition”: {
                     “StringNotEquals”: {
                          “s3:x-amz-server-side-encryption”: true
                     }
                   },
                   “Principal”: “*”
     }]
}
```

Options:

A- Option A

B- Option B

C- Option C

D- Option D

Answer:

B

Explanation:

https://aws.amazon.com/blogs/security/how-to-use-bucket-policies-and-apply-defense-in-depth-to-help-secure-your-amazon-s3-data/

20%
DISCOUNT