

Solutions Architect Professional Questions

Question 1

A company is planning to implement a hybrid DNS system, utilizing Amazon Route 53 private hosted zone for the domain `cloud.example.com` to manage DNS for resources within its VPCs. The system needs to fulfill two key requirements: 1) On-premises systems must be able to resolve and connect to `cloud.example.com`, and 2) All VPCs should resolve `cloud.example.com`. The company has an existing AWS Direct Connect setup linking its on-premises network to an AWS Transit Gateway. Which architecture should be adopted to achieve the highest performance in meeting these DNS resolution requirements?

- A. Link the private hosted zone with all VPCs. Set up a Route 53 inbound resolver in a shared services VPC. Connect all VPCs to the transit gateway, and configure the on-premises DNS server with forwarding rules for `cloud.example.com` directed to the inbound resolver.
- B. Associate the private hosted zone with all VPCs and deploy an Amazon EC2-based conditional forwarder in the shared services VPC. Attach all VPCs to the transit gateway and add forwarding rules on the on-premises DNS server for `cloud.example.com` pointing to this conditional forwarder.
- C. Connect the private hosted zone to the shared services VPC and implement a Route 53 outbound resolver there. Attach all VPCs to the transit gateway and establish forwarding rules on the on-premises DNS server for `cloud.example.com` to target the outbound resolver.
- D. Associate the private hosted zone with the shared services VPC and create a Route 53 inbound resolver within this VPC. Connect only the shared services VPC to the transit gateway and set up forwarding rules in the on-premises DNS server for `cloud.example.com` to point to the inbound resolver.

Correct Answer: **A**

Explanation of Why A is Correct and the Others Are Not:

- **Why A is Correct:**
 - Option A provides a comprehensive solution that integrates the private hosted zone with all VPCs and uses a Route 53 inbound resolver. The inbound resolver is ideal for routing DNS queries from on-premises networks to AWS. By connecting all VPCs to the transit gateway and setting up forwarding rules from the on-premises DNS server to the inbound resolver, the solution ensures that both on-premises systems and all VPCs

can effectively resolve `cloud.example.com`. This approach is efficient and aligns with the requirements for high performance.

- **Why B is Incorrect:**
 - Deploying an EC2 conditional forwarder introduces unnecessary complexity and potential performance bottlenecks compared to using AWS managed services like Route 53 resolvers. While this setup might work, it is less efficient and scalable than Option A.
- **Why C is Incorrect:**
 - An outbound resolver is typically used for routing DNS queries from AWS to external networks, which is the opposite of what's required in this scenario. The company needs to resolve DNS queries from on-premises systems to AWS (and within AWS), making an inbound resolver a more suitable choice.
- **Why D is Incorrect:**
 - This option limits the Route 53 resolver setup to only the shared services VPC, which might not provide the necessary DNS resolution capabilities for all VPCs as required. It lacks the comprehensive VPC coverage provided in Option A.

Question 2

A company provides weather data through a REST-based API, using Amazon API Gateway integrated with AWS Lambda functions for each operation. They use Amazon Route 53 for DNS, with a record for `weather.example.com`, and store API data in Amazon DynamoDB tables. The company seeks a solution enabling their API to switch to a different AWS Region in case of failures.

Which of these solutions would fulfill this requirement?

- A. Create new Lambda functions in another Region, update API Gateway to use an edge-optimized endpoint with Lambda functions from both Regions, and make DynamoDB tables global.
- B. Set up a new API Gateway and Lambda functions in a different Region, change the Route 53 record to a multi-value answer, include both APIs, enable health monitoring, and make DynamoDB tables global.
- C. Implement a new API Gateway and Lambda functions in another Region, modify the Route 53 record to a failover record, enable health monitoring, and make DynamoDB tables global.
- D. Launch a new API Gateway in another Region, modify Lambda functions to be global, change the Route 53 record to a multi-value answer, include both APIs, enable health monitoring, and make DynamoDB tables global.

The correct answer is **C**.

Explanation for Beginners:

- **The Goal:** The company wants their weather API to keep working even if one AWS Region has issues. This is called "failover" – like having a backup plan when the primary plan fails.
- **Components Involved:**
 - **Amazon API Gateway:** This is like a doorkeeper for the API, managing access and requests.
 - **AWS Lambda:** A service where the company's code runs without needing to manage servers.
 - **Amazon DynamoDB:** A database service where the API's data is stored.
 - **Amazon Route 53:** This helps direct users to the API through `weather.example.com`.
- **Solution C (The Correct Answer):**

To understand why Option C is the correct answer, let's first break down the question and then explain how each element of this solution addresses the requirements:

The Question's Key Elements:

1. **Primary Need:** The company's API must continue to function even if the primary AWS Region becomes unavailable. This is known as creating a "failover" capability.
2. **Current Setup:** They are using Amazon API Gateway with AWS Lambda for API operations, Amazon Route 53 for DNS with a record for `weather.example.com`, and Amazon DynamoDB for data storage.
3. **Requirement:** Implement a solution that allows the API to automatically switch ("fail over") to a different AWS Region if the primary Region fails.

Why Option C is the Best Solution:

- **Deploying a New API Gateway and Lambda Functions in Another Region:** This step creates a complete, operational backup of the API in a different geographical area. If the primary Region encounters issues, this backup Region can take over, ensuring continuous service availability.
- **Changing the Route 53 DNS Record to a Failover Record:** This is a crucial part of the solution. A failover record in Route 53 is designed to redirect traffic from the primary endpoint (in this case, the API in the primary Region) to a secondary endpoint (the API in the backup Region) automatically in the event of a failure. This redirection happens seamlessly, ensuring users can still access the API without interruption.
- **Enabling Target Health Monitoring:** This feature continuously checks the health or operational status of the primary API endpoint. If it detects a problem (like an outage

in the primary Region), it triggers the failover mechanism to switch to the backup Region.

- **Converting the DynamoDB Tables to Global Tables:** DynamoDB Global Tables are replicated across multiple AWS Regions. By converting the existing tables to global tables, the company ensures that the backup API in the alternate Region has immediate, up-to-date access to the same data as the primary API. This synchronization is vital for maintaining consistency in the API's responses, regardless of which Region is handling the requests.
- **Why Other Answers are Wrong:**
 - **A and D (Multi-Value Answer):** These options suggest using a multi-value answer in Route 53. This method is more about load balancing (sharing traffic) rather than failover (switching to a backup in case of failure).
 - **B (Multivalue Answer with Both APIs):** While this also sets up a backup, it doesn't automatically switch all traffic to the backup in case of a failure. It's more about distributing users between two options rather than having a clear primary and backup.
 - **D (Global Lambda Functions):** There's no concept of 'global Lambda functions' in AWS as Lambda functions are region-specific. This makes D an impractical solution.

Question 3

A company uses AWS Organizations to manage multiple accounts under a single Organizational Unit (OU) called "Production". To manage access to certain services, they use deny list Service Control Policies (SCPs) at the organization's root level. Recently, they added a new business unit's AWS account to their organization. However, the administrators of this new unit are unable to modify AWS Config rules to align with the company's standards due to these SCPs. What method would allow these administrators to make the necessary changes while maintaining current policies and minimizing long-term maintenance overhead?

A. Remove the organization's root SCPs that limit access to AWS Config. Create AWS Service Catalog products for the company's standard AWS Config rules and deploy them throughout the organization, including the new account.

B. Create a temporary OU named Onboarding for the new account. Apply an SCP to the Onboarding OU to allow AWS Config actions. Move the new account to the Production OU when adjustments to AWS Config are complete.

C. Convert the organization's root SCPs from deny list SCPs to allow list SCPs to allow the required services only. Temporarily apply an SCP to the organization's root that allows AWS Config actions for principals only in the new account.

D. Create a temporary OU named Onboarding for the new account. Apply an SCP to the Onboarding OU to allow AWS Config actions. Move the organization's root SCP to the

Production OU. Move the new account to the Production OU when adjustments to AWS Config are complete.

Correct Answer: **D**

Explanation of the Correct Answer and Analysis of Other Options:

- **Why D is Correct:**
 - **Creating a Temporary 'Onboarding' OU:** This step involves setting up a new Organizational Unit specifically for integrating new accounts. This provides a controlled environment for configuration adjustments without affecting other accounts.
 - **Applying an SCP to the Onboarding OU:** The SCP (Service Control Policy) applied here allows AWS Config actions, which are necessary for the new account to update its AWS Config rules.
 - **Moving the Organization's Root SCP to the Production OU:** This action ensures that the existing accounts in the Production OU continue to operate under the established SCPs, maintaining security and compliance.
 - **Transition to Production OU Post-Adjustment:** Once the new account's AWS Config is aligned with the company's standards, it is moved to the Production OU. This ensures that it too falls under the same compliance and security policies as the rest of the organization, maintaining consistency.
- **Why Other Options are Incorrect:**
 - **A:** Removing SCPs that limit access to AWS Config at the organization's root could weaken security and compliance controls across all accounts. Although creating AWS Service Catalog products is a good practice for standardization, it doesn't directly address the issue of enabling the new account to update AWS Config rules without compromising security.
 - **B:** While creating a temporary 'Onboarding' OU is a good approach, simply moving the new account to the Production OU after adjustments does not address the issue of maintaining the deny list SCPs for other accounts during the onboarding process.
 - **C:** Converting deny list SCPs to allow list SCPs at the organization's root level is a significant change that could disrupt existing access controls and policies for all accounts in the organization. This approach would also introduce considerable long-term maintenance and potential security risks.

Question 4

A company is moving its two-tier web application from an on-premises setup to AWS to handle an expected increase in users. The application consists of a stateful application server and a PostgreSQL database server. In AWS, they plan to use Amazon Aurora PostgreSQL, Amazon EC2 Auto Scaling, and Elastic Load Balancing (ELB). Which configuration will best ensure a smooth

scaling of both the application and database layers while maintaining a consistent user experience?

- A. Utilize Aurora Auto Scaling for the Aurora Read Replicas. Employ a Network Load Balancer (NLB) with the algorithm focusing on the least outstanding requests, and activate sticky sessions.
- B. Implement Aurora Auto Scaling for the Aurora primary instances (writers). Set up an Application Load Balancer (ALB) using the round-robin routing algorithm and enable sticky sessions.
- C. Implement Aurora Auto Scaling for the Aurora Read Replicas. Configure an Application Load Balancer (ALB) using the round-robin routing algorithm and enable sticky sessions.
- D. Activate Aurora Auto Scaling for the Aurora primary instances (writers). Use a Network Load Balancer (NLB) with the least outstanding requests routing algorithm and enable sticky sessions.

Correct Answer: **C**

Explanation of Why C is Correct:

- **Aurora Auto Scaling for Aurora Replicas (C):**
 - **Aurora Replicas** are used for read operations and can help distribute the load as the user base grows. Auto-scaling these replicas ensures the database scales according to demand.
 - **Not for Writers (B & D):** Auto-scaling writers (primary instances) is not typically done because it can lead to complexities in managing write consistency and data integrity.
- **Application Load Balancer (ALB):**
 - **Round Robin Routing (C & B):** This evenly distributes incoming application traffic across multiple targets (like EC2 instances), which is suitable for stateful applications.
 - **ALB over NLB (A & D):** An Application Load Balancer is more apt for HTTP/HTTPS traffic typical of web applications. Network Load Balancers are more suited for TCP traffic where ultra-low latency is required.
- **Sticky Sessions (C):**
 - Essential for stateful applications to ensure a user's session is directed to the same server each time, maintaining a consistent experience.

Question 5

A company needs to migrate an on-premises service, which collects metadata from various applications, to AWS. This service is accessed by consumer devices, including older models that do not support certain HTTP headers and show errors when these headers are included in responses. The company currently uses a load balancer to identify these older devices via their

User-Agent headers and remove the unsupported headers from responses. As they transition to AWS and adopt serverless architecture with AWS Lambda functions, they want to ensure continued support for these older devices. What AWS solution would enable them to achieve this while handling the header issue?

- A. Set up an Amazon CloudFront distribution for the metadata service. Establish an Application Load Balancer (ALB) and configure CloudFront to route requests to the ALB. Set the ALB to trigger the appropriate Lambda function per request type. Implement a CloudFront function to strip out the problematic headers, using the User-Agent header for identification.
- B. Implement an Amazon API Gateway REST API for the metadata service. Adjust API Gateway to call the relevant Lambda function for different requests. Modify the default gateway responses to exclude the problematic headers, utilizing the User-Agent header for device identification.
- C. Establish an Amazon API Gateway HTTP API for the metadata service. Configure API Gateway to connect to the right Lambda function for each request. Utilize a response mapping template to eliminate the problematic headers based on the User-Agent. Apply this response data mapping to the HTTP API.
- D. Create an Amazon CloudFront distribution for the metadata service. Set up an Application Load Balancer (ALB), and configure CloudFront to direct requests to the ALB. Program the ALB to initiate the corresponding Lambda function for each request type. Use a Lambda@Edge function to remove the problematic headers in responses, guided by the User-Agent header values.

Correct Answer: **A**

Explanation of Why A is Correct and Others are Not:

- **Why A is Correct:**
 - **Use of CloudFront and ALB:** CloudFront efficiently manages web traffic and can work seamlessly with an ALB, which in turn can correctly route requests to the corresponding Lambda functions.
 - **CloudFront Function for Header Manipulation:** A CloudFront function is lightweight and ideal for simple tasks like modifying HTTP headers. It can inspect the User-Agent header and remove unsupported headers for older devices. This mirrors the functionality the company used in their on-premises setup.
- **Why B is Incorrect:**
 - API Gateway's REST API does allow for header manipulation, but modifying default gateway responses might not offer the same level of control and specificity as a dedicated function in CloudFront, especially for varying types of devices and headers.
- **Why C is Incorrect:**

- While API Gateway HTTP API can handle response mapping, it's generally more complex and less suited for simple header modifications compared to the straightforward approach offered by CloudFront functions.
- **Why D is Incorrect:**
 - Lambda@Edge, while powerful, is more complex and resource-intensive compared to CloudFront functions. For simple header manipulations based on User-Agent, a CloudFront function (as in option A) is more efficient and cost-effective.

Question 6

A retail company needs to share a set of data files located in an Amazon S3 bucket (owned by them in Account A) with a business partner. The partner company wishes to allow one of their IAM users, named User_DataProcessor, to retrieve the files from their own AWS account (Account B). What steps should both companies undertake to ensure that User_DataProcessor has the necessary access to the S3 bucket in Account A? Select two steps that together would enable this access.

- A. Enable CORS on the S3 bucket in Account A.
- B. Set the S3 bucket policy in Account A to allow actions 's3:GetObject' and 's3:ListBucket' for any requester.
- C. Update the S3 bucket policy in Account A to permit 's3:GetObject' and 's3:ListBucket' actions specifically for the IAM user 'User_DataProcessor' in Account B.
- D. Configure the IAM user permissions in Account B to allow 'User_DataProcessor' to perform 's3:GetObject' and 's3:ListBucket' actions on the S3 bucket in Account A.
- E. Assign permissions to 'User_DataProcessor' in Account B to invoke 's3:GetObject' and 's3:ListBucket' actions only for the principal of the IAM user itself in Account B.

Correct Answer: **C and D**

Explanation of Why C and D are Correct and Others are Not:

- **Why C is Correct:**
 - It correctly specifies the S3 bucket policy in Account A, which owns the bucket. This policy grants the specific IAM user in Account B (User_DataProcessor) the permissions needed to access the S3 bucket. It's essential that the policy is set in the account that owns the bucket and that it explicitly refers to the user in the other account.
- **Why D is Correct:**
 - This step is about setting permissions for the IAM user within Account B. It ensures that the User_DataProcessor has the necessary rights within their own account to

access resources in another account (Account A in this case). This is needed alongside the bucket policy in Account A for the access to be complete.

- **Why A is Incorrect:**
 - CORS (Cross-Origin Resource Sharing) is not relevant in this scenario, as it pertains to web browsers accessing resources from a different domain. It does not apply to IAM users accessing S3 buckets across AWS accounts.
- **Why B is Incorrect:**
 - This bucket policy would grant access to all requesters, not just the specific IAM user from the business partner's account. This is not secure as it could potentially allow access to any user, not just the intended User_DataProcessor from Account B.
- **Why E is Incorrect:**
 - This option incorrectly suggests setting permissions with a focus on the principal within Account B's policy. It does not establish the cross-account access required for Account B's IAM user to access resources in Account A. Moreover, S3 bucket access policies are the standard method for granting cross-account access, not IAM user policies which typically manage permissions within the same account.

Question 7

A company plans to modernize a web application currently on Amazon EC2 by transitioning to microservices that operate within containers. The application operates under two segregated environments: production and testing. The application experiences variable traffic, with both minimum and maximum loads well-defined. A solutions architect is tasked with designing a serverless structure that keeps operational complexity to a minimum while being cost-effective. What is the most cost-effective solution to meet these requirements?

- A. Convert the application into container images and deploy them as AWS Lambda functions. Set concurrency limits on these Lambda functions to manage peak traffic. Use Amazon API Gateway to create two separate integrations, one for the production environment and another for testing.
- B. Store the container images in Amazon Elastic Container Registry (Amazon ECR). Use Amazon Elastic Container Service (Amazon ECS) with Fargate to automatically scale both production and testing environments based on load expectations. Deploy the containers as ECS tasks and manage traffic with two distinct Application Load Balancers, one for each environment.
- C. Save the container images in Amazon Elastic Container Registry (Amazon ECR). Implement auto-scaling Amazon Elastic Kubernetes Service (Amazon EKS) clusters with the Fargate profile for both production and testing, tailored to the known load patterns. Use these clusters to deploy the container tasks and set up two separate Application Load Balancers for traffic routing.

D. Transfer the container images to AWS Elastic Beanstalk and establish separate environments for production and testing. Use Elastic Beanstalk's management capabilities to handle deployments and configure individual Application Load Balancers for each environment to route the traffic.

Correct Answer: **B**

Explanation of Why B is Correct and Others Are Not:

- **Why B is Correct:**
 - Amazon ECS with Fargate provides a serverless container management service, eliminating the need to provision or manage servers and allowing the architecture to scale with the application's load. By using auto-scaling, it adapts to the variable load in a cost-effective manner, and separate Application Load Balancers ensure traffic is correctly routed to the respective environments. This approach meets the serverless and operational simplicity requirements.
- **Why A is Incorrect:**
 - AWS Lambda is not designed for running traditional containerized applications and has limits on the size of the deployment package and execution time, which may not align with the needs of a traditional web application. Furthermore, Lambda functions are not the best fit for applications with a consistent minimum load due to potential cold start issues.
- **Why C is Incorrect:**
 - Amazon EKS with Fargate is more complex and might be overkill for scenarios where the Kubernetes orchestration's advanced features are not required. Additionally, it can be more costly due to the management overhead and typically higher costs associated with EKS compared to ECS.
- **Why D is Incorrect:**
 - AWS Elastic Beanstalk simplifies deployment and management but is not considered a serverless architecture. It requires the provisioning of servers and does not offer the same level of scaling flexibility and cost-effectiveness as ECS with Fargate for containerized microservices.

Question 8

A company's web application, with its data on an Amazon RDS Multi-AZ DB instance and a read replica in a secondary Region, is served by Amazon EC2 instances under an Application Load Balancer (ALB). These instances are scaled by an Auto Scaling group, which, along with the ALB, is duplicated in a backup AWS Region with scaling values set to zero. The application's endpoint is provided to users through an Amazon Route 53 DNS record. The company aims to achieve a Recovery Time Objective (RTO) of under 15 minutes and enable the application to switch

operations to the backup Region automatically in case of failure. Their budget does not allow for an active-active setup. What solution should a solutions architect suggest to meet these requirements cost-effectively?

A. Adjust the application's Route 53 DNS record to use latency-based routing between the two ALBs. In the backup Region, set up an AWS Lambda function to upgrade the read replica to a primary instance and adjust the Auto Scaling group parameters. Use a CloudWatch alarm based on the ALB's HTTPCode_Target_5XX_Count metric in the primary Region to trigger the Lambda function.

B. In the backup Region, deploy an AWS Lambda function to promote the read replica to a primary instance and change the Auto Scaling group settings. Configure Route 53 with a health check for the application, set to notify the Lambda function via Amazon SNS upon detecting an unhealthy status. Modify the application's Route 53 DNS record with a failover policy that redirects traffic to the backup ALB upon health check failure.

C. Set the backup Region's Auto Scaling group to match the primary Region's settings. Change the application's Route 53 DNS record to a latency-based routing policy balancing traffic between the two ALBs. Remove the read replica and instead use a standalone RDS DB instance with Cross-Region Replication via snapshots and Amazon S3.

D. Set up AWS Global Accelerator with both ALBs as targets with equal weights. Implement an AWS Lambda function in the backup Region to promote the read replica and adjust the Auto Scaling group values. Establish a CloudWatch alarm based on the HTTPCode_Target_5XX_Count metric for the primary ALB to initiate the Lambda function.

Correct Answer: **B**

Explanation of Why B is Correct and Others Are Not:

- **Why B is Correct:**
 - **AWS Lambda Function:** It serves the role of automating the promotion of the read replica and adjustment of the Auto Scaling group when needed.
 - **Route 53 Health Check and Failover Policy:** This setup actively monitors the health of the web application and, combined with the failover policy, automatically reroutes traffic to the backup Region's ALB if a failure is detected. It ensures minimal downtime, aligning with the RTO goal.
 - **Cost-Effective:** This solution does not involve additional costs for active-active traffic distribution, making it budget-friendly while still achieving the required RTO.
- **Why A is Incorrect:**
 - Latency-based routing is used for performance optimization rather than failover purposes and does not ensure automatic failover to the backup Region within the

desired RTO.

- **Why C is Incorrect:**
 - It requires the backup Auto Scaling group to be active, which could incur more costs, deviating from the request for a cost-effective solution. Additionally, removing the read replica and setting up cross-region replication is a complex and costly approach that is likely to exceed the desired RTO.
- **Why D is Incorrect:**
 - AWS Global Accelerator with equal-weighted targets would imply an active-active configuration, which contradicts the company's budget constraints. It also does not provide an automatic failover mechanism based on health checks.

Question 9

A company is currently running an essential application on a standalone Amazon EC2 instance. The application's performance relies on a single-node Amazon ElastiCache for Redis cluster for in-memory data storage and an Amazon RDS for MariaDB instance for its relational database needs. It is critical that each infrastructure component remains operational for the application to work effectively. A solutions architect is tasked with enhancing the application's architecture to ensure that it can recover automatically from any point of failure with minimal downtime. Which three actions should be taken to fulfill these requirements?

- A. Implement an Elastic Load Balancer (ELB) to distribute incoming traffic across multiple EC2 instances. Ensure these instances are part of an Auto Scaling group configured with a minimum of two instances to support failover and recovery.
- B. Deploy an ELB to manage traffic to multiple EC2 instances, with those instances set to operate in unlimited mode to handle any amount of load.
- C. Adjust the MariaDB database to include a read replica within the same Availability Zone and set up the system to promote this replica to the primary database in the event of a primary DB failure.
- D. Convert the MariaDB database instance into a Multi-AZ deployment, enabling automatic failover to a standby instance in a different Availability Zone should the primary database encounter issues.
- E. Set up an Auto Scaling group for the ElastiCache for Redis cluster with a minimum of two instances, ensuring automatic scaling in response to changes in demand.
- F. Establish a replication group for the ElastiCache for Redis cluster and activate Multi-AZ support to allow for automatic failover to a replica in another Availability Zone in case of failure.

Correct Answer: **A, D, F**

Explanation of Why A, D, F Are Correct and the Others Are Not:

- **Why A is Correct:**
 - Introducing an ELB to balance the load across multiple EC2 instances can prevent a single point of failure. Utilizing an Auto Scaling group with at least two instances ensures that there is always a backup instance running, which is crucial for achieving high availability and automatic recovery from EC2 instance failures.
- **Why D is Correct:**
 - MariaDB instances configured for Multi-AZ deployments provide enhanced availability and durability. This setup involves a primary instance and a synchronously replicated secondary instance in a different Availability Zone. AWS handles automatic failover to the secondary instance if the primary one fails, minimizing downtime.
- **Why F is Correct:**
 - For ElastiCache for Redis, creating a replication group with Multi-AZ enables automatic failover across Availability Zones. If the primary node fails, ElastiCache will automatically fail over to a read replica in a different Availability Zone, ensuring the in-memory data store remains available.
- **Why B is Incorrect:**
 - While 'unlimited mode' for EC2 instances allows an instance to burst above its baseline level when needed, it does not contribute to automatic recovery from failures or ensure high availability.
- **Why C is Incorrect:**
 - Having a read replica in the same Availability Zone doesn't protect against AZ-wide service disruptions. The goal is to recover from any failure, which is better achieved through a Multi-AZ deployment rather than a single-AZ strategy.
- **Why E is Incorrect:**
 - Auto Scaling groups are not applicable to ElastiCache for Redis clusters as they are used for EC2 instances. For ElastiCache, the emphasis should be on replication and Multi-AZ for failover capabilities, not on scaling based on demand.

Question 10

A retail company is operating its ecommerce application on AWS. The application runs on Amazon EC2 instances behind an Application Load Balancer (ALB). The company uses an Amazon RDS DB instance as the database backend. Amazon CloudFront is configured with one origin that points to the ALB. Static content is cached. Amazon Route 53 is used to host all public zones.

After an update of the application, the ALB occasionally returns a 502 status code (Bad Gateway) error. The root cause is malformed HTTP headers that are returned to the ALB. The webpage

returns successfully when a solutions architect reloads the webpage immediately after the error occurs.

While the company is working on the problem, the solutions architect needs to provide a custom error page instead of the standard ALB error page to visitors.

Which combination of steps will meet this requirement with the LEAST amount of operational overhead?

- A. Set up an Amazon S3 bucket for web hosting and upload the custom error page. This S3 bucket will serve as the location for the error page.
- B. Implement an Amazon CloudWatch alarm that triggers an AWS Lambda function when the ALB reports failed health checks. The Lambda function should then alter the ALB's rules to redirect traffic to an alternate web server.
- C. Adjust the Amazon Route 53 configurations by integrating health checks and specifying an alternate target for when these checks fail. Update the DNS settings to redirect to an alternative webpage if an error is detected.
- D. Configure an Amazon CloudWatch alarm to initiate an AWS Lambda function when the ALB logs internal errors. Have the Lambda function modify the ALB's rules to direct traffic to a different web server that's publicly available.
- E. Add a custom error response in Amazon CloudFront by setting up a CloudFront custom error page. This will present the custom error page when the 502 status code is returned.

Correct Answer: **A, E**

Explanation of Why A, E Are Correct and the Others Are Not:

- **Why A is Correct:**
 - Creating an S3 bucket to host a static webpage is a straightforward method to serve a custom error page. The S3 bucket can be set up quickly with minimal maintenance, fulfilling the requirement for operational simplicity.
- **Why E is Correct:**
 - CloudFront can handle custom error responses, which allows the custom error page hosted on S3 (or another location) to be displayed when a 502 error is encountered. This process requires minimal operational overhead and can be easily integrated into the existing CloudFront distribution.
- **Why B is Incorrect:**

- Using a CloudWatch alarm to trigger a Lambda function for modifying ALB forwarding rules introduces unnecessary complexity and may not be as quick to set up or maintain. It's also not the most operationally simple approach to handling intermittent error responses.
- **Why C is Incorrect:**
 - While Route 53 health checks and DNS failover can redirect traffic during outages, they are not intended for immediate error response handling like the 502 Bad Gateway error. This method also introduces DNS changes, which can lead to propagation delays and additional complexity.
- **Why D is Incorrect:**
 - Similar to option B, setting up a CloudWatch alarm to invoke a Lambda function adds complexity. Additionally, changing ALB rules might not be suitable for handling sporadic HTTP header issues and can disrupt the user experience.

Question 11

A company is utilizing AWS Organizations to oversee its multiple AWS accounts and requires a centralized network management solution. They want to maintain a unified VPC within a dedicated infrastructure account, where the infrastructure team will exclusively manage the network configurations. At the same time, it is necessary for other accounts within the organization to be able to deploy AWS resources within the subnets of this VPC, without granting them network management permissions. What two steps should the solutions architect take to fulfill these requirements effectively?

- A. In the infrastructure account, establish a transit gateway that interconnects VPCs across the various accounts.
- B. Activate resource sharing capabilities within AWS Organizations by configuring it in the management account.
- C. Within each individual AWS account, set up VPCs with identical CIDR blocks and subnet configurations as the VPC in the infrastructure account and establish VPC peering with the infrastructure account's VPC.
- D. In the infrastructure account, initiate a resource share through AWS Resource Access Manager, including the specific Organizational Unit (OU) in AWS Organizations that necessitates access to the shared network, and select the subnets for the resource share.
- E. In the infrastructure account, create a resource share using AWS Resource Access Manager and tie it to the specific OU in AWS Organizations needing network access, choosing the prefix lists as part of the resource share.

Correct Answer: **B, D**

Explanation of Why B, D Are Correct and the Others Are Not:

- **Why B is Correct:**
 - Enabling resource sharing from the AWS Organizations management account is a necessary step to centrally manage resources and share them across accounts. This allows the infrastructure account to share the necessary networking resources (like subnets) with other accounts in the organization without giving up network management control.
- **Why D is Correct:**
 - Utilizing AWS Resource Access Manager (RAM) to create a resource share specifically for the VPC and its subnets is the direct way to grant other accounts within the organization the ability to create resources within those subnets. It aligns with the requirement that individual accounts can deploy resources within the shared network without managing it.
- **Why A is Incorrect:**
 - While a transit gateway does connect VPCs, it's a networking construct that facilitates inter-VPC communication and is not used for the central management of network resources or to permit resource creation within a VPC by external accounts.
- **Why C is Incorrect:**
 - Creating duplicate VPCs with peering does not meet the requirement of a common network managed by the infrastructure team. Peering VPCs would also not allow the other accounts to operate within the infrastructure account's VPC subnets, but rather within their own, which contradicts the shared network requirement.
- **Why E is Incorrect:**
 - Prefix lists in AWS are used for managing security groups and route tables, not for sharing subnets across accounts. This option would not allow other accounts to create resources within the infrastructure account's VPC.

Question 12

A company needs to integrate with a third-party SaaS application for its operations, which is also hosted on AWS within a VPC. The SaaS application will be accessed via API calls from the company's VPC. The company's internal security guidelines require all connectivity to be private, without any data traveling over the public internet, and no external access to the company's VPC resources. Access must also be tightly controlled to adhere to the principle of least privilege. What approach should the company take to establish this secure and private connection?

A. Set up an AWS PrivateLink interface VPC endpoint within the company's VPC that connects to the third-party SaaS application's endpoint service. Implement a security group that restricts access solely to the endpoint and link this security group to the newly created endpoint.

B. Establish an AWS Site-to-Site VPN connection from the company's VPC to the VPC of the third-party SaaS application. Use network Access Control Lists (ACLs) to regulate and restrict traffic flow through the VPN connection.

C. Create a VPC peering connection between the company's VPC and the VPC of the third-party SaaS application. Modify the route tables to include the required routes for the new peering connection.

D. Deploy an AWS PrivateLink endpoint service in the company's VPC and request the third-party SaaS provider to configure an interface VPC endpoint connected to this service. Provide the necessary permissions to the SaaS provider's account for using the endpoint service.

Correct Answer: **A**

Explanation of Why A is Correct and the Others Are Not:

- **Why A is Correct:**
 - AWS PrivateLink provides private connectivity between VPCs, AWS services, and on-premises applications securely on the AWS network. By creating a PrivateLink interface VPC endpoint, the company can access the SaaS application without the traffic going over the public internet, in line with their security policies. The use of a security group further tightens access control, ensuring that only necessary traffic can reach the endpoint, which aligns with the principle of least privilege.
- **Why B is Incorrect:**
 - A Site-to-Site VPN connection would normally route traffic over the internet, which does not comply with the company's requirement for private connectivity. While the VPN connection is encrypted, it doesn't meet the guideline of not traversing the internet.
- **Why C is Incorrect:**
 - VPC peering allows for the direct connection between two VPCs, but it's not suitable when the company has no control over the third-party SaaS application's VPC or when the third-party does not offer peering as a connectivity option. Moreover, it would potentially allow access to the company's resources from outside their VPC, which violates their security policies.
- **Why D is Incorrect:**
 - This option is reversed; the company cannot create an endpoint service for the SaaS provider. Instead, it's the SaaS provider that would create the endpoint service which the company could then connect to via a PrivateLink interface VPC endpoint.

Question 13

A company is seeking a unified solution to manage and report on the patching of its diverse set of servers, which includes both on-premises servers and Amazon EC2 instances. They require a consolidated report that displays the patch status for all servers. What actions should a solutions architect take to establish this process and reporting?

- A. Implement AWS Systems Manager for patch management across both the on-premises servers and EC2 instances, utilizing its capabilities to create comprehensive patch compliance reports.
- B. Employ AWS OpsWorks for patch management activities for all servers, both on-premises and in the cloud. Integrate with Amazon QuickSight to generate the necessary patch compliance reports.
- C. Configure an Amazon EventBridge rule to schedule patching via an AWS Systems Manager patch remediation job, and use Amazon Inspector for the compilation of patch compliance reports.
- D. Utilize AWS OpsWorks for the patching workflow of all servers. Leverage AWS X-Ray to monitor the patch status and forward this information to AWS Systems Manager OpsCenter for consolidated reporting.

Correct Answer: **A**

Explanation of Why A is Correct and the Others Are Not:

- **Why A is Correct:**
 - AWS Systems Manager provides a comprehensive management solution that can be used across both EC2 instances and on-premises servers. It includes a patch management feature that allows you to define patch baselines, schedule patching, and automate the patching process. Furthermore, Systems Manager has built-in reporting capabilities, which can generate the patch compliance reports required by the company's management, satisfying the need for a single report on the patch status of all servers.
- **Why B is Incorrect:**
 - AWS OpsWorks can automate operations, but it doesn't have built-in patch management capabilities similar to Systems Manager. While QuickSight could be used for visualization, the integration process to create patch compliance reports would be more complex and less direct than using Systems Manager's reporting.
- **Why C is Incorrect:**
 - Amazon EventBridge can trigger patching operations, and Systems Manager can manage patch remediation jobs, but Amazon Inspector is primarily used for security

assessments, not patch compliance reporting. This option adds unnecessary complexity and does not provide a built-in solution for patch compliance reporting.

- **Why D is Incorrect:**
 - AWS OpsWorks and AWS X-Ray are not primarily designed for patch management. AWS X-Ray is a service for analyzing and debugging distributed applications, not for managing or reporting patch compliance. Additionally, OpsCenter is designed for managing operations, not specifically for patch compliance reports.

Question 14

A company has an application deployed on Amazon EC2 instances within an Auto Scaling group, managed by an Application Load Balancer. The application's workload fluctuates during the day, causing EC2 instances to frequently scale in (terminate) and out. Logs from these instances are scheduled to be uploaded to an Amazon S3 bucket every 15 minutes, but there are missing logs from the instances that have been terminated. How can the company ensure that logs from terminating EC2 instances are preserved and sent to the S3 bucket?

- A. Develop a script that transfers log files to S3 and place it on the EC2 instances. Set up an Auto Scaling lifecycle hook and an Amazon EventBridge rule to respond to lifecycle transitions. Use an AWS Lambda function triggered by the `autoscaling:EC2_INSTANCE_TERMINATING` event to halt the termination process, execute the log transfer script, and then manually terminate the instance.
- B. Construct an AWS Systems Manager (SSM) document containing a script for S3 log uploads. Establish an Auto Scaling lifecycle hook, coupled with an EventBridge rule to monitor Auto Scaling events. Trigger a Lambda function upon the `autoscaling:EC2_INSTANCE_TERMINATING` event to execute the SSM document's commands to save logs and then allow the instance to terminate normally.
- C. Increase the frequency of log uploads to every 5 minutes. Incorporate a log transfer script into the EC2 instances' user data. Configure an EventBridge rule to detect when EC2 instances are terminated. Trigger a Lambda function from this rule to initiate the user-data script for log transfers and terminate the instance using the command line interface (CLI).
- D. Integrate an AWS Systems Manager (SSM) document with a log-saving script to S3. Generate an Auto Scaling lifecycle hook that triggers an Amazon Simple Notification Service (SNS) topic alert. When the SNS alert is received, invoke the SSM API to run the script from the document to transfer logs and command the Auto Scaling group to terminate the instance.

Correct Answer: **B**

Explanation of Why B is Correct and the Others Are Not:

- **Why B is Correct:**
 - AWS Systems Manager offers a robust way to execute scripts on EC2 instances, making it suitable for operations such as copying logs to S3. An Auto Scaling lifecycle hook can pause the instance termination process, allowing the Lambda function to invoke the Systems Manager to execute the script. Once the logs are copied, the lifecycle hook is instructed to continue, and the instance can terminate as normal. This process ensures that logs are saved without disrupting the Auto Scaling procedures.
- **Why A is Incorrect:**
 - While it uses a lifecycle hook, the method of preventing termination and then manually terminating the instance introduces unnecessary complexity and potential for errors. Additionally, directly invoking a script from Lambda may not be as reliable as using Systems Manager's built-in mechanisms.
- **Why C is Incorrect:**
 - EventBridge cannot directly trigger a script on an instance without another service like Lambda or Systems Manager. Additionally, changing the log transfer frequency does not address the problem of capturing logs from instances that are about to terminate.
- **Why D is Incorrect:**
 - Sending a message to an SNS topic to invoke Systems Manager is an indirect method that could introduce delays and complexity. Also, using ABANDON instead of allowing the instance to terminate normally could interfere with the proper functioning of the Auto Scaling group.

Question 15

A company operates with several AWS accounts. They have DNS records in a private hosted zone in Amazon Route 53 within Account A, while their applications and databases are hosted in Account B. A solutions architect is in the process of setting up a two-tier application in a new VPC and has created a CNAME record set for the Amazon RDS endpoint, `db.example.com`, in the Route 53 private hosted zone. However, during the application's deployment phase, it was observed that the `db.example.com` domain could not be resolved by the Amazon EC2 instance. Despite confirming the correct creation of the record set in Route 53, the application fails to start. What steps should the solutions architect take to troubleshoot and resolve the domain resolution issue? Select two actions.

- Deploy the database service on a standalone EC2 instance within the new VPC, and establish a DNS record for the private IP of this instance in the existing private hosted zone.
- Access the application-tier EC2 instance via SSH and directly insert the RDS endpoint's IP address into the `resolv.conf` file to facilitate domain resolution.

C. Initiate an authorization process to link the private hosted zone in Account A with the VPC that is being set up in Account B.

D. Establish a new private hosted zone within Account B for the `example.com` domain and configure DNS record replication between the two AWS accounts.

E. Integrate the new VPC in Account B with the existing private hosted zone in Account A, and afterward, remove the association authorization within Account A.

Correct Answer: **C, E**

Explanation of Why C, E Are Correct and the Others Are Not:

- **Why C is Correct:**
 - Creating an authorization is a necessary step to associate a private hosted zone from one AWS account (Account A) with a VPC in another AWS account (Account B). This association is required for the EC2 instances within the new VPC in Account B to resolve DNS queries using the private hosted zone from Account A.
- **Why E is Correct:**
 - Once the authorization is in place, you can associate the new VPC in Account B with the existing private hosted zone in Account A. The association is what allows the DNS records from the private hosted zone in Account A to be used by resources in the VPC in Account B, thereby making `db.example.com` resolvable. Removing the association authorization after the association is done is a clean-up step to ensure the principle of least privilege is maintained.
- **Why A is Incorrect:**
 - Deploying the database on a separate EC2 instance and creating a new record for its private IP is not necessary and does not address the issue of the RDS endpoint not being resolvable due to the private hosted zone being in a different account.
- **Why B is Incorrect:**
 - Manually adding entries to the `resolv.conf` file is not a scalable or AWS-recommended approach for managing DNS. It can also be overwritten by the DHCP lease renewals. This does not solve the fundamental issue of the private hosted zone being in a different AWS account.
- **Why D is Incorrect:**
 - There is no feature in AWS Route 53 that supports replication of DNS records between accounts. Instead, the solution is to associate the private hosted zone from one account with the VPC in another account.

Question 16

A company's web fleet, hosting a blog site on Amazon EC2 instances within an Auto Scaling group and an Application Load Balancer (ALB), has recently started experiencing user complaints of buffering and timeout issues, particularly when accessing videos. This increase in issues coincided with a new feature allowing bloggers to add videos to their posts, which led to a tenfold increase in user traffic. The web application uses Amazon EFS for storing all blog content, including videos. What is the most cost-effective and scalable solution to address the performance issues users are experiencing, especially during peak traffic times?

- A. Modify the Amazon EFS configuration to enable its maximum I/O performance mode.
- B. Transition the blog site to use instance store volumes for data storage. Implement a strategy to copy the site's contents to these volumes upon instance launch and back up to Amazon S3 when instances are shut down.
- C. Set up an Amazon CloudFront distribution, directing it to an Amazon S3 bucket, and migrate the video content from Amazon EFS to the S3 bucket.
- D. Create an Amazon CloudFront distribution for the entire site's content and configure it to target the ALB as its origin.

Correct Answer: **C**

Explanation of Why C is Correct and the Others Are Not:

- **Why C is Correct:**
 - Utilizing Amazon CloudFront, a content delivery network (CDN), to distribute the video content stored in an S3 bucket is an effective way to reduce load times and improve the user experience, especially for media content. Migrating videos from EFS to S3 and using CloudFront allows for efficient caching and delivery of content closer to the users, significantly reducing buffering and timeout issues. This approach is scalable and cost-efficient, especially for handling high traffic and large media files like videos.
- **Why A is Incorrect:**
 - While increasing the I/O performance of Amazon EFS might improve file access times, it is not the most cost-effective solution, particularly for delivering high-traffic video content. EFS is generally more expensive for storing and serving static files like videos compared to using S3 with CloudFront.
- **Why B is Incorrect:**
 - Using instance store volumes for dynamic content delivery introduces complexity and potential data loss risks, as instance store volumes are ephemeral. This approach also doesn't address the fundamental issue of delivering high-bandwidth content, like videos, to a large number of users simultaneously.
- **Why D is Incorrect:**

- Setting up CloudFront for the entire site content and pointing it at the ALB might improve performance for static content, but it doesn't address the specific issue of video content delivery. Additionally, serving dynamic content like blog posts through CloudFront without careful configuration can lead to caching issues.

Question 17

A globally operating company currently has a 1 Gbps AWS Direct Connect connection to one AWS Region. This connection is used for communication between their on-premises network and their AWS Cloud resources, through a single private virtual interface linked to one VPC. The company requires an architecture that adds a backup Direct Connect connection within the same Region for redundancy. Additionally, the solution should facilitate future connectivity to multiple AWS Regions using the same Direct Connect connections. What approach should the solutions architect take to satisfy these requirements?

- A. Set up a Direct Connect gateway. Remove the existing private virtual interface from the current connection. Establish a second Direct Connect connection. For each connection, create a new private virtual interface and associate both interfaces with the Direct Connect gateway. Link the gateway to the existing VPC.
- B. Maintain the current private virtual interface. Establish a second Direct Connect connection. On this new connection, configure a new private virtual interface and link it directly to the existing VPC.
- C. Retain the existing private virtual interface. Set up a second Direct Connect connection. On the new connection, create a public virtual interface and connect this interface to the existing VPC.
- D. Implement a transit gateway. Remove the current private virtual interface from the existing connection. Establish a second Direct Connect connection. On both connections, create new private virtual interfaces and connect them to the transit gateway. Associate this gateway with the existing VPC.

Correct Answer: **A**

Explanation of Why A is Correct and the Others Are Not:

- **Why A is Correct:**
 - Setting up a Direct Connect gateway allows for the creation of a centralized connectivity hub that can manage connections to multiple VPCs across different AWS Regions. By removing the existing private virtual interface and creating new private virtual interfaces on both the existing and new Direct Connect connections, the solution ensures redundancy. Connecting both interfaces to the Direct Connect

gateway and then linking the gateway to the VPC enables the desired redundancy and prepares the infrastructure for expansion into other Regions.

- **Why B is Incorrect:**
 - While creating a second Direct Connect connection with a new private virtual interface provides redundancy, it doesn't address the requirement for future multi-region connectivity. Directly connecting to a single VPC limits the flexibility to expand into other Regions.
- **Why C is Incorrect:**
 - Creating a public virtual interface on the new connection would allow access to AWS public services but does not provide the required redundancy for the existing private virtual interface. Additionally, it does not facilitate multi-region VPC connectivity.
- **Why D is Incorrect:**
 - Although a transit gateway can be used to interconnect VPCs and on-premises networks, it's not the optimal solution for this scenario. The question specifies the need for a redundant Direct Connect setup and multi-region connectivity through Direct Connect, not VPC interconnectivity, which is the primary use case of a transit gateway.

Question 18

A company operates a web application where users upload short videos. Currently, these videos are stored on Amazon EBS volumes and categorized using a custom software tool. The application, experiencing variable traffic with peak periods, is hosted on Amazon EC2 instances within an Auto Scaling group. There's also a separate Auto Scaling group of EC2 instances for processing an Amazon SQS queue. The company is looking to streamline its architecture by reducing operational complexity and eliminating third-party software dependencies, favoring AWS managed services instead. What architectural changes should be made to align with these objectives?

- A. Transition the web application to Amazon ECS containers and utilize Spot instances in the Auto Scaling group for SQS queue processing. Replace the custom categorization software with Amazon Rekognition.
- B. Migrate video storage to Amazon EFS, with the file system accessible to the EC2 instances of the web application. Configure an AWS Lambda function to process the SQS queue and integrate it with Amazon Rekognition for video categorization.
- C. Host the web application on Amazon S3 for static content serving. Store uploaded videos in Amazon S3 and set up S3 event notifications to trigger messages to the SQS queue. Utilize an AWS Lambda function to process the SQS queue, invoking Amazon Rekognition for video categorization.

D. Implement AWS Elastic Beanstalk for deploying EC2 instances in an Auto Scaling group for the web application. Use Elastic Beanstalk's worker environment for SQS queue processing and adopt Amazon Rekognition for video categorization.

Correct Answer: **C**

Explanation of Why C is Correct and the Others Are Not:

- **Why C is Correct:**
 - Hosting the static content of the web application on Amazon S3 is a scalable and managed solution for handling variable traffic efficiently. Using Amazon S3 for video storage not only simplifies the architecture but also integrates seamlessly with AWS services like Lambda and Rekognition. S3 event notifications can directly trigger SQS messages, which are then processed by Lambda functions. This setup eliminates the need for EC2 instances and Auto Scaling groups for the application, thereby reducing operational overhead. Amazon Rekognition replaces the custom software for video categorization, further aligning with the goal of using AWS managed services.
- **Why A is Incorrect:**
 - While using Amazon ECS and Spot instances offers a managed environment and cost savings, respectively, it still involves maintaining container infrastructure and does not simplify the architecture as much as the S3-Lambda combination. Also, it doesn't address the static content hosting aspect.
- **Why B is Incorrect:**
 - Amazon EFS is not the most cost-effective or scalable solution for storing user-uploaded videos, especially when compared to Amazon S3. Although using Lambda for processing the SQS queue is a step towards reducing operational overhead, this option doesn't provide a comprehensive managed service solution for the entire architecture.
- **Why D is Incorrect:**
 - AWS Elastic Beanstalk simplifies application deployment and management but still relies on running and managing EC2 instances. This approach does not reduce the operational overhead to the same extent as leveraging fully managed services like S3 and Lambda.

Question 19

A company's serverless application, integrating Amazon CloudFront, Amazon API Gateway, and AWS Lambda, currently relies on manual AWS CLI scripts to deploy new Lambda function versions and to revert to previous versions in case of errors. The company aims to improve the efficiency of deploying and rolling back application logic changes in Lambda functions, while

also enhancing the detection and response time to errors. What approach should be adopted to achieve faster deployments and more effective error handling?

A. Implement and deploy nested AWS CloudFormation stacks, where the parent stack includes the CloudFront distribution and API Gateway, and the child stack comprises the Lambda function. Utilize AWS CloudFormation change sets for Lambda updates, and roll back to a previous version when errors occur.

B. Utilize AWS Serverless Application Model (SAM) along with AWS CodeDeploy for Lambda deployments, gradually shifting traffic to the new version. Incorporate pre-traffic and post-traffic testing to verify the updated code, with automatic rollback based on Amazon CloudWatch alarms.

C. Consolidate the current AWS CLI deployment scripts into a single script for updating the Lambda function. After deploying the new version, perform testing, and roll back to the prior version if errors are identified.

D. Develop an AWS CloudFormation stack with a new API Gateway endpoint linked to the updated Lambda version. Redirect the CloudFront origin to this new endpoint, monitor for errors, and revert the CloudFront origin to the former API Gateway endpoint if issues arise.

Correct Answer: **B**

Explanation of Why B is Correct and the Others Are Not:

- **Why B is Correct:**
 - AWS SAM provides an efficient and streamlined way to manage serverless applications like Lambda functions. Combined with AWS CodeDeploy, it enables controlled deployment strategies, such as canary or linear deployments, which can gradually shift traffic to the new function version. This phased approach allows for effective error detection through pre-traffic and post-traffic testing. If any CloudWatch alarms are triggered, CodeDeploy can automatically roll back to the previous stable version. This method significantly decreases both deployment time and the time to detect and revert errors.
- **Why A is Incorrect:**
 - While AWS CloudFormation is useful for managing infrastructure as code, its change sets are more suited for infrastructure updates rather than application deployment strategies. CloudFormation does not offer the same level of control and monitoring for traffic shifting and automated rollbacks as AWS CodeDeploy in conjunction with SAM.
- **Why C is Incorrect:**
 - Refactoring AWS CLI scripts into a single deployment script does not inherently provide a more efficient deployment process or faster error detection and rollback

mechanism. It still relies on manual intervention for error detection and rollback, unlike the automated processes offered in option B.

- **Why D is Incorrect:**
 - Creating a new API Gateway endpoint for each deployment and changing the CloudFront origin introduces unnecessary complexity and potential for error. It's also not an efficient method for deploying Lambda functions or quickly rolling back changes. This approach does not leverage the benefits of controlled deployments and automated rollbacks.

Question 20

A company needs to store a substantial amount of archived documents, which will be accessed by employees via the corporate intranet. Employees connect to the system through a client VPN service linked to a VPC. It is essential that the data remains private and not accessible to the public. The company has physical copies of these documents; thus, the frequency of digital access will be low. They do not prioritize availability or speedy retrieval. What is the most cost-effective storage solution that meets these criteria?

- A. Set up an Amazon S3 bucket using the S3 One Zone-Infrequent Access (S3 One Zone-IA) storage class by default. Enable website hosting on the S3 bucket. Create an S3 interface endpoint and restrict bucket access solely to this endpoint.
- B. Deploy an Amazon EC2 instance with a web server and connect an Amazon Elastic File System (Amazon EFS) with files stored in the EFS One Zone-Infrequent Access (EFS One Zone-IA) storage class. Adjust instance security groups to permit access only from the private network.
- C. Implement an Amazon EC2 instance with a web server and attach an Amazon Elastic Block Store (Amazon EBS) volume for archiving, opting for the Cold HDD (sc1) volume type. Configure the instance's security groups to limit access to private networks only.
- D. Create an Amazon S3 bucket, setting the S3 Glacier Deep Archive storage class as the default. Enable website hosting on the S3 bucket. Establish an S3 interface endpoint and allow bucket access exclusively through this endpoint.

Correct Answer: **A**

Explanation of Why A is Correct and the Others Are Not:

- **Why A is Correct:**
 - Amazon S3 One Zone-Infrequent Access (S3 One Zone-IA) is a cost-effective storage class designed for data that is accessed less frequently. It provides the same durability and throughput as standard S3 but at a lower cost. Since availability and retrieval speed are not priorities, and considering the low request rate, this option is ideal. The

configuration of an S3 interface endpoint and access restriction ensures that the data remains private and accessible only through the corporate VPN.

- **Why B is Incorrect:**
 - Using Amazon EC2 with EFS One Zone-IA, while suitable for infrequent access, introduces more operational complexity and higher costs compared to S3 One Zone-IA. Managing EC2 instances and EFS would require more maintenance and does not provide the same level of cost-effectiveness as S3.
- **Why C is Incorrect:**
 - Running an EC2 instance with an attached EBS volume (Cold HDD) is more expensive and operationally complex than necessary for the company's requirements. It also does not offer the same level of scalability and ease of management as an S3-based solution.
- **Why D is Incorrect:**
 - S3 Glacier Deep Archive is an extremely low-cost storage service for archiving and long-term backup, suitable for data that can tolerate retrieval times of several hours. However, setting up website hosting on a Glacier Deep Archive bucket is not possible, and this storage class would be overly restrictive for the company's needs, given that immediate data access isn't a priority but might occasionally be required.

Question 21

A company currently utilizes an on-premises Active Directory service for user authentication and wishes to extend this authentication to sign in to its AWS accounts managed under AWS Organizations. They already have AWS Site-to-Site VPN connections set up between their on-premises network and AWS accounts. The company's security policy mandates conditional access based on user groups and roles, with a requirement for managing user identities in a centralized location. What solution should be implemented to integrate their on-premises Active Directory with AWS for user authentication, while adhering to their security policy?

A. Set up AWS IAM Identity Center (AWS Single Sign-On) and integrate it with the on-premises Active Directory using SAML 2.0. Activate automatic user provisioning through the System for Cross-domain Identity Management (SCIM) v2.0 protocol. Implement attribute-based access controls (ABACs) to regulate access to the AWS accounts.

B. Configure AWS IAM Identity Center (AWS Single Sign-On) using IAM Identity Center as the identity source. Enable automatic user provisioning using the System for Cross-domain Identity Management (SCIM) v2.0 protocol. Manage access to AWS accounts using IAM Identity Center permission sets.

C. In one of the AWS accounts, establish AWS Identity and Access Management (IAM) with a SAML 2.0 identity provider linked to the on-premises Active Directory. Create IAM users

corresponding to federated users and assign access based on Active Directory groups. Manage access to other AWS accounts through cross-account IAM users.

D. In one of the AWS accounts, configure AWS Identity and Access Management (IAM) with an OpenID Connect (OIDC) identity provider connected to the on-premises Active Directory. Create IAM roles for federated users based on Active Directory groups and enable access to AWS accounts through cross-account IAM roles.

Correct Answer: **A**

Explanation of Why A is Correct and the Others Are Not:

- **Why A is Correct:**
 - AWS IAM Identity Center (AWS Single Sign-On) with SAML 2.0 integration allows for seamless federation with the on-premises Active Directory. The use of SCIM v2.0 for automatic provisioning simplifies user management by syncing users and groups from Active Directory to AWS. Attribute-based access controls (ABACs) provide the necessary flexibility for conditional access based on user attributes and roles, aligning with the company's requirement for group- and role-based access management. This solution maintains centralized identity management and meets the conditional access policy requirements.
- **Why B is Incorrect:**
 - Using IAM Identity Center as the identity source doesn't leverage the company's existing on-premises Active Directory, contradicting the requirement for centralized user identity management.
- **Why C is Incorrect:**
 - Configuring IAM with a SAML 2.0 identity provider directly in AWS accounts requires the management of IAM users corresponding to Active Directory users, which can become complex and does not centralize identity management as effectively as AWS Single Sign-On.
- **Why D is Incorrect:**
 - The use of an OpenID Connect (OIDC) identity provider is more suited for web identity federation and doesn't align as well with the integration of an existing on-premises Active Directory setup. This approach also adds unnecessary complexity compared to the AWS Single Sign-On solution.

Question 22

A software company's application, which uses Amazon API Gateway, AWS Lambda functions, and an Amazon DynamoDB table to consume a REST API, is experiencing a rise in errors during PUT requests. Analysis shows that most PUT requests, and consequently the errors, are from a limited number of clients using specific API keys, with a significant portion originating from a

single client. The API is not critical and can handle retry attempts for failed calls. However, these errors are being shown to customers, negatively impacting the API's reputation. What strategy should the solutions architect recommend to enhance the customer experience without compromising the API's functionality?

- A. Integrate retry logic with exponential backoff and jitter in the client application, ensuring that errors are intercepted and handled with clear error messages.
- B. Apply API throttling through a usage plan in API Gateway, and ensure that the client application is designed to manage HTTP 429 responses gracefully without displaying an error.
- C. Enable API caching to improve response times at the production stage, and conduct load tests for 10 minutes to confirm the adequacy of cache capacity for the current workload.
- D. Set reserved concurrency limits for the Lambda functions to allocate the necessary resources during sudden spikes in traffic.

Correct Answer: **B**

Explanation of Why B is Correct and the Others Are Not:

- **Why B is Correct:**
 - Implementing API throttling through a usage plan in API Gateway is an effective way to manage and limit the rate of requests from overactive clients. Throttling helps in controlling the load on the backend services like Lambda and DynamoDB, potentially reducing the occurrence of errors. By handling HTTP 429 ("Too Many Requests") responses appropriately in the client application, the visibility of these errors to end users can be minimized, improving their experience. This solution directly addresses the issue of a single client sending a large number of requests, which is identified as the primary cause of the errors.
- **Why A is Incorrect:**
 - While retry logic with exponential backoff is generally a good practice for handling transient errors in client applications, it does not address the root cause of the issue - the high volume of requests from a few clients. It may actually exacerbate the problem by increasing the number of requests.
- **Why C is Incorrect:**
 - API caching can improve response times for GET requests by caching responses, but it does not typically benefit PUT requests, which are causing the errors in this scenario. Additionally, caching does not address the issue of a high number of requests from a single client.
- **Why D is Incorrect:**

- Setting reserved concurrency for Lambda functions can help manage Lambda resources, but it doesn't solve the problem of too many incoming requests from a single client. It might also lead to throttling of other legitimate requests if the concurrency limit is reached, further impacting the user experience.

Question 23

A company's data-intensive application, running on AWS, utilizes a cluster of hundreds of Amazon EC2 instances and a shared file system hosted on several EC2 instances, containing 200 TB of data. The application, which operates within a single AWS Region, generates a report monthly by processing a subset of files from the shared file system over a 72-hour period. The compute instances are managed by an Auto Scaling group, while the storage instances run continuously. The solutions architect is tasked with reducing costs by replacing the shared file system instances, ensuring that the file system delivers high-performance access to the necessary data during the 72-hour job execution. What is the most cost-effective solution to meet these requirements while providing the largest overall cost reduction?

- A. Transfer the data from the current shared file system to an Amazon S3 bucket utilizing S3 Intelligent-Tiering. Each month before the job, use Amazon FSx for Lustre to create a new file system, importing data from S3 using lazy loading. Utilize this file system as shared storage during the job and delete it post-completion.
- B. Move the data from the shared file system to a large Amazon Elastic Block Store (Amazon EBS) volume with Multi-Attach enabled. Automate the attachment of this EBS volume to the EC2 instances through the Auto Scaling group's launch template. Use this volume as shared storage for the job duration and detach it afterward.
- C. Migrate the data from the existing shared file system to an Amazon S3 bucket set to S3 Standard. Before each monthly job, create a new Amazon FSx for Lustre file system, loading data from S3 via batch loading. Use this file system for shared storage during the job and delete it once completed.
- D. Shift the data from the current shared file system to an Amazon S3 bucket. Prior to the monthly job, establish an AWS Storage Gateway file gateway using the data from S3. Employ this file gateway for shared storage during the job and remove it after the job's completion.

Correct Answer: **A**

Explanation of Why A is Correct and the Others Are Not:

- **Why A is Correct:**
 - Option A offers a solution that balances performance and cost. By migrating to Amazon S3 with Intelligent-Tiering, the data is stored cost-effectively, adapting to

access patterns. Amazon FSx for Lustre, integrated with S3, provides a high-performance file system ideal for data-intensive workloads. Lazy loading ensures that only the required data is loaded into FSx for Lustre, optimizing performance and cost. The ability to delete the FSx file system after each run significantly reduces costs compared to maintaining persistent storage.

- **Why B is Incorrect:**
 - Using a large EBS volume with Multi-Attach introduces potential performance bottlenecks, as it is not designed for high-throughput, data-intensive workloads like FSx for Lustre. Also, the continuous running and manual management of an EBS volume are less cost-effective compared to the on-demand creation and deletion of an FSx for Lustre file system.
- **Why C is Incorrect:**
 - While FSx for Lustre integrated with S3 Standard is a viable option, batch loading is less efficient than lazy loading for this use case. Batch loading would load the entire data set, which is unnecessary for a job that only processes a subset of files, leading to increased costs and reduced performance.
- **Why D is Incorrect:**
 - AWS Storage Gateway's file gateway provides a way to integrate on-premises environments with cloud storage, but it is not optimized for high-performance computing tasks. This approach would likely result in suboptimal performance for data-intensive jobs and would not be as cost-effective as using FSx for Lustre.

Question 24

A company is creating a new service accessible via TCP on a static port and needs it to be highly available, redundant across Availability Zones, and accessible through the DNS name my.service.com. This service must have fixed address assignments so that other companies can whitelist its addresses. Assuming deployment across multiple Availability Zones in a single AWS Region, what solution should be implemented to meet these requirements?

- A. Deploy Amazon EC2 instances, each with its own Elastic IP address. Set up a Network Load Balancer (NLB) to handle the static TCP port and register the EC2 instances with it. Create a DNS name server record set for my.service.com, assigning it the Elastic IP addresses of the EC2 instances. Distribute these Elastic IP addresses for whitelisting.
- B. Establish an Amazon ECS cluster with a service definition for the app, assigning public IP addresses to the cluster. Use a Network Load Balancer (NLB) to expose the TCP port. Create a target group and link it to the ECS cluster. Set up a DNS A record for my.service.com, pointing to the ECS cluster's public IP addresses. Share these public IPs for whitelisting.

C. Set up Amazon EC2 instances for the service. Allocate one Elastic IP address per Availability Zone. Configure a Network Load Balancer (NLB) for the TCP port, assigning the Elastic IP addresses to it per Availability Zone. Register the EC2 instances with the NLB. Create a DNS A (alias) record for my.service.com, pointing it to the NLB DNS name.

D. Build an Amazon ECS cluster and a service definition for the app, assigning a public IP address to each host in the cluster. Implement an Application Load Balancer (ALB) for the static TCP port. Link a target group to the ECS service definition in the ALB. Set up a CNAME DNS record set linked to the public IP addresses. Provide the Elastic IP addresses for whitelisting.

Correct Answer: **C**

Explanation of Why C is Correct and the Others Are Not:

- **Why C is Correct:**
 - Option C offers a solution that aligns with the need for high availability, redundancy across Availability Zones, and fixed address assignments. By allocating one Elastic IP address per Availability Zone and associating them with a Network Load Balancer (NLB), the solution ensures that the service is available even if one zone goes down. The NLB handles traffic distribution across EC2 instances. The use of a DNS A (alias) record pointing to the NLB DNS name allows the service to be accessed using the desired DNS name while maintaining the fixed addresses necessary for whitelisting.
- **Why A is Incorrect:**
 - Assigning Elastic IP addresses directly to EC2 instances and using these IPs in a DNS record set does not offer the same level of high availability and load balancing as an NLB. It also complicates management and does not leverage AWS load balancing capabilities.
- **Why B is Incorrect:**
 - While ECS and NLB can be part of a scalable solution, assigning public IP addresses to an ECS cluster and using these IPs in a DNS record doesn't provide the same level of reliability and ease of management as associating Elastic IPs with an NLB. ECS is also not necessary for this scenario.
- **Why D is Incorrect:**
 - Using an Application Load Balancer (ALB) and CNAME records with public IP addresses does not satisfy the requirement for fixed address assignments necessary for whitelisting. ALBs are also generally used for HTTP/HTTPS traffic, not generic TCP traffic.

Question 25

A company is planning to migrate its on-premises data analytics platform, consisting of 12 servers, to Amazon EC2 instances while adopting a consumption-based model to reduce costs.

The platform executes hourly and daily scheduled jobs, which are critical with tight SLAs and make up 65% of system usage. Additionally, it runs user-requested jobs that usually complete in under 5 minutes without any specific SLA, accounting for 35% of system usage. In case of system failures, it's acceptable for user jobs to be delayed, but scheduled jobs must still meet their SLAs. The solution must ensure high availability and adherence to SLAs in a cost-effective manner without long-term commitments. What is the most cost-effective solution to transition this system to EC2 while maintaining its high availability and meeting the SLAs?

- A. Distribute the 12 EC2 instances evenly across two Availability Zones in the selected AWS Region. Configure two instances in each Availability Zone as On-Demand Instances with Capacity Reservations, and the remaining four instances in each zone as Spot Instances.
- B. Allocate the 12 EC2 instances across three Availability Zones in the chosen AWS Region. In one Availability Zone, run all four instances as On-Demand Instances with Capacity Reservations. Use Spot Instances for the remaining instances in the other two zones.
- C. Place the 12 EC2 instances across three Availability Zones in the selected AWS Region. In each Availability Zone, deploy two instances as On-Demand Instances under a Savings Plan, and the other two instances as Spot Instances.
- D. Arrange the 12 EC2 instances across three Availability Zones in the specified AWS Region. In each zone, set up three instances as On-Demand Instances with Capacity Reservations and one instance as a Spot Instance.

Correct Answer: **D**

Explanation of Why D is Correct and the Others Are Not:

- **Why D is Correct:**
 - Option D provides a balanced approach to high availability and cost-effectiveness. By splitting the 12 instances across three Availability Zones, it ensures redundancy and fault tolerance. Using three On-Demand Instances with Capacity Reservations in each zone offers the necessary reliability and performance for the critical scheduled jobs, while one Spot Instance in each zone can handle less critical user jobs. This setup aligns with the need for maintaining SLAs during system failures and offers flexibility to handle variable workloads cost-effectively.
- **Why A is Incorrect:**
 - Having only two On-Demand Instances in each Availability Zone might not provide enough capacity to handle the critical scheduled jobs, especially during peak times or system failures. Additionally, relying heavily on Spot Instances could lead to interruptions, particularly during high-demand periods, which could impact the SLAs.
- **Why B is Incorrect:**

- Concentrating all On-Demand Instances in a single Availability Zone poses a risk to high availability. If the zone with all On-Demand Instances experiences issues, it could jeopardize the SLAs for the scheduled jobs.
- **Why C is Incorrect:**
 - While a Savings Plan can offer cost savings, it requires a commitment, which contradicts the requirement for no long-term commitments. Additionally, having only two On-Demand Instances in each zone may not provide sufficient capacity for the critical jobs, particularly if there's a system failure in one of the zones.

Question 26

A security engineer is enhancing the security of an application that currently retrieves credentials for an Amazon RDS for MySQL database from an encrypted S3 file. For the application's next version, the following improvements are desired: utilize robust, randomly generated database passwords stored in a secure, AWS-managed service; deploy application resources using AWS CloudFormation; and rotate database credentials every 90 days. The task is to create a CloudFormation template to deploy these application updates. Which combination of resources in the CloudFormation template will most effectively fulfill these security enhancements with minimal operational effort?

- A. Create the database password using an AWS Secrets Manager secret. Include an AWS Lambda function for password rotation. Define a Secrets Manager RotationSchedule to rotate the password every 90 days.
- B. Establish the database password as a SecureString type in AWS Systems Manager Parameter Store. Implement an AWS Lambda function for password rotation. Configure a Parameter Store RotationSchedule to update the password every 90 days.
- C. Use AWS Secrets Manager to generate the database password. Deploy an AWS Lambda function for rotating the password. Utilize an Amazon EventBridge scheduled rule to initiate the Lambda function for password rotation at 90-day intervals.
- D. Define the database password as a SecureString parameter in AWS Systems Manager Parameter Store. Employ an AWS AppSync DataSource to automate password rotation every 90 days.

Correct Answer: **A**

Explanation of Why A is Correct and the Others Are Not:

- **Why A is Correct:**
 - AWS Secrets Manager is specifically designed for managing and rotating secrets, making it an ideal choice for handling database credentials securely. It supports

automatic rotation of secrets, which fulfills the requirement for strong, randomly generated passwords. The integration of a Lambda function for custom rotation logic and the use of Secrets Manager RotationSchedule offers an automated, 90-day password rotation cycle. This approach meets all the specified requirements with the least operational overhead, as it leverages built-in AWS capabilities for secret management and rotation.

- **Why B is Incorrect:**
 - While AWS Systems Manager Parameter Store can store secrets, it does not have built-in support for automated secret rotation like AWS Secrets Manager. The need to manually configure a rotation mechanism, including a custom Lambda function and a rotation schedule, adds unnecessary operational complexity.
- **Why C is Incorrect:**
 - Option C correctly utilizes AWS Secrets Manager for secret storage but relies on Amazon EventBridge for triggering the Lambda function for password rotation. This introduces additional complexity compared to the built-in rotation schedule feature of Secrets Manager in option A, which is more straightforward and operationally efficient.
- **Why D is Incorrect:**
 - AWS AppSync DataSource is not a tool designed for secret management or password rotation. Furthermore, AWS Systems Manager Parameter Store does not provide an automated password rotation feature like AWS Secrets Manager, making this option unsuitable for the desired automated, 90-day rotation requirement.

Question 27

A company has its data distributed across multiple Amazon DynamoDB tables and needs a serverless solution to make this data publicly available via a straightforward HTTPS-based API. This solution should be capable of scaling automatically to handle varying levels of user demand. What are two appropriate solutions that fulfill these requirements effectively?

- A. Implement an Amazon API Gateway REST API, configuring it to interface directly with DynamoDB using the AWS integration option available within API Gateway.
- B. Establish an Amazon API Gateway HTTP API and set it up for direct interaction with DynamoDB through API Gateway's AWS integration feature.
- C. Deploy an Amazon API Gateway HTTP API, and integrate it with AWS Lambda functions that retrieve and return data from the DynamoDB tables.
- D. Set up an AWS Global Accelerator and configure it with AWS Lambda@Edge functions, designed to fetch and return data from the DynamoDB tables.

E. Construct a Network Load Balancer and design listener rules that direct incoming requests to relevant AWS Lambda functions.

Correct Answers: **A, C**

Explanation of Why A, C Are Correct and the Others Are Not:

- **Why A is Correct:**
 - Amazon API Gateway REST API with direct integrations to DynamoDB provides a seamless, serverless method to expose DynamoDB data over HTTPS. The AWS integration type allows API Gateway to interact directly with DynamoDB, removing the need for intermediate services or additional coding. This solution scales automatically and fits the requirement for a simple, scalable API.
- **Why C is Correct:**
 - Configuring an Amazon API Gateway HTTP API with AWS Lambda functions to access DynamoDB tables is a highly effective serverless approach. Lambda functions offer flexibility in data processing and retrieval from DynamoDB, and the integration with API Gateway HTTP API facilitates a scalable, secure HTTP/S endpoint. This combination caters to the need for a public API that automatically adjusts to demand.
- **Why B is Incorrect:**
 - While Amazon API Gateway HTTP API is a valid option for creating APIs, the direct integration of HTTP APIs with DynamoDB is not supported. This direct integration is currently a feature specific to REST APIs.
- **Why D is Incorrect:**
 - AWS Global Accelerator and Lambda@Edge are more suited for improving global application availability and performance. This setup is not typically used for creating APIs to directly access DynamoDB data and does not provide the straightforward API solution required by the company.
- **Why E is Incorrect:**
 - Network Load Balancer is primarily used for balancing TCP/UDP traffic and is not designed to create HTTPS APIs. It is more aligned with load balancing across EC2 instances or containers rather than integrating with serverless platforms like Lambda for API purposes.

Question 28

A company has recently acquired 10 domain names for use in its online marketing efforts. They need a solution that will efficiently redirect visitors from each of these domains to specific URLs as defined in a JSON document. All the domain's DNS records are managed using Amazon Route 53. The solution must support both HTTP and HTTPS requests. The objective is to devise a method that minimizes operational effort to set up and manage these redirects. What three

steps should be combined to create a solution that fulfills these requirements with minimal operational complexity?

- A. Deploy a dynamic web page hosted on an Amazon EC2 instance, programmed to parse the JSON document and event message, and then return the appropriate redirect URL based on this data.
- B. Implement an Application Load Balancer configured with listeners for both HTTP and HTTPS traffic.
- C. Develop an AWS Lambda function that processes incoming requests, referencing the JSON document to determine and return the correct redirect URL.
- D. Set up an Amazon API Gateway API using a custom domain to expose the AWS Lambda function externally.
- E. Configure an Amazon CloudFront distribution and integrate a Lambda@Edge function to handle the redirection logic.
- F. Generate an SSL/TLS certificate through AWS Certificate Manager (ACM), including all the new domains as Subject Alternative Names in the certificate.

Correct Answers: **C, E, F**

Explanation of Why C, E, F Are Correct and the Others Are Not:

- **Why C is Correct:**
 - Utilizing an AWS Lambda function for URL redirection based on a JSON document is a serverless, scalable, and low-maintenance approach. Lambda can process incoming requests and perform the redirection logic without the need for managing underlying server infrastructure.
- **Why E is Correct:**
 - Amazon CloudFront with Lambda@Edge allows the redirection logic to be executed closer to the user at the edge locations, enhancing performance. This setup is particularly effective for handling both HTTP and HTTPS traffic globally with minimal latency.
- **Why F is Correct:**
 - Creating an SSL/TLS certificate in AWS Certificate Manager with all the new domains as Subject Alternative Names ensures that HTTPS requests are securely handled. This is crucial for a solution that accepts HTTPS traffic and is essential for securing web redirects.
- **Why A is Incorrect:**

- Hosting a dynamic webpage on an EC2 instance to handle redirection involves more operational effort in terms of setup, maintenance, and scaling. This approach is less efficient compared to a serverless solution like Lambda.
- **Why B is Incorrect:**
 - An Application Load Balancer is primarily used for distributing incoming application traffic across multiple targets. While it can handle HTTP and HTTPS requests, it is not the most straightforward or efficient solution for simple URL redirection based on a JSON document.
- **Why D is Incorrect:**
 - While Amazon API Gateway can expose a Lambda function, it's an unnecessary additional layer for this specific use case. Lambda@Edge with CloudFront directly addresses the requirement more efficiently by executing the redirection logic at edge locations.

Question 29

A multi-account AWS organization utilizes various AWS services, including VPCs, Amazon EC2 instances, and containers, across its accounts. In each VPC, the organization's compliance team has deployed security tools running on EC2 instances. These tools send data to a dedicated AWS account for the compliance team. All compliance-related resources are marked with a "costCenter" tag, assigned the value "compliance". The organization seeks an accurate method to determine the expenses incurred by these security tools on EC2 instances to appropriately bill the compliance team's AWS account. What approach should a solutions architect take to accurately capture and report these costs?

- Activate the "costCenter" user-defined tag in the organization's management account. Set up the AWS Cost and Usage Reports to be delivered monthly to an S3 bucket in the management account. Utilize the tag breakdown in these reports to calculate the total costs associated with resources tagged as "costCenter".
- Enable the "costCenter" user-defined tag in the organization's member accounts. Arrange for monthly AWS Cost and Usage Reports to be saved in an S3 bucket in the management account. Schedule a monthly AWS Lambda function to process these reports and sum up the costs of resources tagged "costCenter".
- In the organization's member accounts, activate the "costCenter" tag. From the management account, configure a monthly AWS Cost and Usage Report. Employ the tag breakdown within the report to ascertain the total costs for resources tagged with "costCenter".
- Generate a custom report using AWS Trusted Advisor's organization view. Set this report to produce a monthly billing summary, specifically targeting the "costCenter" tagged resources within the compliance team's AWS account.

Correct Answer: **A**

Explanation of Why A is Correct and the Others Are Not:

- **Why A is Correct:**
 - Activating the "costCenter" tag in the management account and configuring AWS Cost and Usage Reports to aggregate data into an S3 bucket allows for a centralized view of costs across the organization. The tag breakdown feature in these reports enables precise tracking and calculation of costs associated with the specific "costCenter" tag. This method provides the most accurate and straightforward way to measure and allocate costs for the compliance team's resources.
- **Why B is Incorrect:**
 - While this option also involves using AWS Cost and Usage Reports, the additional step of processing these reports with a Lambda function each month introduces unnecessary complexity. This method is less efficient compared to directly using the tag breakdown feature in the reports as in Option A.
- **Why C is Incorrect:**
 - Activating the "costCenter" tag in member accounts but configuring the Cost and Usage Report from the management account can lead to incomplete data aggregation and might not capture all relevant tagged resource costs accurately. The centralized activation of the tag in the management account (as in Option A) is a more reliable approach.
- **Why D is Incorrect:**
 - AWS Trusted Advisor primarily provides recommendations for optimizing AWS environments and is not specifically designed for detailed cost allocation and reporting. It does not offer the same level of detailed cost tracking and reporting based on specific user-defined tags as the AWS Cost and Usage Reports.

Question 30

In an organization that operates with 50 AWS accounts under AWS Organizations, each containing multiple VPCs, there is a need to establish inter-VPC connectivity within each member account using AWS Transit Gateway. Additionally, the organization seeks to automate the process for future accounts, ensuring that each new member account automatically sets up a new VPC and establishes a transit gateway attachment. What two steps should be combined to achieve these objectives, ensuring seamless integration of new accounts with efficient VPC creation and transit gateway attachment?

A. Utilize AWS Resource Access Manager in the management account to share the transit gateway with the member accounts.

- B. Employ an AWS Organizations Service Control Policy (SCP) from the management account to share the transit gateway with the member accounts.
- C. Execute an AWS CloudFormation stack set from the management account, designed to create a new VPC and a VPC transit gateway attachment in new member accounts, and link this attachment to the transit gateway using its ID.
- D. Initiate an AWS CloudFormation stack set from the management account to generate a new VPC and a peering transit gateway attachment in each new member account, and connect this attachment to the transit gateway via a service-linked role.
- E. Share the transit gateway with member accounts from the management account by leveraging AWS Service Catalog.

Correct Answers: **A, C**

Explanation of Why A, C Are Correct and the Others Are Not:

- **Why A is Correct:**
 - AWS Resource Access Manager (RAM) is the ideal tool for sharing resources like transit gateways across accounts in an AWS Organization. It simplifies the process of sharing and ensures that the transit gateway is available to member accounts without manually replicating it in each account.
- **Why C is Correct:**
 - Utilizing AWS CloudFormation stack sets allows for the automated deployment of resources across multiple AWS accounts. By creating a stack set in the management account that automatically provisions a new VPC and creates a transit gateway attachment in each new member account, this approach streamlines the setup process and ensures consistency. The stack set can use the shared transit gateway ID to make the necessary connections.
- **Why B is Incorrect:**
 - Service Control Policies (SCPs) are used to manage permissions in AWS Organizations, not for resource sharing. SCPs define what actions are allowed or denied in member accounts, but they cannot directly share resources like a transit gateway.
- **Why D is Incorrect:**
 - The concept of a "peering transit gateway attachment" is not a standard approach and does not align with the typical functionality of AWS Transit Gateway. The CloudFormation stack set should focus on creating standard transit gateway attachments, not peering attachments.
- **Why E is Incorrect:**

- AWS Service Catalog is used to create and manage catalogs of IT services that are approved for use on AWS. While it can be used to standardize resource provisioning, it is not the most straightforward or effective method for sharing a transit gateway across multiple accounts.

Question 31

An enterprise-level organization seeks to enable its developers to acquire third-party software from AWS Marketplace, but with certain restrictions. The organization operates under AWS Organizations with full features and has designated shared services accounts within each organizational unit (OU) for procurement managers. The goal is to limit developers to accessing only approved software via AWS Private Marketplace. Furthermore, the organization wants to restrict the administration of the Private Marketplace to individuals with a specific IAM role named procurement-manager-role. Other IAM entities, including users, groups, roles, and account administrators, should not have administrative privileges for Private Marketplace. What architectural solution should be adopted to effectively implement these restrictions while ensuring the procurement team has the necessary administrative control?

- A. In every AWS account across the organization, establish an IAM role named procurement-manager-role with the PowerUserAccess managed policy attached. Implement an inline policy for all IAM users and roles in each account to explicitly deny access to AWSPrivateMarketplaceAdminFullAccess.
- B. In all AWS accounts within the organization, set up an IAM role named procurement-manager-role and assign the AdministratorAccess managed policy. Apply a permissions boundary that includes the AWSPrivateMarketplaceAdminFullAccess policy to all developer roles.
- C. In each shared services account of the organization, create an IAM role called procurement-manager-role with the AWSPrivateMarketplaceAdminFullAccess managed policy attached. Implement an SCP at the root level of the organization that denies the ability to administer Private Marketplace to all except the procurement-manager-role. Also, apply another root-level SCP that prohibits creation of any IAM role named procurement-manager-role across the organization.
- D. In all AWS accounts used by developers, introduce an IAM role named procurement-manager-role and attach the AWSPrivateMarketplaceAdminFullAccess managed policy. Use an SCP in AWS Organizations to restrict administration of Private Marketplace solely to procurement-manager-role, applying this SCP to all shared services accounts.

Correct Answer: **C**

Explanation of Why C is Correct and the Others Are Not:

- **Why C is Correct:**
 - This option correctly centralizes the management of Private Marketplace administration within the shared services accounts by creating a specific role for procurement managers. Attaching the `AWSPriateMarketplaceAdminFullAccess` managed policy to this role ensures that only designated managers can administer the Private Marketplace. The use of SCPs at the organization's root level effectively enforces the policy organization-wide, both denying general administrative access to Private Marketplace and preventing the creation of additional procurement-manager-role roles, thereby ensuring compliance with the procurement team's policy.
- **Why A is Incorrect:**
 - Granting `PowerUserAccess` to the procurement-manager-role in every account is excessive and doesn't align with the principle of least privilege. Additionally, denying permissions through inline policies is less efficient and more complex than using SCPs for organization-wide policy enforcement.
- **Why B is Incorrect:**
 - Assigning `AdministratorAccess` to the procurement-manager-role is too broad and doesn't align with the specific requirement of administering only the Private Marketplace. The use of a permissions boundary for developer roles doesn't address the need to restrict marketplace administration access across the entire organization.
- **Why D is Incorrect:**
 - While this option restricts the administration of the Private Marketplace to the procurement-manager-role, it incorrectly suggests creating this role in all developer accounts, which is unnecessary and could lead to decentralized management. The application of the SCP only to shared services accounts also does not ensure organization-wide enforcement of the policy.

Question 32

A company is in the process of implementing AWS Organizations to constrain its developers to use only Amazon EC2, Amazon S3, and Amazon DynamoDB. The developers account resides in a dedicated organizational unit (OU). The solutions architect has implemented the following SCP on the developers account:


```
1 {
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Sid": "AllowEC2",
6       "Effect": "Allow",
7       "Action": "ec2:*",
8       "Resource": "*"
9     }
10  ]
11 }
```

```

9      },
10     {
11         "Sid": "AllowDynamoDB",
12         "Effect": "Allow",
13         "Action": "dynamodb:*",
14         "Resource": "*"
15     },
16     {
17         "Sid": "AllowS3",
18         "Effect": "Allow",
19         "Action": "s3:*",
20         "Resource": "*"
21     }
22 ]
23 }

```

When this policy is deployed, IAM users in the developers account are still able to use AWS services that are not listed in the policy. What should the solutions architect do to eliminate the developers' ability to use services outside the scope of this policy?

- A. Add individual explicit deny statements for each AWS service that should not be available to the developers.
- B. Detach the FullAWSAccess SCP from the OU that contains the developers' accounts.
- C. Revise the FullAWSAccess SCP to include explicit deny statements for all other services.
- D. Append an explicit deny statement with a wildcard () to the SCP to prevent access to any services not explicitly allowed.

Correct Answer: **B**

Explanation of Why B is Correct and the Others Are Not:

- **Why B is Correct:**
 - AWS Organizations allows you to manage permissions using SCPs, which control the services and actions that users and roles can access within each account. By default, the FullAWSAccess SCP allows full access to all AWS services and resources. To restrict developers to only use specific services, the FullAWSAccess SCP should be removed from the developers' OU. This action ensures that only the SCPs explicitly allowing certain services (like EC2, S3, and DynamoDB) are in effect, effectively limiting the developers to those services.
- **Why A is Incorrect:**

- Creating explicit deny statements for each AWS service is not practical due to the large number of services and the ongoing maintenance required as new services are released or existing services are updated. This approach is also error-prone and can lead to overly complex SCPs.
- **Why C is Incorrect:**
 - Modifying the FullAWSAccess SCP is not recommended as it is a default SCP provided by AWS and is intended to allow access to all AWS services. Instead, custom SCPs should be created or existing ones should be adjusted to enforce specific permissions without altering the default FullAWSAccess SCP.
- **Why D is Incorrect:**
 - While adding a wildcard deny statement to the SCP could potentially block access to all services not listed, it's an implicit best practice to manage access using the least privilege principle—allowing what is needed and denying everything else by default. It's more efficient to remove the FullAWSAccess SCP and rely on other SCPs that explicitly allow the necessary services.

Question 33

A company's mobile application is served via a monolithic REST API, which is deployed on five Amazon EC2 instances located in the public subnets of a VPC. Mobile clients access the API using a domain managed by Amazon Route 53, which is set up with a multivalue answer routing policy linked to the IP addresses of the EC2 instances. The application recently experienced difficulty handling unexpected spikes in traffic. A solutions architect is tasked with devising a strategy that allows the application to manage these new and variable traffic loads efficiently, with a focus on minimizing operational complexity. What approach should be taken to equip the app to deal with the increased and fluctuating demand?

- Decompose the API into separate AWS Lambda functions and establish an Amazon API Gateway REST API with Lambda function integration as the new backend. Redirect the Route 53 domain record to target the API Gateway endpoint.
- Convert the API into a containerized format and set up an Amazon Elastic Kubernetes Service (EKS) cluster to host these containers using Amazon EC2 instances. Implement a Kubernetes ingress resource and reroute the Route 53 domain record to the ingress endpoint.
- Formulate an Auto Scaling group encompassing the current EC2 instances, and configure it to scale based on CPU usage metrics. Develop an AWS Lambda function to dynamically adjust the Route 53 domain record in response to changes in the Auto Scaling group.
- Introduce an Application Load Balancer (ALB) as an intermediary for the API and reposition the EC2 instances to private subnets within the VPC. Assign these instances as targets under the ALB and update the Route 53 domain record to point towards the ALB.

Correct Answer: **A**

Explanation of Why A is Correct and the Others Are Not:

- **Why A is Correct:**
 - Option A suggests transitioning to a serverless architecture, which inherently scales with demand and thus can handle variable traffic loads without the need for manual scaling configurations. By using AWS Lambda in conjunction with Amazon API Gateway, the application can automatically scale to accommodate increases in traffic. Additionally, this setup simplifies operational management since there are no servers to maintain, and AWS manages the scaling automatically. Updating Route 53 to point to the API Gateway consolidates traffic management and simplifies DNS configuration.
- **Why B is Incorrect:**
 - While containerization and the use of Amazon EKS provide a more modern infrastructure that can scale, this option adds significant operational complexity compared to a serverless approach. Managing an EKS cluster, container orchestration, and Kubernetes ingress resources requires deeper expertise and hands-on management.
- **Why C is Incorrect:**
 - Managing Auto Scaling for EC2 instances requires monitoring and configuring scaling policies. Additionally, writing a custom Lambda function to update Route 53 adds to the operational overhead. This option does not minimize the complexity as efficiently as leveraging serverless technologies.
- **Why D is Incorrect:**
 - Introducing an ALB improves traffic distribution across EC2 instances and can help with scaling to an extent. However, the underlying infrastructure still requires management of scaling policies and does not reduce operational overhead as significantly as a serverless approach. Moving EC2 instances to private subnets adds networking complexity and does not contribute to handling traffic spikes as effectively as AWS Lambda with API Gateway.

Question 34

Within a large organization that utilizes AWS Organizations, there are distinct Organizational Units (OUs) for different engineering teams, each managing multiple AWS accounts. With hundreds of accounts in the organization, a solutions architect is tasked with devising a system that allows each OU to monitor and analyze their AWS accounts' cost usage data. What architectural approach should be adopted to enable each OU to have visibility into their respective cost breakdowns?

- A. Generate a separate AWS Cost and Usage Report (CUR) for each OU by leveraging AWS Resource Access Manager, and provide access to each team to review the CUR via Amazon QuickSight dashboards.
- B. Produce a singular AWS Cost and Usage Report (CUR) from the central AWS Organizations management account, and permit each team to examine the CUR using Amazon QuickSight dashboards.
- C. Initiate an individual AWS Cost and Usage Report (CUR) in every member account of AWS Organizations, and facilitate each team's access to review the CUR through Amazon QuickSight dashboards.
- D. Formulate an AWS Cost and Usage Report (CUR) utilizing AWS Systems Manager, and provide access for each team to observe the CUR via Systems Manager OpsCenter dashboards.

Correct Answer: **B**

Explanation of Why B is Correct and the Others Are Not:

- **Why B is Correct:**
 - Creating a centralized AWS Cost and Usage Report (CUR) from the management account of AWS Organizations allows for the aggregation of cost data across all accounts within the organization. This centralized approach is not only efficient but also simplifies the process as it avoids the complexity of generating and managing multiple reports. Amazon QuickSight can then be used to create dashboards that are tailored to each OU, presenting a filtered view of the CUR data relevant to their respective accounts. This method minimizes operational overhead while providing the necessary cost visibility to each OU.
- **Why A is Incorrect:**
 - AWS Resource Access Manager is not a service used to generate CURs. It primarily facilitates the sharing of AWS resources across accounts. Additionally, creating separate reports for each OU would increase complexity and management overhead, contrary to the goal of minimizing operational effort.
- **Why C is Incorrect:**
 - Generating separate CURs for each member account can result in a fragmented view of cost data and complicate the consolidation process. This approach would significantly increase the effort required to provide each OU with a coherent and comprehensive view of their costs.
- **Why D is Incorrect:**
 - AWS Systems Manager is not designed to create CURs, and Systems Manager OpsCenter is a service that helps you view and manage operations data, not cost

usage data. This option does not align with the functionalities provided by AWS services for cost management and reporting.

Question 35

A company currently stores its data on an on-premises Windows file server, generating approximately 5 GB of new data each day. As part of its migration to AWS, the company needs to ensure this data is accessible in a cloud-based file system, especially since they have moved a portion of their Windows workloads to AWS. They have already set up AWS Direct Connect for connectivity between their local network and AWS. What approach should the company adopt for transferring their daily data increments to a cloud file system effectively?

- A. Implement the file gateway feature of AWS Storage Gateway to serve as a replacement for the current Windows file server, redirecting existing file shares to this new file gateway.
- B. Utilize AWS DataSync to create a daily synchronization job that moves data from the on-premises Windows file server to Amazon FSx, which is optimized for Windows-based applications.
- C. Configure AWS Data Pipeline to orchestrate a daily data transfer operation from the on-premises Windows file server to Amazon Elastic File System (Amazon EFS), suitable for Linux-based workloads.
- D. Arrange for AWS DataSync to facilitate a daily data replication task from the local Windows file server to Amazon Elastic File System (Amazon EFS), typically used with Linux-based systems.

Correct Answer: **B**

Explanation of Why B is Correct and the Others Are Not:

- **Why B is Correct:**
 - AWS DataSync is specifically designed for moving large volumes of data between on-premises storage and AWS services, including Amazon FSx for Windows File Server. FSx provides a fully managed native Windows file system in the cloud, which makes it an appropriate choice for Windows-based workloads. DataSync can be scheduled to run daily tasks, automating the transfer of the 5 GB of new data with minimal manual intervention, making it an efficient and effective migration strategy.
- **Why A is Incorrect:**
 - While AWS Storage Gateway's file gateway could be used to integrate on-premises file-based storage with cloud storage services, it's primarily used for ongoing hybrid storage environments rather than a direct migration strategy. Moreover, it doesn't replace the need for a managed file system service like Amazon FSx for Windows workloads in AWS.

- **Why C is Incorrect:**
 - AWS Data Pipeline is a service for processing and moving data between different AWS compute and storage services, but Amazon EFS is not compatible with Windows file systems. Amazon EFS is designed for Linux-based applications and workloads, which does not align with the company's requirement for a Windows-based cloud file system.
- **Why D is Incorrect:**
 - Similar to option C, this answer is incorrect because Amazon EFS does not support Windows file system data, and thus, using DataSync to transfer data to EFS would not be suitable for the company's Windows workloads.

Question 36

A solutions architect is evaluating a web application hosted on AWS that serves static assets from an Amazon S3 bucket located in the us-east-1 Region. To enhance the application's resilience and enable multi-region support, the company has also provisioned an S3 bucket in an additional AWS Region. What is the most streamlined approach to ensure that the application can reliably serve content from both Regions with minimal management effort?

- A. Modify the application to concurrently upload each asset to the two S3 buckets. Configure an Amazon Route 53 public hosted zone with record sets using a weighted routing policy directed at both buckets. Adjust the application to access the assets via the Route 53 domain.
- B. Develop an AWS Lambda function that duplicates assets from the primary S3 bucket in us-east-1 to the secondary bucket in the other Region whenever a new asset is uploaded. Establish an Amazon CloudFront distribution that uses both S3 buckets as origins in an origin group.
- C. Enable S3 cross-region replication from the primary bucket in us-east-1 to replicate content to the secondary bucket in the other Region. Configure an Amazon CloudFront distribution with an origin group featuring both S3 buckets.
- D. Set up cross-region replication for the S3 bucket in us-east-1 to mirror content to the secondary S3 bucket. In the event of a regional failure, modify the application's codebase to fetch assets from the secondary bucket.

Correct Answer: **C**

Explanation of Why C is Correct and the Others Are Not:

- **Why C is Correct:**
 - Enabling S3 cross-region replication automates the process of duplicating assets to a secondary bucket in another Region, ensuring that content is available from both locations with no additional manual steps. Integrating with Amazon CloudFront and setting up an origin group with both S3 buckets allows CloudFront to automatically

serve content from the operational bucket, providing high availability and resiliency with no intervention required. This solution optimizes for operational simplicity by handling failover at the CloudFront level rather than through application changes or complex routing policies.

- **Why A is Incorrect:**
 - Configuring the application to write to both buckets increases complexity and the potential for errors. Using Route 53 with a weighted routing policy adds unnecessary overhead because it requires DNS-level changes rather than leveraging the built-in failover capabilities of CloudFront, which can more efficiently manage content delivery and availability.
- **Why B is Incorrect:**
 - Creating a Lambda function to manually copy objects between buckets introduces additional components that must be managed and monitored, increasing operational overhead. It's also less efficient compared to the automated replication provided by S3. Additionally, CloudFront's failover capabilities are sufficient to handle regional resiliency without the need for a Lambda function.
- **Why D is Incorrect:**
 - Relying on updating the application's code to switch to the secondary bucket in case of a failover is not operationally efficient. It requires active intervention and code deployment, which could lead to downtime and is not an automated solution. This approach would not provide the least operational overhead compared to leveraging CloudFront's automatic failover.

Question 37

Following a critical outage caused by unexpected traffic spikes, a company plans to migrate its existing on-premises .NET-based web application, which relies on a MySQL database, to AWS to accommodate a user base of 200,000 daily visitors. The company requires a solution that not only scales effectively but also maintains high availability. How should the solutions architect structure this deployment on AWS to ensure that the application can reliably handle the increased traffic without downtime?

A. Deploy the application via AWS Elastic Beanstalk, setting up a web server environment with an Amazon RDS MySQL Multi-AZ DB instance for database redundancy. Arrange for a Network Load Balancer (NLB) to distribute traffic to an EC2 Auto Scaling group across multiple Availability Zones. Direct the company's domain traffic to the NLB using an Amazon Route 53 alias record.

B. Construct a scalable infrastructure using AWS CloudFormation, which includes an Application Load Balancer (ALB) directing traffic to an EC2 Auto Scaling group across three Availability Zones for fault tolerance. Incorporate a Multi-AZ Amazon Aurora MySQL DB cluster with a

Retain deletion policy to safeguard the database. Point the company's domain to the ALB with an Amazon Route 53 alias record.

C. Implement an AWS Elastic Beanstalk environment with auto-scaling capabilities that extends over two AWS Regions, each with its own ALB. Establish a Multi-AZ Amazon Aurora MySQL DB cluster with a cross-Region read replica to support the database workload. Configure Amazon Route 53 with a geoproximity routing policy to manage traffic between the two Regions.

D. Initiate an AWS CloudFormation stack that sets up an ALB in front of an Amazon ECS cluster configured with Spot instances spanning three Availability Zones. Add an Amazon RDS MySQL single-instance database with a Snapshot deletion policy to the architecture. Utilize an Amazon Route 53 alias record to redirect the company's domain traffic to the ALB.

Correct Answer: **B**

Explanation of Why B is Correct and the Others Are Not:

- **Why B is Correct:**
 - Option B suggests a highly resilient AWS infrastructure that leverages AWS CloudFormation for automated infrastructure deployment, ensuring consistent and repeatable setups. An ALB paired with an EC2 Auto Scaling group across three Availability Zones maximizes the application's availability and scalability. The use of Amazon Aurora MySQL, which is compatible with MySQL and offers enhanced performance and reliability features, further improves the database tier's resiliency. The Multi-AZ deployment provides fault tolerance, while the Retain deletion policy ensures data persistence. The use of Amazon Route 53 with an alias record offers a reliable DNS solution to route user traffic to the ALB, providing a seamless user experience.
- **Why A is Incorrect:**
 - While AWS Elastic Beanstalk simplifies application deployment, the NLB setup described here is not as feature-rich as an ALB, which offers advanced routing and high availability for HTTP/HTTPS applications, making it better suited for a .NET web application.
- **Why C is Incorrect:**
 - This option extends the architecture across two separate Regions, which may not be necessary and could introduce additional complexity and cost. A single-region, multi-AZ architecture is typically sufficient for high availability and fault tolerance. Moreover, cross-Region replication for Aurora is generally used for global applications or disaster recovery, not for handling traffic surges.
- **Why D is Incorrect:**
 - Using Spot instances within an ECS cluster can offer cost savings but may also introduce the risk of interruption, which is not ideal for a critical application requiring

high availability. Additionally, the use of a single RDS MySQL instance without a Multi-AZ deployment lacks the high availability requirement specified in the scenario, and a Snapshot deletion policy doesn't provide the same level of data protection as a Retain policy.

Question 38

A company, utilizing AWS Organizations for managing several AWS accounts, has a requirement to establish an Amazon Simple Notification Service (Amazon SNS) topic in every member account. This SNS topic is crucial for linking with an external third-party alerting system. A solutions architect has prepared an AWS CloudFormation template to standardize the SNS topic creation and plans to use CloudFormation StackSets for streamlined deployment across all accounts. With trusted access already activated in AWS Organizations, what actions should the solutions architect take to efficiently roll out the CloudFormation StackSets to all associated AWS accounts?

- A. In each member account of the organization, initiate a stack set with service-managed permissions. Opt for organization-wide deployment settings in the stack set configurations. Employ CloudFormation StackSets drift detection to maintain stack consistency.
- B. Individually create stacks within all member accounts. Opt for self-managed permissions. Select organization-wide deployment settings. Turn on automatic deployment for CloudFormation StackSets.
- C. From the management account of AWS Organizations, generate a stack set with service-managed permissions. Choose to deploy across the organization in the stack set configurations. Activate automatic deployment for CloudFormation StackSets.
- D. In the management account of AWS Organizations, establish individual stacks with service-managed permissions. Configure them to be deployed organization-wide. Implement CloudFormation StackSets drift detection to monitor stack alignment.

Correct Answer: **C**

Explanation of Why C is Correct and the Others Are Not:

- **Why C is Correct:**
 - Initiating a stack set from the management account with service-managed permissions allows the solutions architect to leverage the built-in integration of CloudFormation StackSets with AWS Organizations. By setting deployment options to target the entire organization, the solutions architect can ensure that the SNS topic is automatically created in all existing and future member accounts. Enabling automatic deployment simplifies the process by handling the deployment logistics, thus meeting the

requirement with minimal manual intervention and ensuring the SNS topic is consistently deployed across all accounts.

- **Why A is Incorrect:**
 - While service-managed permissions are appropriate for organization-wide deployment, creating a stack set in each member account is not practical. It requires unnecessary repetition and doesn't take advantage of the centralized management capabilities offered by deploying from the management account.
- **Why B is Incorrect:**
 - Self-managed permissions require the manual setup of IAM roles and permissions, increasing the complexity of the deployment process. This method also implies individual stack creation rather than a centralized stack set, which contradicts the goal of streamlined deployment across the organization.
- **Why D is Incorrect:**
 - Creating individual stacks in the management account does not facilitate automatic distribution to all member accounts. Service-managed permissions are specifically designed to work with stack sets, not individual stacks, for organization-wide deployment. Moreover, while drift detection is useful for monitoring, it doesn't contribute to the initial deployment process.

Question 39

A company is planning to transfer its diverse set of workloads, which are currently running on a mix of physical servers and virtual machines hosting a variety of applications on both Linux and Windows platforms, from its on-site data center to AWS. To facilitate a smooth transition, the company needs to collect comprehensive information about their existing system configurations, performance metrics, active processes, and network connectivity. Additionally, the company wants to categorize its on-premises applications into logical groupings to organize the migration process to AWS. Furthermore, they require guidance on selecting the most economical Amazon EC2 instance types suitable for their workloads. What three actions should the solutions architect execute to fulfill these requirements and ensure an efficient migration strategy?

- A. Deploy the AWS Application Discovery Agent on each of the on-premises physical servers and virtual machines to gather detailed insights into the current infrastructure.
- B. Install the AWS Systems Manager Agent on all on-premises servers and virtual machines to evaluate the existing applications.
- C. Utilize AWS Systems Manager Application Manager to categorize the on-premises servers into distinct application sets that are ready for migration.

D. Employ AWS Migration Hub to systematically group servers based on their respective applications in preparation for the AWS migration.

E. Leverage AWS Migration Hub to generate EC2 instance type recommendations and estimated cost considerations tailored to the company's workloads.

F. Input server sizing details into AWS Trusted Advisor and implement its cost optimization suggestions for the migration.

Correct Answers: **A, D, E**

Explanation of Why A, D, E Are Correct and the Others Are Not:

- **Why A is Correct:**
 - The AWS Application Discovery Agent is specifically designed to collect and send detailed system information, including configuration, performance data, and running processes, which is essential for migration planning. This tool helps identify dependencies and performance baselines, aiding in the selection of appropriate EC2 instance types for cost-effective migrations.
- **Why D is Correct:**
 - AWS Migration Hub provides a centralized location to track the progress of migrations across multiple AWS and partner solutions. It allows for the grouping of servers into applications, aiding in the organization and simplification of migration efforts, which can be critical for complex environments with numerous applications.
- **Why E is Correct:**
 - AWS Migration Hub not only tracks migrations but also assists in identifying suitable Amazon EC2 instance types for the workloads being migrated. It provides recommendations that can help the company run its applications on AWS both effectively and cost-efficiently.
- **Why B is Incorrect:**
 - While AWS Systems Manager Agent offers management capabilities for AWS and on-premises environments, it is not the primary tool for performing initial assessments of on-premises workloads for migration purposes.
- **Why C is Incorrect:**
 - AWS Systems Manager Application Manager is a feature within Systems Manager that helps manage applications, but it is not specifically designed for grouping servers for the purpose of migration like AWS Migration Hub.
- **Why F is Incorrect:**
 - AWS Trusted Advisor is a tool that provides recommendations for AWS resources already running in the cloud. It's not designed to import on-premises server data or provide detailed assessments and cost optimization strategies for migrations.

Question 40

A company operates an image-processing application on AWS within a Virtual Private Cloud (VPC) that spans two Availability Zones. Each zone consists of a public and a private subnet. Amazon EC2 instances located in the private subnets run the service, with an Application Load Balancer in the public subnets directing traffic to it. The service requires internet access, currently facilitated by two NAT gateways, and it heavily relies on Amazon S3, downloading around 1 TB of data from an S3 bucket daily. The service is marketed with a strong emphasis on security. The company now seeks to minimize cloud costs without compromising on security or increasing operational burden. What approach should be taken to align with these financial and security objectives?

- A. Transition from using NAT gateways to more cost-effective NAT instances. Update the VPC routing table to direct traffic from the private subnets through the newly configured NAT instances.
- B. Relocate the EC2 instances from the private to the public subnets and eliminate the NAT gateways to reduce the associated costs.
- C. Implement an S3 gateway endpoint within the VPC. Attach a policy to the endpoint that permits the necessary interactions with the S3 bucket.
- D. Connect an Amazon Elastic File System (EFS) volume to the EC2 instances and use it to store the images, replacing the S3 bucket.

Correct Answer: **C**

Explanation of Why C is Correct and the Others Are Not:

- **Why C is Correct:**
 - Introducing an S3 gateway endpoint in the VPC is a cost-saving measure as it enables instances in private subnets to access S3 without going through NAT gateways, which incur costs for data processing. The S3 gateway endpoint allows private access to S3, which maintains the application's high-security standards. This approach also reduces operational overhead as it simplifies the architecture without compromising security and eliminates the data transfer costs associated with NAT gateways.
- **Why A is Incorrect:**
 - NAT instances may be less expensive than NAT gateways but require additional configuration and management, which increases the operational burden. Furthermore, they generally offer lower bandwidth and less automatic scalability compared to NAT gateways, potentially impacting the service's performance.
- **Why B is Incorrect:**

- Moving EC2 instances to public subnets would expose them directly to the internet, which could significantly compromise the service's security posture. This action would go against the company's promise of a highly secure service and is not recommended for sensitive workloads.
- **Why D is Incorrect:**
 - Using Amazon EFS instead of S3 for image storage could result in higher costs, especially considering the volume of data involved (1 TB daily). Additionally, it would not address the service's need to access the internet, which is currently facilitated by the NAT gateways.

Question 41

Following the launch of a new application on AWS, which primarily utilizes Amazon DynamoDB, a company has identified that their application experiences a peak load once a week for a 4-hour duration. This peak load is twice as high as the usual load, which is consistent throughout the remainder of the week. Notably, the application's interaction with the DynamoDB table involves significantly more write operations than read operations. To optimize costs while accommodating these usage patterns, what strategy should the solutions architect adopt specifically for the DynamoDB table?

- A. Implement AWS Application Auto Scaling to dynamically adjust capacity during the high-demand periods. Secure reserved Read Capacity Units (RCUs) and Write Capacity Units (WCUs) aligned with the average load level.
- B. Switch to the on-demand capacity mode for the DynamoDB table to automatically handle capacity based on actual usage.
- C. Deploy DynamoDB Accelerator (DAX) alongside the table and subsequently reduce the provisioned read capacity to be in line with the recalculated peak load.
- D. Integrate DynamoDB Accelerator (DAX) with the table and shift to using the on-demand capacity mode for handling table operations.

Correct Answer: **A**

Explanation of Why A is Correct and the Others Are Not:

- **Why A is Correct:**
 - Option A presents a balanced approach to cost optimization by combining the predictability of reserved capacity with the flexibility of auto-scaling. By purchasing reserved capacity units for the average load, the company can benefit from cost savings associated with reserved pricing. Concurrently, AWS Application Auto Scaling can automatically increase capacity during the weekly peak periods, ensuring the table

can handle the doubled load without any performance issues. This approach effectively manages costs while still providing the necessary resources during peak and average load times.

- **Why B is Incorrect:**
 - While the on-demand capacity mode in DynamoDB offers simplicity and scalability, it may not be the most cost-effective for predictable and cyclical load patterns. In this scenario, the peak load is consistent and predictable, making the reserved capacity with auto-scaling a more cost-efficient choice than on-demand, which typically costs more per unit of capacity.
- **Why C is Incorrect:**
 - DynamoDB Accelerator (DAX) is primarily a caching service that enhances read performance and wouldn't significantly benefit a write-heavy workload. Reducing provisioned read capacity based on DAX would not address the key cost driver, which in this case is the write capacity during peak periods.
- **Why D is Incorrect:**
 - Combining DAX with on-demand capacity doesn't directly address the need to optimize costs for a write-heavy workload. DAX focuses on improving read performance and does not significantly impact the cost associated with write operations, which is the primary concern in this scenario.

Question 42

A company is looking to move its data processing application, currently based on-premises, to AWS. In its existing setup, users upload files through a web interface, which are then stored on NAS. A message queue notifies the processing server about these new files. Each file takes up to an hour to process. The company has noticed a pattern where the queue of files for processing peaks during business hours and diminishes significantly afterwards. The company seeks a migration strategy to AWS that optimizes costs while accommodating this fluctuating workload. What migration plan should be recommended for the most cost-effective solution?

- A. Implement Amazon SQS for queue management. Adjust the current web server to send messages to this SQS queue. Utilize AWS Lambda to process files from the queue as they arrive and save the processed files in Amazon S3.
- B. Set up Amazon MQ for the queue system. Modify the existing web server to interact with this MQ queue. Deploy an Amazon EC2 instance for processing the files from the queue and save the output in Amazon EFS. Terminate the EC2 instance once processing is complete.
- C. Utilize Amazon MQ for queue creation. Update the existing web server to connect with this MQ queue. Employ AWS Lambda to handle file processing from the queue and store the processed files in Amazon EFS.

D. Establish Amazon SQS for managing the queue. Configure the current web server to send messages to this SQS queue. Leverage Amazon EC2 instances within an EC2 Auto Scaling group to process files from the queue, scaling based on queue length, and store the results in Amazon S3.

Correct Answer: **D**

Explanation of Why D is Correct and the Others Are Not:

- **Why D is Correct:**
 - Option D provides a scalable and flexible solution by using Amazon SQS and EC2 Auto Scaling. Amazon SQS efficiently manages the queue of processing tasks, while EC2 Auto Scaling dynamically adjusts the number of EC2 instances based on the length of the SQS queue. This means that during peak hours, more instances will be running to handle the load, and fewer instances will be used during off-peak hours, optimizing cost efficiency. Storing processed files in Amazon S3 is a cost-effective choice for storing large amounts of data. This setup aligns well with the variable workload pattern described by the company.
- **Why A is Incorrect:**
 - AWS Lambda has a maximum execution time limit (15 minutes at the time of this response), which might not be sufficient for processing tasks that can take up to an hour. Therefore, using Lambda for this workload may lead to incomplete processing or require complex workarounds, making it less suitable for this scenario.
- **Why B is Incorrect:**
 - While this option provides a method to process files, manually managing the lifecycle of EC2 instances (creating and shutting down after tasks) introduces operational overhead and lacks the scalability and cost-effectiveness of an auto-scaling solution. Additionally, storing files in Amazon EFS might not be as cost-effective as S3 for processed media files.
- **Why C is Incorrect:**
 - Similar to option A, using AWS Lambda for processing may not be feasible due to the Lambda execution time limit, which could be insufficient for the up to 1-hour processing requirement per file. Moreover, Amazon EFS storage might not be necessary for processed files and could incur higher costs compared to using S3.

Question 43

A company currently utilizes Amazon OpenSearch Service for data analysis, loading data from an S3 Standard storage bucket into an OpenSearch Service cluster with 10 data nodes. This data is kept in the cluster for a one-month period for read-only analytical purposes, after which the relevant index in the cluster is deleted. However, for compliance reasons, the company needs to

maintain a copy of all original input data. Faced with concerns about ongoing operational costs, the company seeks a recommendation on how to restructure this setup in a more cost-effective manner, without violating data retention requirements.

A. Convert the current data nodes in the OpenSearch Service cluster to UltraWarm nodes to manage the expected data volume. Simultaneously, change the storage class of the input data from S3 Standard to S3 Glacier Deep Archive as soon as it is loaded into the OpenSearch cluster.

B. Downsize the OpenSearch Service cluster to only 2 data nodes and integrate UltraWarm nodes to support the expected data volume. Configure the system so that indexes automatically move to UltraWarm upon ingestion by OpenSearch Service. Implement an S3 Lifecycle policy to transition the original input data to S3 Glacier Deep Archive after the one-month period.

C. Decrease the number of data nodes in the OpenSearch Service cluster to 2 and add UltraWarm nodes for the anticipated data load. Set up the system to move indexes to UltraWarm on ingestion, and further transfer these indexes from UltraWarm to cold storage nodes within the cluster. Implement an S3 Lifecycle policy to delete the original data from the S3 bucket after one month.

D. Reduce the number of data nodes in the OpenSearch Service cluster to 2, but add instance-backed data nodes for the required capacity. Change the storage class of the input data from S3 Standard to S3 Glacier Deep Archive concurrently with its upload to the OpenSearch cluster.

Correct Answer: **B**

Explanation of Why B is Correct and the Others Are Not:

- **Why B is Correct:**
 - Option B offers a balanced solution that addresses both cost-effectiveness and compliance. By reducing the number of data nodes and adding UltraWarm nodes, the company can handle the required data volume more efficiently, as UltraWarm nodes are designed for less frequent access and are more cost-effective for long-term storage. Transitioning indexes to UltraWarm after ingestion optimizes performance during the active analysis period. Using an S3 Lifecycle policy to move data to Glacier Deep Archive after one month aligns with the compliance need to retain data, while also reducing storage costs compared to S3 Standard.
- **Why A is Incorrect:**
 - Transitioning input data directly to S3 Glacier Deep Archive as soon as it's loaded into the OpenSearch cluster may not be feasible if the data needs to be readily accessible for analysis. Moreover, replacing all data nodes with UltraWarm nodes might not offer the required performance for active analysis.
- **Why C is Incorrect:**

- Adding cold storage nodes to the cluster in addition to UltraWarm nodes can lead to unnecessary complexity and might not be cost-effective. Deleting the original data from the S3 bucket after one month contradicts the compliance requirement to retain a copy of all input data.
- **Why D is Incorrect:**
 - Using instance-backed data nodes instead of UltraWarm nodes could result in higher operational costs, as instance-backed nodes are typically used for more active data. Transitioning input data directly to Glacier Deep Archive upon upload may hinder immediate analysis requirements.

Question 44

Within an organization utilizing AWS Organizations, a company operates 10 accounts, each equipped with AWS Config. These accounts are categorized under either the Production (Prod) or Non-Production (NonProd) Organizational Units (OUs). Currently, each account has an Amazon EventBridge rule that alerts an Amazon SNS topic whenever an Amazon EC2 security group inbound rule is created with the source IP range 0.0.0.0/0. The security team is subscribed to this SNS topic. The company now seeks to prohibit the creation of EC2 security group inbound rules with the source IP 0.0.0.0/0 in all NonProd OU accounts, aiming for a solution with minimal operational complexity. What approach should be taken to achieve this?

- A. Adjust the EventBridge rule in the NonProd OU accounts to trigger an AWS Lambda function, which will remove the security group inbound rule and then send a notification to the SNS topic.
- B. Implement the 'vpc-sg-open-only-to-authorized-ports' AWS Config managed rule across all accounts within the NonProd OU.
- C. Create a Service Control Policy (SCP) that permits the 'ec2:AuthorizeSecurityGroupIngress' action only when the 'aws:SourceIp' condition key does not match 0.0.0.0/0. Apply this SCP to the NonProd OU.
- D. Establish an SCP that explicitly denies the 'ec2:AuthorizeSecurityGroupIngress' action when the 'aws:SourceIp' condition key is set to 0.0.0.0/0. Apply this SCP to all accounts within the NonProd OU.

Correct Answer: **D**

Explanation of Why D is Correct and the Others Are Not:

- **Why D is Correct:**
 - Creating an SCP that specifically denies the action 'ec2:AuthorizeSecurityGroupIngress' when 'aws:SourceIp' is 0.0.0.0/0 directly addresses the requirement by preventing the

creation of such security group rules. Applying this SCP to the NonProd OU ensures that this restriction is uniformly enforced across all accounts within that OU. This approach provides a centralized, enforceable policy with minimal operational overhead, as it does not require any additional monitoring or intervention once the SCP is in place.

- **Why A is Incorrect:**
 - Modifying the EventBridge rule to invoke a Lambda function for removing the rule and sending notifications introduces additional complexity and operational tasks. This approach reacts after the rule is created, rather than preventing its creation, which does not align with the proactive security requirement.
- **Why B is Incorrect:**
 - The 'vpc-sg-open-only-to-authorized-ports' AWS Config rule is designed to check whether security groups allow unrestricted access to specified ports, not to prevent the creation of security group rules. Additionally, this rule would only evaluate the configuration after creation, which is reactive rather than preventive.
- **Why C is Incorrect:**
 - An SCP that allows 'ec2:AuthorizeSecurityGroupIngress' except for a specific 'aws:SourceIp' does not effectively prevent the creation of rules with 0.0.0.0/0. Instead, it allows all other configurations, which does not strictly enforce the desired security policy.

Question 45

A company currently operates a Git repository in its local data center and uses webhooks to trigger processes within the AWS Cloud. These webhooks are processed by logic hosted on Amazon EC2 instances within an Auto Scaling group, connected to an Application Load Balancer (ALB). The Git server interacts with this ALB to execute webhook-related tasks. The company is now looking to transition this setup to a serverless architecture, aiming for a solution that minimizes ongoing management efforts. What serverless approach should be adopted to effectively meet these needs?

- A. Set up a unique AWS Lambda function URL for each webhook and reconfigure the on-premises Git server to directly invoke these Lambda function URLs.
- B. Establish an Amazon API Gateway HTTP API and associate each webhook with a dedicated AWS Lambda function. Redirect webhook calls from the Git server to this new API Gateway endpoint.
- C. Migrate the webhook logic to AWS App Runner and configure an ALB to route traffic to the App Runner service. Modify the Git server to direct webhook requests to the updated ALB endpoint.

D. Convert the webhook logic into containerized applications and deploy them on Amazon Elastic Container Service (Amazon ECS) using AWS Fargate. Create an Amazon API Gateway REST API with the ECS service as the backend target and update the Git server to call this new API Gateway endpoint.

Correct Answer: **B**

Explanation of Why B is Correct and the Others Are Not:

- **Why B is Correct:**
 - Implementing Amazon API Gateway with separate AWS Lambda functions for each webhook is a highly scalable and serverless approach. API Gateway acts as a single entry point for all webhook requests, efficiently routing each request to the corresponding Lambda function based on the configured routing rules. This solution significantly reduces operational overhead by eliminating the need to manage EC2 instances, Auto Scaling groups, or ALBs, and it leverages the fully managed nature of both API Gateway and Lambda.
- **Why A is Incorrect:**
 - While using AWS Lambda function URLs is a serverless approach, it lacks the centralized management and routing capabilities provided by API Gateway. Directly calling individual Lambda function URLs for each webhook can lead to a more fragmented architecture and might complicate webhook management.
- **Why C is Incorrect:**
 - AWS App Runner is a service for running containerized applications without requiring in-depth infrastructure management. However, it is not fully serverless in the same way as Lambda and API Gateway, as it still involves managing containers. Additionally, using an ALB in this context does not reduce operational overhead compared to the existing EC2-based solution.
- **Why D is Incorrect:**
 - Containerizing the webhook logic and running it on Amazon ECS with AWS Fargate, while more managed than traditional EC2 instances, still involves container management. Furthermore, setting up an API Gateway REST API to communicate with Fargate adds complexity and does not fully leverage the serverless capabilities that AWS offers.

Question 46

A company is preparing to shift 1,000 servers from its local data centers, which currently operate on multiple VMware clusters, to AWS. Part of this migration strategy involves collecting detailed metrics from these servers, including CPU specifications, memory utilization, operating system details, and active processes. The company aims to not only gather this data but also to

perform analysis and querying on it. What approach should the company adopt to effectively gather, store, and analyze this server information within the AWS ecosystem?

- A. Install the AWS Agentless Discovery Connector as a virtual appliance on the existing on-premises servers. Activate Data Exploration in AWS Migration Hub, utilize AWS Glue for ETL operations on the collected data, and perform queries using Amazon S3 Select.
- B. Manually extract the performance metrics of the virtual machines from the current infrastructure. Import this data directly into AWS Migration Hub and fill in any gaps as needed. Use Amazon QuickSight to analyze and query the information.
- C. Develop a custom script to automatically retrieve server data from the on-premises environment. Employ the AWS CLI to execute the 'put-resource-attributes' command, uploading the detailed server information to AWS Migration Hub. Access and query the data through the Migration Hub interface.
- D. Implement the AWS Application Discovery Agent on each of the on-premises servers. Enable Data Exploration within AWS Migration Hub and use Amazon Athena for conducting predefined queries on the data stored in Amazon S3.

Correct Answer: **D**

Explanation of Why D is Correct and the Others Are Not:

- **Why D is Correct:**
 - Deploying the AWS Application Discovery Agent on each server allows for comprehensive data collection, covering all the required metrics. Integrating this with Data Exploration in AWS Migration Hub provides a centralized view of the gathered data. Amazon Athena is then a suitable tool for querying this data as it is stored in Amazon S3, offering flexibility and powerful data analysis capabilities without the need for additional data transformation steps.
- **Why A is Incorrect:**
 - While the AWS Agentless Discovery Connector can collect some infrastructure information, it might not gather as detailed data as the Application Discovery Agent. Moreover, using AWS Glue for ETL and S3 Select for querying adds unnecessary complexity and might not provide the same level of detailed analysis compared to using Athena.
- **Why B is Incorrect:**
 - Manually exporting VM performance data and updating AWS Migration Hub is not only labor-intensive but also prone to errors and omissions. Amazon QuickSight is effective for visualization and analysis, but this approach lacks the automated, comprehensive data collection that the Discovery Agent provides.

- **Why C is Incorrect:**
 - Creating a custom script for data collection is a more complex and error-prone approach compared to using AWS's pre-built tools like the Application Discovery Agent. Additionally, querying data directly in the Migration Hub console may not offer the same depth of analysis and querying capabilities as Athena.

Question 47

A company is developing a serverless application that operates on AWS Lambda within a Virtual Private Cloud (VPC). The application needs to connect to a service offered by an external provider, which only accepts incoming requests from specific public IPv4 addresses listed in their allow list. The company has to provide a consistent public IP address to the external provider to enable the connection from their application. What approach should the company adopt to ensure that their serverless application can reliably access this external service?

- A. Implement a Network Address Translation (NAT) gateway within the VPC, assigning it a stable Elastic IP address. Adjust the VPC's routing to direct outbound traffic through this NAT gateway.
- B. Set up an egress-only internet gateway in the VPC and link it to an Elastic IP address. Modify the Lambda function's elastic network interface to route its traffic via this egress-only internet gateway.
- C. Introduce an internet gateway to the VPC and attach an Elastic IP address to it. Directly configure the Lambda function to use this internet gateway for its network communications.
- D. Install an internet gateway in the VPC and connect it to an Elastic IP address. Alter the VPC's public route table to route traffic through this internet gateway.

Correct Answer: **A**

Explanation of Why A is Correct and the Others Are Not:

- **Why A is Correct:**
 - A NAT gateway allows resources in a private subnet, such as a Lambda function in a VPC, to initiate outbound traffic to the internet while maintaining a static public IP address through an Elastic IP association. This setup enables the Lambda function to communicate with the external service using a consistent public IP address, fulfilling the requirement set by the external provider. The NAT gateway effectively masks the private IP addresses of the internal resources, providing controlled access to external services.
- **Why B is Incorrect:**
 - An egress-only internet gateway is designed specifically for IPv6 traffic and does not support the association of Elastic IP addresses, which are IPv4. Additionally, AWS

Lambda functions cannot be configured to explicitly use an egress-only internet gateway.

- **Why C is Incorrect:**
 - Directly configuring a Lambda function to use an internet gateway is not possible, as Lambda functions in a VPC do not have the capability to be assigned public IP addresses. Internet gateways are used for routing traffic to and from the internet for resources in public subnets, not for providing a static public IP for outbound connections.
- **Why D is Incorrect:**
 - While adding an internet gateway and updating the route table enables resources in public subnets to access the internet, it does not address the requirement for a single, consistent public IP address. This setup would not provide the static IP address needed for the external service's allow list.

Question 48

A web application, designed by a solutions architect, incorporates an Amazon API Gateway Regional endpoint and an AWS Lambda function. The target audience for this application is located near the AWS Region of deployment. The Lambda function's primary role is to query data from an Amazon Aurora MySQL database, which is configured with three read replicas. However, during high-traffic periods, the application struggles with performance, primarily due to a surge in the number of database connections. The solutions architect is tasked with enhancing the application's performance while addressing the issue of numerous database connections. What two measures should the architect implement to achieve this improvement?

- A. Opt for the cluster endpoint when connecting to the Aurora database.
- B. Implement RDS Proxy to manage a connection pool targeting the reader endpoint of the Aurora database.
- C. Activate the Lambda Provisioned Concurrency feature.
- D. Refactor the Lambda function by placing the database connection initiation code outside the main event handler function.
- E. Switch the API Gateway endpoint from a regional to an edge-optimized type.

Correct Answers: **B, D**

Explanation of Why B and D Are Correct and the Others Are Not:

- **Why B is Correct:**

- RDS Proxy is specifically designed to manage database connections efficiently, especially under high load. By setting up a connection pool to the reader endpoint of the Aurora database, RDS Proxy can significantly reduce the number of direct connections to the database. This helps in managing spikes in traffic more efficiently, improving application performance.
- **Why D is Correct:**
 - In AWS Lambda, moving the code for opening database connections outside the event handler (i.e., creating a persistent connection) can enhance performance. This approach utilizes Lambda's execution context reuse, meaning the database connection can be reused for subsequent invocations of the same function, reducing the overhead of establishing a new connection for each function execution.
- **Why A is Incorrect:**
 - Using the cluster endpoint primarily directs traffic to the primary instance and is not suitable for read-heavy workloads. This option does not address the issue of too many open connections and may not lead to the desired performance improvement.
- **Why C is Incorrect:**
 - While Lambda Provisioned Concurrency can improve performance by keeping a specified number of function instances warm, it does not directly address the issue of numerous database connections, which is the primary cause of the performance bottleneck in this scenario.
- **Why E is Incorrect:**
 - Changing the API Gateway endpoint to an edge-optimized endpoint is beneficial for latency reduction in scenarios where the consumers are globally distributed. However, as the application's consumers are close to the AWS Region of deployment, this change would not significantly impact the performance issue related to database connections.

Question 49

A company aims to deploy a web application on AWS, requiring the distribution of web traffic across several Amazon EC2 instances. A critical security mandate for this deployment is to maintain encryption of data in transit from the client all the way to the web servers. What is the most appropriate solution to ensure end-to-end encryption for this web application?

- A. Configure an Application Load Balancer (ALB) in front of the EC2 instances. Use AWS Certificate Manager (ACM) to issue an SSL certificate and attach it to the ALB. Attempt to export this SSL certificate for installation on each EC2 instance. Set the ALB to receive traffic on port 443 and reroute it to port 443 on the EC2 instances.
- B. Link the EC2 instances to a target group. Obtain an SSL certificate from AWS Certificate Manager (ACM) and integrate it with an Amazon CloudFront distribution. Designate the

CloudFront distribution to interact with the target group as its source.

C. Deploy the EC2 instances behind an Application Load Balancer (ALB). Secure an SSL certificate from AWS Certificate Manager (ACM) and assign it to the ALB. Acquire a separate third-party SSL certificate and install it on every EC2 instance. Configure the ALB to accept connections on port 443 and direct this traffic to port 443 on the instances.

D. Situate the EC2 instances behind a Network Load Balancer (NLB). Install a third-party SSL certificate on both the NLB and each EC2 instance. Adjust the NLB settings to listen on port 443 and forward traffic to port 443 on the EC2 instances.

Correct Answer: **C**

Explanation of Why C is Correct and the Others Are Not:

- **Why C is Correct:**
 - This solution effectively establishes end-to-end encryption. The ALB handles the initial SSL termination from the client, and the third-party SSL certificate installed on each EC2 instance ensures that the traffic remains encrypted as it travels from the ALB to the EC2 instances. This dual-layer of encryption (SSL certificate at ALB for client-to-ALB traffic and third-party SSL certificate on EC2 instances for ALB-to-instance traffic) fulfills the requirement of maintaining encryption throughout the entire data transit path.
- **Why A is Incorrect:**
 - While the ALB setup is correct for handling SSL termination, AWS Certificate Manager (ACM) does not support exporting certificates for installation on EC2 instances. Therefore, this setup fails to ensure end-to-end encryption as required.
- **Why B is Incorrect:**
 - Amazon CloudFront with an ACM SSL certificate can provide encryption from the client to the CloudFront distribution. However, CloudFront primarily serves as a content delivery network and may not be the optimal choice for load balancing EC2 instances. Additionally, this setup does not explicitly ensure encryption from CloudFront to the EC2 instances.
- **Why D is Incorrect:**
 - Network Load Balancers (NLBs) operate at the transport layer and do not handle SSL termination. While installing a third-party SSL certificate on the NLB and EC2 instances might seem like it provides end-to-end encryption, the NLB itself does not decrypt and re-encrypt traffic, so it cannot ensure the required encryption between the client and the EC2 instances through the load balancer.

Question 50

A company is transitioning its on-premises data analytics platform to AWS. This platform includes two Node.js applications: one for gathering sensor data into a MySQL database, and another for generating data reports through aggregation processes. Currently, when aggregation tasks are executed, some data collection jobs encounter issues. The company seeks a solution to rectify these data loading problems while ensuring a seamless migration experience for their customers, without any disruptions or noticeable changes.

A. Establish an Amazon Aurora MySQL database connected to the on-site database through replication. Set up an Aurora Replica of this database and shift the data aggregation tasks to this replica. Construct AWS Lambda functions for data collection, accessible via a Network Load Balancer (NLB), and utilize Amazon RDS Proxy for writing to the Aurora MySQL database. After synchronizing the databases, deactivate the replication, switch the Aurora Replica to the primary role, and redirect the data collection DNS to the NLB.

B. Deploy an Amazon Aurora MySQL database. Utilize AWS Database Migration Service (AWS DMS) for ongoing replication from the local database to the Aurora database. Redirect aggregation operations to the Aurora MySQL database. Configure data collection endpoints on Amazon EC2 instances within an Auto Scaling group behind an Application Load Balancer (ALB). Once synchronization is complete, update the data collection DNS to the ALB and stop the AWS DMS replication.

C. Implement an Amazon Aurora MySQL database. Employ AWS Database Migration Service (AWS DMS) for continuous replication from the on-premises database to Aurora. Create an Aurora Replica and transfer the data aggregation tasks to this replica. Establish data collection endpoints using AWS Lambda functions behind an Application Load Balancer (ALB), with Amazon RDS Proxy managing writes to Aurora MySQL. After achieving database synchronization, update the data collection DNS to the ALB and discontinue the AWS DMS replication.

D. Configure an Amazon Aurora MySQL database. Create an Aurora Replica for this database and allocate the aggregation jobs to it. Set up data collection points using an Amazon Kinesis data stream and replicate data to Aurora MySQL via Amazon Kinesis Data Firehose. Once the databases are aligned, stop the replication process, convert the Aurora Replica to the primary instance, and adjust the data collection DNS to target the Kinesis data stream.

Correct Answer: **C**

Explanation of Why C is Correct and the Others Are Not:

- **Why C is Correct:**
 - This solution effectively balances the needs for continuous replication, minimal interruption, and load management. AWS DMS ensures real-time data replication to Aurora MySQL without disrupting the existing on-premises setup. The Aurora Replica

offloads the aggregation workload, ensuring the primary database's performance isn't affected. AWS Lambda functions behind an ALB provide a scalable and serverless approach for data collection, and Amazon RDS Proxy optimizes database writes by managing connections efficiently. After synchronization, the seamless cutover to AWS is achieved by updating the DNS, ensuring uninterrupted service for customers.

- **Why A is Incorrect:**

- While this option includes a replication approach and the use of an Aurora Replica, the use of an NLB with AWS Lambda functions is not standard practice and might not provide the desired results. Additionally, the process of disabling replication and restarting the Aurora Replica as the primary instance could lead to service interruptions, which doesn't align with the requirement for a seamless migration.

- **Why B is Incorrect:**

- This option also utilizes AWS DMS for replication and moves aggregation to Aurora. However, the use of EC2 instances behind an ALB for data collection might not be as scalable and efficient as using serverless AWS Lambda functions. Additionally, the process of disabling the DMS task after cutover might not be as seamless as in Option C.

- **Why D is Incorrect:**

- Incorporating Amazon Kinesis for data collection introduces unnecessary complexity for the given scenario. While Kinesis Data Firehose can handle data streaming well, it's not required for this use case, and the use of a Kinesis data stream as a collection endpoint may not align with the existing application architecture. This setup could also potentially disrupt the current user experience.

Question 51

A health insurance company maintains an Amazon S3 bucket to store sensitive personal data (PII). Currently, the data is encrypted using server-side encryption with S3 managed keys (SSE-S3). However, a new policy mandates that all existing and new data in the bucket be encrypted using keys controlled by the company's security team. The bucket does not have versioning enabled. What method should be used to comply with this new encryption requirement?

A. Update the S3 bucket settings to employ SSE-S3 encryption with a key managed by the company. Re-upload all data in the bucket using the AWS Command Line Interface (CLI). Implement a bucket policy that blocks any upload requests (PutObject) that do not use encryption.

B. Modify the S3 bucket settings to use server-side encryption with AWS Key Management Service (KMS) managed keys (SSE-KMS). Enforce a bucket policy that prohibits uploads without encryption. Re-upload all existing data in the bucket via the AWS CLI.

C. Change the S3 bucket settings to server-side encryption with AWS KMS managed keys (SSE-KMS). Implement a bucket policy that automatically applies encryption to both uploads (PutObject) and downloads (GetObject) of data.

D. Switch the S3 bucket's default encryption to AES-256 with a key managed by the company. Apply a policy that denies unencrypted upload requests to any users accessing the bucket. Re-upload all existing objects in the bucket using the AWS CLI.

Correct Answer: **B**

Explanation of Why B is Correct and the Others Are Not:

- **Why B is Correct:**
 - Option B meets the requirement by changing the encryption method to SSE-KMS, which allows the company's security team to manage the encryption keys via AWS KMS. This offers greater control and auditing capabilities over the encryption keys compared to SSE-S3. The re-upload of all objects ensures that existing data is encrypted with the new KMS keys. The bucket policy that denies unencrypted uploads (PutObject requests) enforces the policy for all new objects, ensuring compliance with the new encryption standard.
- **Why A is Incorrect:**
 - While this option includes re-uploading objects and setting a bucket policy, SSE-S3 with a customer-managed key is not a valid S3 encryption option. The valid options are SSE-S3 (where S3 manages the keys), SSE-C (where the customer manages the keys and provides them in each request), and SSE-KMS (where AWS KMS manages the keys).
- **Why C is Incorrect:**
 - S3 bucket policies cannot automatically encrypt objects on GetObject and PutObject requests. Encryption needs to be specified at the time of object creation (PutObject). Therefore, this option does not provide a feasible solution.
- **Why D is Incorrect:**
 - AES-256 with a customer-managed key as described is not a standard encryption option offered by S3. The available options are SSE-S3, SSE-C, and SSE-KMS. Also, the policy description does not align with standard S3 bucket policy capabilities.

Question 52

A company has deployed a dynamic web application on AWS, comprising Amazon EC2 instances within an Auto Scaling group linked to an Application Load Balancer (ALB). This setup is globally distributed using an Amazon CloudFront distribution, which has the ALB set as its origin. Amazon Route 53 is used for DNS management with an A record pointing to the CloudFront distribution at www.example.com. The company now seeks to enhance this

application's availability and fault tolerance. What configuration should be applied to achieve high availability and fault tolerance for this application?

A. Establish a complete secondary version of the application in another AWS Region. Modify the Route 53 A record to function as a failover record, including both CloudFront distributions. Implement Route 53 health checks for monitoring.

B. Set up a new ALB, Auto Scaling group, and EC2 instances in a separate AWS Region. Adapt the existing CloudFront distribution by adding a new origin linked to the new ALB. Form an origin group with these two origins, designating one as the primary and the other as the secondary.

C. Deploy a new Auto Scaling group and EC2 instances in a different AWS Region. Configure a second target in the ALB for the new Auto Scaling group. Employ the failover routing feature on the ALB.

D. Implement an entirely separate application setup in a different AWS Region. Create an additional CloudFront distribution and set this new application deployment as its origin. Utilize AWS Global Accelerator to manage both CloudFront distributions as endpoints.

Correct Answer: **B**

Explanation of Why B is Correct and the Others Are Not:

- **Why B is Correct:**
 - This solution effectively ensures high availability and fault tolerance by having a backup ALB, Auto Scaling group, and EC2 instances in a different AWS Region. By adding the new ALB as a secondary origin in the existing CloudFront distribution and creating an origin group, CloudFront can seamlessly switch between the two origins based on availability. This setup provides a robust mechanism for fault tolerance without the need for additional DNS-level configurations, ensuring that the application remains accessible even if one region experiences issues.
- **Why A is Incorrect:**
 - While setting up a secondary application deployment and using Route 53 for failover is a viable approach, it does not integrate well with the existing CloudFront distribution setup. Managing failover at the DNS level with Route 53 does not leverage CloudFront's capabilities to seamlessly switch between origins.
- **Why C is Incorrect:**
 - ALBs do not support failover routing algorithms across different regions. Therefore, setting up an additional Auto Scaling group and EC2 instances in a different region and adding them as a target in the ALB would not address the requirement for cross-regional high availability and fault tolerance.

- **Why D is Incorrect:**
 - Creating a separate CloudFront distribution and using AWS Global Accelerator introduces unnecessary complexity and redundancy. Global Accelerator is primarily used to optimize global traffic routing to AWS endpoints, which CloudFront already effectively manages. This approach also does not utilize the existing CloudFront distribution efficiently.

Question 53

A large company with numerous AWS accounts within AWS Organizations utilizes one account as a transit account, containing a transit gateway that connects to all other accounts. This setup also includes Site-to-Site VPN connections linking the company's worldwide offices to the transit account. AWS Config is active across all accounts. The company's network team requires a centralized method to manage and distribute a list of internal IP addresses, representing their global offices, to developers who need secure access to their applications. What approach can achieve this with minimal ongoing management efforts?

- A. Construct a JSON file with all internal IP addresses and store it in an Amazon S3 bucket. Set up an Amazon SNS topic in each account that triggers whenever the JSON file is updated. This topic should activate an AWS Lambda function that updates the security group rules with the new IP addresses in each account.
- B. Implement an AWS Config managed rule in each account that encompasses all internal IP address ranges. This rule should audit the compliance of security groups with the listed IP ranges and automatically correct any security groups found to be non-compliant.
- C. In the designated transit account, generate a VPC prefix list that includes all internal IP addresses. Share this prefix list with the other accounts using AWS Resource Access Manager. Leverage this shared prefix list to set up security group rules in the various accounts.
- D. Create a security group within the transit account that lists all internal IP addresses. In the other accounts, configure security groups to refer to the transit account's security group through a nested security group reference system.

Correct Answer: **C**

Explanation of Why C is Correct and the Others Are Not:

- **Why C is Correct:**
 - Creating a VPC prefix list in the transit account and sharing it across other accounts using AWS Resource Access Manager offers a centralized, efficient way to manage IP address ranges. This method allows security groups in all accounts to reference a single, shared list, ensuring consistency and reducing the complexity of updates. Any

changes to the IP addresses need only be made in the transit account's prefix list, automatically propagating to all accounts that use the shared list in their security group configurations.

- **Why A is Incorrect:**
 - While using a JSON file in S3 and SNS topics for updates is a possible solution, it introduces significant operational overhead and complexity. This method requires maintaining Lambda functions in every account to update security groups, which can be cumbersome and prone to errors, especially in large organizations with many accounts.
- **Why B is Incorrect:**
 - AWS Config managed rules are designed for configuration compliance checks and remediation but are not suited for centrally managing and distributing IP address lists. This approach would be more reactive than proactive, focusing on correcting non-compliance rather than efficiently distributing and referencing IP address ranges.
- **Why D is Incorrect:**
 - Security groups in AWS do not support nested references from other accounts. Thus, referencing a security group in the transit account from other accounts is not feasible. This option does not provide a viable mechanism for sharing IP address information across accounts.

Question 54

A company has launched a static website hosted on Amazon S3, accessible via Amazon CloudFront, and featuring a backend powered by Amazon API Gateway and AWS Lambda functions. The company intends to generate a bi-weekly CSV report detailing each Lambda function's memory and cost recommendations, including potential savings, and to store these reports in an S3 bucket. What approach should be adopted to produce these reports with minimal development effort?

A. Develop an AWS Lambda function to gather and analyze performance metrics from Amazon CloudWatch Logs for each API-connected Lambda function over the past two weeks. Format the data into a CSV file and save it in an S3 bucket. Schedule this Lambda function to execute bi-weekly using Amazon EventBridge.

B. Enable AWS Compute Optimizer and create a Lambda function to invoke the `ExportLambdaFunctionRecommendations` API. Configure this function to generate a CSV report and save it to an S3 bucket. Use Amazon EventBridge to trigger this Lambda function every two weeks.

C. Activate AWS Compute Optimizer and enable enhanced infrastructure metrics. From within the Compute Optimizer dashboard, arrange for bi-weekly exports of Lambda function

recommendations as a CSV file, saving these to an S3 bucket.

D. Subscribe to the AWS Business Support plan for the production account and opt into AWS Compute Optimizer through AWS Trusted Advisor. Use Trusted Advisor's interface to schedule bi-weekly exports of cost optimization reports as CSV files, which are then stored in an S3 bucket.

Correct Answer: **B**

Explanation of Why B is Correct and the Others Are Not:

- **Why B is Correct:**
 - Using AWS Compute Optimizer with a Lambda function calling the `ExportLambdaFunctionRecommendations` operation is a streamlined approach to generate the desired reports. Compute Optimizer provides specific recommendations for Lambda memory sizing and cost optimizations. The EventBridge rule to schedule the Lambda function every two weeks automates the process with minimal development time, as it leverages AWS services directly without the need for extensive custom coding or manual intervention.
- **Why A is Incorrect:**
 - While feasible, this option involves significant development effort to extract and process CloudWatch Logs data manually. Creating a custom solution to analyze logs, generate reports, and manage scheduling is more complex and time-consuming compared to leveraging Compute Optimizer.
- **Why C is Incorrect:**
 - AWS Compute Optimizer does not currently offer a feature to automatically schedule and export reports directly from its console. This would require manual intervention every two weeks, which does not meet the requirement for minimal development time.
- **Why D is Incorrect:**
 - While AWS Trusted Advisor provides valuable insights, including cost optimizations, it does not specifically focus on Lambda memory and cost recommendations in the context required here. Additionally, Trusted Advisor does not have a built-in feature to schedule and export such reports automatically, and opting for AWS Business Support adds unnecessary cost without directly addressing the requirement.

Question 55

A company operates its factory and automation applications within a single AWS Virtual Private Cloud (VPC), utilizing Amazon EC2, Amazon ECS, and Amazon RDS services. These applications are managed by three distinct teams, each accountable for the costs and performance of their respective applications. Each application and team is identified by specific tags. Team members utilize IAM for daily operations. The company needs a method to assign monthly AWS costs to

each application or team accurately. Additionally, it requires the capability to generate cost reports for historical analysis and future projections over 12-month periods. The solutions architect is tasked with suggesting an appropriate AWS Billing and Cost Management solution to produce these detailed cost reports. What three steps should be taken to fulfill these requirements?

- A. Enable user-defined cost allocation tags that correspond to each application and team.
- B. Turn on AWS-generated cost allocation tags that represent each application and team.
- C. Establish a distinct cost category for every application within the AWS Billing and Cost Management system.
- D. Provide IAM users access to the AWS Billing and Cost Management service.
- E. Set up a cost budget within AWS.
- F. Activate AWS Cost Explorer to analyze and report on costs.

Correct Answers: **A, C, F**

Explanation of Why A, C, and F Are Correct and the Others Are Not:

- **Why A is Correct:**
 - Activating user-defined cost allocation tags allows the company to track expenses based on specific applications and teams. These tags are essential for attributing costs accurately to each team and application, enabling detailed and accurate cost reporting.
- **Why C is Correct:**
 - Creating a cost category for each application within AWS Billing and Cost Management helps in organizing costs according to each application. This categorization is crucial for detailed cost analysis and attributing expenses to the respective teams and applications.
- **Why F is Correct:**
 - Enabling Cost Explorer is vital for analyzing and reporting AWS costs. It provides the functionality to view historical data and forecast future costs based on current trends, meeting the company's requirement for past and future cost analysis.
- **Why B is Incorrect:**
 - AWS-generated cost allocation tags typically include tags automatically applied by AWS services, like 'aws:createdBy', and might not directly correspond to specific applications or teams as required by the company.
- **Why D is Incorrect:**

- While providing IAM access to Billing and Cost Management is useful for access control, it does not directly contribute to the cost allocation, reporting, and forecasting capabilities required by the company.
- **Why E is Incorrect:**
 - Setting up a cost budget is a tool for monitoring and managing AWS spending, but it does not provide the detailed cost attribution or historical and forecasting analysis needed for this scenario.

Question 56

A customer, currently operating a web application on-premises, wishes to migrate to AWS Cloud. The application relies on fetching data from a third-party API that is secured behind a firewall and only permits connections from a single public CIDR block listed on the client's allow list. The planned AWS setup involves deploying the application on Amazon EC2 instances placed behind an Application Load Balancer (ALB) within a VPC. The ALB will reside in public subnets, and the EC2 instances in private subnets, with NAT gateways facilitating internet access for the private subnets. What approach should a solutions architect recommend to ensure uninterrupted access to the third-party API after migrating the web application to AWS?

- A. Link a customer-owned block of public IP addresses to the VPC and enable the assignment of public IP addresses in the VPC's public subnets.
- B. Import a customer-owned block of public IP addresses into the AWS account. Create Elastic IP addresses from this block and assign them to the VPC's NAT gateways.
- C. Generate Elastic IP addresses from the customer-owned block of public IPs and attach these static Elastic IP addresses directly to the ALB.
- D. Import a customer-owned block of public IP addresses into the AWS account and configure AWS Global Accelerator to use Elastic IPs from this block. Designate the ALB as the endpoint for the accelerator.

Correct Answer: **B**

Explanation of Why B is Correct and the Others Are Not:

- **Why B is Correct:**
 - This option aligns with the requirement to maintain a consistent public IP address that the third-party API recognizes. By importing a customer-owned IP block into AWS and creating Elastic IP addresses for the NAT gateways, the web application's outbound traffic routed through the NAT gateways will consistently come from the known public IP addresses. This setup ensures that the third-party API continues to recognize and allow requests from the migrated application.

- **Why A is Incorrect:**
 - While associating a block of customer-owned public IP addresses with the VPC is possible, simply enabling public IP addressing in the VPC's public subnets does not guarantee a consistent public IP address for outbound requests to the third-party API. It also exposes the instances in public subnets directly to the internet, which may not be desirable.
- **Why C is Incorrect:**
 - ALBs do not support the assignment of Elastic IP addresses. Additionally, even if this were technically feasible, it would address only the inbound traffic to the ALB and not the outbound requests from the application to the third-party API.
- **Why D is Incorrect:**
 - AWS Global Accelerator is primarily used to improve the performance and availability of applications by directing traffic through AWS's global network infrastructure. While it can use static IP addresses, this service is more about managing inbound application traffic rather than controlling outbound requests to a third-party API.

Question 57

A business with multiple AWS accounts is leveraging AWS Organizations and Service Control Policies (SCPs) for governance. An SCP, shown below, has been applied to an Organizational Unit (OU) that includes the AWS account with the ID 1111-1111-1111:

```
1  {
2    "Version": "2012-10-17",
3    "Statement": [
4      {
5        "Sid": "AllowsAllActions",
6        "Effect": "Allow",
7        "Action": "*",
8        "Resource": "*"
9      },
10     {
11       "Sid": "DenyCloudTrail",
12       "Effect": "Deny",
13       "Action": "cloudtrail:*",
14       "Resource": "*"
15     }
16   ]
17 }
```

Developers within the AWS account 1111-1111-1111 have reported that they are unable to create Amazon S3 buckets. How should the administrator rectify this situation?

- A. Modify the SCP by incorporating the s3:CreateBucket action with an "Allow" effect.
- B. Remove the specified account from the OU, then directly attach the SCP to the AWS account 1111-1111-1111.
- C. Guide the developers to add the necessary Amazon S3 permissions to their IAM policies.
- D. Detach the SCP from the AWS account 1111-1111-1111.

Correct Answer: **C**

Explanation of Why C is Correct and the Others Are Not:

- **Why C is Correct:**
 - The SCP in question allows all actions ("Action": "*") except for actions related to AWS CloudTrail ("Action": "cloudtrail:*"), which are explicitly denied. Since the SCP doesn't restrict S3 actions, developers' inability to create S3 buckets is not due to the SCP but likely due to their individual IAM permissions. Therefore, the developers should review and adjust their IAM permissions to include the s3:CreateBucket permission.
- **Why A is Incorrect:**
 - Adding s3:CreateBucket with "Allow" effect to the SCP is unnecessary since the SCP already allows all actions except those related to CloudTrail. The SCP does not prevent S3 bucket creation, so modifying it would not resolve the issue.
- **Why B is Incorrect:**
 - Removing the account from the OU would eliminate any management and control that the SCP provides over the account, potentially affecting other services and compliance. Since the issue is not caused by the SCP, this action would not address the developers' issue and could lead to broader unintended consequences.
- **Why D is Incorrect:**
 - Detaching the SCP from the account would remove all restrictions and controls that the SCP provides, which could have security and governance implications. Moreover, it's unnecessary since the SCP, as it stands, does not block the creation of S3 buckets.

Question 58

A business-critical, monolithic application operates on an Amazon EC2 instance utilizing Amazon Linux 2 and stores data on an encrypted EBS volume. The company's legal team has mandated that the data from the EBS volume be backed up to an S3 bucket. The application team lacks the administrative SSH key for the instance but needs to ensure the application remains available to users while complying with the backup directive.

- A. Assign an IAM role to the EC2 instance granting S3 write access. Utilize AWS Systems Manager Session Manager to log into the instance without SSH and execute commands to transfer the data to S3.
- B. Generate an AMI from the EC2 instance, ensuring the instance reboots during this process. Spin up a new EC2 instance from this AMI, associate an IAM role with S3 write access, and execute a data transfer to S3.
- C. Utilize Amazon Data Lifecycle Manager (DLM) to create a snapshot of the encrypted EBS volume and then transfer this snapshot's data to an S3 bucket.
- D. Create an AMI of the current EC2 instance. Launch a new EC2 instance from this AMI, assign an IAM role with S3 write privileges, and initiate a command to move data into S3.

Correct Answer: **C**

Explanation of Why C is Correct and the Others Are Not:

- **Why C is Correct:**
 - Amazon DLM automates the process of creating snapshots of EBS volumes, which can then be used as a backup. These snapshots can be copied directly to S3, providing a backup solution without the need for SSH access to the instance. This method satisfies the backup requirement while ensuring the application remains available to users.
- **Why A is Incorrect:**
 - While AWS Systems Manager Session Manager allows access to an instance without an SSH key, it would require installing and configuring the Systems Manager agent, and the manual process of copying files to S3 could interrupt the application service. It's also less efficient compared to automated snapshot management.
- **Why B is Incorrect:**
 - Creating an AMI and launching a new instance would involve unnecessary steps and could potentially lead to downtime, which is not acceptable for a business-critical application. Additionally, the reboot could disrupt the service, which violates the requirement that the application must continue serving users.
- **Why D is Incorrect:**
 - Similar to option B, creating an AMI, launching a new instance, and manually transferring data to S3 is an inefficient approach that introduces complexity without any added benefit. It does not align with the goal of minimal disruption to the service.

Question 59

A solutions architect is tasked with transferring data from an S3 bucket located in one AWS account to a new S3 bucket in a different AWS account. The transfer must be carried out using

the AWS Command Line Interface (CLI). What steps should be taken to ensure a successful data transfer involving these two S3 buckets across separate AWS accounts?

- A. Formulate a bucket policy that permits the originating bucket to enumerate its files and to upload objects and modify object access control lists (ACLs) in the target bucket. Apply this policy to the target bucket.
- B. Construct a bucket policy that enables a user from the target AWS account to list the contents of the original bucket and read its objects. Apply this policy to the original bucket.
- C. Generate an IAM policy within the original account allowing a user from the same account to list the original bucket, retrieve objects, and in the target bucket, list, upload objects, and modify object ACLs. Link this policy to the user in the original account.
- D. Draft an IAM policy in the new account authorizing a user from this account to list and retrieve objects from the original bucket, and to list, upload objects, and adjust object ACLs in the target bucket. Attach this policy to the user in the new account.
- E. Execute the `aws s3 sync` command logged in as a user from the original account. Designate the originating and target buckets for the data transfer.
- F. Use the `aws s3 sync` command while authenticated as a user in the new account. Define the original and target buckets for the data transfer process.

Correct Answers: **B, D, F**

Explanation of Why B, D, and F Are Correct and the Others Are Not:

- **Why B is Correct:**
 - A bucket policy that grants a user from the target account permission to list and read objects from the source bucket is essential. This enables the user in the new account to access the data in the original bucket.
- **Why D is Correct:**
 - An IAM policy in the new account that provides permissions to access the source bucket and to write to the target bucket is required. This policy gives the new account's user the necessary permissions to carry out the data transfer.
- **Why F is Correct:**
 - Running the `aws s3 sync` command as a user in the new account is the appropriate action, as this user will have been granted the necessary permissions to read from the source and write to the target bucket as per the policies defined in options B and D.
- **Why A is Incorrect:**
 - A bucket policy attached to the target bucket allowing actions from the source bucket is not necessary. Permissions need to be granted to users, not buckets, for cross-

account access.

- **Why C is Incorrect:**
 - Creating an IAM policy in the source account for a user in the same account is unnecessary because the data transfer is being initiated from the new account, not the original one. The source account's user does not need permissions for the target bucket.
- **Why E is Incorrect:**
 - Executing the `aws s3 sync` command as a user in the source account does not align with the need for cross-account access. The source account's user typically does not have permissions to write to the new account's bucket.

Question 60

A company has experienced an outage due to a problematic release of their web application, which is powered by AWS Lambda and managed through AWS CloudFormation. To mitigate such issues in the future, a solutions architect has been tasked with modifying the deployment strategy to incorporate canary releases, a technique that allows new versions to be rolled out gradually.

- A. Generate a new alias for each Lambda function version released. Utilize the AWS CLI command `update-alias` with the `routing-config` option to incrementally shift traffic to the new version.
- B. Launch each new application release using a separate CloudFormation stack. Employ Amazon Route 53's weighted routing policy to incrementally adjust traffic to the new stack.
- C. For each update, create a distinct version of the Lambda function. Implement the AWS CLI command `update-function-configuration` with the `routing-config` option to manage traffic distribution between versions.
- D. Employ AWS CodeDeploy and specify the `CodeDeployDefault.OneAtATime` deployment configuration to progressively route traffic to the updated Lambda function.

Correct Answer: **A**

Explanation of Why A is Correct and the Others Are Not:

- **Why A is Correct:**
 - AWS Lambda aliases can be used to point to different function versions and control the percentage of traffic that each version receives. Using the `update-alias` AWS CLI command with the `routing-config` parameter allows for the gradual rollout of new versions, enabling a canary deployment strategy. This approach provides a controlled

way to introduce and monitor new changes before fully adopting them, which is ideal for mitigating potential outages caused by new releases.

- **Why B is Incorrect:**
 - Deploying a new CloudFormation stack for each release and using Route 53's weighted routing is more complex and is not the standard method for canary deployments of Lambda functions. This approach is more suited to situations where you have immutable infrastructure and need complete environment isolation.
- **Why C is Incorrect:**
 - The `update-function-configuration` AWS CLI command is used for updating configuration details of a Lambda function, not for routing traffic between different versions. Therefore, it does not support the `routing-config` parameter for traffic distribution.
- **Why D is Incorrect:**
 - While AWS CodeDeploy does support canary deployments, the `CodeDeployDefault.OneAtATime` deployment configuration mentioned here is not designed for canary releases. Instead, it is used to ensure that the deployment happens to one instance at a time, which is a different deployment strategy.

Question 61

A financial institution operates a data repository within Amazon S3 and nightly acquires financial records through SFTP from various external entities. Currently, the institution maintains a self-managed SFTP server hosted on an Amazon EC2 instance situated in a VPC's public subnet. Files received via SFTP are transferred to the data lake by a scheduled cron job on the same server. Access to the SFTP server is facilitated by the DNS entry `sftp.example.com` managed by Amazon Route 53. The institution seeks to enhance the dependability and scalability of their SFTP infrastructure.

- A. Incorporate the existing EC2-based SFTP server into an Auto Scaling group and route traffic through an Application Load Balancer (ALB). Modify the Route 53 DNS entry for `sftp.example.com` to direct traffic to the ALB.
- B. Transition from the self-managed SFTP server to the managed AWS Transfer for SFTP service. Redirect the `sftp.example.com` DNS entry in Route 53 to the AWS Transfer for SFTP server endpoint.
- C. Replace the self-hosted SFTP service with an AWS Storage Gateway file gateway for SFTP transactions. Update the Route 53 DNS record `sftp.example.com` to resolve to the file gateway's endpoint.
- D. Deploy a Network Load Balancer (NLB) to handle traffic to the SFTP server hosted on the EC2 instance. Change the `sftp.example.com` DNS record in Route 53 to point towards the NLB.

Correct Answer: **B**

Explanation of Why B is Correct and the Others Are Not:

- **Why B is Correct:**
 - AWS Transfer for SFTP is a managed service that handles SFTP traffic directly to and from Amazon S3 without the need to run and manage EC2 instances. By migrating to AWS Transfer for SFTP and updating the DNS record to point to the service endpoint, the company can achieve greater reliability and scalability without the overhead of managing server infrastructure.
- **Why A is Incorrect:**
 - While an Auto Scaling group can improve the scalability of EC2 instances, SFTP does not work natively with ALBs due to its requirement for maintaining stateful connections. This setup would not provide the necessary reliability or scalability improvements for SFTP transfers.
- **Why C is Incorrect:**
 - AWS Storage Gateway's file gateway is designed to integrate on-premises environments with cloud storage, providing a file system interface to S3. It is not intended to replace an SFTP server and does not natively support SFTP protocols.
- **Why D is Incorrect:**
 - A Network Load Balancer can handle TCP traffic, and could theoretically improve the availability of the SFTP server. However, NLBs are designed for stateless traffic and might not properly handle the stateful nature of SFTP connections. Additionally, this solution doesn't inherently provide scalability improvements and still requires managing EC2 instances.

Question 62

A corporation seeks to relocate an application hosted on an on-premises VMware infrastructure to an Amazon EC2 environment, with a key requirement being the retention of the application's existing software and configurations. What steps should the solutions architect take to ensure a seamless transition that maintains the integrity of the application's setup?

- A. Set up AWS DataSync to synchronize the VMware data store with Amazon FSx for Windows File Server using the SMB protocol. Subsequently, migrate the virtual machines to Amazon EC2 using the VM Import/Export service.
- B. Utilize the VMware vSphere interface to convert the application into an OVF image. Create an S3 bucket within the target AWS Region to store this image. Assign an IAM role equipped for VM Import tasks and execute the EC2 import operation using the AWS Command Line Interface.

C. Implement AWS Storage Gateway as a file gateway to establish a CIFS share. Generate a backup to this CIFS share, then go to the AWS Console to create an Amazon Machine Image (AMI) from this backup. Proceed to initialize an EC2 instance from this newly created AMI.

D. Arrange for a hybrid environment activation through AWS Systems Manager. Install the Systems Manager Agent on the VMware-based virtual machine and register it as a managed instance. Use AWS Backup to capture a snapshot of the virtual machine and generate an AMI. Start an EC2 instance from this AMI.

Correct Answer: **B**

Explanation of Why B is Correct and the Others Are Not:

- **Why B is Correct:**
 - Exporting the application VM as an OVF file is a recognized method for capturing the entire state of the VM, including its software and configurations. Storing the image in S3 and using the AWS VM Import service to bring the VM into EC2 is a direct migration path that preserves the application's integrity.
- **Why A is Incorrect:**
 - While AWS DataSync can replicate data stores, and FSx for Windows File Server could theoretically be used to host VMware data stores, VM Import/Export does not work directly with SMB shares or FSx, making this option unsuitable for a direct migration.
- **Why C is Incorrect:**
 - AWS Storage Gateway's file gateway does not natively support the migration of VMware VMs. Creating an AMI from a backup copy on CIFS share does not ensure the VM's configuration and software are preserved accurately in the EC2 environment.
- **Why D is Incorrect:**
 - AWS Systems Manager is not typically used for migrating VMs from VMware to EC2. While it can manage hybrid environments, it does not facilitate the direct import of VMs into EC2. Additionally, AWS Backup cannot create snapshots of on-premises VMs to create AMIs directly.

Question 63

A company specializing in video processing has a Node.js application that automatically downloads and processes new images uploaded to an S3 bucket, saves the processed images to another S3 bucket, and logs image metadata in a DynamoDB table. This process is triggered by an AWS Lambda function set to maximum timeout settings. Recently, the application has started to fail due to the increased size of the incoming images, leading to timeout errors. The company seeks to redesign the application's architecture to eliminate these errors without the burden of managing servers.

- A. Update the deployment process to include the creation of a Docker container image encapsulating the application's code. Push this Docker image to Amazon ECR.
- B. Establish a new ECS task definition with AWS Fargate compatibility. Use this task definition and the image from ECR to run tasks in ECS. Modify the Lambda function to trigger an ECS Fargate task when new images are uploaded to the S3 bucket.
- C. Implement an AWS Step Functions state machine with a Parallel state to call the Lambda function and increase the provisioned concurrency settings for the Lambda.
- D. Introduce a new ECS task definition that is compatible with Amazon EC2 instances. Use this task definition with the ECR image to run tasks in ECS. Change the Lambda function so it starts an ECS EC2 task upon the arrival of new images in the S3 bucket.
- E. Refactor the application to save images on Amazon EFS and to record metadata in an Amazon RDS database instance. Update the Lambda function to connect to the EFS share for file access.

Correct Answers: **A, B**

Explanation of Why A and B Are Correct and the Others Are Not:

- **Why A is Correct:**
 - Packaging the application into a Docker image and storing it in Amazon ECR is a modern approach to application deployment. It allows for portability and is a prerequisite step for running the application on AWS Fargate, which is a serverless compute engine for containers and can handle larger workloads that exceed Lambda's maximum timeout.
- **Why B is Correct:**
 - ECS with AWS Fargate compatibility allows for running containerized applications without managing servers or clusters. By modifying the Lambda function to trigger an ECS Fargate task, it can handle larger payloads that Lambda cannot, thus overcoming the timeout issues.
- **Why C is Incorrect:**
 - AWS Step Functions can orchestrate AWS services, but the issue is not with parallelism or concurrency; it's with the timeout limitations of Lambda. Step Functions would not address the core problem of Lambda timing out due to large image sizes.
- **Why D is Incorrect:**
 - While ECS with EC2 compatibility is a valid container management solution, it requires managing EC2 instances. The company specifically stated it does not want to manage the underlying infrastructure, which makes this option unsuitable.
- **Why E is Incorrect:**

- Modifying the application to use Amazon EFS and RDS would not address the Lambda timeout issue. This would also introduce additional complexity and the need for managing storage and database services, which the company wants to avoid.

Question 64

A corporation utilizes AWS Organizations along with AWS Control Tower for orchestrating a landing zone architecture. The corporation seeks to enforce governance across its AWS environment, specifically aiming to ensure all Amazon RDS database instances within its production organizational unit (OU) are encrypted at rest. The corporation requires a strategy to identify any RDS instances that do not meet this encryption criterion.

- A. Activate the compulsory governance mechanisms provided by AWS Control Tower and enforce these across the production OU.
- B. Select and enforce the relevant recommended governance mechanism from AWS Control Tower's selection of suggested guardrails for the production OU.
- C. Configure a new compulsory governance rule using AWS Config and implement this rule across all accounts in the production OU.
- D. Craft a tailored SCP within AWS Control Tower and assign it to the production OU.

Correct Answer: **B**

Explanation of Why B is Correct and the Others Are Not:

- **Why B is Correct:**
 - AWS Control Tower offers a set of guardrails that are "strongly recommended" for ensuring compliance and security best practices. These guardrails include checks for unencrypted RDS instances. By enabling the appropriate guardrail specifically for RDS encryption, the company can easily monitor compliance within the production OU.
- **Why A is Incorrect:**
 - Mandatory guardrails in AWS Control Tower are predefined and cannot be turned on or off. While they automatically apply to all OUs, including production, they may not specifically address the RDS encryption requirement if it is not part of the mandatory guardrails.
- **Why C is Incorrect:**
 - AWS Config allows for the creation of custom rules, but AWS Control Tower guardrails are designed to provide governance at a higher, more abstracted level. While AWS Config rules can enforce specific compliance checks, the question suggests using AWS Control Tower's built-in capabilities, making this a more complex approach than necessary.

- **Why D is Incorrect:**
 - While creating a custom SCP is a viable way to enforce policies, it is a lower-level control than AWS Control Tower guardrails. SCPs are also not directly related to compliance checks, such as detecting unencrypted RDS instances, which is more aligned with the functionality provided by guardrails.

Question 65

A rising business operates numerous Amazon Linux 2 AMI-based EC2 instances located within private subnets and heavily utilizes SSH for debugging by its engineering team. Their infrastructure includes a VPC with segregated public and private zones, a NAT gateway, and a Site-to-Site VPN that connects with their internal network. The EC2 instances are configured to permit direct SSH access from the company's internal environment through security groups. The company aims to enhance its security measures for SSH access and establish a method to track the engineers' command activities.

- A. Implement EC2 Instance Connect across all EC2 instances and eliminate any security group rules that permit inbound SSH connections on TCP port 22. Instruct engineers to connect to instances using the EC2 Instance Connect command-line interface.
- B. Refine EC2 security group settings to restrict inbound SSH traffic on TCP port 22 exclusively to the engineers' device IPs. Deploy the Amazon CloudWatch agent on all instances to capture and forward OS-level audit logs to CloudWatch Logs.
- C. Adjust EC2 security group configurations to limit inbound SSH traffic on TCP port 22 solely to the IPs of the engineers' devices. Activate AWS Config to monitor changes in EC2 security group settings. Utilize AWS Firewall Manager to enforce consistent security group policies and automatically correct any rule modifications.
- D. Provision an IAM role with the AmazonSSMManagedInstanceCore managed policy and assign it to all EC2 instances. Remove all inbound SSH rules from the EC2 security groups. Guide engineers to install the AWS Systems Manager Session Manager plugin on their devices and facilitate remote access via Systems Manager's start-session command.

Correct Answer: **D**

Explanation of Why D is Correct and the Others Are Not:

- **Why D is Correct:**
 - AWS Systems Manager Session Manager is a secure way to manage EC2 instances without opening SSH ports. It allows for secure instance management, as well as logging and auditing of session activity through integration with AWS CloudTrail and CloudWatch, without the need for an open SSH port, hence increasing security. The

AmazonSSMManagedInstanceCore policy provides the necessary permissions for the instances to be managed by Systems Manager.

- **Why A is Incorrect:**
 - While EC2 Instance Connect provides a secure method of using SSH without permanently opening SSH ports, it does not by itself provide auditing of the commands run by the engineers. It also requires additional steps every time an engineer needs to connect.
- **Why B is Incorrect:**
 - Simply limiting SSH access to specific IP addresses does not fundamentally change the security posture, as IP addresses can be spoofed or changed, and it still exposes port 22. Additionally, while CloudWatch can capture logs, this does not prevent or provide the same level of security and auditing as AWS Systems Manager.
- **Why C is Incorrect:**
 - While AWS Config and AWS Firewall Manager provide compliance monitoring and security management, respectively, they don't directly enhance the security of SSH access, nor do they offer auditing of command execution on the instances.

Question 66

An organization that employs AWS Organizations has set up a system permitting its development team to innovate within AWS, using their corporate email to initiate account requests. The company needs to impose a limit on AWS spending by the developers to prevent high expenses and avoid running services when they're not needed. The organization seeks a method to allocate a predefined monthly AWS spending allowance for each developer. Choose three options.

- A. Institute an SCP that imposes a monthly spending cap at the account level. Attach this SCP to the developer-specific accounts.
- B. Utilize AWS Budgets to set up a distinct monthly spending budget for every developer account as a part of the process of account initiation.
- C. Draft an SCP that blocks access to high-cost services and features. Link this SCP to the developer accounts.
- D. Generate an IAM policy that restricts access to high-cost services and features. Assign this IAM policy to the developer accounts.
- E. Configure AWS Budgets to create an alert action that will shut down services upon reaching the set budget limit. Set this action to end all running services.

F. Arrange for AWS Budgets to trigger an alert action that sends a notification via Amazon SNS when the spending limit is reached. Set up an AWS Lambda function to stop all active services in response to the notification.

Correct Answers: **B, C, F**

Explanation of Why B, C, and F Are Correct and the Others Are Not:

- **Why B is Correct:**
 - AWS Budgets is designed to track and manage AWS costs. Setting up a monthly budget for each developer account directly addresses the company's requirement to limit spending. It's an administrative task that can be automated and enforced as accounts are created.
- **Why C is Correct:**
 - SCPs can effectively prevent access to certain AWS services that may incur high costs. Applying such SCPs to developer accounts is a proactive way to control costs at the organizational level without daily oversight.
- **Why F is Correct:**
 - Setting up AWS Budgets with alert actions that trigger a Lambda function to terminate services when the predefined spending limit is reached can enforce the budgetary restrictions in real-time and prevent additional costs from accruing.
- **Why A is Incorrect:**
 - SCPs can restrict actions but cannot enforce fixed spending limits or terminate services based on budget thresholds. They control permissions rather than spending caps.
- **Why D is Incorrect:**
 - IAM policies restrict user permissions within an AWS account but are not designed to manage costs directly. An IAM policy cannot enforce spending caps or automatically take action when a budget is exceeded.
- **Why E is Incorrect:**
 - AWS Budgets can alert when a budget threshold is reached, but the service doesn't have the built-in capability to terminate all services. This action would have to be performed by an external process, like an AWS Lambda function as described in Option F.

Question 67

A business currently operates applications within a Source AWS account that is part of AWS Organizations. This includes an application utilizing AWS Lambda for function execution and an Amazon Aurora database for inventory data storage. Lambda functions are deployed via deployment packages, and the Aurora database is set up for automated backups. The goal is to transition these Lambda functions and the Aurora database to a new AWS account, named

Target, while ensuring the process minimizes downtime, especially since the application handles critical data.

A. Retrieve the deployment package for the Lambda functions from the Source account and utilize it to establish identical Lambda functions in the Target account. Make the automated Aurora DB cluster snapshot available to the Target account.

B. Obtain the Lambda function deployment package from the Source account, then use this package to create new Lambda functions in the Target account. Employ AWS Resource Access Manager (AWS RAM) to share the Aurora DB cluster with the Target account and authorize the Target account to clone the Aurora DB cluster.

C. Utilize AWS Resource Access Manager (AWS RAM) to share both the Lambda functions and the Aurora DB cluster with the Target account. Provide the Target account the necessary permissions to clone the Aurora DB cluster.

D. Apply AWS Resource Access Manager (AWS RAM) to share the Lambda functions with the Target account. Share the automated Aurora DB cluster snapshot with the Target account.

Correct Answer: **B**

Explanation of Why B is Correct and the Others Are Not:

- **Why B is Correct:**
 - This option effectively combines the migration of Lambda functions and the Aurora database with minimal downtime. The manual transfer of the Lambda deployment package ensures that the functions are recreated accurately in the Target account. Using AWS RAM to share the Aurora DB cluster and allowing the Target account to clone it is a streamlined way to migrate the database while ensuring data integrity and minimizing downtime.
- **Why A is Incorrect:**
 - While this option addresses the migration of the Lambda functions, sharing an automated Aurora DB snapshot does not provide the same level of ease and efficiency in database migration as cloning the DB cluster.
- **Why C is Incorrect:**
 - AWS RAM cannot be used to share Lambda functions directly between accounts. This option does not provide a feasible method for migrating Lambda functions.
- **Why D is Incorrect:**
 - Similar to option C, AWS RAM does not support direct sharing of Lambda functions. Additionally, sharing only the Aurora DB snapshot may not be as efficient as cloning the entire DB cluster for migration purposes.

Question 68

A business currently operates a Python script on an Amazon EC2 instance that activates every 10 minutes to handle data. This script retrieves and processes files from an Amazon S3 bucket, averaging around 5 minutes per file, and ensures no file is processed more than once. Upon evaluating Amazon CloudWatch metrics, it was observed that the EC2 instance remains underutilized for about 40% of the time due to the script's processing time. The business aims to adapt this system to be more scalable, highly available, and reduce the need for extensive ongoing management, all while being cost-effective.

- A. Convert the existing data processing script into an AWS Lambda function. Set up the Lambda function to be triggered by S3 event notifications when new files are uploaded to the S3 bucket.
- B. Establish an Amazon Simple Queue Service (Amazon SQS) queue and configure S3 to send event notifications to this SQS queue. Implement an EC2 Auto Scaling group with at least one instance running continuously. Adapt the script to continuously check the SQS queue and process the indicated S3 objects.
- C. Transform the script into a Docker container image and execute it on an EC2 instance set to continuously poll the S3 bucket for new files, processing them as they appear.
- D. Package the script into a Docker container to be run on Amazon Elastic Container Service (Amazon ECS) using AWS Fargate. Develop an AWS Lambda function that triggers the Fargate RunTask API operation to process each file, triggered by S3 event notifications.

Correct Answer: **A**

Explanation of Why A is Correct and the Others Are Not:

- **Why A is Correct:**
 - Migrating the script to an AWS Lambda function provides a serverless solution that scales automatically with the workload. Using S3 event notifications for triggering Lambda removes the need for continuous polling, thereby optimizing resource utilization and reducing costs. This approach minimizes management overhead and maximizes cost-effectiveness by only running the function when needed.
- **Why B is Incorrect:**
 - While using an SQS queue with an EC2 Auto Scaling group can make the workload scalable, it doesn't efficiently address the issue of the instance being idle. This approach also involves managing EC2 instances, which increases long-term management overhead.
- **Why C is Incorrect:**
 - Containerizing the script and running it on an EC2 instance still involves continuous polling of the S3 bucket and doesn't effectively reduce idle time. Additionally, it

requires ongoing management of the EC2 instance.

- **Why D is Incorrect:**
 - Although running the container on ECS with Fargate provides scalability and reduces management overhead compared to EC2, using Lambda to trigger Fargate tasks adds unnecessary complexity and potential latency. Directly triggering a Lambda function from S3 events is a more streamlined and cost-effective solution.

Question 69

A North American financial services company is preparing to launch a new web application for its customers on AWS, initially in the us-east-1 Region using Amazon EC2 instances. The application must be designed for high availability, capable of scaling dynamically in response to varying user traffic. Additionally, the company seeks to establish a disaster recovery setup in the us-west-1 Region, utilizing an active-passive failover approach.

- A. Establish separate VPCs in the us-east-1 and us-west-1 regions and set up VPC peering between them. In the us-east-1 VPC, deploy an Application Load Balancer (ALB) covering multiple Availability Zones across both VPCs. Use an Auto Scaling group for deploying EC2 instances across these Availability Zones, integrating it with the ALB.
- B. Construct VPCs in both the us-east-1 and us-west-1 regions. In us-east-1, set up an Application Load Balancer (ALB) that spans multiple Availability Zones within that region only, and link it to an Auto Scaling group with EC2 instances distributed across these zones. Mirror this setup in us-west-1. Use Amazon Route 53 for DNS management, creating individual records for each ALB and implementing health checks to maintain inter-region availability.
- C. Create VPCs in both us-east-1 and us-west-1 regions. In us-east-1, establish an ALB that covers multiple Availability Zones within the region and connect it to an Auto Scaling group with EC2 instances distributed across these zones. Replicate this configuration in us-west-1. Implement Amazon Route 53 with separate DNS records for each ALB, enabling health checks and configuring failover routing policies for each record.
- D. Formulate VPCs in the us-east-1 and us-west-1 regions and enable VPC peering. In us-east-1, configure an ALB extending across multiple Availability Zones in both regions. Set up an Auto Scaling group to deploy EC2 instances across these Availability Zones and link it to the ALB. Use Amazon Route 53 for DNS, creating a single record for the ALB.

Correct Answer: **C**

Explanation of Why C is Correct and the Others Are Not:

- **Why C is Correct:**

- This solution effectively utilizes two separate VPCs, one in each region, with ALBs and Auto Scaling groups configured in each VPC. This setup ensures high availability and dynamic scalability in the primary region (us-east-1). For disaster recovery, the mirrored setup in us-west-1 stands ready for failover. Amazon Route 53 is configured with health checks and failover routing, ensuring traffic is directed to the secondary region only if the primary is down, aligning with the active-passive failover strategy.
- **Why A is Incorrect:**
 - ALBs cannot extend across multiple regions, and VPC peering does not facilitate this. This option fails to provide an effective disaster recovery mechanism between the two regions.
- **Why B is Incorrect:**
 - While this option sets up a similar configuration in both regions, it lacks the failover mechanism. Without a failover routing policy in Route 53, it does not adequately provide for active-passive disaster recovery.
- **Why D is Incorrect:**
 - Similar to option A, this approach is flawed because an ALB cannot span multiple regions. Additionally, without Route 53 failover routing, it doesn't support the required active-passive failover for disaster recovery.

Question 70

A company utilizing a single AWS account is seeking improvements in its AWS Management Console access strategy. Currently, the company's IT support staff use individual IAM users for console access, with these users aligned with their specific job roles. The staff now wishes to simplify their login process by using their existing on-premises Active Directory credentials instead of managing separate IAM user accounts. The solutions architect is considering the use of AWS IAM Identity Center (AWS Single Sign-On) to achieve this integration. The goal is to find a solution that effectively integrates Active Directory for console access in the most cost-effective manner.

A. Set up an AWS Organizations instance and enable the IAM Identity Center feature. Establish and configure an AWS Directory Service for Microsoft Active Directory (AWS Managed Microsoft AD) with a bi-directional trust relationship with the on-premises Active Directory. Configure IAM Identity Center to use this AWS Managed Microsoft AD as the identity source, creating permission sets that align with the groups in AWS Managed Microsoft AD.

B. Create an AWS Organizations instance and activate IAM Identity Center. Implement an AD Connector to establish a connection with the company's on-premises Active Directory. In IAM Identity Center, choose the AD Connector as the identity source and associate permission sets with groups existing in the company's Active Directory.

C. Develop an AWS Organizations instance with all features enabled. Set up and configure an AWS Directory Service for Microsoft Active Directory (AWS Managed Microsoft AD) that forms a bi-directional trust with the company's on-premises Active Directory. In IAM Identity Center, select this AWS Managed Microsoft AD as the identity source and map permission sets to groups within it.

D. Establish an AWS Organizations instance with all features active. Configure an AD Connector for connectivity with the on-premises Active Directory. In IAM Identity Center, set the AD Connector as the identity source and create permission sets corresponding to the groups in the on-premises Active Directory.

Correct Answer: **D**

Explanation of Why D is Correct:

- **Option D** is the most cost-effective and straightforward solution for integrating the company's existing on-premises Active Directory with AWS IAM Identity Center. By enabling all features within AWS Organizations and using an AD Connector, this approach leverages the existing Active Directory infrastructure without the need for additional AWS directory services. The AD Connector serves as a bridge between the on-premises Active Directory and AWS IAM Identity Center, allowing the IT support team to access the AWS Management Console using their existing Active Directory credentials. This eliminates the need for separate IAM accounts, thus simplifying access management.

Explanation of Why Other Options Are Incorrect:

- **Options A and C**: Both involve setting up an AWS Managed Microsoft AD, which incurs additional costs and complexity. This is not necessary for integrating with the existing on-premises Active Directory and is less cost-effective compared to using an AD Connector.
- **Option B**: While it also suggests using an AD Connector, this option does not include enabling all features in AWS Organizations, which is recommended for maximizing the capabilities and benefits of AWS IAM Identity Center.

Question 71

A company specializing in online video streaming has recently introduced a mobile application dedicated to video sharing. This application is designed to upload a variety of files, whose sizes range between 1 GB and 10 GB, to an Amazon S3 bucket located in the useast-1 Region. However, users in Australia have been facing issues with prolonged upload times and incomplete file uploads. A solution is needed to enhance the performance of the app for these users, particularly for the upload process. What two solutions could effectively address these requirements?

- A. Activate S3 Transfer Acceleration for the bucket and modify the app to utilize the Transfer Acceleration endpoint for file uploads.
- B. Create an S3 bucket in multiple regions to initially receive uploads, and then use S3 Cross-Region Replication to transfer the files to the primary distribution bucket.
- C. Implement Amazon Route 53 with latency-based routing to direct uploads to the closest S3 bucket region.
- D. Alter the app to segment video files into smaller parts and employ multipart uploads for transferring these segments to Amazon S3. E. Adjust the app to append random prefixes to the file names prior to the upload process.

Correct Answer: **A, D**

Correct Answers and Explanations:

- **A (Enable S3 Transfer Acceleration):** This is a suitable solution because S3 Transfer Acceleration optimizes file transfer paths using Amazon CloudFront's globally distributed edge locations. By sending data to the nearest edge location, which then forwards it to the S3 bucket, the upload process becomes faster and more reliable, especially for users in distant locations like Australia. This makes the uploads quicker and reduces the likelihood of incomplete uploads.
- **D (Multipart Upload):** Splitting large video files into smaller chunks and using multipart upload is an effective strategy. This approach allows simultaneous uploading of multiple file parts, thereby speeding up the overall upload process. If an upload of a particular chunk fails, only that chunk needs to be reuploaded instead of the entire file, making this method more reliable for users experiencing unstable network conditions.

Why Other Answers are Incorrect:

- **B (S3 Cross-Region Replication):** This option involves unnecessary complexity and costs. While it creates redundancy and local access points, it doesn't directly address the issue of initial upload performance for users in Australia.
- **C (Route 53 Latency-Based Routing):** While this could potentially route users to the nearest bucket region, the main issue lies in the upload to the initial S3 bucket. Without addressing this first point of upload, simply rerouting won't effectively solve the problem.
- **E (Random Prefixes):** Adding random prefixes to file names does not have a direct impact on upload speeds or reliability. This approach is more related to optimizing the organization and retrieval of files within the S3 bucket, rather than improving upload performance.

Question 72

An application currently utilizes a Multi-AZ DB instance of Amazon RDS for MySQL, located in the us-east-1 Region. Following a failover testing procedure, the application experienced a loss of database connectivity and was unable to re-establish this connection. It was only after restarting the application that the connection was successfully re-established. A solution is sought that would allow the application to regain database connectivity post-failover without necessitating a restart. What solution would fulfill these criteria?

- A. Implement an Amazon Aurora MySQL Serverless v1 DB instance, migrate the existing RDS DB instance to this Aurora Serverless v1 instance, and then redirect the application's database connection settings to the Aurora reader endpoint.
- B. Establish an RDS proxy, linking it to the current RDS endpoint, and modify the application's connection settings to use this RDS proxy endpoint.
- C. Set up a two-node Amazon Aurora MySQL DB cluster, transfer the RDS DB instance to this Aurora cluster, create an RDS proxy targeting the original RDS endpoint, and adjust the application's connection settings to this RDS proxy endpoint.
- D. Generate an Amazon S3 bucket, migrate the database to this bucket using AWS Database Migration Service (AWS DMS), configure Amazon Athena to access the S3 bucket as a data repository, install the latest version of the Open Database Connectivity (ODBC) driver for the application, and update the application's connection settings to the Athena endpoint.

Correct Answer: **B**

Correct Answer and Explanation:

- **B (Create an RDS proxy):** This is the most appropriate solution. By setting up an RDS proxy, the application can maintain persistent connections to the database, which helps in managing the connections more efficiently and in a more resilient manner, especially during failover events. The RDS proxy keeps connections alive and quickly reroutes them to the new primary instance after a failover, eliminating the need for the application to restart to re-establish connections.

Why Other Answers are Incorrect:

- **A (Migrate to Amazon Aurora Serverless):** While Aurora Serverless can provide scalability and high availability, simply migrating to an Aurora Serverless DB instance doesn't directly address the immediate reconnection issue that occurs during failover. The focus is on maintaining connections, not on the database's scalability or serverless capabilities.
- **C (Migrate to a two-node Aurora Cluster and create RDS Proxy):** This solution adds unnecessary complexity. Migrating to an Aurora cluster and creating an RDS proxy doesn't offer a significant advantage over directly creating an RDS proxy for the existing RDS instance in terms of maintaining database connections during failover.

- **D (Migrate to Amazon S3 and use Athena):** This approach involves a complete overhaul of the database architecture, moving from a transactional database to a setup using S3 and Athena, which is geared more towards analytics and querying large datasets. This does not address the problem of maintaining database connections during failovers and is an overly complex and inappropriate solution for the issue at hand.

Question 73

A business is developing a cloud-based solution on AWS, where a large number of devices (in the thousands) will establish connections to send and receive data. It's crucial that these devices are capable of real-time communication using the MQTT protocol and authenticate using individual X.509 certificates. What is the most suitable solution that achieves these requirements while ensuring minimal operational complexity?

- A. Implement AWS IoT Core and create an Amazon MQ queue for each device, along with provisioning a unique certificate. Establish connections from each device to Amazon MQ.
- B. Deploy a Network Load Balancer (NLB) paired with an AWS Lambda authorizer. Operate an MQTT broker on a cluster of Amazon EC2 instances managed by an Auto Scaling group. Set this group as the NLB's target and connect each device to the NLB.
- C. Configure AWS IoT Core and create an AWS IoT 'thing' for every device, providing each with a certificate. Directly connect each device to AWS IoT Core.
- D. Establish an Amazon API Gateway HTTP API and a Network Load Balancer (NLB). Create an integration between the API Gateway and NLB, and configure a mutual TLS certificate authorizer on the HTTP API. Run an MQTT broker on an Amazon EC2 instance targeted by the NLB and connect each device to the NLB.

Correct Answer: **C**

Correct Answer and Explanation:

- **C (Set up AWS IoT Core with individual IoT things and certificates):** This is the optimal solution. AWS IoT Core is specifically designed for scenarios like this, offering a managed service that can easily handle thousands of devices connecting via MQTT. By creating an individual 'thing' for each device and provisioning unique certificates, AWS IoT Core ensures secure, authenticated communication with minimal operational overhead. It natively supports MQTT and X.509 certificates, making it a straightforward and efficient choice.

Why Other Answers are Incorrect:

- **A (Use AWS IoT Core with Amazon MQ):** While AWS IoT Core is a suitable platform for device management and communication, pairing it with Amazon MQ for each device adds unnecessary complexity. Amazon MQ is generally used for message queuing and not for direct device-to-cloud communication, making it less efficient in this context.
- **B (NLB with Lambda authorizer and MQTT broker on EC2):** This solution involves significant operational complexity. Setting up and managing an MQTT broker on EC2 instances, along with an NLB and a Lambda-based authorization mechanism, requires more setup and maintenance than a managed service like AWS IoT Core. It also lacks the out-of-the-box integration and streamlined device management provided by IoT Core.
- **D (API Gateway, NLB, and MQTT broker on EC2):** Like option B, this solution introduces unnecessary complexity and operational overhead. Managing API Gateway integrations, NLB configurations, and EC2-based MQTT brokers detracts from the simplicity and efficiency offered by a fully managed service like AWS IoT Core. This approach would require substantial setup and ongoing maintenance, which is not ideal for a scenario prioritizing minimal operational overhead.

Question 74

A corporation is operating multiple workloads within a unified AWS account. Recently, a new organizational policy has been introduced, stipulating that engineers are restricted to deploying only pre-approved resources, and they must utilize AWS CloudFormation for these deployments. A solutions architect is tasked with devising a strategy to implement this new limitation on the IAM roles utilized by the engineers for access. What steps should the solutions architect take to establish this system?

- A. Store AWS CloudFormation templates, featuring sanctioned resources, in an Amazon S3 bucket. Amend the engineers' IAM role policy to permit access solely to Amazon S3 and AWS CloudFormation. Deploy resources using these specific AWS CloudFormation templates.
- B. Modify the engineers' IAM role policy to grant permissions exclusively for deploying approved resources and for using AWS CloudFormation. Construct stacks using AWS CloudFormation templates that consist of only approved resources.
- C. Revise the IAM role policy for the engineers to permit solely AWS CloudFormation actions. Formulate a new IAM policy that allows deploying approved resources, and link this policy to a new IAM service role. Utilize this IAM service role when generating AWS CloudFormation stacks.
- D. Use AWS CloudFormation stacks for resource deployment. Update the IAM policy of the engineers' roles to restrict access to only their individual AWS CloudFormation stacks.

Correct Answer: **C**

Correct Answer and Explanation:

- **C (Update IAM role for CloudFormation actions and create a new IAM service role):** This is the most suitable solution. By updating the engineers' IAM role to limit actions to AWS CloudFormation, engineers are restricted to using CloudFormation for resource provisioning. The creation of a new IAM service role with permissions to deploy only approved resources ensures that when CloudFormation is used, it can only create resources that are allowed by the new service role policy. This approach effectively enforces the policy of deploying only approved resources through CloudFormation, in line with the new company policy.

Why Other Answers are Incorrect:

- **A (Use S3 for storing CloudFormation templates):** While storing approved templates in S3 and restricting IAM role access to S3 and CloudFormation enforces the use of CloudFormation, it does not effectively enforce the restriction to approved resources. Engineers could potentially modify or use other templates that may not comply with the approved resource list.
- **B (Restrict IAM role to approved resources and CloudFormation):** Directly restricting the IAM role to only allow provisioning approved resources is difficult to manage and enforce. It requires constant updating of the IAM policy as approved resources change and does not leverage the capability of CloudFormation to manage these permissions more dynamically.
- **D (Limit access to individual CloudFormation stacks):** Restricting engineers to access only their CloudFormation stacks doesn't ensure that only approved resources are provisioned. It only limits the scope of their access within CloudFormation but does not enforce the policy of using approved resources.

Question 75

A company is preparing to launch a new application that will collect millions of small-sized records (each under 4 KB) every minute from devices globally. These records must be stored reliably and accessed with low latency. They are temporary, with a retention period of only 120 days before deletion is permitted. The anticipated storage requirement for a year is estimated to be between 10 and 15 TB. What would be the most economical storage strategy that also fulfills these criteria?

A. Program the application to save each individual record as a .csv file in an Amazon S3 bucket, enabling indexed retrieval. Implement a lifecycle policy in S3 to remove data that exceeds 120 days.

B. Configure the application to deposit each record into an Amazon DynamoDB table, scaling it appropriately. Utilize the DynamoDB Time to Live (TTL) function to automatically discard records older than 120 days.

C. Arrange for the application to insert each record into a single table within an Amazon RDS MySQL database. Schedule a daily cron job to execute a query that erases records older than 120 days.

D. Set up the application to accumulate records in batches before transferring them to an Amazon S3 bucket. Adjust the metadata for these objects to reflect the batched records, and use Amazon S3's metadata search capability for data retrieval. Set a lifecycle policy in S3 for the removal of data after 120 days.

Correct Answer: **B**

Correct Answer and Explanation:

- **B (Store records in Amazon DynamoDB with TTL):** This is the best solution. DynamoDB is highly efficient for handling large volumes of small records, which fits the application's need to ingest millions of small records per minute. It's scalable, provides low-latency access, and is cost-effective for this type of workload. The TTL feature in DynamoDB simplifies data management by automatically deleting records after 120 days, aligning with the company's data retention policy. This approach minimizes operational overhead and aligns well with the application's requirements.

Why Other Answers are Incorrect:

- **A (Individual .csv files in Amazon S3):** Storing each record as a separate .csv file in S3 could result in massive overhead and higher costs due to the sheer number of files generated. Additionally, S3 is not optimized for high-frequency, small-file writes and indexed retrieval at this scale.
- **C (Amazon RDS MySQL database with cron job):** Using an RDS MySQL database for such high-frequency, small-record storage is not cost-effective. RDS incurs higher costs for storage and compute, and managing deletions with a nightly cron job adds unnecessary complexity and operational overhead.
- **D (Batch records in Amazon S3 with metadata search):** While batching could reduce the number of writes, using S3 metadata search for retrieval is not ideal for real-time data access and can be inefficient for the described use case. S3 is not primarily designed for the high-write frequency of small records and does not provide the low-latency access required for this scenario.

Question 76

A retail organization is operating an e-commerce platform on AWS, spread across several regions, with a requirement for uninterrupted service for online transactions. The platform's data is managed using an Amazon RDS for MySQL database instance. To ensure maximum availability of the database, what is the most effective strategy?

- A. Enable automated backups in Amazon RDS. If there's a service disruption, elevate one of these backups to a new standalone database instance and reroute the database traffic to this instance. Subsequently, establish a new read replica with the newly promoted database instance as its source.
- B. Set up global tables and read replicas in Amazon RDS and enable cross-region functionality. In case of a service interruption, employ AWS Lambda to transfer read replicas from one region to another.
- C. Implement global tables and automated backups in Amazon RDS. Should a disruption occur, use AWS Lambda to relocate the read replicas from one region to another.
- D. Establish cross-region read replicas in Amazon RDS. Upon experiencing a disruption, promote one of the cross-region read replicas to become a standalone database instance and redirect the database traffic to this instance. Then, generate a new read replica sourcing from the newly promoted database instance.

Correct Answer and Explanation:

- **D (Configure cross-region read replicas in Amazon RDS):** This is the most appropriate solution for ensuring high availability. Cross-region read replicas in RDS provide a robust mechanism for maintaining database availability, especially in a multi-region setup like this. In the event of a regional outage or disruption, promoting a cross-region read replica to a standalone DB instance allows for a quick recovery of database services. This approach ensures that there is always a synchronized copy of the database available in another region, ready to be activated, thus maintaining high availability.

Why Other Answers are Incorrect:

- **A (Automated backups promotion):** While automated backups are useful for disaster recovery, they do not offer the same level of availability as cross-region read replicas. Promoting a backup to a standalone instance can be more time-consuming and might not meet the high availability requirements for an e-commerce platform.
- **B (Global tables and Lambda for replica copying):** Amazon RDS does not support global tables, which are a feature specific to Amazon DynamoDB. The concept of using AWS Lambda to copy read replicas between regions is not practical and introduces unnecessary complexity and potential latency.
- **C (Global tables with automated backups and Lambda):** As mentioned, global tables are not a feature of Amazon RDS but of Amazon DynamoDB. This option, therefore, does not apply to an RDS for MySQL scenario and is based on a misunderstanding of RDS capabilities.

Question 77

Example Corp., which has an on-premises data center and a VPC (VPC A) in its AWS account, is currently connected to VPC A via an AWS Site-to-Site VPN. This setup allows seamless access to VPC A from its on-premises servers. Recently, Example Corp. acquired AnyCompany, which possesses its own VPC, named VPC B. Notably, there's no IP address conflict between these networks. VPC A and VPC B have been peered together. Now, Example Corp. aims to establish connectivity from its on-premises servers to VPC B, ensuring that network ACLs and security groups are aptly configured. What is the most efficient method to achieve this connectivity?

- A. Implement a transit gateway and link the Site-to-Site VPN, VPC A, and VPC B to this transit gateway. Revise the route tables within the transit gateway to incorporate routes for the IP ranges of each network.
- B. Set up a transit gateway and establish a new Site-to-Site VPN connection between the on-premises network and VPC B, associating this VPN connection with the transit gateway. Add routing for traffic to the peered VPCs and establish authorization rules for client access to both VPC A and VPC B.
- C. Adjust the route tables for the Site-to-Site VPN and both VPCs, encompassing all three networks. Enable BGP propagation across all networks and wait for the BGP propagation process, which may take up to 5 minutes.
- D. Alter the virtual private gateway definition of the Site-to-Site VPN to incorporate both VPC A and VPC B. Distribute the two routers of the virtual private gateway across the two VPCs.

Correct Answer and Explanation:

- **A (Create a transit gateway and attach networks):** This is the optimal solution with the least operational effort. A transit gateway acts as a network transit hub, efficiently connecting VPCs and on-premises networks. By attaching the Site-to-Site VPN, VPC A, and VPC B to a single transit gateway and updating the route tables accordingly, Example Corp. can facilitate direct network routing between its on-premises data center and both VPCs. This approach simplifies the network architecture and reduces the complexity of managing multiple individual connections.

Why Other Answers are Incorrect:

- **B (New VPN connection to VPC B):** Establishing a new Site-to-Site VPN connection to VPC B and connecting it to the transit gateway adds unnecessary complexity. It requires setting up and managing an additional VPN connection when the existing VPN can be utilized via a transit gateway.
- **C (Update route tables and configure BGP propagation):** This option involves significant changes to route tables and BGP settings, which could be more complicated and time-

consuming compared to using a transit gateway. It also doesn't inherently solve the challenge of connecting the on-premises network to both VPCs efficiently.

- **D (Modify virtual private gateway):** Modifying the virtual private gateway to include both VPC A and VPC B is not a viable solution in AWS. Virtual private gateways cannot be split across multiple VPCs. This approach does not align with AWS networking capabilities and would not achieve the desired connectivity.

Question 78

A company has recently transitioned from an on-premises data center to AWS, adopting a replatforming approach. Part of this migration involved a server running an outdated SMTP service, crucial for an application that dispatches emails to customers. This older SMTP server lacks TLS encryption support and operates over TCP port 25. The application exclusively uses SMTP for email sending. The organization plans to switch to Amazon Simple Email Service (Amazon SES) for its email needs, aiming to retire the old SMTP server. They have already set up and authenticated their SES domain and have had the SES usage limitations removed. What modifications are necessary for the application to integrate with Amazon SES for sending emails?

- A. Adjust the application to establish a connection with Amazon SES using TLS Wrapper. Create an IAM role with permissions for `ses:SendEmail` and `ses:SendRawEmail`, and associate this role with an Amazon EC2 instance.
- B. Reconfigure the application to interface with Amazon SES using STARTTLS. Acquire SMTP credentials for Amazon SES and utilize these credentials for SES authentication.
- C. Alter the application to leverage the SES API for email dispatch. Establish an IAM role endowed with `ses:SendEmail` and `ses:SendRawEmail` permissions, and employ this role as a service role for Amazon SES.
- D. Revise the application to send emails using AWS SDKs. Generate an IAM user specifically for Amazon SES, along with API access keys, and authenticate with Amazon SES using these keys.

Correct Answer and Explanation:

- **B (Configure application with Amazon SES using STARTTLS):** This is the most fitting solution. Since the application can only use SMTP, configuring it to connect to Amazon SES using STARTTLS is the appropriate choice. STARTTLS is an extension to plain SMTP that allows a client to upgrade a plain text connection to a secure connection (TLS) on the same port. Acquiring Amazon SES SMTP credentials and using them for authentication aligns with the application's SMTP-only constraint and the need for secure email transmission, effectively replacing the legacy SMTP server.

Why Other Answers are Incorrect:

- **A (TLS Wrapper with IAM role on EC2 instance):** This option is not suitable because the legacy SMTP service does not support TLS encryption, and the application is limited to using SMTP. Using a TLS Wrapper would require modifications beyond the SMTP-only constraint of the application.
- **C (Use SES API with IAM service role):** While using the SES API is a valid approach for sending emails, it deviates from the application's restriction to using only SMTP. This choice would necessitate significant changes to the application, which is not in line with the scenario's constraints.
- **D (AWS SDKs with IAM user and API keys):** This option, similar to C, involves using AWS SDKs and deviates from the application's SMTP-only limitation. It would require more extensive alterations to the application than simply configuring it to use SES with SMTP, which is not ideal given the scenario.

Question 79

Following the acquisition of several companies, each having distinct AWS accounts with varying billing and reporting methods, a company has merged these accounts into a single AWS Organizations entity. This consolidation has presented challenges in generating a comprehensive cost report that effectively categorizes expenses for all teams involved. The company's finance team requires a solution that enables them to generate detailed cost reports for all the acquired companies through a self-managed application. What approach should be taken to fulfill this requirement?

- A. Generate an AWS Cost and Usage Report for the entire organization, incorporating defined tags and cost categories. Construct a table in Amazon Athena based on this report. Then, create an Amazon QuickSight dataset derived from the Athena table and share this dataset with the finance team.
- B. Produce an AWS Cost and Usage Report for the whole organization, applying specific tags and cost categories. Develop a custom template within AWS Cost Explorer for use by the finance department to create their reports.
- C. Develop an Amazon QuickSight dataset that gathers expenditure data via the AWS Price List Query API, and make this dataset accessible to the finance team.
- D. Utilize the AWS Price List Query API to gather spending data for each account. Create a customized template in AWS Cost Explorer for the finance department's use in report generation.

Correct Answer and Explanation:

- **A (AWS Cost and Usage Report with Athena and QuickSight):** This is the most suitable solution. By creating an AWS Cost and Usage Report, the company can capture detailed billing information across the entire organization. The use of tags and cost categories allows for the segmentation of costs in a way that is meaningful to the finance team. Leveraging Amazon Athena to query this data and then using Amazon QuickSight to visualize and share the data provides a powerful, flexible, and self-managed reporting solution. This approach enables the finance team to create customized, detailed reports that align with their specific needs.

Why Other Answers are Incorrect:

- **B (Custom template in AWS Cost Explorer):** While AWS Cost Explorer is a useful tool for analyzing AWS spend, it lacks the flexibility and depth provided by a combination of AWS Cost and Usage Report, Athena, and QuickSight. Cost Explorer's predefined templates may not offer the granular control and customization required for detailed reporting across multiple acquired companies.
- **C (QuickSight dataset with AWS Price List Query API):** Relying solely on the AWS Price List Query API for spending information is insufficient for this scenario. This API provides pricing information for AWS services, but it does not offer the detailed usage and cost data across multiple accounts that the finance team needs for comprehensive cost analysis.
- **D (AWS Price List Query API with Cost Explorer template):** Similar to option C, this approach is limited by the scope of the AWS Price List Query API, which doesn't provide the necessary detailed cost and usage data required for thorough reporting of the organization's AWS expenses. Moreover, using only AWS Cost Explorer for this purpose may not provide the level of detail and customization needed for effective cost management across multiple accounts.

Question 80

A company operating an IoT platform on AWS has observed a significant increase in the number of deployed field sensors, a trend expected to continue. These sensors transmit data to Node.js API servers on Amazon EC2 instances, which are organized behind an Application Load Balancer. The data is then stored in an Amazon RDS MySQL database utilizing a 4 TB General Purpose SSD volume. However, the company is facing challenges: the API servers are persistently overburdened, and the RDS system is exhibiting high write latency. What combination of measures should be taken to permanently address these challenges and support future sensor expansion, while also maintaining cost-efficiency? Select two options.

A. Increase the storage capacity of the MySQL General Purpose SSD to 6 TB, which will enhance the volume's IOPS.

- B. Transition the database system to Amazon Aurora from RDS MySQL, incorporating read replicas into the architecture.
- C. Implement Amazon Kinesis Data Streams alongside AWS Lambda for the ingestion and processing of the incoming raw data.
- D. Utilize AWS X-Ray for the analysis and troubleshooting of application problems and augment the number of API servers to handle the increased load.
- E. Restructure the database architecture to use Amazon DynamoDB in place of the RDS MySQL database.

Correct Answer and Explanation:

- **C (Use Amazon Kinesis Data Streams and AWS Lambda):** This is an effective solution as it offloads the immediate data processing workload from the API servers. Kinesis Data Streams can efficiently handle large volumes of incoming data, while AWS Lambda can process this data in a scalable manner. This setup reduces the load on the API servers and improves overall system performance.
- **E (Switch to Amazon DynamoDB):** DynamoDB is a highly scalable NoSQL database service that can efficiently handle the write and read demands of growing IoT data. It's more suited for scenarios with high throughput and large-scale data, which is characteristic of IoT platforms. Transitioning to DynamoDB can provide the scalability and performance needed for the increasing sensor data, making it a cost-effective solution for long-term growth.

Why Other Answers are Incorrect:

- **A (Increase MySQL SSD storage to 6 TB):** While increasing the storage size will indeed provide more IOPS, it's a temporary solution that doesn't address the underlying scalability issues. As the number of sensors and data volume continues to grow, this approach will become increasingly costly and less effective.
- **B (Migrate to Amazon Aurora with read replicas):** While Aurora offers better performance and scalability compared to standard MySQL instances, this solution primarily improves read performance with the addition of read replicas. However, the primary issue in this scenario is related to write latency and API server overload, which Aurora read replicas do not directly address.
- **D (Use AWS X-Ray and add more API servers):** Although AWS X-Ray can help identify bottlenecks and issues in the application, and adding more API servers can distribute the load, this approach does not fundamentally resolve the scalability issues with the database. It might improve performance in the short term but doesn't provide a scalable, cost-effective solution for the growing data volume.

A company has developed a serverless electronic document management system hosted on AWS in the eu-central-1 Region. This system, which includes a web app using Amazon CloudFront with S3 as the origin, allows users to upload documents. The web app interacts with Amazon API Gateway Regional endpoints, which in turn trigger AWS Lambda functions. These functions manage metadata storage in an Amazon Aurora Serverless database and handle document uploads to an S3 bucket. Following successful proof of concept with a major client, the company is experiencing growth and now seeks to reduce latency for users outside Europe. What two measures should be implemented to achieve improved latency globally?

- A. Activate S3 Transfer Acceleration for the S3 bucket and modify the web application to utilize Transfer Acceleration signed URLs for uploads.
- B. Set up an AWS Global Accelerator and link it to the CloudFront distribution.
- C. Switch the API Gateway from Regional endpoints to edge-optimized endpoints.
- D. Deploy the entire serverless stack in two additional global locations, incorporating Aurora Serverless global databases.
- E. Introduce an Amazon RDS proxy between the Lambda functions and the Aurora Serverless database for improved database interaction.

Correct Answer and Explanation:

- **A (Enable S3 Transfer Acceleration):** This is an effective solution for reducing upload latency for users worldwide. S3 Transfer Acceleration speeds up uploads to S3 by routing data through Amazon CloudFront's globally distributed edge locations. By modifying the web app to use Transfer Acceleration signed URLs, users can upload documents faster regardless of their geographical location.
- **C (Switch to edge-optimized API Gateway endpoints):** Changing the API Gateway from Regional to edge-optimized endpoints enhances global performance. Edge-optimized endpoints allow API calls to be routed through CloudFront edge locations, which reduces latency by bringing the requests closer to the user, improving response times for users outside Europe.

Why Other Answers are Incorrect:

- **B (Create AWS Global Accelerator for CloudFront):** Linking AWS Global Accelerator to a CloudFront distribution is not necessary and doesn't provide additional benefits. CloudFront is already a global content delivery network designed to optimize delivery and reduce latency. AWS Global Accelerator is more suited for non-HTTP/S use cases or for improving availability and performance across regions, which is not the primary concern here.

- **D (Provision stack in additional locations with global Aurora databases):** While deploying the application stack in multiple regions can reduce latency, it significantly increases the complexity and cost of the solution. Managing global databases and multiple instances of the serverless stack would require substantial effort and might not be the most cost-effective approach.
- **E (Add Amazon RDS proxy):** Incorporating an RDS proxy between Lambda and Aurora Serverless is beneficial for managing database connections and scaling, but it does not address the issue of latency for users outside Europe. This solution is more related to database efficiency rather than improving global access latency.

Question 82

An adventure company has recently updated its mobile application to allow users to upload their hiking and rafting media, including photos and videos. These files are stored in Amazon S3 Standard and are made available to users via Amazon CloudFront. The company is seeking ways to reduce storage costs without compromising on access speed. Analysis shows that most of the uploaded content is rarely accessed after the initial 30 days, though some items continue to be frequently viewed even after this period. The company requires a cost-effective solution that still ensures immediate (millisecond-level) access to these photos and videos. What is the most suitable approach to meet these criteria?

- A. Implement S3 Intelligent-Tiering for the S3 bucket.
- B. Set up an S3 Lifecycle policy to move both photos and videos from S3 Standard to S3 Glacier Deep Archive after a period of 30 days.
- C. Transition from Amazon S3 to an Amazon Elastic File System (Amazon EFS), with the EFS file system mounted on Amazon EC2 instances.
- D. Apply a Cache-Control: max-age header to the S3 photos and videos, configuring the header to 30 days.

Correct Answer and Explanation:

- **A (Configure S3 Intelligent-Tiering):** This solution is ideal for the company's needs. S3 Intelligent-Tiering automatically moves data between two access tiers — one for frequently accessed data and another for infrequently accessed data — depending on how often the data is accessed. This ensures that the company pays a lower storage rate for infrequently accessed files while maintaining immediate access to all files. Since some media remains frequently accessed post-30 days, Intelligent-Tiering dynamically adjusts to usage patterns without manual intervention, providing a cost-efficient storage solution while ensuring millisecond retrieval availability.

Why Other Answers are Incorrect:

- **B (Transition to S3 Glacier Deep Archive):** Moving files to S3 Glacier Deep Archive significantly reduces storage costs but does not provide millisecond-level access. Retrieval from Glacier Deep Archive can take hours, which does not align with the need for immediate access to the media files.
- **C (Use Amazon EFS with EC2 instances):** Transitioning to Amazon EFS involves more complexity and potential cost than necessary. While EFS provides immediate access to files, it's generally more expensive than using S3 with Intelligent-Tiering for the described use case. Additionally, managing EFS with EC2 instances adds operational overhead.
- **D (Set Cache-Control: max-age header):** Adding a Cache-Control header with a max-age of 30 days controls how long the content is cached in CloudFront, but it doesn't address the underlying requirement of reducing storage costs in S3. This solution does not impact the storage tier or cost of storing the files in S3.

Question 83

A company is experiencing a significant rise in expenses related to the use of various Amazon S3 storage classes for storing their files and images. To address this issue, a solutions architect is tasked with analyzing the past year's data trends to determine the most cost-effective storage classes for the stored objects. What approach should the architect take to effectively review the 12-month data trends and identify the most suitable storage classes?

- A. Obtain and examine the AWS Cost and Usage Reports for S3 usage over the previous 12 months. Additionally, consult AWS Trusted Advisor for recommendations on cost optimization.
- B. Utilize S3 storage class analysis and transfer the data into an Amazon QuickSight dashboard for an in-depth examination of storage patterns.
- C. Employ Amazon S3 Storage Lens, upgrading the standard dashboard to incorporate advanced metrics specifically for assessing storage trends.
- D. Implement Access Analyzer for S3, retrieve its report for the previous year, and import this data into an Amazon QuickSight dashboard for analysis.

Correct Answer and Explanation:

- **C (Use Amazon S3 Storage Lens with advanced metrics):** This is the most effective solution for the given requirements. Amazon S3 Storage Lens provides comprehensive visibility into object storage usage and activity trends across the company's S3 resources. By upgrading to include advanced metrics, the solutions architect can access detailed analytics that assist in identifying usage patterns and determining the most appropriate storage classes for

their files and images. This tool is specifically designed for S3 storage analysis, making it a fitting choice for reviewing storage class utilization and optimizing costs.

Why Other Answers are Incorrect:

- **A (AWS Cost and Usage Reports and Trusted Advisor):** While AWS Cost and Usage Reports provide valuable information about S3 usage costs, and Trusted Advisor offers general cost-saving recommendations, this approach may not offer the specific, detailed analysis of storage class usage trends necessary to make informed decisions about the most appropriate storage classes.
- **B (S3 storage class analysis and QuickSight):** S3 storage class analysis does provide insights into how data is accessed and which storage class may be most cost-effective. However, it's not as comprehensive as S3 Storage Lens for detailed trend analysis. While importing data into Amazon QuickSight for visualization is beneficial, Storage Lens offers more advanced and specific insights directly related to storage class optimization.
- **D (Access Analyzer for S3 and QuickSight):** Access Analyzer for S3 is primarily designed to review and analyze access policies and ensure that S3 resources are not inadvertently shared with unintended recipients. It is not specifically focused on storage class trends and cost optimization, making it less suitable for the company's need to analyze storage classes based on usage trends.

Question 84

A company currently hosting its cloud setup on AWS in a single region is looking to expand its operations. This expansion involves deploying their infrastructure across multiple AWS accounts and regions. A solutions architect is tasked with managing this process using infrastructure as code. The architect must find a solution that accommodates the company's need for multi-region, multi-account deployment. What approach should the architect adopt to fulfill these requirements effectively?

- A. Implement AWS CloudFormation templates and integrate IAM policies to manage different accounts. Apply these templates across the various regions.
- B. Utilize AWS Organizations and deploy AWS CloudFormation templates from the central management account. Manage the multi-account deployments using AWS Control Tower.
- C. Combine AWS Organizations with AWS CloudFormation StackSets, executing a CloudFormation template from an account endowed with the appropriate IAM permissions.
- D. Employ nested AWS CloudFormation templates, altering the region settings through the use of these nested stacks.

Correct Answer and Explanation:

- **C (Use AWS Organizations and CloudFormation StackSets):** This solution is the most fitting. AWS Organizations allows for efficient management of multiple AWS accounts, and AWS CloudFormation StackSets is an extension of CloudFormation that enables the architect to deploy a single template across multiple accounts and regions simultaneously. By using StackSets, the solutions architect can define the infrastructure once and then replicate it across different accounts and regions, aligning with the company's expansion plans. This approach ensures consistency in infrastructure deployment and simplifies the management of multi-region, multi-account environments.

Why Other Answers are Incorrect:

- **A (CloudFormation templates with IAM policies across regions):** While AWS CloudFormation templates are effective for defining infrastructure as code, this approach lacks the inherent capability to easily manage deployments across multiple AWS accounts and regions. It would require manual replication of templates in each region and account, which is less efficient and more error-prone compared to using StackSets.
- **B (Use AWS Organizations with Control Tower and CloudFormation):** AWS Control Tower provides governance and best practices for multi-account AWS environments, but it is more focused on setting up new accounts and ensuring compliance rather than deploying specific infrastructure as code across existing accounts and regions. While Control Tower is useful for account management, it does not directly address the need for deploying CloudFormation templates across multiple regions and accounts.
- **D (Nested CloudFormation stacks for region changes):** Nested stacks in AWS CloudFormation are used to manage complex templates by breaking them down into smaller, reusable components. However, nested stacks do not inherently provide a simple way to deploy infrastructure across multiple AWS accounts and regions. This approach would still require manual intervention to manage deployments in different regions, which is not optimal for the company's expansion requirements.

Question 85

A company is transitioning from a monolithic architecture to a more contemporary application design on AWS. This transformation includes an enhancement of the existing CI/CD pipeline to align with the modernized application. The upgraded pipeline must support frequent releases (several times per hour) and offer the capability for rapid rollbacks. What architectural design for the CI/CD pipeline would best fulfill these specific requirements?

A. Construct a CI/CD pipeline that utilizes Amazon Machine Images (AMIs) to encapsulate the application and its configurations. Execute application deployments through the replacement of Amazon EC2 instances.

B. Implement AWS Elastic Beanstalk for staging in a secondary environment, designated as the target for the application's CI/CD pipeline. Perform deployments by switching between staging and production environment URLs.

C. Adopt AWS Systems Manager for infrastructure re-provisioning at each deployment. Modify the EC2 user data to fetch the newest code version from Amazon S3, and employ Amazon Route 53 weighted routing to direct traffic to the updated environment.

D. Implement application updates as part of an Auto Scaling event, using predefined AMIs. Utilize new AMI versions to introduce instances and systematically remove instances running on older AMIs based on the defined termination policy during deployment.

Correct Answer and Explanation:

- **B (Use AWS Elastic Beanstalk with staging and production environments):** This option is the most suitable for the company's requirements. AWS Elastic Beanstalk allows for easy management of application deployments, including the setup of separate staging and production environments. By deploying changes to a staging environment first and then swapping URLs with the production environment, the company can achieve rapid, frequent deployments. Additionally, this approach allows for quick rollbacks in case of issues, as it simply requires swapping the URLs back, minimizing downtime and risk.

Why Other Answers are Incorrect:

- **A (Deployment with AMIs and EC2 instance replacement):** While deploying new AMIs and replacing EC2 instances can be effective, this method tends to be slower and less agile compared to Elastic Beanstalk's environment swapping. It may not support the high frequency of deployments and rapid rollbacks required, as instance replacement can be time-consuming.
- **C (AWS Systems Manager with Route 53 weighted routing):** Using AWS Systems Manager to re-provision infrastructure and Route 53 for routing introduces complexity and potential delays in the deployment process. This method might not support the high frequency of updates and quick rollbacks as efficiently as Elastic Beanstalk's environment swapping approach.
- **D (Auto Scaling with AMI updates):** While using Auto Scaling with updated AMIs is a valid strategy for some scenarios, it is not as efficient for scenarios requiring multiple deployments per hour and rapid rollbacks. This approach can lead to longer deployment times and may not provide the agility required for quick reversions to previous versions in case of issues.

A solutions architect is tasked with configuring the VPC infrastructure for a company's application running on Amazon EC2 instances in an AWS Region. This application requires access to an Amazon Aurora DB Cluster. Each set of EC2 instances and the DB Cluster are associated with their respective security groups. The architect's objective is to establish rules within these security groups that allow the application minimal and necessary access to the DB Cluster. What two steps should the architect take to achieve this goal while adhering to the principle of least privilege?

- A. Implement an inbound rule in the security group of the EC2 instances, designating the security group of the DB cluster as the source for the default Aurora port.
- B. Configure an outbound rule in the security group of the EC2 instances, setting the security group of the DB cluster as the destination for the default Aurora port.
- C. Set up an inbound rule in the DB cluster's security group, identifying the security group of the EC2 instances as the source for the default Aurora port.
- D. Establish an outbound rule in the DB cluster's security group, pointing to the security group of the EC2 instances as the destination for the default Aurora port.
- E. Create an outbound rule in the DB cluster's security group, specifying the security group of the EC2 instances as the destination over ephemeral ports.

Correct Answer and Explanation:

- **B (Outbound rule from EC2 instances to DB cluster):** This step is essential because it allows outbound traffic from the EC2 instances to reach the DB cluster. Specifying the DB cluster's security group as the destination ensures that the EC2 instances can only communicate with the DB cluster over the Aurora's default port, adhering to the principle of least privilege.
- **C (Inbound rule to DB cluster from EC2 instances):** This rule is crucial as it allows the DB cluster to receive incoming traffic from the EC2 instances. By specifying the EC2 instances' security group as the source for the default Aurora port, the rule ensures that only traffic originating from the application on the EC2 instances can access the DB cluster, maintaining a tight security posture.

Why Other Answers are Incorrect:

- **A (Inbound rule from DB cluster to EC2 instances):** This rule is not necessary because the communication initiated by the EC2 instances to the DB cluster does not require an inbound rule in the EC2 instances' security group. The response from the DB cluster to the EC2 instances is covered by the outbound rule of the DB cluster's security group.

- **D (Outbound rule from DB cluster to EC2 instances):** Outbound rules in the DB cluster's security group to the EC2 instances are generally not required for this kind of setup, as the DB cluster typically does not initiate communication with the EC2 instances.
- **E (Outbound rule from DB cluster to EC2 instances over ephemeral ports):** This configuration is unnecessary in the context of enabling the application on EC2 instances to access the DB cluster. Ephemeral ports are not typically used in this scenario, and the focus should be on allowing the EC2 instances to initiate communication with the DB cluster, not the other way around.

Question 87

A company is revising its internal approach to cloud cost management for different business units. Using AWS Organizations, it currently operates separate AWS accounts for each unit and follows a tagging standard (application, environment, owner) across the organization. The goal is to implement a centralized system where each business unit receives detailed monthly cloud expenditure reports and notifications when spending exceeds predetermined limits. What is the most economical method to establish this system of detailed reporting and spending alerts for each business unit?

- A. Set up AWS Budgets in every individual account with alerts based on application, environment, and owner tags. Link each business unit to a corresponding Amazon SNS topic for alerts. Use AWS Cost Explorer in each account to generate monthly spending reports for the respective units.
- B. Implement AWS Budgets within the central management account of AWS Organizations, creating alerts categorized by application, environment, and owner tags. Associate each business unit with a specific Amazon SNS topic for these alerts. Utilize AWS Cost Explorer in the management account to produce monthly spending reports for each business unit.
- C. Arrange AWS Budgets in each separate account, forming alerts categorized by application, environment, and owner tags. Connect each business unit to an Amazon SNS topic for these alerts. Rely on the AWS Billing and Cost Management dashboard in every account to compile monthly reports for each unit.
- D. Activate AWS Cost and Usage Reports in the management account of AWS Organizations, tailoring the reports by application, environment, and owner tags. Develop an AWS Lambda function to process these reports, issue spending alerts, and disseminate monthly reports to the email lists of each business unit.

Correct Answer and Explanation:

- **B (Use AWS Budgets in the management account with Cost Explorer):** This option is the most cost-effective and efficient. By configuring AWS Budgets in the central management

account, the company can streamline the process of monitoring and reporting cloud spending across all business units. Grouping the budget alerts by application, environment, and owner, and linking them to SNS topics, ensures that relevant notifications are sent out. Additionally, using AWS Cost Explorer within the management account allows for centralized generation of detailed monthly reports for each business unit. This approach leverages the consolidated view and control offered by the management account, reducing the need for repetitive tasks across multiple accounts.

Why Other Answers are Incorrect:

- **A (AWS Budgets in each account with Cost Explorer):** Setting up AWS Budgets in each individual account is less efficient and could lead to higher operational overhead. It requires managing budgets and reports separately in each account, which is more complex compared to a centralized approach in the management account.
- **C (AWS Budgets in each account with Billing and Cost Management dashboard):** Similar to option A, configuring budgets in each separate account increases the management complexity. Additionally, using the Billing and Cost Management dashboard in each account for monthly reports is less streamlined than using Cost Explorer in a centralized management account.
- **D (Cost and Usage Reports with AWS Lambda):** While this method could technically meet the requirements, it is overly complex and likely more costly. Developing and maintaining a custom Lambda function for processing reports and sending notifications requires additional development and operational efforts, making it less cost-effective compared to leveraging existing AWS Budgets and Cost Explorer functionalities in the management account.

Question 88

A company utilizes AWS CloudFormation for its infrastructure deployment. There is a concern that the deletion of a production CloudFormation stack might inadvertently result in the loss of critical data stored in Amazon RDS databases or Amazon EBS volumes. The company seeks a method to prevent accidental data deletion when a CloudFormation stack is removed. What approach can ensure that data in RDS and EBS resources is safeguarded against accidental deletion during stack removal?

- A. Update the CloudFormation templates by incorporating a `DeletionPolicy` attribute for RDS and EBS resources within them.
- B. Establish a stack policy that explicitly prohibits the deletion of RDS and EBS resources.
- C. Revise IAM policies to specifically forbid the deletion of RDS and EBS resources that are tagged with `"aws:cloudformation:stack-name"`.

D. Implement AWS Config rules to restrict the deletion of RDS and EBS resources.

Correct Answer and Explanation:

- **A (Add DeletionPolicy attribute in CloudFormation templates):** This is the most direct and effective solution. By adding a DeletionPolicy attribute to the RDS and EBS resources in the CloudFormation templates, the company can control the behavior of these resources when the stack is deleted. For instance, setting the DeletionPolicy to "Retain" ensures that the specific resources are not deleted even when the stack is removed. This approach directly ties the protection mechanism to the resource definition within CloudFormation, ensuring that the data remains intact regardless of stack deletion actions.

Why Other Answers are Incorrect:

- **B (Configure stack policy to disallow deletion):** Stack policies in AWS CloudFormation primarily govern updates to the stack resources but do not provide a mechanism to prevent the deletion of specific resources when a stack is deleted. Therefore, a stack policy would not be effective in preventing the accidental deletion of RDS and EBS resources.
- **C (Modify IAM policies with "aws:cloudformation:stack-name" tag):** While IAM policies can control what actions a user can perform on AWS resources, using IAM to prevent the deletion of resources tagged with specific CloudFormation stack names can be complex and may not provide the necessary level of granularity or direct control needed to protect specific RDS and EBS resources during stack deletion.
- **D (Use AWS Config rules):** AWS Config rules are used for assessing, auditing, and evaluating the configurations of AWS resources. Although Config rules can help identify non-compliant resources, they are not designed to actively prevent the deletion of resources. They are more suited for monitoring and reporting rather than enforcing operational restrictions like preventing deletion.

Question 89

A company, utilizing VPC flow logs for its NAT gateway, has observed inbound traffic marked as "ACCEPT" originating from the public IP address 198.51.100.2 towards a private Amazon EC2 instance. The initial segments of the VPC's CIDR block are 203.0. A solutions architect is tasked with investigating if this traffic signifies unexpected inbound connections from the internet. What approach should the architect take to determine the nature of this traffic?

A. Navigate to the AWS CloudTrail console and select the log group containing logs for both the NAT gateway's and the private instance's elastic network interfaces. Execute a query to isolate records with the destination address patterned as "203.0" and the source address as "198.51.100.2". Utilize the stats command to calculate the total data transferred by these source and destination addresses.

B. Access the Amazon CloudWatch console and choose the log group that includes logs from the NAT gateway's and the private instance's elastic network interfaces. Conduct a query filtering for logs where the destination address is similar to "203.0" and the source address matches "198.51.100.2". Apply the stats command to aggregate the data volume transferred associated with these addresses.

C. Go to the AWS CloudTrail console, select the log group for the NAT gateway's and the private instance's elastic network interfaces. Filter the logs with the destination address as "198.51.100.2" and the source address resembling "203.0". Employ the stats command to sum up the bytes transferred by these source and destination addresses.

D. Use the Amazon CloudWatch console, opt for the log group with logs from the NAT gateway's and the private instance's elastic network interfaces. Filter the logs to identify those with the destination address "198.51.100.2" and the source address similar to "203.0". Apply the stats command for aggregating the total bytes transferred for these addresses.

Correct Answer and Explanation:

- **B (Use Amazon CloudWatch for NAT gateway and EC2 instance interface logs):** This option is the most appropriate. Amazon CloudWatch is the correct service for accessing and analyzing VPC flow logs, not AWS CloudTrail. VPC flow logs capture information about IP traffic going to and from network interfaces in the VPC, including NAT gateways and EC2 instances. By filtering logs in CloudWatch for those with the destination address in the range "203.0" (the VPC CIDR block) and the source address "198.51.100.2" (the public IP in question), the architect can examine the specific traffic of interest. The use of the stats command to summarize the total bytes transferred provides insight into the volume of data exchanged, aiding in determining the nature of the traffic.

Why Other Answers are Incorrect:

- **A (AWS CloudTrail for interface logs):** AWS CloudTrail is not the correct service for this task. CloudTrail is used for auditing AWS account activity and API usage, not for analyzing VPC flow logs, which are available in Amazon CloudWatch.
- **C and D (Inverted address filtering in CloudTrail and CloudWatch):** These options incorrectly suggest filtering logs with the destination address as the public IP ("198.51.100.2") and the source address as the VPC CIDR block ("203.0"). In the given scenario, the traffic originates from the public IP and is destined for the VPC, making this filtering approach incorrect. Additionally, option C incorrectly suggests using AWS CloudTrail, which is not suitable for analyzing VPC flow logs.

Question 90

Within a single AWS Organizations setup, a company operates two distinct business units, each with its own AWS account. These units frequently exchange confidential documents via dedicated Amazon S3 buckets in their respective accounts, linked by two-way replication. A recent security evaluation revealed that neither of these S3 buckets employs encryption at rest, a requirement according to the company's data storage policy. The company intends to implement server-side encryption using Amazon S3 managed encryption keys (SSE-S3) for these buckets. Considering the large volume of objects in the buckets, what is the most efficient method to meet this encryption requirement?

- A. Enable SSE-S3 for both S3 buckets. Employ S3 Batch Operations to replicate and apply encryption to the existing objects within their current locations.
- B. Establish an AWS Key Management Service (AWS KMS) key in each AWS account. Activate server-side encryption using AWS KMS keys (SSE-KMS) for each S3 bucket, utilizing the respective account's KMS key. Encrypt the existing objects by executing an S3 copy command via the AWS CLI.
- C. Activate SSE-S3 for both S3 buckets. Encrypt the current objects using an S3 copy command executed in the AWS CLI.
- D. Generate an AWS Key Management Service (AWS KMS) key in each account. Implement server-side encryption with AWS KMS keys (SSE-KMS) for both S3 buckets, applying the corresponding KMS key in each account. Use S3 Batch Operations to replicate the objects within the same location.

Correct Answer and Explanation:

- **A (Enable SSE-S3 and use S3 Batch Operations):** This is the most operationally efficient solution. Enabling SSE-S3 for both buckets ensures that all future objects will be encrypted using Amazon S3 managed encryption keys. For the existing objects, S3 Batch Operations can be used to efficiently encrypt these objects in place. This approach doesn't require creating and managing additional KMS keys and is simpler compared to manually copying objects via the AWS CLI. S3 Batch Operations is designed to handle operations on millions of objects efficiently, making it ideal for the scenario described.

Why Other Answers are Incorrect:

- **B (Use SSE-KMS and AWS CLI for encryption):** This option is less efficient because it involves creating and managing separate KMS keys in each account and manually copying objects using the AWS CLI. Managing KMS keys adds complexity, and the process of copying objects via the CLI is operationally more intensive, especially for millions of objects.
- **C (Enable SSE-S3 and use AWS CLI for encryption):** While enabling SSE-S3 is correct, using the AWS CLI to copy objects for encryption is less efficient compared to using S3 Batch

Operations, particularly when dealing with millions of objects. The manual copy process is time-consuming and operationally demanding.

- **D (Use SSE-KMS and S3 Batch Operations):** This option introduces unnecessary complexity by requiring the creation and management of KMS keys. Since the company's requirement can be met with SSE-S3, which uses S3-managed keys, creating KMS keys for SSE-KMS is an additional operational burden that does not provide a significant advantage in this context.

Question 91

A company's application in the AWS Cloud is amassing a significant volume of unstructured data, stored in an Amazon S3 bucket using the S3 Standard storage class. This data, already amounting to several terabytes, is expanding daily by gigabytes. The company actively queries and analyzes this data, but it only accesses data less than a year old. For compliance purposes, it is mandatory to keep all data indefinitely. What is the most cost-effective approach to meet these data querying and long-term retention needs?

- A. Implement S3 Select for data querying and establish an S3 Lifecycle policy to transition data older than one year to S3 Glacier Deep Archive.
- B. Utilize Amazon Redshift Spectrum for data querying and create an S3 Lifecycle policy that moves data older than one year to S3 Glacier Deep Archive.
- C. Adopt AWS Glue Data Catalog and Amazon Athena for data querying and configure an S3 Lifecycle policy to transfer data older than one year to S3 Glacier Deep Archive.
- D. Apply Amazon Redshift Spectrum for querying the data and set up an S3 Lifecycle policy to shift data over one year old to S3 Intelligent-Tiering.

Correct Answer and Explanation:

- **C (AWS Glue Data Catalog and Amazon Athena with S3 Glacier Deep Archive):** This solution is the most cost-effective and functional for the company's requirements. AWS Glue Data Catalog alongside Amazon Athena provides a powerful and flexible way to query and analyze large amounts of unstructured data stored in S3. Athena allows for serverless queries directly against the data in S3, which is suitable for the type of data and usage pattern described. Additionally, setting up an S3 Lifecycle policy to transition older data (more than 1 year old) to S3 Glacier Deep Archive aligns with the need for long-term, cost-effective data retention, especially considering the data is infrequently accessed after a year.

Why Other Answers are Incorrect:

- **A (S3 Select with S3 Glacier Deep Archive):** While S3 Select is useful for querying within individual objects, it is less efficient for querying across a large dataset of several terabytes, especially when compared to the capabilities of AWS Glue Data Catalog and Amazon Athena. S3 Select is more suited for scenarios where the querying is limited to a smaller subset of data within objects.
- **B (Amazon Redshift Spectrum with S3 Glacier Deep Archive):** Although Amazon Redshift Spectrum can be used to query large datasets in S3, it is generally part of a larger, more complex, and potentially more expensive solution involving Amazon Redshift. For the company's needs, a simpler and more cost-effective solution like Athena, which doesn't require Redshift clusters, would be more appropriate.
- **D (Amazon Redshift Spectrum with S3 Intelligent-Tiering):** Using Amazon Redshift Spectrum for querying is potentially more costly and complex than necessary. Moreover, transitioning data to S3 Intelligent-Tiering is not as cost-effective for long-term storage of data that is not accessed, as Intelligent-Tiering is designed for data with unknown or changing access patterns, not for data that is known to be infrequently accessed. S3 Glacier Deep Archive is a more suitable choice for data that is retained for compliance reasons and is not accessed after a certain period.

Question 92

A company specializing in video processing is looking to develop a machine learning model and needs to transfer 600 TB of compressed data from their on-premises network attached storage to AWS. This data, consisting of thousands of files, is beyond the processing capabilities of their current infrastructure. The company aims to complete this one-time data transfer within three weeks and requires that the data be encrypted during transit. Their internet connection, which is shared across multiple departments, has a bandwidth of 100 Mbps. What is the most cost-effective and feasible solution for transferring this large amount of data to AWS within the specified timeframe?

- Acquire multiple AWS Snowball Edge Storage Optimized devices through the AWS Management Console. Set these devices to target an S3 bucket, transfer the data to them, and then return the devices to AWS.
- Establish a 10 Gbps AWS Direct Connect link between the company's site and the closest AWS Region. Use a VPN to transfer the data securely into Amazon S3 in that region.
- Implement a VPN connection from the on-premises network attached storage to the nearest AWS Region and transfer the data through this VPN.
- Install an AWS Storage Gateway file gateway on the company's premises, directing it to an S3 bucket in AWS. Migrate the data to this file gateway.

Correct Answer and Explanation:

- **A (Use AWS Snowball Edge Storage Optimized devices):** This solution is the most practical and cost-effective for the company's needs. Given the large volume of data (600 TB) and the limited internet bandwidth (100 Mbps), transferring the data over the internet would be time-prohibitive. AWS Snowball Edge devices are designed for securely transferring large amounts of data into AWS. They provide a fast, physical solution for large-scale data migration, and the data is encrypted during transit. Once the data is copied to these devices, they can be shipped back to AWS, ensuring the transfer is completed within the three-week timeframe.

Why Other Answers are Incorrect:

- **B (Set up AWS Direct Connect):** While AWS Direct Connect provides a dedicated network connection to AWS, setting up a 10 Gbps connection for a one-time transfer is not cost-effective. Additionally, the logistics and time required to establish a Direct Connect link would likely exceed the three-week timeframe.
- **C (Create a VPN connection for data transfer):** Transferring 600 TB of data over a 100 Mbps shared internet connection, even with a VPN for encryption, would take an impractical amount of time, far exceeding the three-week limit. This approach is not feasible given the bandwidth constraints.
- **D (Deploy AWS Storage Gateway file gateway):** While AWS Storage Gateway facilitates on-premises to cloud data transfer, moving 600 TB of data over a 100 Mbps internet connection would be extremely slow and inefficient. The time required for such a transfer would likely surpass the three-week deadline, making this option unsuitable for the company's requirements.

Question 93

A company's forms-processing application, now hosted on AWS, allows users to upload scanned forms via a web interface. These forms are stored in Amazon S3, while user metadata and file references are maintained in an Amazon RDS for PostgreSQL database. Currently, form uploads trigger Amazon SNS notifications to a team, whose members manually process each form, validate data, and enter it into another system via an API. The company seeks an automated solution for processing these forms to increase accuracy, reduce time to market, and lower long-term operational effort. What automated solution would best fulfill these criteria for form processing?

A. Implement an application tier using custom-developed optical character recognition (OCR) libraries on an Amazon Elastic Kubernetes Service (Amazon EKS) cluster. This application tier will process uploaded forms, store results in Amazon S3, and then parse and store extracted data in

an Amazon DynamoDB table before submitting it to the target system's API. This tier will be hosted on EC2 instances.

B. Expand the current system with an additional application tier comprising AWS Step Functions and AWS Lambda, using AI/ML models trained and hosted on EC2 instances for OCR processing of uploaded forms. Results will be saved in Amazon S3, parsed, and the necessary data extracted within this application tier before being sent to the target system's API.

C. Create a new application tier on EC2 instances, utilizing this tier to access AI/ML model endpoints hosted in Amazon SageMaker for OCR of the forms. Output will be stored in Amazon ElastiCache, parsed, and required data extracted within the application tier, then submitted to the target system's API.

D. Enhance the system with an application tier using AWS Step Functions and AWS Lambda, configured to employ Amazon Textract and Amazon Comprehend for OCR processing of forms upon upload. Store the processed data in Amazon S3, parse and extract the necessary information within the application tier, and then submit it to the target system's API.

Correct Answer and Explanation:

- **D (Use AWS Step Functions and AWS Lambda with Amazon Textract and Amazon Comprehend):** This solution aligns well with the company's requirements for automation, accuracy, speed, and reduced operational overhead. By integrating AWS Step Functions and AWS Lambda, the company can set up a scalable and serverless workflow. Amazon Textract is specifically designed for OCR and can accurately extract text and data from scanned documents, while Amazon Comprehend can be used for understanding and analyzing the extracted text. This combination of services allows for efficient processing of forms without the need for managing underlying infrastructure or developing custom OCR solutions. Storing processed data in Amazon S3 and handling the data extraction and API submission within the application tier ensures a streamlined process.

Why Other Answers are Incorrect:

- **A (Custom OCR libraries on Amazon EKS):** Developing custom OCR libraries and deploying them on an Amazon EKS cluster would require significant upfront development and ongoing maintenance efforts. This approach is less cost-effective and would entail a longer time to market compared to using pre-built AWS services like Textract and Comprehend.
- **B (AI/ML models on EC2 for OCR):** Training and hosting AI/ML models on EC2 instances for OCR is more complex and operationally intensive than utilizing pre-built and managed services like Amazon Textract. It would require more time for development, training, and maintenance, which is contrary to the company's goals of minimizing time to market and operational overhead.

- **C (AI/ML models in Amazon SageMaker with output in ElastiCache):** While using Amazon SageMaker for AI/ML models is a viable approach for OCR, storing the output in Amazon ElastiCache and managing an additional application tier on EC2 instances adds unnecessary complexity and operational overhead. This method also does not leverage the fully managed capabilities of services like Textract and Comprehend, which are more aligned with the company's requirements for efficiency and cost-effectiveness.

Question 94

A company is transitioning its on-premise order-processing platform to AWS. The existing system comprises a web frontend on VMs, RabbitMQ for frontend-backend communication, and a Kubernetes-based backend for order processing. The company wishes to migrate without significant modifications to the application architecture. What migration solution would meet these needs while minimizing operational complexity?

- A. Convert the web server VM into an Amazon Machine Image (AMI). Establish an Amazon EC2 Auto Scaling group using this AMI, paired with an Application Load Balancer. Replace the on-premises RabbitMQ with Amazon MQ and utilize Amazon Elastic Kubernetes Service (Amazon EKS) for the containerized order-processing backend.
- B. Develop a custom AWS Lambda runtime to replicate the web server environment. Replace the web front end with an Amazon API Gateway API. Substitute the on-premises RabbitMQ with Amazon MQ and use Amazon EKS for the containerized backend.
- C. Transform the web server VM into an AMI. Create an EC2 Auto Scaling group based on this AMI, along with an Application Load Balancer. Substitute the on-premises RabbitMQ with Amazon MQ and deploy Kubernetes on a separate EC2 instance fleet for the backend.
- D. Convert the web server VM to an AMI. Set up an EC2 Auto Scaling group with the AMI and an Application Load Balancer. Use Amazon Simple Queue Service (Amazon SQS) in place of the on-premises RabbitMQ, and configure Amazon EKS for managing the backend order processing.

Correct Answer and Explanation:

- **A (AMI with Auto Scaling, Amazon MQ, and Amazon EKS):** This solution effectively mirrors the company's existing architecture in the AWS environment with minimal changes and low operational overhead. Creating an AMI from the current VMs allows for a seamless transition of the web front end to AWS, maintaining its current setup. The use of an EC2 Auto Scaling group and an Application Load Balancer ensures scalability and high availability. Replacing RabbitMQ with Amazon MQ is a suitable choice as it offers managed message queuing services compatible with RabbitMQ, facilitating easy migration without major application changes. Finally, adopting Amazon EKS for the Kubernetes-based

backend allows the company to leverage a managed Kubernetes service, reducing the complexity of managing the Kubernetes infrastructure.

Why Other Answers are Incorrect:

- **B (Custom AWS Lambda runtime and API Gateway):** Creating a custom Lambda runtime to mimic the web server environment and using API Gateway to replace the web front end would require significant alterations to the application architecture, contradicting the company's desire to avoid major changes. Additionally, this approach may introduce complexities in adapting the existing VM-based setup to a serverless model.
- **C (AMI with separate EC2-hosted Kubernetes):** While the first part of this solution is aligned with the company's architecture, installing and managing Kubernetes on a separate fleet of EC2 instances introduces higher operational overhead. Managing Kubernetes clusters manually is more complex compared to using a managed service like Amazon EKS.
- **D (AMI and Amazon SQS):** This option replaces RabbitMQ with Amazon SQS, which may not be directly compatible with the existing messaging patterns used in the application. Unlike Amazon MQ, SQS is a different type of messaging service and might require significant application changes to integrate, thereby increasing operational complexity.

Question 95

A solutions architect is working on setting up a client-side encryption mechanism for storing objects in an Amazon S3 bucket, utilizing an AWS KMS Customer Master Key (CMK) for encryption. An IAM policy has been crafted and assigned to an IAM role to facilitate this.

The IAM policy is as follows:

```
1  {
2    "Version": "2012-10-17",
3    "Statement": [
4      {
5        "Sid": "DownloadUpload",
6        "Action": [
7          "s3:GetObject",
8          "s3:GetObjectVersion",
9          "s3:PutObject",
10         "s3:PutObjectAcl"
11        ],
12        "Effect": "Allow",
13        "Resource": "arn:aws:s3:::BucketName/*"
14      },
15      {
16        "Sid": "KMSAccess",
17        "Action": [
```



```

18     "kms:Decrypt",
19     "kms:Encrypt"
20 ],
21     "Effect": "Allow",
22     "Resource": "arn:aws:kms:Region:Account:key/KeyID"
23 }
24 ]
25 }

```

During testing, the architect was able to retrieve existing objects from the S3 bucket without issues. However, they encountered a 'Forbidden' error when trying to upload new objects to the S3 bucket.

Which permission should be added to the IAM policy to satisfy the complete set of requirements?

- A. kms:GenerateDataKey
- B. kms:GetKeyPolicy
- C. kms:GetPublicKey
- D. kms:Sign

The correct answer is:

A. kms:GenerateDataKey

This is the appropriate action to add because the `kms:GenerateDataKey` permission is used to create data keys that are employed for client-side encryption. This action allows the IAM role to get a new data encryption key encrypted under the specified CMK. Since the task involves client-side encryption, the application must be able to generate a new data key for each object uploaded to S3, which it then uses to encrypt the data.

The other options are incorrect for the following reasons:

B. `kms:GetKeyPolicy` is not relevant to the encryption process for S3 objects. It is used to retrieve a key policy, which is not required for the upload process.

C. `kms:GetPublicKey` is used to retrieve the public key from an asymmetric CMK, which is not applicable in this scenario since the context suggests symmetric encryption with S3 and KMS.

D. `kms:Sign` is utilized to digitally sign a message or a digest with a CMK. This operation is not related to the encryption for storing objects in S3.

Question 96

A company has created a web application, which is hosted on a set of Amazon EC2 instances placed behind an Application Load Balancer. To enhance the application's security without mistakenly impeding valid traffic, the company is considering the implementation of AWS WAF web ACLs. What configuration should the solutions architect adopt for the web ACLs to bolster security while safeguarding legitimate traffic?

- A. Initially set the web ACL rules to "Count" mode. Enable AWS WAF logs to monitor incoming requests and check for any incorrect rejections (false positives). Refine the rules to eliminate these inaccuracies. Gradually transition the web ACL rules from "Count" to "Block" as confidence in the rule accuracy increases.
- B. Employ exclusively rate-based rules within the web ACLs, choosing the highest feasible rate limit. Temporarily block requests that surpass this threshold. Construct detailed nested rules to more precisely monitor request rates.
- C. Configure the web ACL rules to "Block" immediately. Use only AWS managed rule groups within the web ACLs. Assess the effectiveness of these groups by reviewing Amazon CloudWatch metrics in conjunction with AWS WAF sampled request data or logs.
- D. Rely solely on tailor-made rule groups in the web ACLs and set them to "Allow." Turn on AWS WAF logging to observe the requests and identify false positives. Tweak the rules to prevent these errors. Eventually, modify the web ACL rules from "Allow" to "Block" after ensuring rule accuracy.

The correct answer is: A.

This is the best option because it allows the solutions architect to safely test and refine WAF rules without disrupting legitimate traffic. By setting the action to "Count," the WAF will tally the matches to the rules without actually blocking the requests. This enables a period of observation and analysis, using WAF logging to identify and correct any false positives. Once the rules are verified to be accurate and free of false positives, they can then be confidently set to "Block" to enforce security measures.

Why the Other Answers are Incorrect:

- B.** Utilizing rate-based rules with a high limit might not effectively improve the security posture as it could allow potentially malicious requests under the threshold. Also, blocking based on rate alone doesn't provide the granularity needed to avoid false positives affecting legitimate traffic.
- C.** Setting the rules to "Block" outright could inadvertently block legitimate traffic, especially if the managed rule groups are not tailored to the specific traffic patterns of the web application. Without testing and refinement, this approach is risky and could lead to business disruption.

D. Using only custom rule groups set to "Allow" would not enhance security because it would not block any traffic. While AWS WAF logging can be used to analyze requests, the strategy does not inherently improve security until the rules are eventually changed to "Block," at which point legitimate traffic might be impacted without prior rule refinement.

Question 97

A company with multiple AWS accounts in AWS Organizations needs a more streamlined approach to managing shared security group rules. These rules include a set of IP CIDR ranges that allow connectivity to the company's internal network. Developers in each AWS account currently add new IP CIDR ranges to their security groups manually, and the security team, which operates from a separate AWS account, informs the account owners about updates to the list. The solutions architect is tasked with devising a system to uniformly distribute the shared CIDR ranges across all accounts in a way that minimizes operational effort.

- A. Implement an Amazon SNS topic in the security team's account and deploy a Lambda function in every AWS account within the organization. Configure this function to update security groups with new IP addresses whenever a message is published to the SNS topic by the security team.
- B. Establish customer-managed prefix lists in each AWS account containing all internal CIDR ranges and inform each account owner to include these lists in their security groups. The security team would communicate updates directly to each account owner.
- C. In the security team's account, create a single customer-managed prefix list with all internal CIDR ranges and share it across the organization using AWS Resource Access Manager. Then, instruct each account owner to update their security groups to reference this shared prefix list.
- D. Set up an IAM role with security group update permissions in all organization accounts and a Lambda function in the security team's account. The Lambda function, upon receiving a list of internal IP addresses, would assume the IAM role in each account and update the security groups accordingly.

The correct answer is: C.

This option is the most efficient because it centralizes the management of the CIDR ranges in one location and leverages AWS Resource Access Manager to share these prefix lists with the entire organization. This method greatly reduces the operational overhead by eliminating the need to manage updates across multiple accounts manually.

Why the Other Answers are Incorrect:

A. While using SNS and Lambda in each account could work, it creates significant overhead by requiring a Lambda function in every single account, which must be maintained and monitored.

B. Managing separate customer-managed prefix lists in each account involves a lot of redundant effort and does not streamline the process. Each account would need to be individually updated, which is not operationally efficient.

D. Creating an IAM role in each account and a central Lambda function introduces complexity, as the function would need permissions to assume roles across all accounts and would represent a single point of failure. It also adds the overhead of maintaining the cross-account trust and permissions.

Question 98

A company has recently updated its work policy to permit employees to work remotely, provided they connect via a VPN. The company's internal applications are spread across VPCs in various AWS accounts, which are currently reachable from the on-premise office network through an established AWS Site-to-Site VPN connection. There are existing peering connections between the main AWS account's VPC and those in other accounts. A solutions architect has been tasked with creating a scalable Client VPN solution that allows remote employees to access these applications.

A. Set up an individual Client VPN endpoint in every AWS account and arrange the necessary routing to permit access to the internal applications.

B. Establish a single Client VPN endpoint in the primary AWS account and configure the necessary routing to ensure access to the internal applications.

C. Implement a Client VPN endpoint in the primary AWS account and deploy a transit gateway connected to each AWS account, with the appropriate routing configurations for access to the internal applications.

D. Install a Client VPN endpoint in the primary AWS account and create a link between this Client VPN endpoint and the existing AWS Site-to-Site VPN.

The correct answer is: B.

This solution is the most cost-effective because it consolidates the Client VPN endpoint into a single account, leveraging the existing peering connections to provide access to applications hosted in various VPCs. This approach minimizes the administrative overhead and cost associated with managing multiple Client VPN endpoints across different accounts.

Why the Other Answers are Incorrect:

A. Creating a Client VPN endpoint in each AWS account would incur unnecessary costs and operational complexity. Managing multiple VPN endpoints and routing configurations is more complex and does not offer any advantages over a centralized solution.

C. While a transit gateway can centralize connectivity across multiple VPCs and AWS accounts, it is not the most cost-effective solution for this scenario. Implementing a transit gateway incurs additional costs and is more suitable for complex networking requirements that are not specified in the current scenario.

D. Linking a Client VPN endpoint directly to a Site-to-Site VPN does not inherently provide client VPN functionality for remote workers. The Client VPN endpoint is intended for individual remote connections, not for linking to a Site-to-Site VPN, which is typically used for connecting entire networks rather than individual clients.

Question 99

A company's application, hosted on AWS, is experiencing unpredictable response times and a rise in error rates due to synchronous calls to external third-party services. The current setup involves directly invoking an AWS Lambda function for these calls. To improve the application's stability, a solutions architect has been tasked with restructuring the architecture to asynchronously handle interactions with the third-party services while ensuring all requests are eventually processed.

A. Implement Amazon SQS to queue the events, which will then trigger the Lambda function. B. Leverage AWS Step Functions to orchestrate event flow to the Lambda function. C. Employ Amazon EventBridge to route events to the Lambda function. D. Utilize Amazon SNS to hold the events, which will subsequently invoke the Lambda function.

The correct answer is: A.

This solution effectively decouples the third-party service calls from the application's primary workflow. By using Amazon SQS as a buffer, the events can be queued, ensuring that the Lambda function can process them at a consistent pace, independent of the third-party service's availability or response time. This approach also provides resilience, as SQS can retain messages if the Lambda function is unable to process them immediately, thereby ensuring that no calls are lost and that they are all eventually completed.

Why the Other Answers are Incorrect:

B. While AWS Step Functions can manage complex workflows, it is not the most straightforward or cost-effective solution for simply decoupling Lambda function invocations. Step Functions is better suited for orchestrating multiple AWS services in scenarios that require complex state management and coordination.

C. Amazon EventBridge is a serverless event bus service that routes events from AWS services to various targets. While it could be used to pass events to Lambda, it does not inherently provide

a queuing mechanism to ensure that all events are eventually processed in case of third-party service delays or failures.

D. Amazon SNS is a publish/subscribe service that can trigger Lambda functions. However, it doesn't offer the same level of message handling and retry capabilities as SQS. If the Lambda function fails to process a message, SNS does not provide a built-in mechanism to retain and retry the message automatically.

Question 100

A company utilizes AWS to run its applications within a structure that includes multiple accounts. Specifically, the sales and marketing teams operate under separate AWS accounts within AWS Organizations. The sales team has accumulated petabytes of data within an Amazon S3 bucket, and this data is secured using encryption with an AWS KMS key. The marketing team, which uses Amazon QuickSight for creating data visualizations, requires access to the sales team's data in the S3 bucket. The marketing team has already configured an IAM service role for QuickSight within their own AWS account. The company is seeking a method to securely share the sales team's S3 data with the marketing team while minimizing administrative effort.

- A. Set up a new S3 bucket within the marketing team's AWS account and establish an S3 replication rule from the sales account to replicate the data. Modify QuickSight's permissions in the marketing account to access the newly created S3 bucket.
- B. Implement a Service Control Policy (SCP) to provide the marketing account access to the S3 bucket. Use AWS Resource Access Manager (AWS RAM) to share the KMS key from the sales account with the marketing account. Update QuickSight's permissions in the marketing account to access the original S3 bucket.
- C. Amend the S3 bucket policy in the sales account to allow access for the marketing account's QuickSight role. Set up a KMS grant for the encryption key used on the S3 bucket, granting decryption privileges to the QuickSight role. Update QuickSight's permissions in the marketing account for S3 bucket access.
- D. Construct an IAM role within the sales team's AWS account with permissions to the S3 bucket. In the marketing account, assume the IAM role from the sales account to gain S3 access. Adjust the QuickSight role to trust the newly established IAM role from the sales account.

The correct answer is:

D. Construct an IAM role within the sales team's AWS account with permissions to the S3 bucket. In the marketing account, assume the IAM role from the sales account to gain S3 access. Adjust the QuickSight role to trust the newly established IAM role from the sales account.

This solution is the most efficient because it allows the marketing team to access the sales team's S3 bucket by assuming a role that has the necessary permissions. This approach leverages AWS's existing role assumption and trust relationship capabilities to facilitate cross-account access without needing to replicate the data or directly share encryption keys. It simplifies the operation by using the existing QuickSight role and modifying it to assume the cross-account role.

Why the Other Answers are Incorrect:

- A.** Creating a new S3 bucket and setting up replication involves unnecessary data duplication and additional storage costs, especially given the large volume of data (petabytes). This approach would significantly increase operational overhead and is not cost-effective.
- B.** AWS Resource Access Manager (AWS RAM) does not support sharing AWS KMS keys. Furthermore, SCPs does not directly grant access to resources; they are used to place guardrails on accounts within an organization. This option is not feasible with the current AWS service capabilities.
- C.** S3 bucket policies are configured within the account that owns the bucket. The policy in the marketing account cannot grant access to resources in the sales account. Additionally, the marketing team's QuickSight role would need permissions to assume a role that has access to the sales account's resources, which is not addressed in this option.