AWS Certified Solutions Architect Associate Practice Test 2 - Results

Return to review
Chart
Pie chart with 4 slices.
End of interactive chart.
Attempt 2
All knowledge areas
All questions
Question 1: **Correct**
A company has a top priority requirement to monitor a few database metrics and then afterward, send email notifications to the Operations team in case there is an issue. Which AWS services can accomplish this requirement? (Select TWO.)

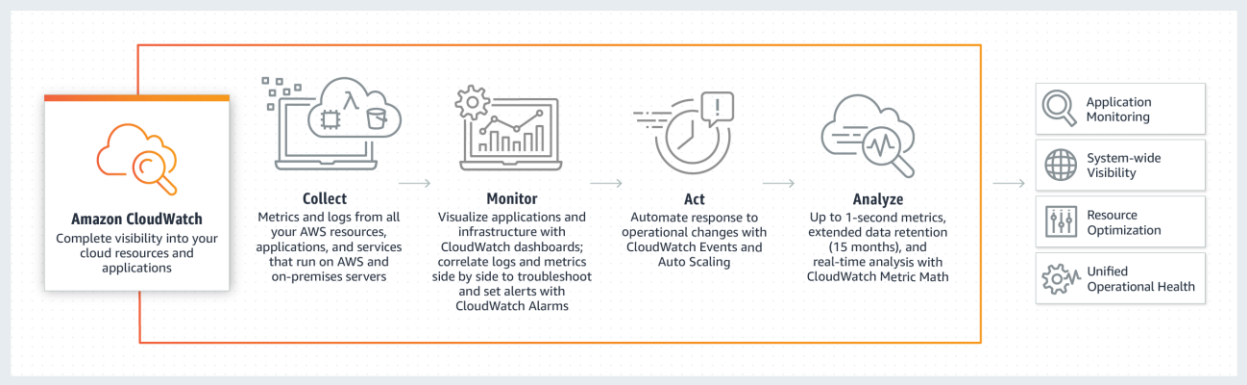
- ☐ **Amazon Simple Queue Service (SQS)**
- ☐ **Amazon EC2 Instance with a running Berkeley Internet Name Domain (BIND) Server.**
- ☐ **Amazon Simple Email Service**
- ☑ **Amazon CloudWatch**

  (Correct)
- ☑ **Amazon Simple Notification Service (SNS)**

  (Correct)

Explanation
Amazon CloudWatch and **Amazon Simple Notification Service (SNS)** are correct. In this requirement, you can use Amazon CloudWatch to monitor the database and then Amazon SNS to send the emails to the Operations team. Take note that you should use

SNS instead of SES (Simple Email Service) when you want to monitor your EC2 instances.



CloudWatch collects monitoring and operational data in the form of logs, metrics, and events, providing you with a unified view of AWS resources, applications, and services that run on AWS, and on-premises servers.

SNS is a highly available, durable, secure, fully managed pub/sub messaging service that enables you to decouple microservices, distributed systems, and serverless applications.

Amazon Simple Email Service is incorrect. SES is a cloud-based email sending service designed to send notifications and transactional emails.

Amazon Simple Queue Service (SQS) is incorrect. SQS is a fully-managed message queuing service. It does not monitor applications nor send email notifications, unlike SES.

Amazon EC2 Instance with a running Berkeley Internet Name Domain (BIND) Server is incorrect because BIND is primarily used as a Domain Name System (DNS) web service. This is only applicable if you have a private hosted zone in your AWS account. It does not monitor applications nor send email notifications.

References:

https://aws.amazon.com/cloudwatch/

https://aws.amazon.com/sns/

Check out this Amazon CloudWatch Cheat Sheet:

https://tutorialsdojo.com/amazon-cloudwatch/

Question 2: **Correct**
A start-up company that offers an intuitive financial data analytics service has consulted you about their AWS architecture. They have a fleet of Amazon EC2 worker instances that process financial data and then outputs reports which are used by their clients. You must store the generated report files in a durable storage. The number of files to be stored can grow over time as the start-up company is expanding rapidly overseas and hence, they also need a way to distribute the reports faster to clients located across the globe.

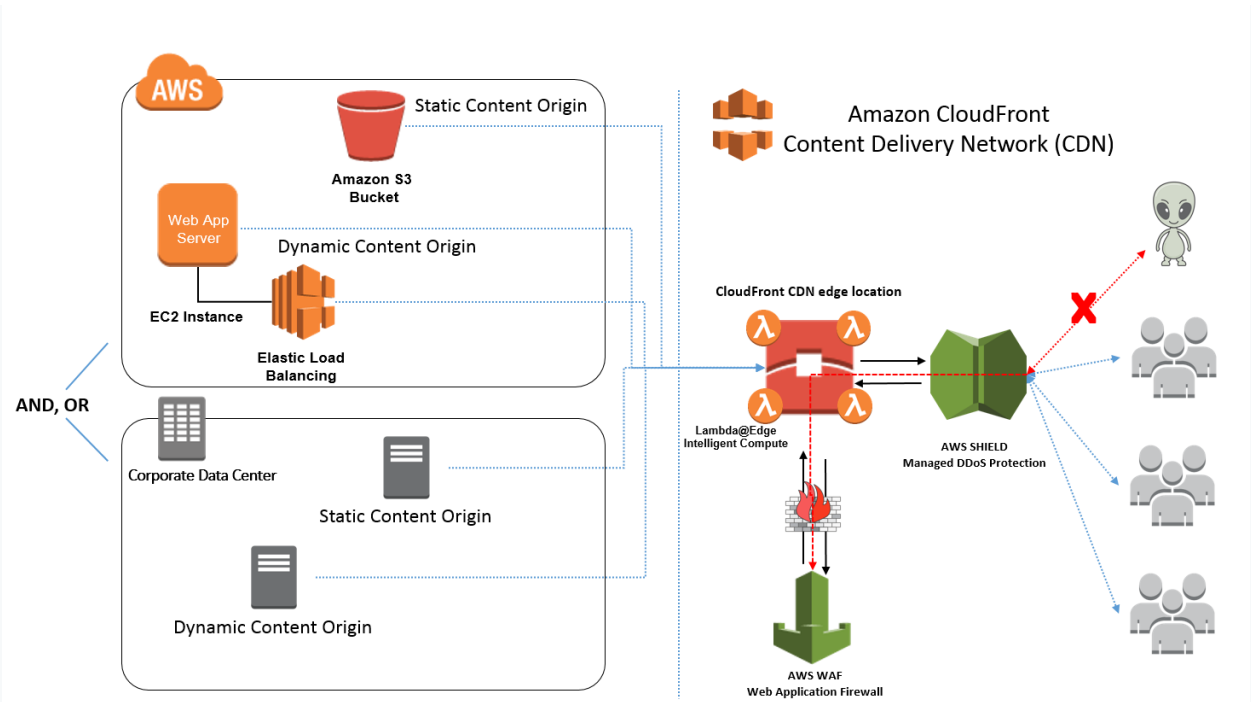
Which of the following is a cost-efficient and scalable storage option that you should use for this scenario?

- **Use Amazon Glacier as the data storage and ElastiCache as the CDN.**
- **Use Amazon Redshift as the data storage and CloudFront as the CDN.**
- **Use Amazon S3 as the data storage and CloudFront as the CDN.**

  (Correct)
- **Use multiple EC2 instance stores for data storage and ElastiCache as the CDN.**

Explanation
A Content Delivery Network (CDN) is a critical component of nearly any modern web application. It used to be that CDN merely improved the delivery of content by replicating commonly requested files (static content) across a globally distributed set of caching servers. However, CDNs have become much more useful over time.

For caching, a CDN will reduce the load on an application origin and improve the experience of the requestor by delivering a local copy of the content from a nearby cache edge, or Point of Presence (PoP). The application origin is off the hook for opening the connection and delivering the content directly as the CDN takes care of the heavy lifting. The end result is that the application origins don't need to scale to meet demands for static content.



Amazon CloudFront is a fast content delivery network (CDN) service that securely delivers data, videos, applications, and APIs to customers globally with low latency, high transfer speeds, all within a developer-friendly environment. CloudFront is integrated with AWS – both physical locations that are directly connected to the AWS global infrastructure, as well as other AWS services.

Amazon S3 offers a highly durable, scalable, and secure destination for backing up and archiving your critical data. This is the correct option as the start-up company is looking for a durable storage to store the audio and text files. In addition, ElastiCache is only used for caching and not specifically as a Global Content Delivery Network (CDN).

Using Amazon Redshift as the data storage and CloudFront as the CDN is incorrect as Amazon Redshift is usually used as a Data Warehouse.

Using Amazon S3 Glacier as the data storage and ElastiCache as the CDN is incorrect as Amazon S3 Glacier is usually used for data archives.

Using multiple EC2 instance stores for data storage and ElastiCache as the CDN is incorrect as data stored in an instance store is not durable.

References:

https://aws.amazon.com/s3/

https://aws.amazon.com/caching/cdn/

Check out this Amazon S3 Cheat Sheet:

https://tutorialsdojo.com/amazon-s3/

Tutorials Dojo's AWS Certified Solutions Architect Associate Exam Study Guide:

https://tutorialsdojo.com/aws-certified-solutions-architect-associate/

Question 3: **Incorrect**
A company is building an internal application that serves as a repository for images uploaded by a couple of users. Whenever a user uploads an image, it would be sent to Kinesis Data Streams for processing before it is stored in an S3 bucket. If the upload was successful, the application will return a prompt informing the user that the operation was successful. The entire processing typically takes about 5 minutes to finish.

Which of the following options will allow you to asynchronously process the request to the application from upload request to Kinesis, S3, and return a reply in the most cost-effective manner?

- **Use a combination of SNS to buffer the requests and then asynchronously process them using On-Demand EC2 Instances.**
- **Use a combination of Lambda and Step Functions to orchestrate service components and asynchronously process the requests.**

  (Incorrect)
- **Use a combination of SQS to queue the requests and then asynchronously process them using On-Demand EC2 Instances.**

- **Replace the Kinesis Data Streams with an Amazon SQS queue. Create a Lambda function that will asynchronously process the requests.**

  (Correct)

Explanation

AWS Lambda supports the synchronous and asynchronous invocation of a Lambda function. You can control the invocation type only when you invoke a Lambda function. When you use an AWS service as a trigger, the invocation type is predetermined for each service. You have no control over the invocation type that these event sources use when they invoke your Lambda function. Since processing only takes 5 minutes, Lambda is also a cost-effective choice.

You can use an AWS Lambda function to process messages in an Amazon Simple Queue Service (Amazon SQS) queue. Lambda event source mappings support standard queues and first-in, first-out (FIFO) queues. With Amazon SQS, you can offload tasks from one component of your application by sending them to a queue and processing them asynchronously.

Kinesis Data Streams is a real-time data streaming service that requires the provisioning of shards. Amazon SQS is a cheaper option because you only pay for what you use. Since there is no requirement for real-time processing in the scenario given, replacing Kinesis Data Streams with Amazon SQS would save more costs.

Hence, the correct answer is: **Replace the Kinesis stream with an Amazon SQS queue. Create a Lambda function that will asynchronously process the requests.**

Using a combination of Lambda and Step Functions to orchestrate service components and asynchronously process the requests is incorrect. The AWS Step Functions service lets you coordinate multiple AWS services into serverless workflows so you can build and update apps quickly. Although this can be a valid solution, it is not cost-effective since the application does not have a lot of components to orchestrate.

Lambda functions can effectively meet the requirements in this scenario without using Step Functions. This service is not as cost-effective as Lambda.

Using a combination of SQS to queue the requests and then asynchronously processing them using On-Demand EC2 Instances and **Using a combination of SNS to buffer the requests and then asynchronously processing them using On-Demand EC2 Instances** are both incorrect as using On-Demand EC2 instances is not cost-effective. It is better to use a Lambda function instead.

References:

https://docs.aws.amazon.com/lambda/latest/dg/welcome.html

https://docs.aws.amazon.com/lambda/latest/dg/lambda-invocation.html

https://aws.amazon.com/blogs/compute/new-aws-lambda-controls-for-stream-processing-and-asynchronous-invocations/

AWS Lambda Overview - Serverless Computing in AWS:
https://youtu.be/bPVX1zHwAnY

Tutorials Dojo's AWS Certified Solutions Architect Associate Exam Study Guide:

https://tutorialsdojo.com/aws-certified-solutions-architect-associate/

Question 4: **Correct**
A company owns a photo-sharing app that stores user uploads on Amazon S3. There has been an increase in the number of explicit and offensive images being reported. The company currently relies on human efforts to moderate content, and they want to streamline this process by using Artificial Intelligence to only flag images for review. For added security, any communication with your resources on your Amazon VPC must not traverse the public Internet.

How can this task be accomplished with the LEAST amount of effort?

- **Use Amazon Rekognition to detect images with graphic nudity or violence in Amazon S3. Create an Interface VPC endpoint for Amazon Rekognition with**

the necessary policies to prevent any traffic from traversing the public Internet.

(Correct)

- **Use an image classification model in Amazon SageMaker. Set up Amazon GuardDuty and connect it with Amazon SageMaker to ensure that all communications do not traverse the public Internet.**

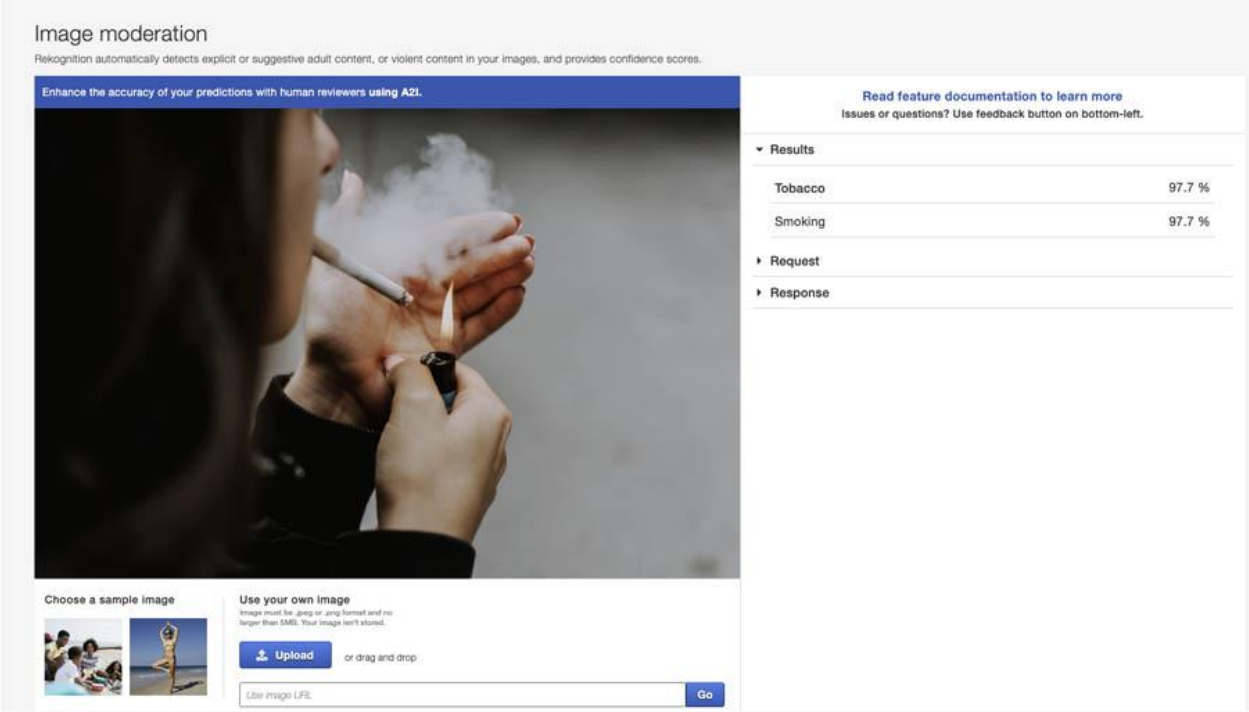
- **Use Amazon Monitron to monitor each user upload in S3. Use the AWS Transit Gateway Network Manager to block any outbound requests to the public Internet.**

- **Use Amazon Detective to detect images with graphic nudity or violence in Amazon S3. Ensure that all communications made by your AWS resources do not traverse the public Internet via the AWS Audit Manager service.**

Explanation

Amazon Rekognition can help you streamline or automate image and video moderation workflows using machine learning. Using fully managed image and video moderation APIs, you can proactively detect inappropriate, unwanted, or offensive content containing nudity, suggestiveness, violence, and other such categories.

Amazon Rekognition returns a hierarchical taxonomy of moderation-related labels that make it easy for you to define granular business rules as per your own Standards and Practices (S&P), User Safety, or compliance guidelines - without requiring any machine learning experience.



If you use Amazon Virtual Private Cloud (Amazon VPC) to host your AWS resources, you can establish a private connection between your VPC and Amazon Rekognition. You can use this connection to enable Amazon Rekognition to communicate with your resources on your VPC without going through the public internet.

To connect your VPC to Amazon Rekognition, you define an interface VPC endpoint for Amazon Rekognition. An interface endpoint is an elastic network interface with a private IP address that serves as an entry point for traffic destined to a supported AWS service. The endpoint provides reliable, scalable connectivity to Amazon Rekognition—and it doesn't require an internet gateway, a network address translation (NAT) instance, or a VPN connection.

In this scenario, it is best to use Amazon Rekognition to automatically analyze images for you instead of manually scanning them and tagging those that you find offensive. Of course, this is not a holy grail solution, as you'd still have to go over those flagged images for further review, but it would definitely help speed up the process of content moderation.

Hence, the correct answer is: **Use Amazon Rekognition to detect images with graphic nudity or violence in Amazon S3. Create an Interface VPC endpoint for Amazon Rekognition with the necessary policies to prevent any traffic from traversing the public Internet.**

The option that says: **Use an image classification model in Amazon SageMaker. Set up Amazon GuardDuty and connect it with Amazon SageMaker to ensure that all**

communications do not traverse the public Internet is incorrect. Using Amazon SageMaker will require you to actually train a machine learning model; it does not come off the shelf, unlike Amazon Rekognition. Take note that the scenario explicitly mentioned that the task must be accomplished with the least amount of effort. In addition, the Amazon GuardDuty service is not capable of ensuring that all traffic in Amazon SageMaker is private. Amazon GuardDuty is primarily used as an intelligent threat detection solution and not a networking service.

The option that says: **Use Amazon Detective to detect images with graphic nudity or violence in Amazon S3. Ensure that all communications made by your AWS resources do not traverse the public Internet via the AWS Audit Manager service** is incorrect. Amazon Detective is commonly used to analyze, investigate, and quickly identify the root cause of potential security issues in your AWS workloads, as well as for detecting suspicious activities. This service can't detect any graphic images. Moreover, the AWS Audit Manager just continuously audits your AWS usage to simplify how you assess risk and compliance with regulations and industry standards. The AWS Audit Manager, by itself, cannot halt any outbound traffic traversing the public Internet from your VPC.

The option that says **Use Amazon Monitron to monitor each user upload in S3. Use the AWS Transit Gateway Network Manager to block any outbound requests to the public Internet** is incorrect. Amazon Monitron is simply a service that detects abnormal conditions in industrial equipment such as fans, compressors, motors, etc. In addition, the AWS Transit Gateway Network Manager is simply a feature of AWS Transit Gateway that centralizes the management and monitoring of networking resources and connections to remote branch locations.

References:

https://aws.amazon.com/rekognition/content-moderation/

https://aws.amazon.com/blogs/machine-learning/amazon-rekognition-adds-support-for-six-new-content-moderation-categories/

https://docs.aws.amazon.com/rekognition/latest/dg/vpc.html

Check out this Amazon Rekognition Cheat Sheet:

https://tutorialsdojo.com/amazon-rekognition/

Question 5: **Incorrect**

A company has an application that continually sends encrypted documents to Amazon S3. The company requires that the configuration for data access is in line with their strict compliance standards. They should also be alerted if there is any risk of unauthorized access or suspicious access patterns.

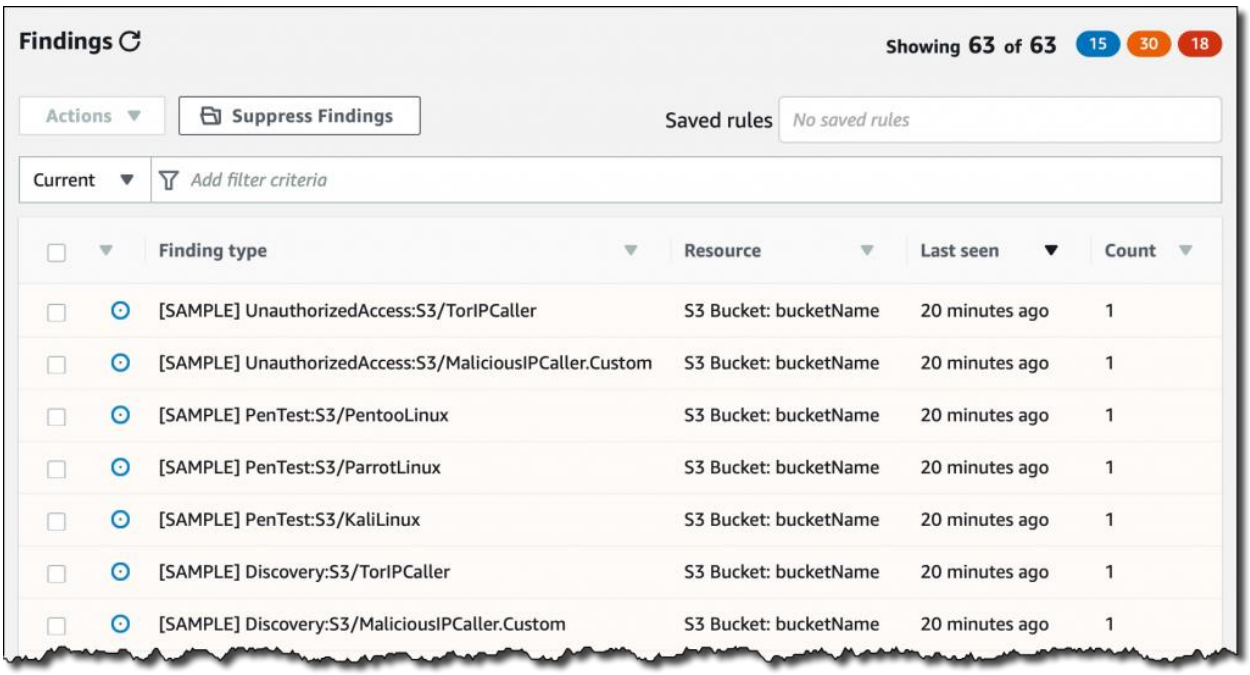
Which step is needed to meet the requirements?

- **Use Amazon Inspector to alert whenever a security violation is detected on S3.**
- **Use Amazon Rekognition to monitor and recognize patterns on S3.**
- **Use Amazon Macie to monitor and detect access patterns on S3.**

  (Incorrect)
- **Use Amazon GuardDuty to monitor malicious activity on S3.**

  (Correct)

Explanation
Amazon GuardDuty can generate findings based on suspicious activities such as requests coming from known malicious IP addresses, changing of bucket policies/ACLs to expose an S3 bucket publicly, or suspicious API call patterns that attempt to discover misconfigured bucket permissions.



To detect possibly malicious behavior, GuardDuty uses a combination of anomaly detection, machine learning, and continuously updated threat intelligence.

Hence, the correct answer is: **Use Amazon GuardDuty to monitor malicious activity on S3.**

The option that says: **Use Amazon Rekognition to monitor and recognize patterns on S3** is incorrect because Amazon Rekognition is simply a service that can identify the objects, people, text, scenes, and activities on your images or videos, as well as detect any inappropriate content.

The option that says: **Use Amazon Macie to monitor and detect access patterns on S3** is incorrect because Macie cannot detect usage patterns on S3 data. While Amazon Macie is capable of detecting policy changes in S3 buckets, this is not enough to detect unauthorized or suspicious access patterns.

The option that says: **Use Amazon Inspector to alert whenever a security violation is detected on S3** is incorrect because Inspector is basically an automated security assessment service that helps improve the security and compliance of applications deployed on AWS.

References:

https://aws.amazon.com/guardduty/

https://aws.amazon.com/blogs/aws/new-using-amazon-guardduty-to-protect-your-s3-buckets/

Check out this Amazon GuardDuty Cheat Sheet:

https://tutorialsdojo.com/amazon-guardduty/

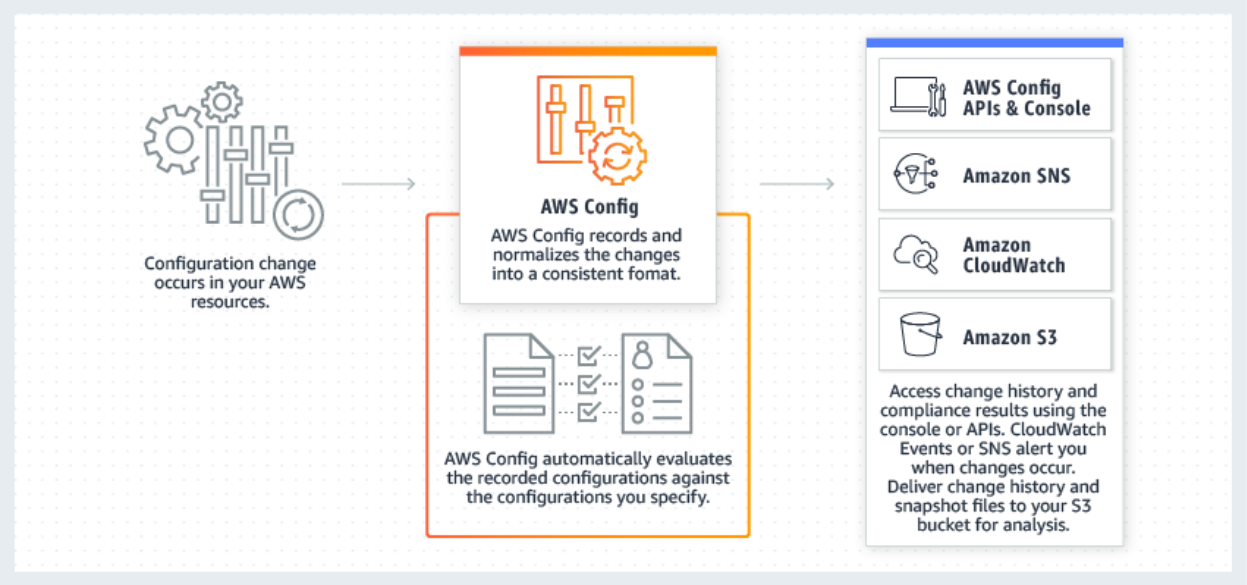
Question 6: **Correct**
A company needs to assess and audit all the configurations in their AWS account. It must enforce strict compliance by tracking all configuration changes made to any of its Amazon S3 buckets. Publicly accessible S3 buckets should also be identified automatically to avoid data breaches.

Which of the following options will meet this requirement?

- **Use AWS Trusted Advisor to analyze your AWS environment.**
- **Use AWS CloudTrail and review the event history of your AWS account.**
- **Use AWS IAM to generate a credential report.**
- **Use AWS Config to set up a rule in your AWS account.**

  (Correct)

Explanation
AWS Config is a service that enables you to assess, audit, and evaluate the configurations of your AWS resources. Config continuously monitors and records your AWS resource configurations and allows you to automate the evaluation of recorded configurations against desired configurations. With Config, you can review changes in configurations and relationships between AWS resources, dive into detailed resource configuration histories, and determine your overall compliance against the configurations specified in your internal guidelines. This enables you to simplify compliance auditing, security analysis, change management, and operational troubleshooting.



You can use AWS Config to evaluate the configuration settings of your AWS resources. By creating an AWS Config rule, you can enforce your ideal configuration in your AWS account. It also checks if the applied configuration in your resources violates any of the conditions in your rules. The AWS Config dashboard shows the compliance status of your rules and resources. You can verify if your resources comply with your desired configurations and learn which specific resources are noncompliant.

Hence, the correct answer is: **Use AWS Config to set up a rule in your AWS account**.

The option that says: **Use AWS Trusted Advisor to analyze your AWS environment** is incorrect because AWS Trusted Advisor only provides best practice recommendations. It cannot define rules for your AWS resources.

The option that says: **Use AWS IAM to generate a credential report** is incorrect because this report will not help you evaluate resources. The IAM credential report is just a list of all IAM users in your AWS account.

The option that says: **Use AWS CloudTrail and review the event history of your AWS account** is incorrect. Although it can track changes and store a history of what happened to your resources, this service still cannot enforce rules to comply with your organization's policies.

References:

https://aws.amazon.com/config/

https://docs.aws.amazon.com/config/latest/developerguide/evaluate-config.html

Check out this AWS Config Cheat Sheet:

https://tutorialsdojo.com/aws-config/

Tutorials Dojo's AWS Certified Solutions Architect Associate Exam Study Guide:

https://tutorialsdojo.com/aws-certified-solutions-architect-associate-saa-c02/

Question 7: **Correct**

A company has two On-Demand EC2 instances inside the Virtual Private Cloud in the same Availability Zone but are deployed to different subnets. One EC2 instance is running a database and the other EC2 instance a web application that connects with the database. You need to ensure that these two instances can communicate with each other for the system to work properly.

What are the things you have to check so that these EC2 instances can communicate inside the VPC? (Select TWO.)

- ☑ **Check the Network ACL if it allows communication between the two subnets.**

  (Correct)

- ☐ **Check if the default route is set to a NAT instance or Internet Gateway (IGW) for them to communicate.**

- ☐ **Ensure that the EC2 instances are in the same Placement Group.**

- ☑ **Check if all security groups are set to allow the application host to communicate to the database on the right port and protocol.**

  (Correct)

- ☐ **Check if both instances are the same instance class.**

Explanation
First, the Network ACL should be properly set to allow communication between the two subnets. The security group should also be properly configured so that your web server can communicate with the database server.

Security Group	Network Access Control List
Acts as a firewall for associated Amazon EC2 instances	Acts as a firewall for associated subnets
Controls both inbound and outbound traffic at the instance level	Controls both inbound and outbound traffic at the subnet level
You can secure your VPC instances using only security groups	Network ACLs are an additional layer of defense.
Supports allow rules only	Supports allow rules and deny rules
Stateful (Return traffic is automatically allowed, regardless of any rules)	Stateless (Return traffic must be explicitly allowed by rules)
Evaluates all rules before deciding whether to allow traffic	Evaluates rules in number order when deciding whether to allow traffic, starting with the lowest numbered rule.
Applies only to the instance that is associated to it	Applies to all instances in the subnet it is associated with
Has separate rules for inbound and outbound traffic	Has separate rules for inbound and outbound traffic
A newly created security group denies all inbound traffic by default	A newly created nACL denies all inbound traffic by default
A newly created security group has an outbound rule that allows all outbound traffic by default	A newly created nACL denies all outbound traffic default
Instances associated with a security group can't talk to each other unless you add rules allowing it	Each subnet in your VPC must be associated with a network ACL. If none is associated, the default nACL is selected.
Security groups are associated with network interfaces	You can associate a network ACL with multiple subnets; however, a subnet can be associated with only one network ACL at a time.

Hence, these are the correct answers:

1. **Check if all security groups are set to allow the application host to communicate to the database on the right port and protocol.**

2. **Check the Network ACL if it allows communication between the two subnets.**

The option that says: **Check if both instances are the same instance class** is incorrect because the EC2 instances do not need to be of the same class in order to communicate with each other.

The option that says: **Check if the default route is set to a NAT instance or Internet Gateway (IGW) for them to communicate** is incorrect because an Internet gateway is primarily used to communicate to the Internet.

The option that says: **Ensure that the EC2 instances are in the same Placement Group** is incorrect because Placement Group is mainly used to provide low-latency network performance necessary for tightly-coupled node-to-node communication.

Reference:

http://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_Subnets.html

Check out this Amazon VPC Cheat Sheet:

https://tutorialsdojo.com/amazon-vpc/

Tutorials Dojo's AWS Certified Solutions Architect Associate Exam Study Guide:

https://tutorialsdojo.com/aws-certified-solutions-architect-associate/

Question 8: **Correct**
A company plans to conduct a network security audit. The web application is hosted on an Auto Scaling group of EC2 Instances with an Application Load Balancer in front to evenly distribute the incoming traffic. A Solutions Architect has been tasked to enhance the security posture of the company's cloud infrastructure and minimize the impact of DDoS attacks on its resources.

Which of the following is the most effective solution that should be implemented?

- 

Configure Amazon CloudFront distribution and set Application Load Balancer as the origin. Create a rate-based web ACL rule using AWS WAF and associate it with Amazon CloudFront.

(Correct)

- **Configure Amazon CloudFront distribution and set a Network Load Balancer as the origin. Use Amazon GuardDuty to block suspicious hosts based on its security findings. Set up a custom AWS Lambda function that processes the security logs and invokes Amazon SNS for notification.**

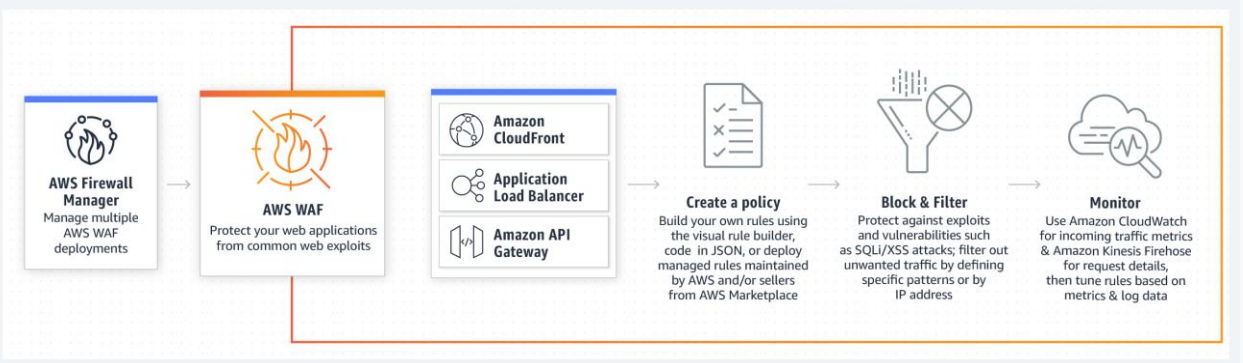
- **Configure Amazon CloudFront distribution and set a Network Load Balancer as the origin. Use VPC Flow Logs to monitor abnormal traffic patterns. Set up a custom AWS Lambda function that processes the flow logs and invokes Amazon SNS for notification.**

- **Configure Amazon CloudFront distribution and set an Application Load Balancer as the origin. Create a security group rule and deny all the suspicious addresses. Use Amazon SNS for notification.**

Explanation

AWS WAF is a web application firewall that helps protect your web applications or APIs against common web exploits that may affect availability, compromise security, or consume excessive resources. AWS WAF gives you control over how traffic reaches your applications by enabling you to create security rules that block common attack patterns, such as SQL injection or cross-site scripting, and rules that filter out specific traffic patterns you define. You can deploy AWS WAF on Amazon CloudFront as part of your CDN solution, the Application Load Balancer that fronts your web servers or origin servers running on EC2, or Amazon API Gateway for your APIs.

To detect and mitigate DDoS attacks, you can use **AWS WAF** in addition to AWS Shield. AWS WAF is a web application firewall that helps detect and mitigate web application layer DDoS attacks by inspecting traffic inline. Application layer DDoS attacks use well-formed but malicious requests to evade mitigation and consume application resources. You can define custom security rules that contain a set of conditions, rules, and actions to block attacking traffic. After you define web ACLs, you can apply them to CloudFront distributions, and web ACLs are evaluated in the priority order you specified when you configured them.



By using AWS WAF, you can configure web access control lists (Web ACLs) on your CloudFront distributions or Application Load Balancers to filter and block requests based on request signatures. Each Web ACL consists of rules that you can configure to string match or regex match one or more request attributes, such as the URI, query-string, HTTP method, or header key. In addition, by using AWS WAF's rate-based rules, you can automatically block the IP addresses of bad actors when requests matching a rule exceed a threshold that you define. Requests from offending client IP addresses will receive 403 Forbidden error responses and will remain blocked until request rates drop below the threshold. This is useful for mitigating HTTP flood attacks that are disguised as regular web traffic.

It is recommended that you add web ACLs with rate-based rules as part of your AWS Shield Advanced protection. These rules can alert you to sudden spikes in traffic that might indicate a potential DDoS event. A rate-based rule counts the requests that arrive from any individual address in any five-minute period. If the number of requests exceeds the limit that you define, the rule can trigger an action such as sending you a notification.

Hence, the correct answer is: **Configure Amazon CloudFront distribution and set Application Load Balancer as the origin. Create a rate-based web ACL rule using AWS WAF and associate it with Amazon CloudFront.**

The option that says: **Configure Amazon CloudFront distribution and set a Network Load Balancer as the origin. Use VPC Flow Logs to monitor abnormal traffic patterns. Set up a custom AWS Lambda function that processes the flow logs and invokes Amazon SNS for notification** is incorrect because this option only allows you to monitor the traffic that is reaching your instance. You can't use VPC Flow Logs to mitigate DDoS attacks.

The option that says: **Configure Amazon CloudFront distribution and set an Application Load Balancer as the origin. Create a security group rule and deny all the suspicious addresses. Use Amazon SNS for notification** is incorrect. To deny suspicious addresses, you must manually insert the IP addresses of these hosts. This is a manual task which is not a sustainable solution. Take note that attackers generate large

volumes of packets or requests to overwhelm the target system. Using a security group in this scenario won't help you mitigate DDoS attacks.

The option that says: **Configure Amazon CloudFront distribution and set a Network Load Balancer as the origin. Use Amazon GuardDuty to block suspicious hosts based on its security findings. Set up a custom AWS Lambda function that processes the security logs and invokes Amazon SNS for notification** is incorrect because Amazon GuardDuty is just a threat detection service. You should use AWS WAF and create your own AWS WAF rate-based rules for mitigating HTTP flood attacks that are disguised as regular web traffic.

References:

https://docs.aws.amazon.com/waf/latest/developerguide/ddos-overview.html

https://docs.aws.amazon.com/waf/latest/developerguide/ddos-get-started-rate-based-rules.html

https://d0.awsstatic.com/whitepapers/Security/DDoS_White_Paper.pdf

Check out this AWS WAF Cheat Sheet:

https://tutorialsdojo.com/aws-waf/

AWS Security Services Overview - WAF, Shield, CloudHSM, KMS:

https://youtu.be/-1S-RdeAmMo

Question 9: **Correct**
A company plans to migrate its suite of containerized applications running on-premises to a container service in AWS. The solution must be cloud-agnostic and use an open-source platform that can automatically manage containerized workloads and services. It should also use the same configuration and tools across various production environments.

What should the Solution Architect do to properly migrate and satisfy the given requirement?

- 

Migrate the application to Amazon Elastic Container Service with ECS tasks that use the AWS Fargate launch type.

- **Migrate the application to Amazon Elastic Kubernetes Service with EKS worker nodes.**

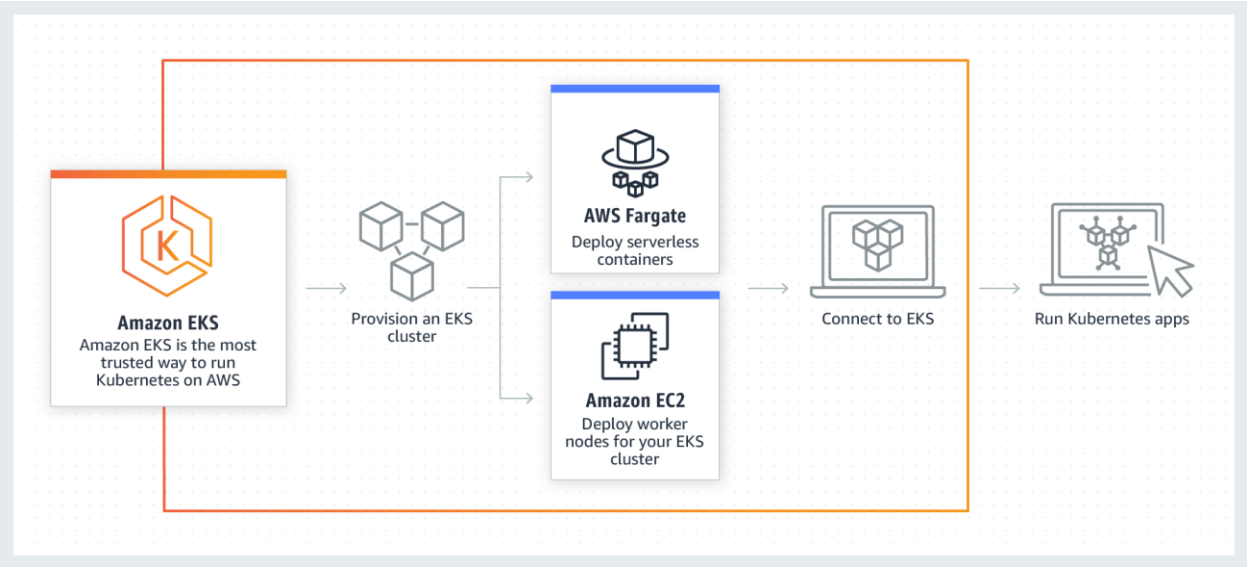
  (Correct)

- **Migrate the application to Amazon Elastic Container Service with ECS tasks that use the Amazon EC2 launch type.**

- **Migrate the application to Amazon Container Registry (ECR) with Amazon EC2 instance worker nodes.**

Explanation

Amazon EKS provisions and scales the Kubernetes control plane, including the API servers and backend persistence layer, across multiple AWS availability zones for high availability and fault tolerance. Amazon EKS automatically detects and replaces unhealthy control plane nodes and provides patching for the control plane. Amazon EKS is integrated with many AWS services to provide scalability and security for your applications. These services include Elastic Load Balancing for load distribution, IAM for authentication, Amazon VPC for isolation, and AWS CloudTrail for logging.



To migrate the application to a container service, you can use Amazon ECS or Amazon EKS. But the key point in this scenario is a cloud-agnostic and open-source platforms. Take note that Amazon ECS is an AWS proprietary container service. This means that it is not an open-source platform. Amazon EKS is a portable, extensible, and open-source platform for managing containerized workloads and services. Kubernetes is considered cloud-agnostic because it allows you to move your containers to other cloud service providers.

Amazon EKS runs up-to-date versions of the open-source Kubernetes software, so you can use all of the existing plugins and tools from the Kubernetes community. Applications running on Amazon EKS are fully compatible with applications running on any standard Kubernetes environment, whether running in on-premises data centers or public clouds. This means that you can easily migrate any standard Kubernetes application to Amazon EKS without any code modification required.

Hence, the correct answer is: **Migrate the application to Amazon Elastic Kubernetes Service with EKS worker nodes.**

The option that says: **Migrate the application to Amazon Container Registry (ECR) with Amazon EC2 instance worker nodes** is incorrect because Amazon ECR is just a fully-managed Docker container registry. Also, this option is not an open-source platform that can manage containerized workloads and services.

The option that says: **Migrate the application to Amazon Elastic Container Service with ECS tasks that use the AWS Fargate launch type** is incorrect because it is stated in the scenario that you have to migrate the application suite to an open-source platform. AWS Fargate is just a serverless compute engine for containers. It is not cloud-agnostic since you cannot use the same configuration and tools if you move it to another cloud service provider such as Microsoft Azure or Google Cloud Platform (GCP).

The option that says: **Migrate the application to Amazon Elastic Container Service with ECS tasks that use the Amazon EC2 launch type** is incorrect because Amazon ECS is an AWS proprietary managed container orchestration service. You should use Amazon EKS since Kubernetes is an open-source platform and is considered cloud-agnostic. With Kubernetes, you can use the same configuration and tools that you're currently using in AWS even if you move your containers to another cloud service provider.

References:

https://docs.aws.amazon.com/eks/latest/userguide/what-is-eks.html

https://aws.amazon.com/eks/faqs/

Check out our library of AWS Cheat Sheets:

https://tutorialsdojo.com/links-to-all-aws-cheat-sheets/

Question 10: **Correct**
An insurance company plans to implement a message filtering feature in their web application. To implement this solution, they need to create separate Amazon SQS queues for each type of quote request. The entire message processing should not exceed 24 hours.

As the Solutions Architect of the company, which of the following should you do to meet the above requirement?

- **Create a data stream in Amazon Kinesis Data Streams. Use the Amazon Kinesis Client Library to deliver all the records to the designated SQS queues based on the quote request type.**

- **Create one Amazon SNS topic and configure the Amazon SQS queues to subscribe to the SNS topic. Publish the same messages to all SQS queues. Filter the messages in each queue based on the quote request type.**

- **Create one Amazon SNS topic and configure the Amazon SQS queues to subscribe to the SNS topic. Set the filter policies in the SNS subscriptions to publish the message to the designated SQS queue based on its quote request type.**

  (Correct)

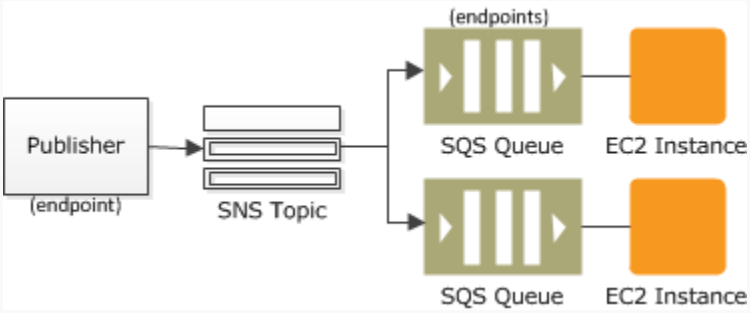
- **Create multiple Amazon SNS topics and configure the Amazon SQS queues to subscribe to the SNS topics. Publish the message to the designated SQS queue based on the quote request type.**

Explanation
Amazon SNS is a fully managed pub/sub messaging service. With Amazon SNS, you can use topics to simultaneously distribute messages to multiple subscribing endpoints

such as Amazon SQS queues, AWS Lambda functions, HTTP endpoints, email addresses, and mobile devices (SMS, Push).

Amazon SQS is a message queue service used by distributed applications to exchange messages through a polling model. It can be used to decouple sending and receiving components without requiring each component to be concurrently available.

A fanout scenario occurs when a message published to an SNS topic is replicated and pushed to multiple endpoints, such as Amazon SQS queues, HTTP(S) endpoints, and Lambda functions. This allows for parallel asynchronous processing.



For example, you can develop an application that publishes a message to an SNS topic whenever an order is placed for a product. Then, two or more SQS queues that are subscribed to the SNS topic receive identical notifications for the new order. An Amazon Elastic Compute Cloud (Amazon EC2) server instance attached to one of the SQS queues can handle the processing or fulfillment of the order. And you can attach another Amazon EC2 server instance to a data warehouse for analysis of all orders received.

By default, an Amazon SNS topic subscriber receives every message published to the topic. You can use Amazon SNS message filtering to assign a filter policy to the topic subscription, and the subscriber will only receive a message that they are interested in. Using Amazon SNS and Amazon SQS together, messages can be delivered to applications that require immediate notification of an event. This method is known as fanout to Amazon SQS queues.

Hence, the correct answer is: **Create one Amazon SNS topic and configure the Amazon SQS queues to subscribe to the SNS topic. Set the filter policies in the SNS subscriptions to publish the message to the designated SQS queue based on its quote request type.**

The option that says: **Create one Amazon SNS topic and configure the Amazon SQS queues to subscribe to the SNS topic. Publish the same messages to all SQS queues. Filter the messages in each queue based on the quote request type** is incorrect because this option will distribute the same messages on all SQS queues instead of its designated queue. You need to fan-out the messages to multiple SQS queues using a

filter policy in Amazon SNS subscriptions to allow parallel asynchronous processing. By doing so, the entire message processing will not exceed 24 hours.

The option that says: **Create multiple Amazon SNS topics and configure the Amazon SQS queues to subscribe to the SNS topics. Publish the message to the designated SQS queue based on the quote request type** is incorrect because to implement the solution asked in the scenario, you only need to use one Amazon SNS topic. To publish it to the designated SQS queue, you must set a filter policy that allows you to fanout the messages. If you didn't set a filter policy in Amazon SNS, the subscribers would receive all the messages published to the SNS topic. Thus, using multiple SNS topics is not an appropriate solution for this scenario.

The option that says: **Create a data stream in Amazon Kinesis Data Streams. Use the Amazon Kinesis Client Library to deliver all the records to the designated SQS queues based on the quote request type** is incorrect because Amazon KDS is not a message filtering service. You should use Amazon SNS and SQS to distribute the topic to the designated queue.

References:

https://aws.amazon.com/getting-started/hands-on/filter-messages-published-to-topics/

https://docs.aws.amazon.com/sns/latest/dg/sns-message-filtering.html

https://docs.aws.amazon.com/sns/latest/dg/sns-sqs-as-subscriber.html

Check out these Amazon SNS and SQS Cheat Sheets:

https://tutorialsdojo.com/amazon-sns/

https://tutorialsdojo.com/amazon-sqs/

Amazon SNS Overview:

https://youtu.be/ft5R45lEUJ8

Question 11: **Correct**
A company has an enterprise web application hosted on Amazon ECS Docker containers that use an Amazon FSx for Lustre filesystem for its high-performance computing workloads. A warm standby environment is running in another AWS region

for disaster recovery. A Solutions Architect was assigned to design a system that will automatically route the live traffic to the disaster recovery (DR) environment only in the event that the primary application stack experiences an outage.

What should the Architect do to satisfy this requirement?

- **Set up a CloudWatch Alarm to monitor the primary Route 53 DNS endpoint and create a custom Lambda function. Execute the `ChangeResourceRecordSets` API call using the function to initiate the failover to the secondary DNS record.**

- **Set up a failover routing policy configuration in Route 53 by adding a health check on the primary service endpoint. Configure Route 53 to direct the DNS queries to the secondary record when the primary resource is unhealthy. Configure the network access control list and the route table to allow Route 53 to send requests to the endpoints specified in the health checks. Enable the `Evaluate Target Health` option by setting it to `Yes`.**

  (Correct)

- **Set up a Weighted routing policy configuration in Route 53 by adding health checks on both the primary stack and the DR environment. Configure the network access control list and the route table to allow Route 53 to send requests to the endpoints specified in the health checks. Enable the `Evaluate Target Health` option by setting it to `Yes`.**

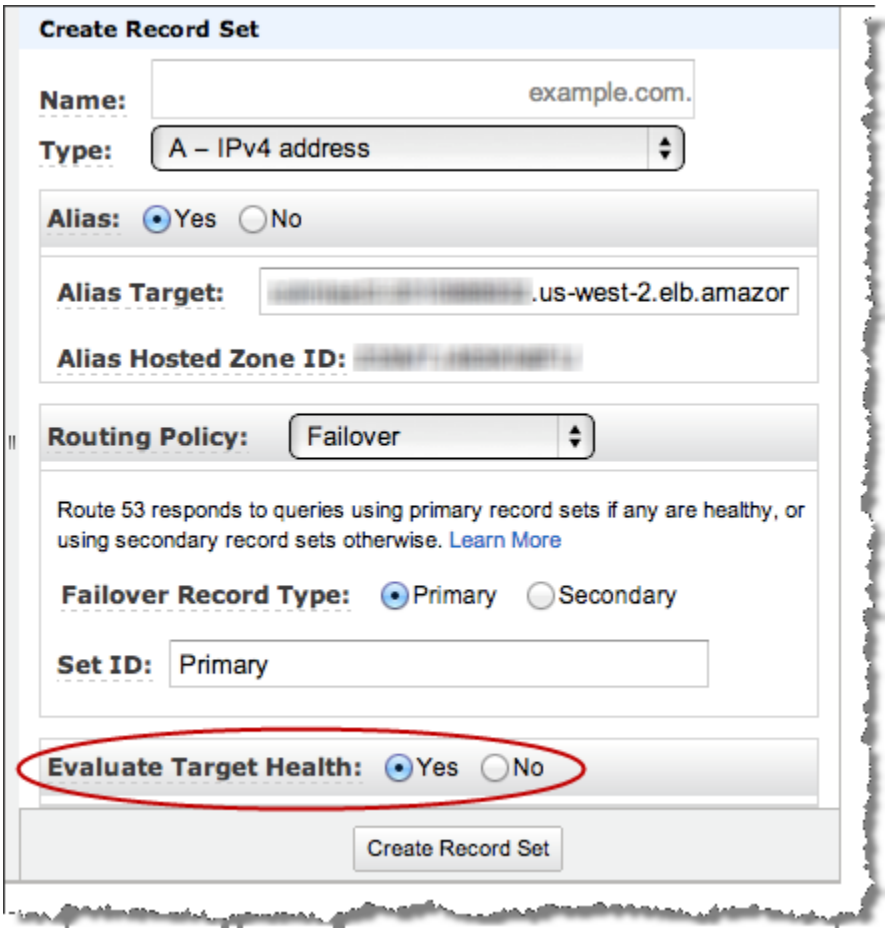
- **Set up a CloudWatch Events rule to monitor the primary Route 53 DNS endpoint and create a custom Lambda function. Execute the `ChangeResourceRecordSets` API call using the function to initiate the failover to the secondary DNS record.**

Explanation

Use an active-passive failover configuration when you want a primary resource or group of resources to be available majority of the time and you want a secondary resource or group of resources to be on standby in case all the primary resources become unavailable. When responding to queries, Route 53 includes only the healthy primary resources. If all the primary resources are unhealthy, Route 53 begins to include only the healthy secondary resources in response to DNS queries.

To create an active-passive failover configuration with one primary record and one secondary record, you just create the records and specify **Failover** for the routing policy. When the primary resource is healthy, Route 53 responds to DNS queries using the primary record. When the primary resource is unhealthy, Route 53 responds to DNS queries using the secondary record.

You can configure a health check that monitors an endpoint that you specify either by IP address or by domain name. At regular intervals that you specify, Route 53 submits automated requests over the Internet to your application, server, or other resource to verify that it's reachable, available, and functional. Optionally, you can configure the health check to make requests similar to those that your users make, such as requesting a web page from a specific URL.



When Route 53 checks the health of an endpoint, it sends an HTTP, HTTPS, or TCP request to the IP address and port that you specified when you created the health check. For a health check to succeed, your router and firewall rules must allow inbound traffic from the IP addresses that the Route 53 health checkers use.

Hence, the correct answer is: **Set up a failover routing policy configuration in Route 53 by adding a health check on the primary service endpoint. Configure Route 53 to direct**

the DNS queries to the secondary record when the primary resource is unhealthy. Configure the network access control list and the route table to allow Route 53 to send requests to the endpoints specified in the health checks. Enable the `Evaluate Target Health` option by setting it to `Yes`.

The option that says: **Set up a Weighted routing policy configuration in Route 53 by adding health checks on both the primary stack and the DR environment. Configure the network access control list and the route table to allow Route 53 to send requests to the endpoints specified in the health checks. Enable the `Evaluate Target Health` option by setting it to `Yes`** is incorrect because Weighted routing simply lets you associate multiple resources with a single domain name (tutorialsdojo.com) or subdomain name (blog.tutorialsdojo.com) and choose how much traffic is routed to each resource. This can be useful for a variety of purposes, including load balancing and testing new versions of software, but not for a failover configuration. Remember that the scenario says that the solution should automatically route the live traffic to the disaster recovery (DR) environment only in the event that the primary application stack experiences an outage. This configuration is incorrectly distributing the traffic on both the primary and DR environment.

The option that says: **Set up a CloudWatch Alarm to monitor the primary Route 53 DNS endpoint and create a custom Lambda function. Execute the `ChangeResourceRecordSets` API call using the function to initiate the failover to the secondary DNS record** is incorrect because setting up a CloudWatch Alarm and using the Route 53 API is not applicable nor useful at all in this scenario. Remember that CloudWatch Alam is primarily used for monitoring CloudWatch metrics. You have to use a Failover routing policy instead.

The option that says: **Set up a CloudWatch Events rule to monitor the primary Route 53 DNS endpoint and create a custom Lambda function. Execute the`ChangeResourceRecordSets` API call using the function to initiate the failover to the secondary DNS record** is incorrect because the Amazon CloudWatch Events service is commonly used to deliver a near real-time stream of system events that describe changes in **some** Amazon Web Services (AWS) resources. There is no direct way for CloudWatch Events to monitor the status of your Route 53 endpoints. You have to configure a health check and a failover configuration in Route 53 instead to satisfy the requirement in this scenario.

References:

https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/dns-failover-types.html

https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/health-checks-types.html

https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/dns-failover-router-firewall-rules.html

Check out this Amazon Route 53 Cheat Sheet:

https://tutorialsdojo.com/amazon-route-53/

Question 12: **Correct**
An application is hosted in AWS Fargate and uses RDS database in Multi-AZ Deployments configuration with several Read Replicas. A Solutions Architect was instructed to ensure that all of their database credentials, API keys, and other secrets are encrypted and rotated on a regular basis to improve data security. The application should also use the latest version of the encrypted credentials when connecting to the RDS database.

Which of the following is the MOST appropriate solution to secure the credentials?

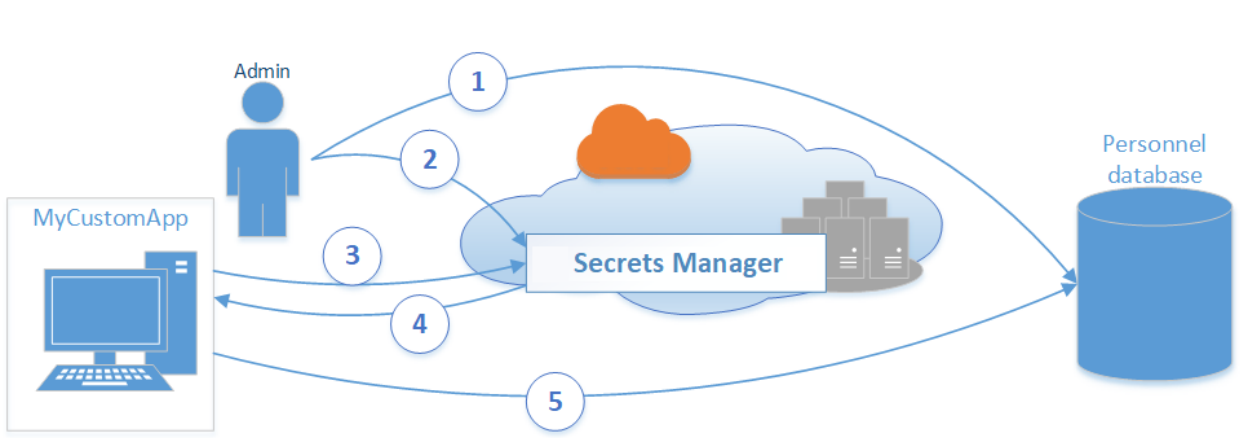
- **Store the database credentials, API keys, and other secrets to AWS ACM.**
- **Use AWS Secrets Manager to store and encrypt the database credentials, API keys, and other secrets. Enable automatic rotation for all of the credentials.**

  (Correct)
- **Store the database credentials, API keys, and other secrets in AWS KMS.**
- **Store the database credentials, API keys, and other secrets to Systems Manager Parameter Store each with a `SecureString` data type. The credentials are automatically rotated by default.**

Explanation
AWS Secrets Manager is an AWS service that makes it easier for you to manage secrets. *Secrets* can be database credentials, passwords, third-party API keys, and even

arbitrary text. You can store and control access to these secrets centrally by using the Secrets Manager console, the Secrets Manager command line interface (CLI), or the Secrets Manager API and SDKs.

In the past, when you created a custom application that retrieves information from a database, you typically had to embed the credentials (the secret) for accessing the database directly in the application. When it came time to rotate the credentials, you had to do much more than just create new credentials. You had to invest time in updating the application to use the new credentials. Then you had to distribute the updated application. If you had multiple applications that shared credentials and you missed updating one of them, the application would break. Because of this risk, many customers have chosen not to regularly rotate their credentials, which effectively substitutes one risk for another.



Secrets Manager enables you to replace hardcoded credentials in your code (including passwords), with an API call to Secrets Manager to retrieve the secret programmatically. This helps ensure that the secret can't be compromised by someone examining your code because the secret simply isn't there. Also, you can configure Secrets Manager to automatically rotate the secret for you according to the schedule that you specify. This enables you to replace long-term secrets with short-term ones, which helps to significantly reduce the risk of compromise.

Hence, the most appropriate solution for this scenario is: **Use AWS Secrets Manager to store and encrypt the database credentials, API keys, and other secrets. Enable automatic rotation for all of the credentials.**

The option that says: **Store the database credentials, API keys, and other secrets to Systems Manager Parameter Store each with a `SecureString` data type. The credentials are automatically rotated by default** is incorrect because the Systems Manager Parameter Store doesn't rotate its parameters by default.

The option that says: **Store the database credentials, API keys, and other secrets to AWS ACM** is incorrect because it is just a managed private CA service that helps you easily and securely manage the lifecycle of your private certificates to allow SSL communication to your application. This is not a suitable service for storing databases or any other confidential credentials.

The option that says: **Store the database credentials, API keys, and other secrets in AWS KMS** is incorrect because this only makes it easy for you to create and manage encryption keys and control the use of encryption across a wide range of AWS services. This is primarily used for encryption and not for hosting your credentials.

References:

https://aws.amazon.com/secrets-manager/

https://aws.amazon.com/blogs/security/how-to-securely-provide-database-credentials-to-lambda-functions-by-using-aws-secrets-manager/

Check out these AWS Secrets Manager and Systems Manager Cheat Sheets:

https://tutorialsdojo.com/aws-secrets-manager/

https://tutorialsdojo.com/aws-systems-manager/

AWS Security Services Overview - Secrets Manager, ACM, Macie:

https://youtu.be/ogVamzF2Dzk

Question 13: **Correct**
A company is hosting its web application in an Auto Scaling group of EC2 instances behind an Application Load Balancer. Recently, the Solutions Architect identified a series of SQL injection attempts and cross-site scripting attacks to the application, which had adversely affected their production data.

Which of the following should the Architect implement to mitigate this kind of attack?

- **Set up security rules that block SQL injection and cross-site scripting attacks in AWS Web Application Firewall (WAF). Associate the rules to the Application Load Balancer.**

(Correct)

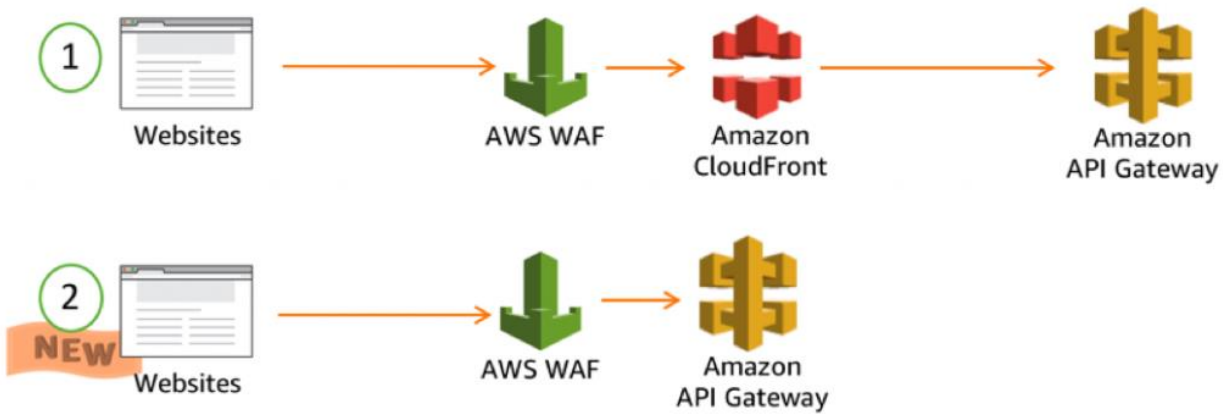
- **Use Amazon GuardDuty to prevent any further SQL injection and cross-site scripting attacks in your application.**

- **Using AWS Firewall Manager, set up security rules that block SQL injection and cross-site scripting attacks. Associate the rules to the Application Load Balancer.**

- **Block all the IP addresses where the SQL injection and cross-site scripting attacks originated using the Network Access Control List.**

Explanation

AWS WAF is a web application firewall that lets you monitor the HTTP and HTTPS requests that are forwarded to an Amazon API Gateway API, Amazon CloudFront or an Application Load Balancer. AWS WAF also lets you control access to your content. Based on conditions that you specify, such as the IP addresses that requests originate from or the values of query strings, API Gateway, CloudFront or an Application Load Balancer responds to requests either with the requested content or with an HTTP 403 status code (Forbidden). You also can configure CloudFront to return a custom error page when a request is blocked.



At the simplest level, AWS WAF lets you choose one of the following behaviors:

Allow all requests except the ones that you specify – This is useful when you want CloudFront or an Application Load Balancer to serve content for a public website, but you also want to block requests from attackers.

Block all requests except the ones that you specify – This is useful when you want to serve content for a restricted website whose users are readily identifiable by properties in web requests, such as the IP addresses that they use to browse to the website.

Count the requests that match the properties that you specify – When you want to allow or block requests based on new properties in web requests, you first can configure AWS WAF to count the requests that match those properties without allowing or blocking those requests. This lets you confirm that you didn't accidentally configure AWS WAF to block all the traffic to your website. When you're confident that you specified the correct properties, you can change the behavior to allow or block requests.

Hence, the correct answer in this scenario is: **Set up security rules that block SQL injection and cross-site scripting attacks in AWS Web Application Firewall (WAF). Associate the rules to the Application Load Balancer.**

Using Amazon GuardDuty to prevent any further SQL injection and cross-site scripting attacks in your application is incorrect because Amazon GuardDuty is just a threat detection service that continuously monitors for malicious activity and unauthorized behavior to protect your AWS accounts and workloads.

Using AWS Firewall Manager to set up security rules that block SQL injection and cross-site scripting attacks, then associating the rules to the Application Load Balancer is incorrect because AWS Firewall Manager just simplifies your AWS WAF and AWS Shield Advanced administration and maintenance tasks across multiple accounts and resources.

Blocking all the IP addresses where the SQL injection and cross-site scripting attacks originated using the Network Access Control List is incorrect because this is an optional layer of security for your VPC that acts as a firewall for controlling traffic in and out of one or more subnets. NACLs are not effective in blocking SQL injection and cross-site scripting attacks

References:

https://aws.amazon.com/waf/

https://docs.aws.amazon.com/waf/latest/developerguide/what-is-aws-waf.html

Check out this AWS WAF Cheat Sheet:

https://tutorialsdojo.com/aws-waf/

AWS Security Services Overview - WAF, Shield, CloudHSM, KMS:

https://youtu.be/-1S-RdeAmMo

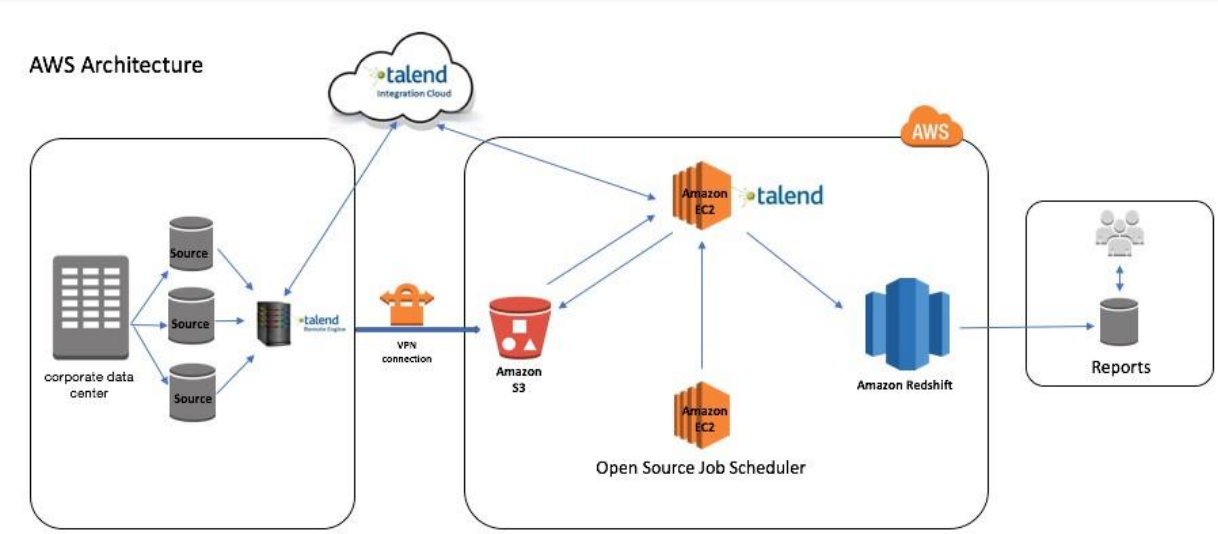
Question 14: **Correct**
A Solutions Architect is working for an online hotel booking firm with terabytes of customer data coming from the websites and applications. There is an annual corporate meeting where the Architect needs to present the booking behavior and acquire new insights from the customers' data. The Architect is looking for a service to perform super-fast analytics on massive data sets in near real-time.

Which of the following services gives the Architect the ability to store huge amounts of data and perform quick and flexible queries on it?

- **Amazon ElastiCache**
- **Amazon DynamoDB**
- **Amazon RDS**
- **Amazon Redshift**

  (Correct)

Explanation
Amazon Redshift is a fast, scalable data warehouse that makes it simple and cost-effective to analyze all your data across your data warehouse and data lake. Redshift delivers ten times faster performance than other data warehouses by using machine learning, massively parallel query execution, and columnar storage on a high-performance disk.



You can use Redshift to analyze all your data using standard SQL and your existing Business Intelligence (BI) tools. It also allows you to run complex analytic queries against terabytes to petabytes of structured and semi-structured data, using sophisticated query optimization, columnar storage on high-performance storage, and massively parallel query execution.

Hence, the correct answer is **Amazon Redshift.**

Amazon DynamoDB is incorrect because DynamoDB is a NoSQL database that is based on key-value pairs used for fast processing of small data that dynamically grows and changes. But if you need to scan large amounts of data (i.e., a lot of keys all in one query), the performance will not be optimal.

Amazon ElastiCache is incorrect because this is used to increase the performance, speed, and redundancy with which applications can retrieve data by providing an in-memory database caching system, and not for database analytical processes.

Amazon RDS is incorrect because this is mainly used for On-Line Transaction Processing (OLTP) applications and not for Online Analytics Processing (OLAP).

References:

https://docs.aws.amazon.com/redshift/latest/mgmt/welcome.html

https://docs.aws.amazon.com/redshift/latest/gsg/getting-started.html

Amazon Redshift Overview:

https://youtu.be/oIDHKeNxvQQ

Check out this Amazon Redshift Cheat Sheet:

https://tutorialsdojo.com/amazon-redshift/

Question 15: **Incorrect**
An organization is currently using a tape backup solution to store its application data on-premises. They plan to use a cloud storage service to preserve the backup data for up to 10 years that may be accessed about once or twice a year.

Which of the following is the most cost-effective option to implement this solution?

- **Use Amazon S3 to store the backup data and add a lifecycle rule to transition the current version to Amazon S3 Glacier.**

  (Incorrect)

- **Use AWS Storage Gateway to backup the data directly to Amazon S3 Glacier Deep Archive.**

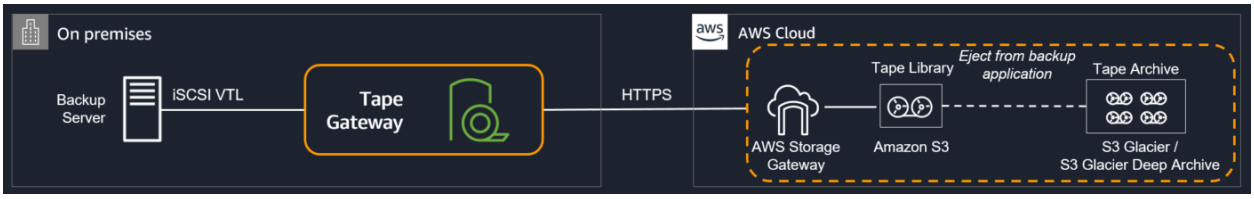
  (Correct)

- **Order an AWS Snowball Edge appliance to import the backup directly to Amazon S3 Glacier.**

- **Use AWS Storage Gateway to backup the data directly to Amazon S3 Glacier.**

Explanation
Tape Gateway enables you to replace using physical tapes on-premises with virtual tapes in AWS without changing existing backup workflows. Tape Gateway supports all leading backup applications and caches virtual tapes on-premises for low-latency data access. Tape Gateway encrypts data between the gateway and AWS for secure data

transfer and compresses data and transitions virtual tapes between Amazon S3 and Amazon S3 Glacier, or Amazon S3 Glacier Deep Archive, to minimize storage costs.



The scenario requires you to backup your application data to a cloud storage service for long-term retention of data that will be retained for 10 years. Since it uses a tape backup solution, an option that uses AWS Storage Gateway must be the possible answer. Tape Gateway can move your virtual tapes archived in Amazon S3 Glacier or Amazon S3 Glacier Deep Archive storage class, enabling you to further reduce the monthly cost to store long-term data in the cloud by up to 75%.

Hence, the correct answer is: **Use AWS Storage Gateway to backup the data directly to Amazon S3 Glacier Deep Archive**.

The option that says: **Use AWS Storage Gateway to backup the data directly to Amazon S3 Glacier** is incorrect. Although this is a valid solution, moving to S3 Glacier is more expensive than directly backing it up to Glacier Deep Archive.

The option that says: **Order an AWS Snowball Edge appliance to import the backup directly to Amazon S3 Glacier** is incorrect because Snowball Edge can't directly integrate backups to S3 Glacier. Moreover, you have to use the Amazon S3 Glacier Deep Archive storage class as it is more cost-effective than the regular Glacier class.

The option that says: **Use Amazon S3 to store the backup data and add a lifecycle rule to transition the current version to Amazon S3 Glacier** is incorrect. Although this is a possible solution, it is difficult to directly integrate a tape backup solution to S3 without using Storage Gateway.

References:

https://aws.amazon.com/storagegateway/faqs/

https://aws.amazon.com/s3/storage-classes/

AWS Storage Gateway Overview:

https://www.youtube.com/watch?v=pNb7xOBJjHE

Check out this AWS Storage Gateway Cheat Sheet:

https://tutorialsdojo.com/aws-storage-gateway/

Question 16: **Correct**
An organization needs to control the access for several S3 buckets. They plan to use a gateway endpoint to allow access to trusted buckets.

Which of the following could help you achieve this requirement?

- **Generate an endpoint policy for trusted S3 buckets.**

  (Correct)

- **Generate an endpoint policy for trusted VPCs.**

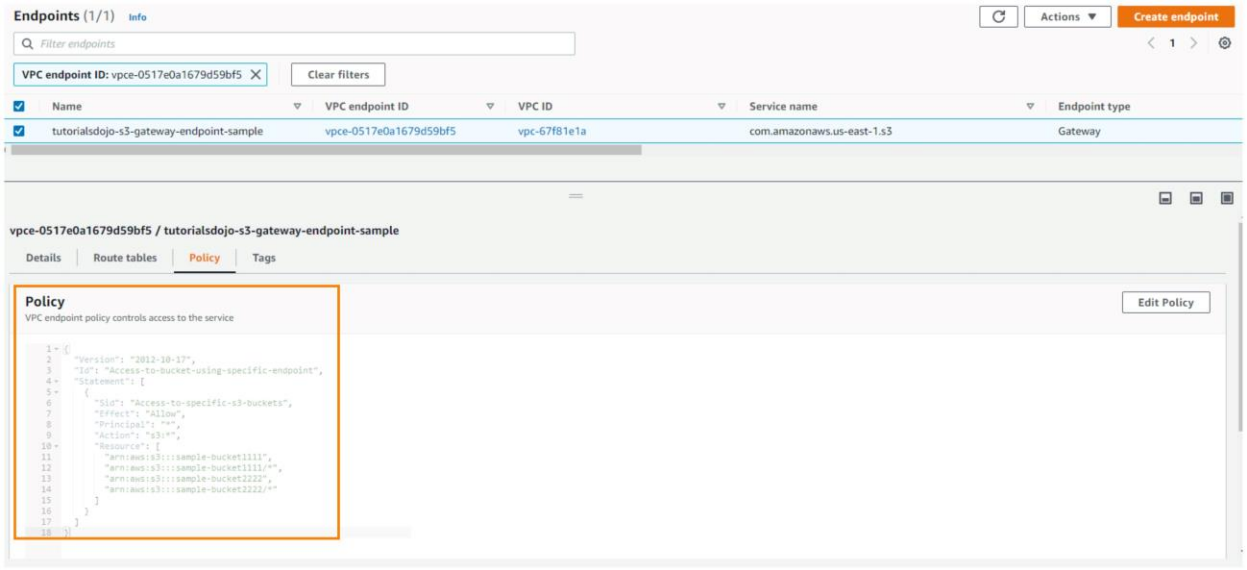
- **Generate a bucket policy for trusted S3 buckets.**

- **Generate a bucket policy for trusted VPCs.**

Explanation
A Gateway endpoint is a type of VPC endpoint that provides reliable connectivity to Amazon S3 and DynamoDB without requiring an internet gateway or a NAT device for your VPC. Instances in your VPC do not require public IP addresses to communicate with resources in the service.

When you create a Gateway endpoint, you can attach an endpoint policy that controls access to the service to which you are connecting. You can modify the endpoint policy attached to your endpoint and add or remove the route tables used by the endpoint. An endpoint policy does not override or replace IAM user policies or service-specific policies (such as S3 bucket policies). It is a separate policy for controlling access from

the endpoint to the specified service.



We can use a bucket policy or an endpoint policy to allow the traffic to trusted S3 buckets. The options that have 'trusted S3 buckets' key phrases will be the possible answer in this scenario. It would take you a lot of time to configure a bucket policy for each S3 bucket instead of using a single endpoint policy. Therefore, you should use an endpoint policy to control the traffic to the trusted Amazon S3 buckets.

Hence, the correct answer is: **Generate an endpoint policy for trusted S3 buckets.**

The option that says: **Generate a bucket policy for trusted S3 buckets** is incorrect. Although this is a valid solution, it takes a lot of time to set up a bucket policy for each and every S3 bucket. This can be simplified by whitelisting access to trusted S3 buckets in a single S3 endpoint policy.

The option that says: **Generate a bucket policy for trusted VPCs** is incorrect because you are generating a policy for trusted VPCs. Remember that the scenario only requires you to allow the traffic for trusted S3 buckets, not to the VPCs.

The option that says: **Generate an endpoint policy for trusted VPCs** is incorrect because it only allows access to trusted VPCs, and not to trusted Amazon S3 buckets.

References:

https://docs.aws.amazon.com/vpc/latest/userguide/vpc-endpoints-s3.html

https://aws.amazon.com/premiumsupport/knowledge-center/connect-s3-vpc-endpoint/

Amazon VPC Overview:

https://youtu.be/oIDHKeNxvQQ

Check out this Amazon VPC Cheat Sheet:

https://tutorialsdojo.com/amazon-vpc/

Question 17: **Correct**
A company is running a custom application in an Auto Scaling group of Amazon EC2 instances. Several instances are failing due to insufficient swap space. The Solutions Architect has been instructed to troubleshoot the issue and effectively monitor the available swap space of each EC2 instance.

Which of the following options fulfills this requirement?

- **Install the CloudWatch agent on each instance and monitor the `SwapUtilization` metric.**

  (Correct)

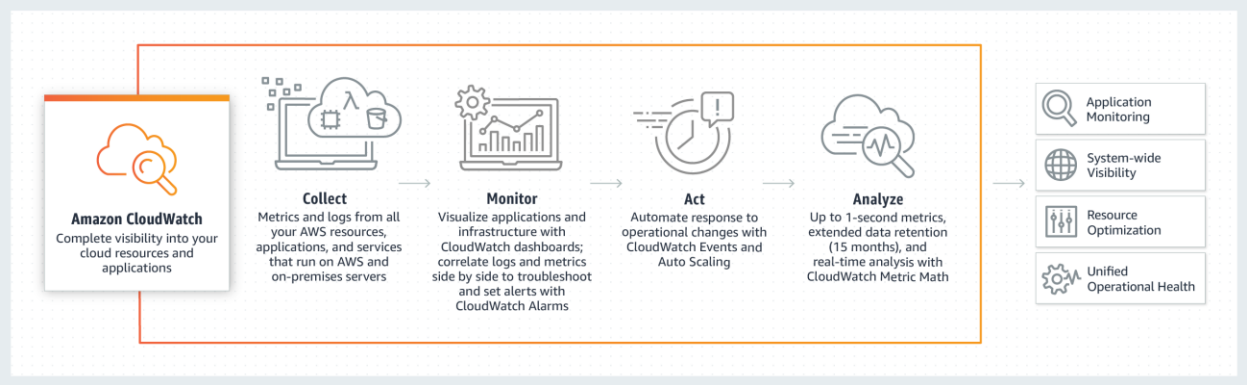
- **Create a new trail in AWS CloudTrail and configure Amazon CloudWatch Logs to monitor your trail logs.**

- **Create a CloudWatch dashboard and monitor the `SwapUsed` metric.**

- **Enable detailed monitoring on each instance and monitor the `SwapUtilization` metric.**

Explanation
Amazon CloudWatch is a monitoring service for AWS cloud resources and the applications you run on AWS. You can use Amazon CloudWatch to collect and track metrics, collect and monitor log files, and set alarms. Amazon CloudWatch can monitor AWS resources such as Amazon EC2 instances, Amazon DynamoDB tables, and Amazon RDS DB instances, as well as custom metrics generated by your applications and services and any log files your applications generate. You can use Amazon

CloudWatch to gain system-wide visibility into resource utilization, application performance, and operational health.



The main requirement in the scenario is to monitor the `SwapUtilization` metric. Take note that you can't use the default metrics of CloudWatch to monitor the `SwapUtilization` metric. To monitor custom metrics, you must install the CloudWatch agent on the EC2 instance. After installing the CloudWatch agent, you can now collect system metrics and log files of an EC2 instance.

Hence, the correct answer is: **Install the CloudWatch agent on each instance and monitor the `SwapUtilization` metric.**

The option that says: **Enable detailed monitoring on each instance and monitor the `SwapUtilization` metric** is incorrect because you can't monitor the `SwapUtilization` metric by just enabling the detailed monitoring option. You must install the CloudWatch agent on the instance.

The option that says: **Create a CloudWatch dashboard and monitor the `SwapUsed` metric** is incorrect because you must install the CloudWatch agent first to add the custom metric in the dashboard.

The option that says: **Create a new trail in AWS CloudTrail and configure Amazon CloudWatch Logs to monitor your trail logs** is incorrect because CloudTrail won't help you monitor custom metrics. CloudTrail is specifically used for monitoring API activities in an AWS account.

References:

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/mon-scripts.html

https://aws.amazon.com/cloudwatch/faqs/

Check out this Amazon CloudWatch Cheat Sheet:

https://tutorialsdojo.com/amazon-cloudwatch/

Amazon CloudWatch Overview:

https://youtu.be/q0DmxfyGkeU

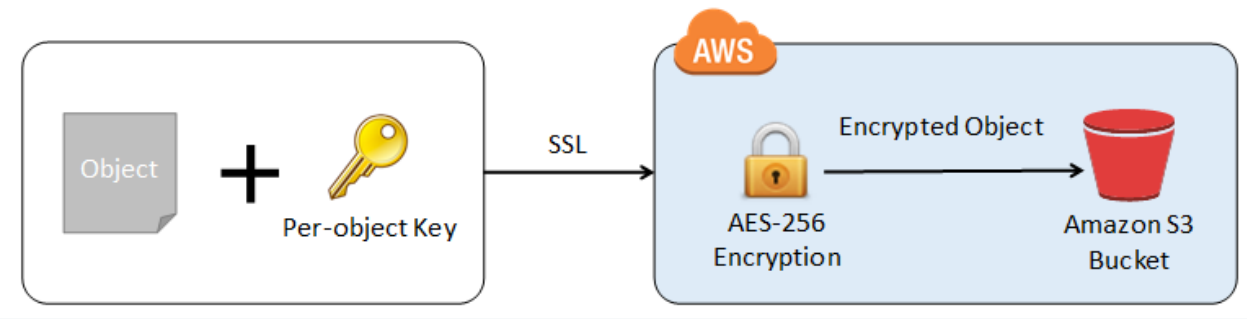
Question 18: **Correct**
All objects uploaded to an Amazon S3 bucket must be encrypted for security compliance. The bucket will use server-side encryption with Amazon S3-Managed encryption keys (SSE-S3) to encrypt data using 256-bit Advanced Encryption Standard (AES-256) block cipher.

Which of the following request headers must be used?

- `x-amz-server-side-encryption-customer-key`
- `x-amz-server-side-encryption-customer-algorithm`
- `x-amz-server-side-encryption-customer-key-MD5`
- `x-amz-server-side-encryption`

  (Correct)

Explanation
Server-side encryption protects data at rest. If you use Server-Side Encryption with Amazon S3-Managed Encryption Keys (SSE-S3), Amazon S3 will encrypt each object with a unique key and as an additional safeguard, it encrypts the key itself with a master key that it rotates regularly. Amazon S3 server-side encryption uses one of the strongest block ciphers available, 256-bit Advanced Encryption Standard (AES-256), to encrypt your data.



If you need server-side encryption for all of the objects that are stored in a bucket, use a bucket policy. For example, the following bucket policy denies permissions to upload an object unless the request includes the **x-amz-server-side-encryption** header to request server-side encryption:

However, if you choose to use server-side encryption with customer-provided encryption keys (SSE-C), you must provide encryption key information using the following request headers:

`x-amz-server-side-encryption-customer-algorithm`

`x-amz-server-side-encryption-customer-key`

`x-amz-server-side-encryption-customer-key-MD5`

Hence, using the **x-amz-server-side-encryption** header is correct as this is the one being used for Amazon S3-Managed Encryption Keys (SSE-S3).

All other options are incorrect since they are used for SSE-C.

References:

https://docs.aws.amazon.com/AmazonS3/latest/dev/serv-side-encryption.html

https://docs.aws.amazon.com/AmazonS3/latest/dev/UsingServerSideEncryption.html

https://docs.aws.amazon.com/AmazonS3/latest/dev/ServerSideEncryptionCustomerKeys.html

Check out this Amazon S3 Cheat Sheet:

https://tutorialsdojo.com/amazon-s3/

Question 19: **Correct**
A company has a dynamic web app written in MEAN stack that is going to be launched in the next month. There is a probability that the traffic will be quite high in the first couple of weeks. In the event of a load failure, how can you set up DNS failover to a static website?

- **Use Route 53 with the failover option to a static S3 website bucket or CloudFront distribution.**

  (Correct)

- **Enable failover to an application hosted in an on-premises data center.**

- **Add more servers in case the application fails.**

- **Duplicate the exact application architecture in another region and configure DNS weight-based routing.**

Explanation
For this scenario, **using Route 53 with the failover option to a static S3 website bucket or CloudFront distribution** is correct. You can create a new Route 53 with the failover option to a static S3 website bucket or CloudFront distribution as an alternative.

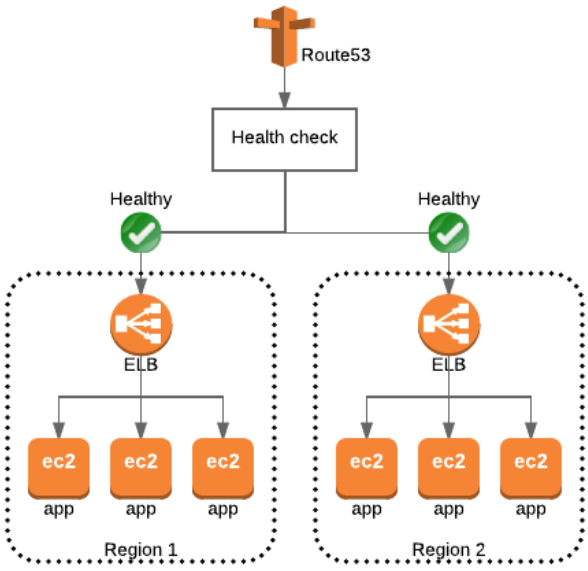


Figure 1 - Both regions operating normally

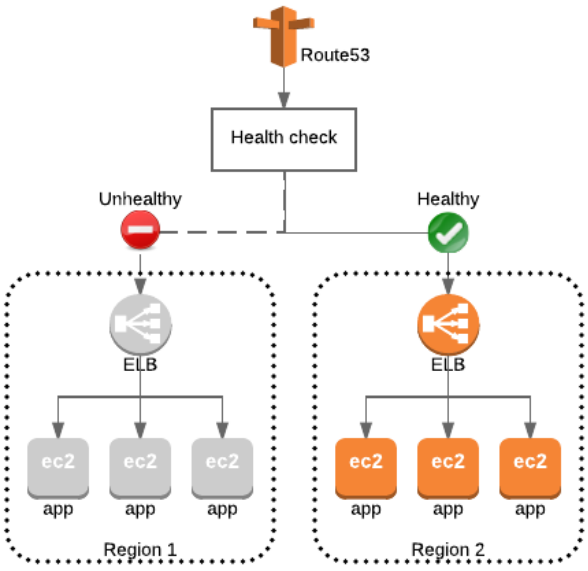


Figure 2 - region 1 experiencing issues

Duplicating the exact application architecture in another region and configuring DNS weight-based routing is incorrect because running a duplicate system is not a cost-effective solution. Remember that you are trying to build a failover mechanism for your web app, not a distributed setup.

Enabling failover to an application hosted in an on-premises data center is incorrect. Although you can set up failover to your on-premises data center, you are not maximizing the AWS environment such as using Route 53 failover.

Adding more servers in case the application fails is incorrect because this is not the best way to handle a failover event. If you add more servers only in case the application fails, then there would be a period of downtime in which your application is unavailable. Since there are no running servers on that period, your application will be unavailable for a certain period of time until your new server is up and running.

Reference:

https://aws.amazon.com/premiumsupport/knowledge-center/fail-over-s3-r53/

http://docs.aws.amazon.com/Route53/latest/DeveloperGuide/dns-failover.html

Check out this Amazon Route 53 Cheat Sheet:

https://tutorialsdojo.com/amazon-route-53/

Question 20: **Incorrect**

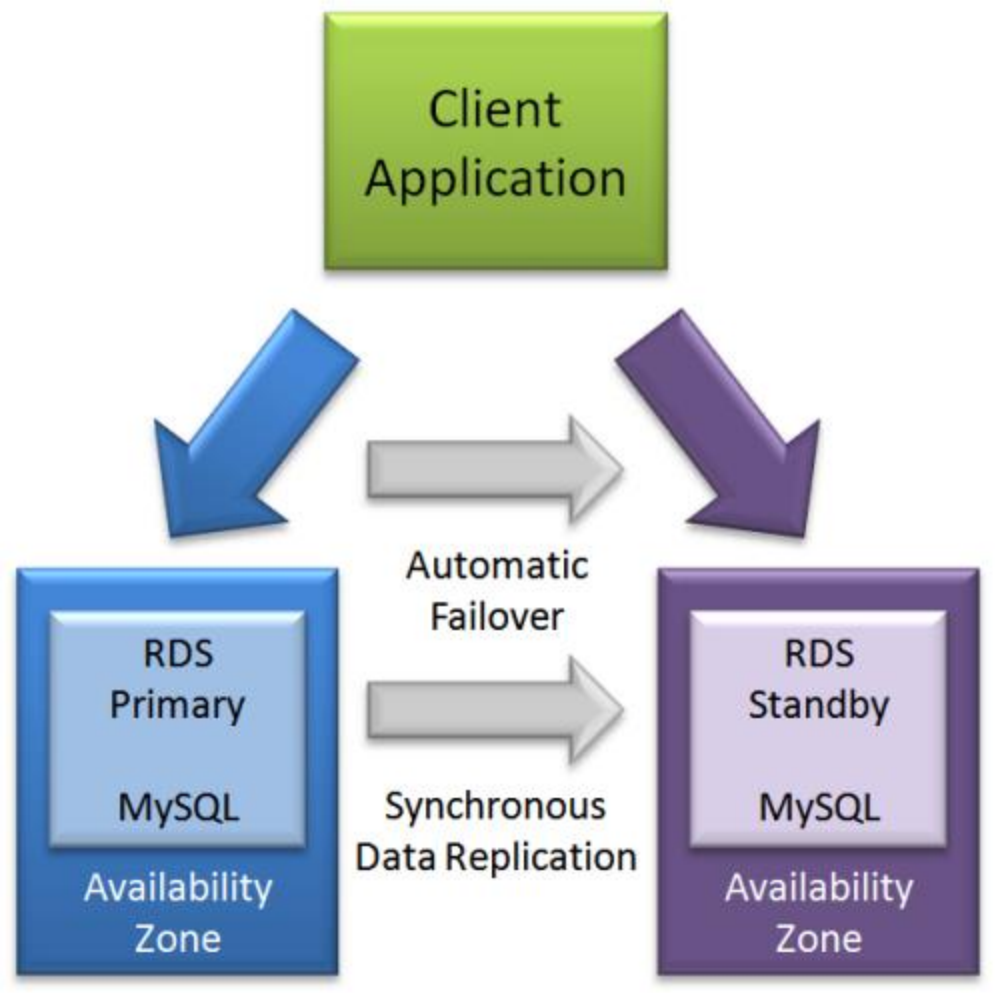
An accounting application uses an RDS database configured with Multi-AZ deployments to improve availability. What would happen to RDS if the primary database instance fails?

- **A new database instance is created in the standby Availability Zone.**
- **The primary database instance will reboot.**
- **The canonical name record (CNAME) is switched from the primary to standby instance.**

  (Correct)
- **The IP address of the primary DB instance is switched to the standby DB instance.**

  (Incorrect)

Explanation
In **Amazon RDS**, failover is automatically handled so that you can resume database operations as quickly as possible without administrative intervention in the event that your primary database instance goes down. When failing over, Amazon RDS simply flips the canonical name record (CNAME) for your DB instance to point at the standby, which is in turn promoted to become the new primary.



The option that says: **The IP address of the primary DB instance is switched to the standby DB instance** is incorrect since IP addresses are per subnet, and subnets cannot span multiple AZs.

The option that says: **The primary database instance will reboot** is incorrect since in the event of a failure, there is no database to reboot with.

The option that says: **A new database instance is created in the standby Availability Zone** is incorrect since with multi-AZ enabled, you already have a standby database in another AZ.

References:

https://aws.amazon.com/rds/details/multi-az/

https://aws.amazon.com/rds/faqs/

Amazon RDS Overview:

https://youtu.be/aZmpLl8K1UU

Check out this Amazon RDS Cheat Sheet:

https://tutorialsdojo.com/amazon-relational-database-service-amazon-rds/

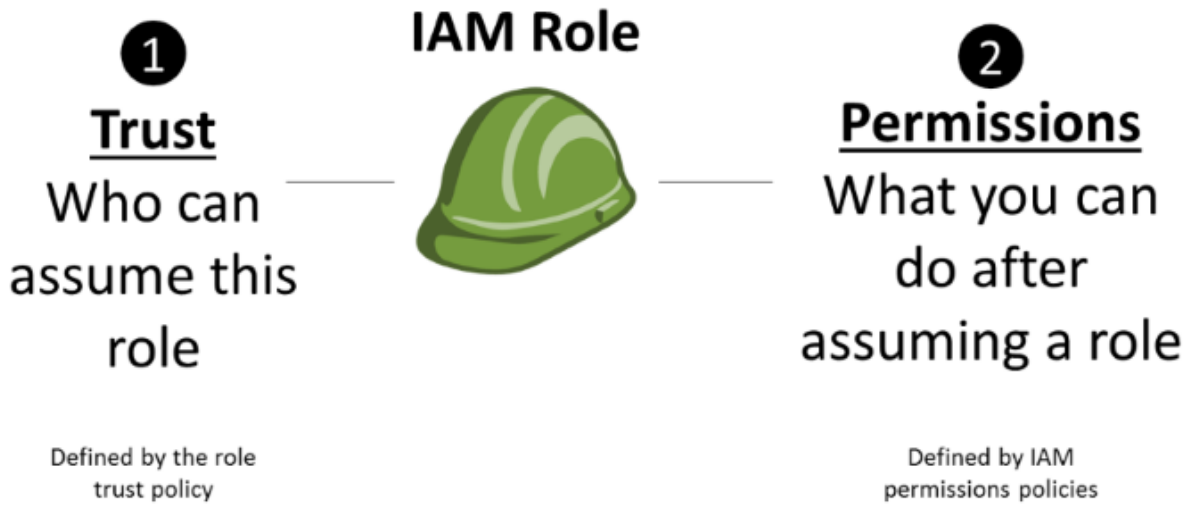
Question 21: **Correct**
A company developed a meal planning application that provides meal recommendations for the week as well as the food consumption of the users. The application resides on an EC2 instance which requires access to various AWS services for its day-to-day operations.

Which of the following is the best way to allow the EC2 instance to access the S3 bucket and other AWS services?

- **Add the API Credentials in the Security Group and assign it to the EC2 instance.**
- **Store the API credentials in a bastion host.**
- **Create a role in IAM and assign it to the EC2 instance.**

  (Correct)
- **Store the API credentials in the EC2 instance.**

Explanation
The best practice in handling API Credentials is to create a new role in the Identity Access Management (IAM) service and then assign it to a specific EC2 instance. In this way, you have a secure and centralized way of storing and managing your credentials.



Storing the API credentials in the EC2 instance, **adding the API Credentials in the Security Group and assigning it to the EC2 instance**, and **storing the API credentials in a bastion host** are incorrect because it is not secure to store nor use the API credentials from an EC2 instance. You should use IAM service instead.

Reference:

http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/iam-roles-for-amazon-ec2.html

Check out this AWS IAM Cheat Sheet:

https://tutorialsdojo.com/aws-identity-and-access-management-iam/

Question 22: **Correct**
A start-up company has an EC2 instance that is hosting a web application. The volume of users is expected to grow in the coming months, and hence, you need to add more elasticity and scalability in your AWS architecture to cope with the demand.

Which of the following options can satisfy the above requirement for the given scenario? (Select TWO.)

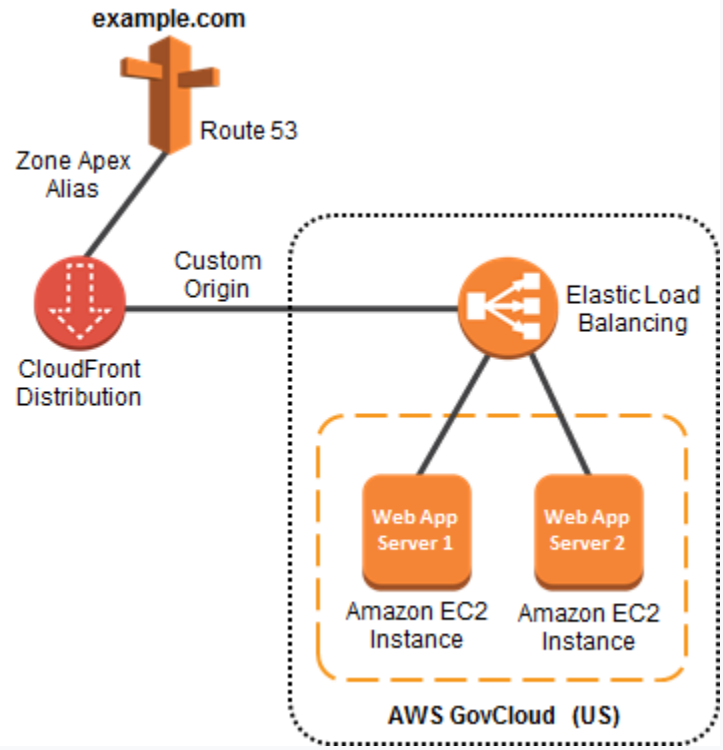
- ☐ **Set up two EC2 instances deployed using Launch Templates and integrated with AWS Glue.**

- ☐ **Set up an AWS WAF behind your EC2 Instance.**
- ☐ **Set up an S3 Cache in front of the EC2 instance.**
- ☑ **Set up two EC2 instances and then put them behind an Elastic Load balancer (ELB).**

  (Correct)
- ☑ **Set up two EC2 instances and use Route 53 to route traffic based on a Weighted Routing Policy.**

  (Correct)

Explanation
Using an Elastic Load Balancer is an ideal solution for adding elasticity to your application. Alternatively, you can also create a policy in Route 53, such as a Weighted routing policy, to evenly distribute the traffic to 2 or more EC2 instances. Hence, **setting up two EC2 instances and then put them behind an Elastic Load balancer (ELB)** and **setting up two EC2 instances and using Route 53 to route traffic based on a Weighted Routing Policy** are the correct answers.



Setting up an S3 Cache in front of the EC2 instance is incorrect because doing so does not provide elasticity and scalability to your EC2 instances.

Setting up an AWS WAF behind your EC2 Instance is incorrect because AWS WAF is a web application firewall that helps protect your web applications from common web exploits. This service is more about providing security to your applications.

Setting up two EC2 instances deployed using Launch Templates and integrated with AWS Glue is incorrect because AWS Glue is a fully managed extract, transform, and load (ETL) service that makes it easy for customers to prepare and load their data for analytics. It does not provide scalability or elasticity to your instances.

References:

https://aws.amazon.com/elasticloadbalancing

http://docs.aws.amazon.com/Route53/latest/DeveloperGuide/Welcome.html

Check out this AWS Elastic Load Balancing Cheat Sheet:

https://tutorialsdojo.com/aws-elastic-load-balancing-elb/

Check out this Amazon Route 53 Cheat Sheet:

https://tutorialsdojo.com/amazon-route-53/

Question 23: **Correct**
An online events registration system is hosted in AWS and uses ECS to host its front-end tier and an RDS configured with Multi-AZ for its database tier. What are the events that will make Amazon RDS automatically perform a failover to the standby replica? (Select TWO.)

- ☑ **Storage failure on primary**

  (Correct)

- ☐ **In the event of Read Replica failure**

- ☐ **Compute unit failure on secondary DB instance**

- ☐ **Storage failure on secondary DB instance**

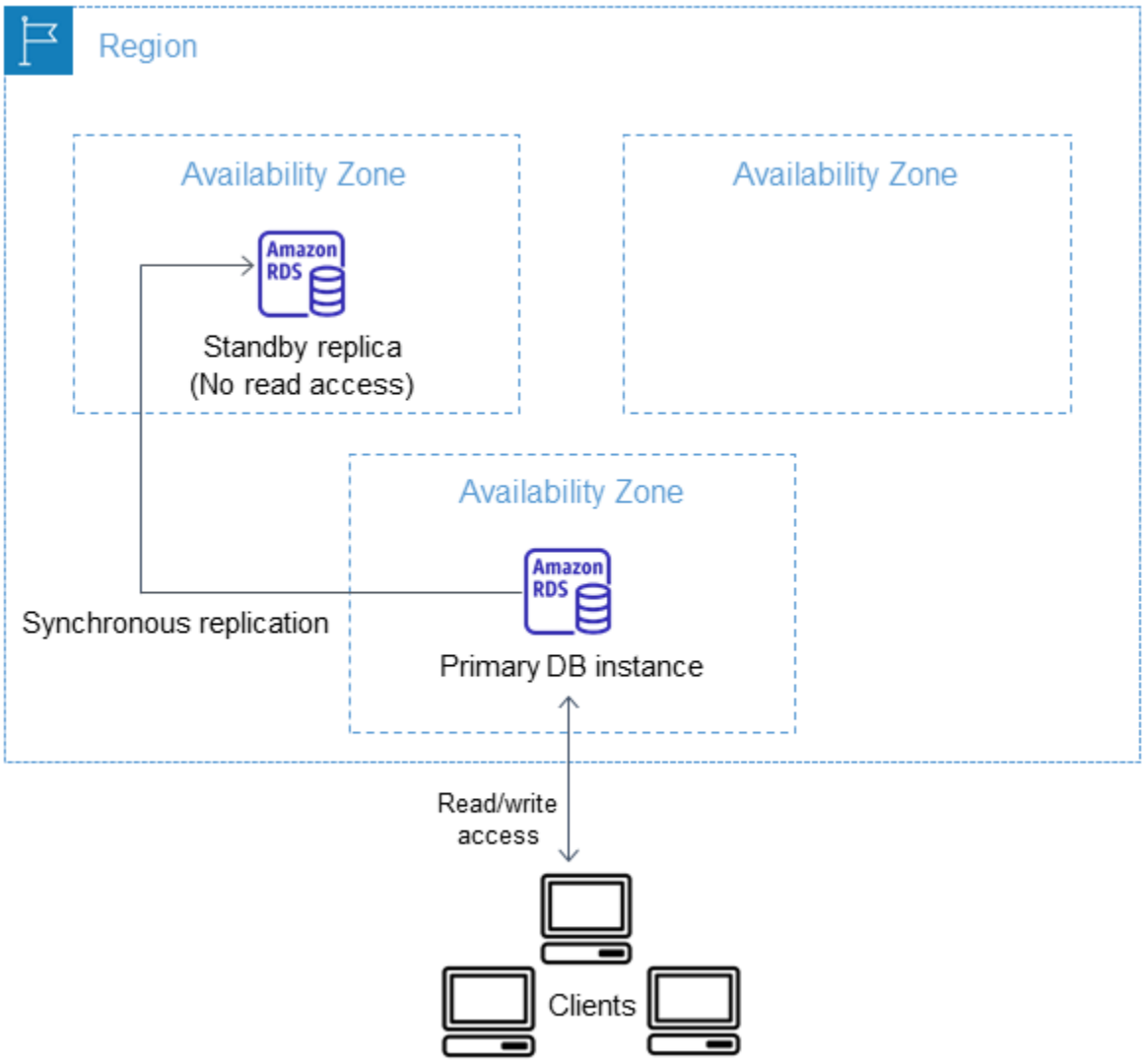
- ☑ **Loss of availability in primary Availability Zone**

  (Correct)

Explanation
Amazon RDS provides high availability and failover support for DB instances using Multi-AZ deployments. Amazon RDS uses several different technologies to provide failover support. Multi-AZ deployments for Oracle, PostgreSQL, MySQL, and MariaDB DB instances use Amazon's failover technology. SQL Server DB instances use SQL Server Database Mirroring (DBM).

In a Multi-AZ deployment, Amazon RDS automatically provisions and maintains a synchronous standby replica in a different Availability Zone. The primary DB instance is synchronously replicated across Availability Zones to a standby replica to provide data redundancy, eliminate I/O freezes, and minimize latency spikes during system backups. Running a DB instance with high availability can enhance availability during planned system maintenance and help protect your databases against DB instance failure and Availability Zone disruption.

Amazon RDS detects and automatically recovers from the most common failure scenarios for Multi-AZ deployments so that you can resume database operations as quickly as possible without administrative intervention.



The high-availability feature is not a scaling solution for read-only scenarios; you cannot use a standby replica to serve read traffic. To service read-only traffic, you should use a Read Replica.

Amazon RDS automatically performs a failover in the event of any of the following:

1. Loss of availability in primary Availability Zone.
2. Loss of network connectivity to primary.
3. Compute unit failure on primary.
4. Storage failure on primary.

Hence, the correct answers are:

- Loss of availability in primary Availability Zone
- Storage failure on primary

The following options are incorrect because all these scenarios do not affect the primary database. Automatic failover only occurs if the primary database is the one that is affected.

- Storage failure on secondary DB instance

- In the event of Read Replica failure

- Compute unit failure on secondary DB instance

References:

https://aws.amazon.com/rds/details/multi-az/

https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/Concepts.MultiAZ.html

Check out this Amazon RDS Cheat Sheet:

https://tutorialsdojo.com/amazon-relational-database-service-amazon-rds/

Question 24: **Correct**
A company hosts its web application on a set of Amazon EC2 instances in an Auto Scaling group behind an Application Load Balancer (ALB). The application has an embedded NoSQL database. As the application receives more traffic, the application becomes overloaded mainly due to database requests. The management wants to ensure that the database is eventually consistent and highly available.

Which of the following options can meet the company requirements with the least operational overhead?

- **Change the ALB with a Network Load Balancer (NLB) to handle more traffic and integrate AWS Global Accelerator to ensure high availability. Configure replication of the NoSQL database on the set of Amazon EC2 instances to spread the database load.**

- **Configure the Auto Scaling group to spread the Amazon EC2 instances across three Availability Zones. Use the AWS Database Migration Service (DMS) with a replication server and an ongoing replication task to migrate the embedded NoSQL database to Amazon DynamoDB**

  (Correct)

- **Configure the Auto Scaling group to spread the Amazon EC2 instances across three Availability Zones. Configure replication of the NoSQL database on the set of Amazon EC2 instances to spread the database load.**

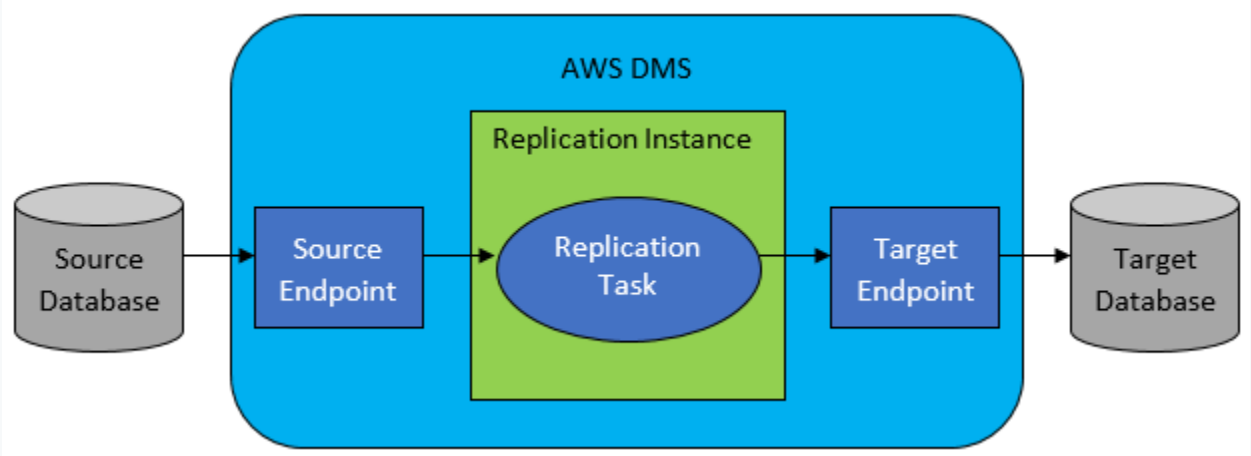
- **Change the ALB with a Network Load Balancer (NLB) to handle more traffic. Use the AWS Migration Service (DMS) to migrate the embedded NoSQL database to Amazon DynamoDB.**

Explanation

AWS Database Migration Service (AWS DMS) is a cloud service that makes it easy to migrate relational databases, data warehouses, NoSQL databases, and other types of data stores. You can use AWS DMS to migrate your data into the AWS Cloud or between combinations of cloud and on-premises setups.

With AWS DMS, you can perform one-time migrations, and you can replicate ongoing changes to keep sources and targets in sync. If you want to migrate to a different database engine, you can use the AWS Schema Conversion Tool (AWS SCT) to translate your database schema to the new platform. You then use AWS DMS to migrate the data. Because AWS DMS is a part of the AWS Cloud, you get the cost efficiency, speed to market, security, and flexibility that AWS services offer.

You can use AWS DMS to migrate data to an Amazon DynamoDB table. Amazon DynamoDB is a fully managed NoSQL database service that provides fast and predictable performance with seamless scalability. AWS DMS supports using a relational database or MongoDB as a source.



Therefore, the correct answer is: **Configure the Auto Scaling group to spread the Amazon EC2 instances across three Availability Zones. Use the AWS Database Migration Service (DMS) with a replication server and an ongoing replication task to migrate the embedded NoSQL database to Amazon DynamoDB.** Using an Auto Scaling group of EC2 instances and migrating the embedded database to Amazon DynamoDB will ensure that both the application and database are highly available with low operational overhead.

The option that says: **Change the ALB with a Network Load Balancer (NLB) to handle more traffic and integrate AWS Global Accelerator to ensure high availability. Configure replication of the NoSQL database on the set of Amazon EC2 instances to spread the database load** is incorrect. It is not recommended to run a production system with an embedded database on EC2 instances. A better option is to migrate the database to a managed AWS service such as Amazon DynamoDB, so you won't have to manually maintain, patch, provision and scale your database yourself. In addition, using an AWS Global Accelerator is not warranted since the architecture is only hosted in a single AWS region and not in multiple regions.

The option that says: **Change the ALB with a Network Load Balancer (NLB) to handle more traffic. Use the AWS Migration Service (DMS) to migrate the embedded NoSQL database to Amazon DynamoDB** is incorrect. The scenario did not require handling millions of requests per second or very low latency to justify the use of NLB. The ALB should be able to able to scale and handle scaling traffic.

The option that says: **Configure the Auto Scaling group to spread the Amazon EC2 instances across three Availability Zones. Configure replication of the NoSQL database on the set of Amazon EC2 instances to spread the database load** is incorrect. This may be possible, but it entails an operational overhead of manually configuring the embedded database to replicate and scale with the EC2 instances. It would be better to migrate the database to a managed AWS database service such as Amazon DynamoDB.

References:

https://docs.aws.amazon.com/dms/latest/userguide/CHAP_Target.DynamoDB.html

https://docs.aws.amazon.com/dms/latest/userguide/Welcome.html

https://docs.aws.amazon.com/autoscaling/ec2/userguide/as-add-availability-zone.html

Check out these Amazon DynamoDB and AWS DMS Cheat Sheets:

https://tutorialsdojo.com/amazon-dynamodb/

https://tutorialsdojo.com/aws-database-migration-service/

Question 25: **Incorrect**
An insurance company utilizes SAP HANA for its day-to-day ERP operations. Since they can't migrate this database due to customer preferences, they need to integrate it with the current AWS workload in the VPC in which they are required to establish a site-to-site VPN connection.

What needs to be configured outside of the VPC for them to have a successful site-to-site VPN connection?

- **A dedicated NAT instance in a public subnet**
- **The main route table in your VPC to route traffic through a NAT instance**
- **An Internet-routable IP address (static) of the customer gateway's external interface for the on-premises network**

  (Correct)
-

An EIP to the Virtual Private Gateway

(Incorrect)

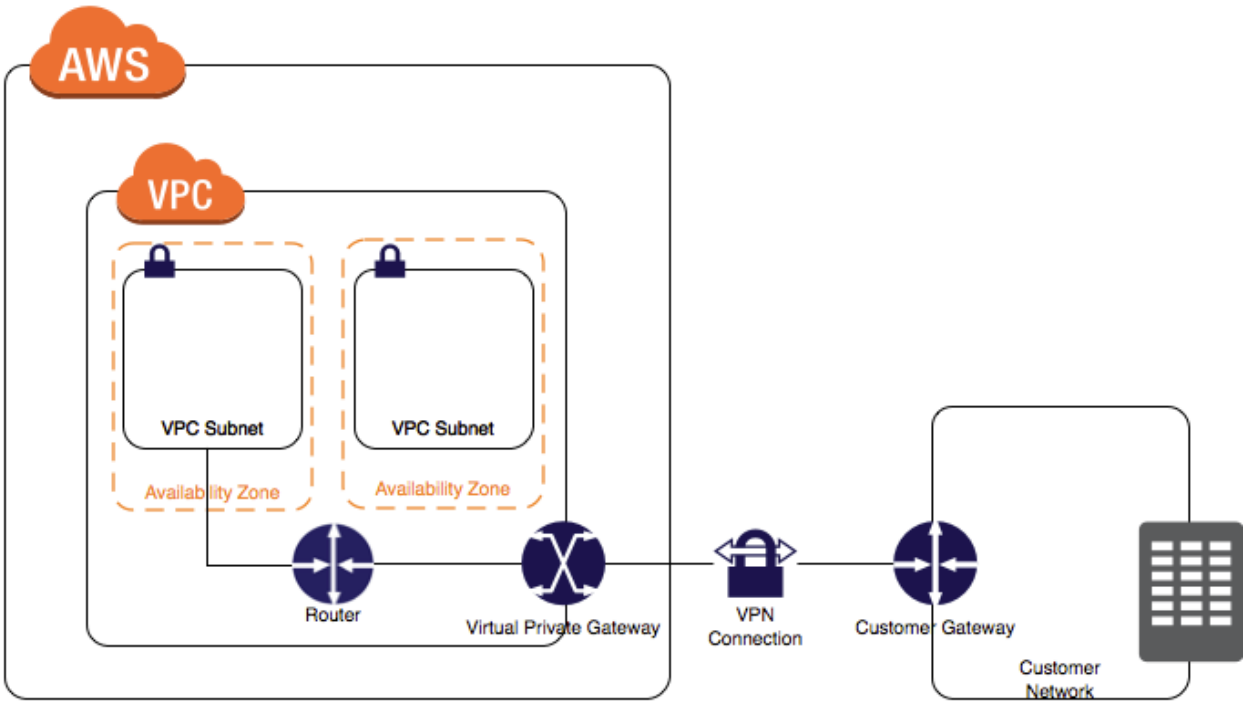
Explanation
By default, instances that you launch into a virtual private cloud (VPC) can't communicate with your own network. You can enable access to your network from your VPC by attaching a virtual private gateway to the VPC, creating a custom route table, updating your security group rules, and creating an AWS managed VPN connection.

Although the term **VPN connection** is a general term, in the Amazon VPC documentation, a VPN connection refers to the connection between your VPC and your own network. AWS supports Internet Protocol security (IPsec) VPN connections.

A **customer gateway** is a physical device or software application on your side of the VPN connection.

To create a VPN connection, you must create a customer gateway resource in AWS, which provides information to AWS about your customer gateway device. Next, you have to set up an Internet-routable IP address (static) of the customer gateway's external interface.

The following diagram illustrates single VPN connections. The VPC has an attached virtual private gateway, and your remote network includes a customer gateway, which you must configure to enable the VPN connection. You set up the routing so that any traffic from the VPC bound for your network is routed to the virtual private gateway.



The options that say: **A dedicated NAT instance in a public subnet** and **the main route table in your VPC to route traffic through a NAT instance** are incorrect since you don't need a NAT instance for you to be able to create a VPN connection.

An EIP to the Virtual Private Gateway is incorrect since you do not attach an EIP to a VPG.

References:

https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_VPN.html

https://docs.aws.amazon.com/vpc/latest/userguide/SetUpVPNConnections.html

Check out this Amazon VPC Cheat Sheet:

https://tutorialsdojo.com/amazon-vpc/

Question 26: **Incorrect**
A company is running a dashboard application on a Spot EC2 instance inside a private subnet. The dashboard is reachable via a domain name that maps to the private IPv4 address of the instance's network interface. A solutions architect needs to increase

network availability by allowing the traffic flow to resume in another instance if the primary instance is terminated.

Which solution accomplishes these requirements?

- **Attach an elastic IP address to the instance's primary network interface and point its IP address to the application's domain name. Automatically move the EIP to a secondary instance if the primary instance becomes unavailable using the AWS Transit Gateway.**

  (Incorrect)

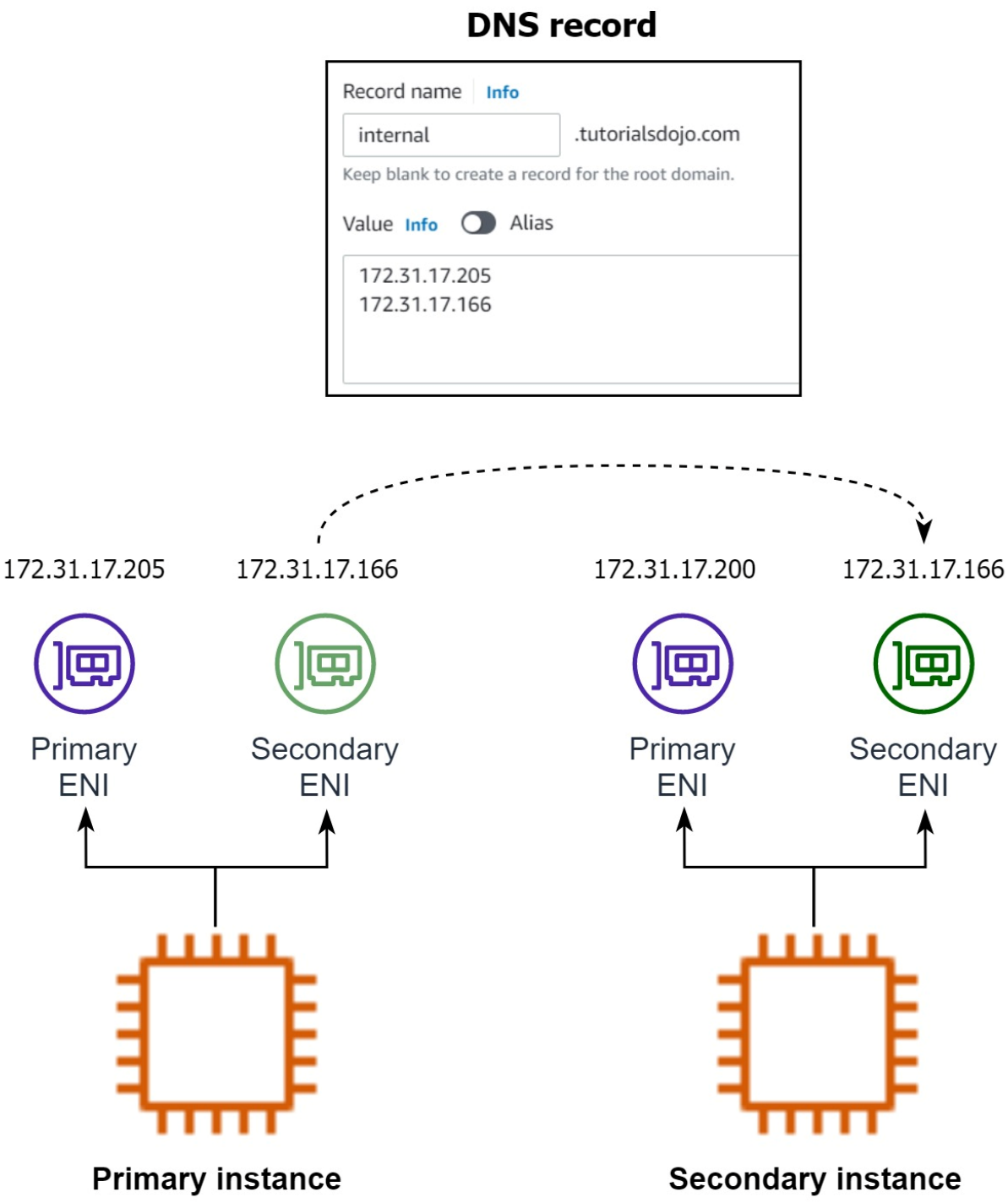
- **Set up AWS Transfer for FTPS service in Implicit FTPS mode to automatically disable the `source/destination` checks on the instance's primary elastic network interface and reassociate it to another instance.**

- **Use the AWS Network Firewall to detach the instance's primary elastic network interface and move it to a new instance upon failure.**

- **Create a secondary elastic network interface and point its private IPv4 address to the application's domain name. Attach the new network interface to the primary instance. If the instance goes down, move the secondary network interface to another instance.**

  (Correct)

Explanation
If one of your instances serving a particular function fails, its network interface can be attached to a replacement or hot standby instance pre-configured for the same role in order to rapidly recover the service. For example, you can use a network interface as your primary or secondary network interface to a critical service such as a database instance or a NAT instance. If the instance fails, you (or more likely, the code running on your behalf) can attach the network interface to a hot standby instance.



Because the interface maintains its private IP addresses, Elastic IP addresses, and MAC address, network traffic begins flowing to the standby instance as soon as you attach the network interface to the replacement instance. Users experience a brief loss of connectivity between the time the instance fails and the time that the network interface is attached to the standby instance, but no changes to the route table or your DNS server are required.

Hence, the correct answer is **Create a secondary elastic network interface and point its private IPv4 address to the application's domain name. Attach the new network interface to the primary instance. If the instance goes down, move the secondary network interface to another instance.**

The option that says: **Attach an elastic IP address to the instance's primary network interface and point its IP address to the application's domain name. Automatically move the EIP to a secondary instance if the primary instance becomes unavailable using the AWS Transit Gateway** is incorrect. Elastic IPs are not needed in the solution since the application is private. Furthermore, an AWS Transit Gateway is primarily used to connect your Amazon Virtual Private Clouds (VPCs) and on-premises networks through a central hub. This particular networking service cannot be used to automatically move an Elastic IP address to another EC2 instance.

The option that says: **Set up AWS Transfer for FTPS service in Implicit FTPS mode to automatically disable the `source/destination` checks on the instance's primary elastic network interface and reassociate it to another instance** is incorrect. First of all, the AWS Transfer for FTPS service is not capable of automatically disabling the source/destination checks and it only supports Explicit FTPS mode. Disabling the source/destination check only allows the instance to which the ENI is connected to act as a gateway (both a sender and a receiver). It is not possible to make the primary ENI of any EC2 instance detachable. A more appropriate solution would be to use an Elastic IP address which can be reassociated with your secondary instance.

The option that says: **Use the AWS Network Firewall to detach the instance's primary elastic network interface and move it to a new instance upon failure** is incorrect. It's not possible to detach the primary network interface of an EC2 instance. In addition, the AWS Network Firewall is only used for filtering traffic at the perimeter of your VPC and not for detaching ENIs.

References:

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-eni.html

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/scenarios-enis.html

https://aws.amazon.com/aws-transfer-family/faqs/

Check out this Amazon EC2 Cheat Sheet:

https://tutorialsdojo.com/amazon-elastic-compute-cloud-amazon-ec2/

Question 27: **Correct**
A company developed a web application and deployed it on a fleet of EC2 instances that uses Amazon SQS. The requests are saved as messages in the SQS queue, which is configured with the maximum message retention period. However, after thirteen days of operation, the web application suddenly crashed and there are 10,000 unprocessed messages that are still waiting in the queue. Since they developed the application, they can easily resolve the issue but they need to send a communication to the users on the issue.

What information should they provide and what will happen to the unprocessed messages?

- **Tell the users that unfortunately, they have to resubmit all of the requests since the queue would not be able to process the 10,000 messages together.**

- **Tell the users that the application will be operational shortly however, requests sent over three days ago will need to be resubmitted.**

- **Tell the users that the application will be operational shortly and all received requests will be processed after the web application is restarted.**

  (Correct)

- **Tell the users that unfortunately, they have to resubmit all the requests again.**

Explanation
In **Amazon SQS**, you can configure the message retention period to a value from 1 minute to 14 days. The default is 4 days. Once the message retention limit is reached, your messages are automatically deleted.

A single Amazon SQS message queue can contain an unlimited number of messages. However, there is a 120,000 limit for the number of inflight messages for a standard queue and 20,000 for a FIFO queue. Messages are inflight after they have been received from the queue by a consuming component, but have not yet been deleted from the queue.

In this scenario, it is stated that the SQS queue is configured with the maximum message retention period. The maximum message retention in SQS is 14 days that is why the option that says: **Tell the users that the application will be operational shortly and all received requests will be processed after the web application is restarted** is the correct answer i.e. there will be no missing messages.

The options that say: **Tell the users that unfortunately, they have to resubmit all the requests again** and **Tell the users that the application will be operational shortly, however, requests sent over three days ago will need to be resubmitted** are incorrect as there are no missing messages in the queue thus, there is no need to resubmit any previous requests.

The option that says: **Tell the users that unfortunately, they have to resubmit all of the requests since the queue would not be able to process the 10,000 messages together** is incorrect as the queue can contain an unlimited number of messages, not just 10,000 messages.

Reference:

https://aws.amazon.com/sqs/

Check out this Amazon SQS Cheat Sheet:

https://tutorialsdojo.com/amazon-sqs/

Question 28: **Correct**
For data privacy, a healthcare company has been asked to comply with the Health Insurance Portability and Accountability Act (HIPAA). The company stores all its backups on an Amazon S3 bucket. It is required that data stored on the S3 bucket must be encrypted.

What is the best option to do this? (Select TWO.)

- ☐ **Enable Server-Side Encryption on an S3 bucket to make use of AES-128 encryption.**
- ☐ **Store the data in encrypted EBS snapshots.**

- ☐

  Store the data on EBS volumes with encryption enabled instead of using Amazon S3.

- ☑

  Before sending the data to Amazon S3 over HTTPS, encrypt the data locally first using your own encryption keys.

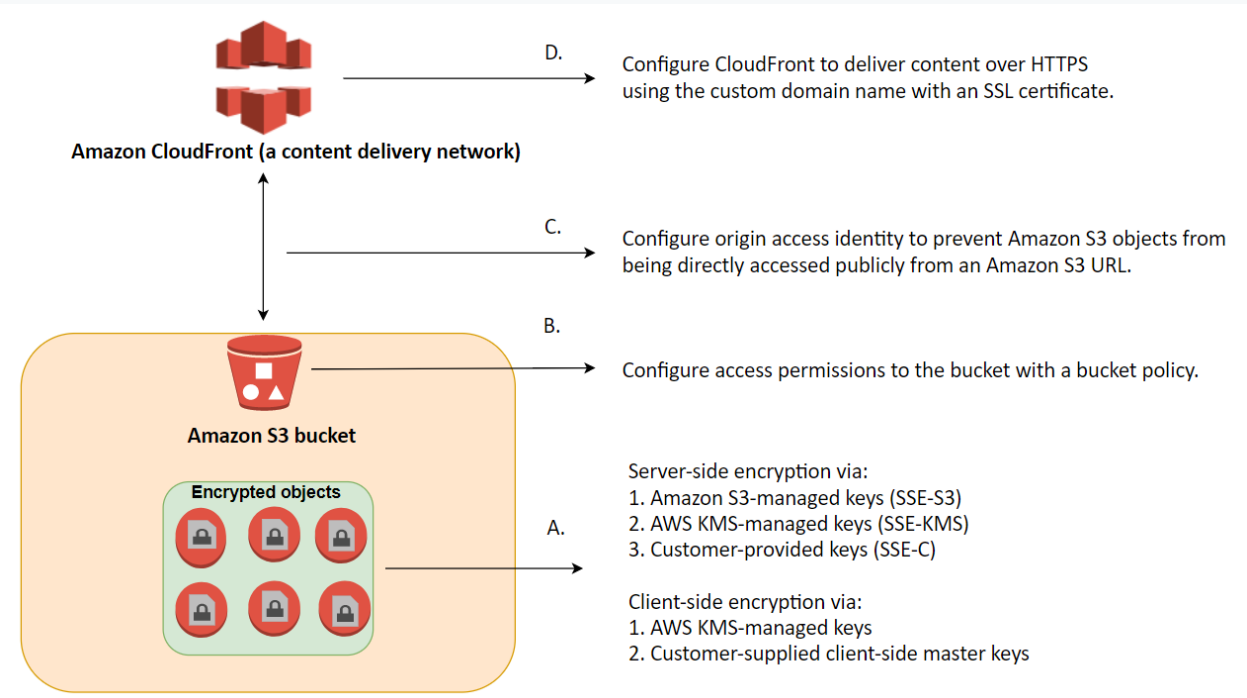
  (Correct)

- ☑

  Enable Server-Side Encryption on an S3 bucket to make use of AES-256 encryption.

  (Correct)

Explanation
Server-side encryption is about data encryption at rest—that is, Amazon S3 encrypts your data at the object level as it writes it to disks in its data centers and decrypts it for you when you access it. As long as you authenticate your request and you have access permissions, there is no difference in the way you access encrypted or unencrypted objects. For example, if you share your objects using a pre-signed URL, that URL works the same way for both encrypted and unencrypted objects.



You have three mutually exclusive options depending on how you choose to manage the encryption keys:

1. Use Server-Side Encryption with Amazon S3-Managed Keys (SSE-S3)
2. Use Server-Side Encryption with AWS KMS-Managed Keys (SSE-KMS)
3. Use Server-Side Encryption with Customer-Provided Keys (SSE-C)

The options that say: **Before sending the data to Amazon S3 over HTTPS, encrypt the data locally first using your own encryption keys** and **Enable Server-Side Encryption on an S3 bucket to make use of AES-256 encryption** are correct because these options are using client-side encryption and Amazon S3-Managed Keys (SSE-S3) respectively. *Client-side encryption* is the act of encrypting data before sending it to Amazon S3 while SSE-S3 uses AES-256 encryption.

Storing the data on EBS volumes with encryption enabled instead of using Amazon S3 and **storing the data in encrypted EBS snapshots** are incorrect because both options use EBS encryption and not S3.

Enabling Server-Side Encryption on an S3 bucket to make use of AES-128 encryption is incorrect as S3 doesn't provide AES-128 encryption, only AES-256.

References:

http://docs.aws.amazon.com/AmazonS3/latest/dev/UsingEncryption.html

https://docs.aws.amazon.com/AmazonS3/latest/dev/UsingClientSideEncryption.html

Check out this Amazon S3 Cheat Sheet:

https://tutorialsdojo.com/amazon-s3/

Tutorials Dojo's AWS Certified Solutions Architect Associate Exam Study Guide:

https://tutorialsdojo.com/aws-certified-solutions-architect-associate/

Question 29: **Incorrect**
An Intelligence Agency developed a missile tracking application that is hosted on both development and production AWS accounts. The Intelligence agency's junior developer only has access to the development account. She has received security clearance to access the agency's production account but the access is only temporary and only write access to EC2 and S3 is allowed.

Which of the following allows you to issue short-lived access tokens that act as temporary security credentials to allow access to your AWS resources?

- **All of the given options are correct.**
- **Use AWS Cognito to issue JSON Web Tokens (JWT)**

  (Incorrect)
- **Use AWS STS**

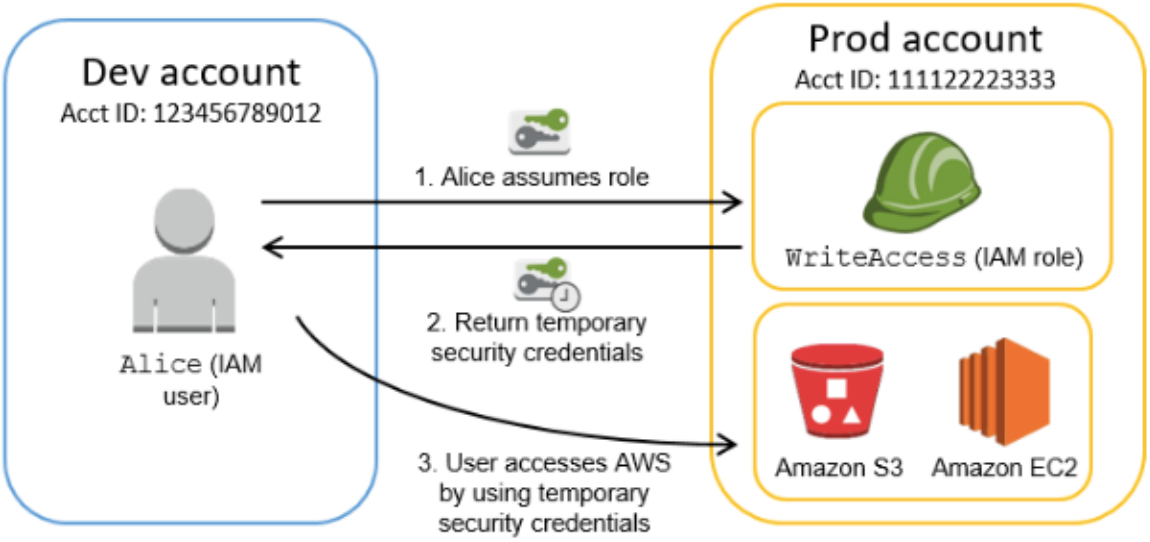
  (Correct)
- **Use AWS SSO**

Explanation

AWS Security Token Service (AWS STS) is the service that you can use to create and provide trusted users with temporary security credentials that can control access to your AWS resources. Temporary security credentials work almost identically to the long-term access key credentials that your IAM users can use.

In this diagram, IAM user Alice in the Dev account (the role-assuming account) needs to access the Prod account (the role-owning account). Here's how it works:

1. Alice in the Dev account assumes an IAM role (WriteAccess) in the Prod account by calling AssumeRole.
2. STS returns a set of temporary security credentials.
3. Alice uses the temporary security credentials to access services and resources in the Prod account. Alice could, for example, make calls to Amazon S3 and Amazon EC2, which are granted by the WriteAccess role.



Using AWS Cognito to issue JSON Web Tokens (JWT) is incorrect because the Amazon Cognito service is primarily used for user authentication and not for providing access to your AWS resources. A JSON Web Token (JWT) is meant to be used for user authentication and session management.

Using AWS SSO is incorrect. Although the AWS SSO service uses STS, it does not issue short-lived credentials by itself. AWS Single Sign-On (SSO) is a cloud SSO service that makes it easy to centrally manage SSO access to multiple AWS accounts and business applications.

The option that says **All of the above** is incorrect as only STS has the ability to provide temporary security credentials.

Reference:

https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_temp.html

AWS Identity Services Overview:

https://youtu.be/AIdUw0i8rr0

Check out this AWS IAM Cheat Sheet:

https://tutorialsdojo.com/aws-identity-and-access-management-iam/

Tutorials Dojo's AWS Certified Solutions Architect Associate Exam Study Guide:

https://tutorialsdojo.com/aws-certified-solutions-architect-associate/

Question 30: **Correct**
A solutions architect is designing a cost-efficient, highly available storage solution for company data. One of the requirements is to ensure that the previous state of a file is preserved and retrievable if a modified version of it is uploaded. Also, to meet regulatory compliance, data over 3 years must be retained in an archive and will only be accessible once a year.

How should the solutions architect build the solution?

- **Create an S3 Standard bucket and enable S3 Object Lock in governance mode.**
- **Create an S3 Standard bucket with object-level versioning enabled and configure a lifecycle rule that transfers files to Amazon S3 Glacier Deep Archive after 3 years.**

  (Correct)

- **Create a One-Zone-IA bucket with object-level versioning enabled and configure a lifecycle rule that transfers files to Amazon S3 Glacier Deep Archive after 3 years.**

- **Create an S3 Standard bucket with S3 Object Lock in compliance mode enabled then configure a lifecycle rule that transfers files to Amazon S3 Glacier Deep Archive after 3 years.**

Explanation

Versioning in Amazon S3 is a means of keeping multiple variants of an object in the same bucket. You can use the S3 Versioning feature to preserve, retrieve, and restore every version of every object stored in your buckets. With versioning, you can recover more easily from both unintended user actions and application failures. After versioning is enabled for a bucket, if Amazon S3 receives multiple write requests for the same object simultaneously, it stores all of those objects.

Hence, the correct answer is: **Create an S3 Standard bucket with object-level versioning enabled and configure a lifecycle rule that transfers files to Amazon S3 Glacier Deep Archive after 3 years.**

The S3 Object Lock feature allows you to store objects using a write-once-read-many (WORM) model. In the scenario, changes to objects are allowed, but their previous versions should be preserved and remain retrievable. If you enable the S3 Object Lock feature, you won't be able to upload new versions of an object. This feature is only helpful when you want to prevent objects from being deleted or overwritten for a fixed amount of time or indefinitely.

Therefore, the following options are incorrect:

- Create an S3 Standard bucket and enable S3 Object Lock in governance mode.

- Create an S3 Standard bucket with S3 Object Lock in compliance mode enabled then configure a lifecycle rule that transfers files to Amazon S3 Glacier Deep Archive after 3 years.

The option that says: **Create a One-Zone-IA bucket with object-level versioning enabled and configure a lifecycle rule that transfers files to Amazon S3 Glacier Deep Archive after 3 years** is incorrect. One-Zone-IA is not highly available as it only relies on one availability zone for storing data.

References:

https://docs.aws.amazon.com/AmazonS3/latest/userguide/Versioning.html

https://aws.amazon.com/blogs/aws/new-amazon-s3-storage-class-glacier-deep-archive/

Check out this Amazon S3 Cheat Sheet:

https://tutorialsdojo.com/amazon-s3/

Question 31: **Incorrect**

A technology company has a suite of container-based web applications and serverless solutions that are hosted in AWS. The Solutions Architect must define a standard infrastructure that will be used across development teams and applications. There are application-specific resources too that change frequently, especially during the early stages of application development. Developers must be able to add supplemental resources to their applications, which are beyond what the architects predefined in the system environments and service templates.

Which of the following should be implemented to satisfy this requirement?

- **Use the Amazon EKS Anywhere service for deploying container applications and serverless solutions. Create a service instance for each application-specific resource.**

  (Incorrect)

- **Set up AWS Proton for deploying container applications and serverless solutions. Create components from the AWS Proton console and attach them to their respective service instance.**

  (Correct)

- **Set up AWS Control Tower to automate container-based application deployments. Use AWS Config for application-specific resources that change frequently.**

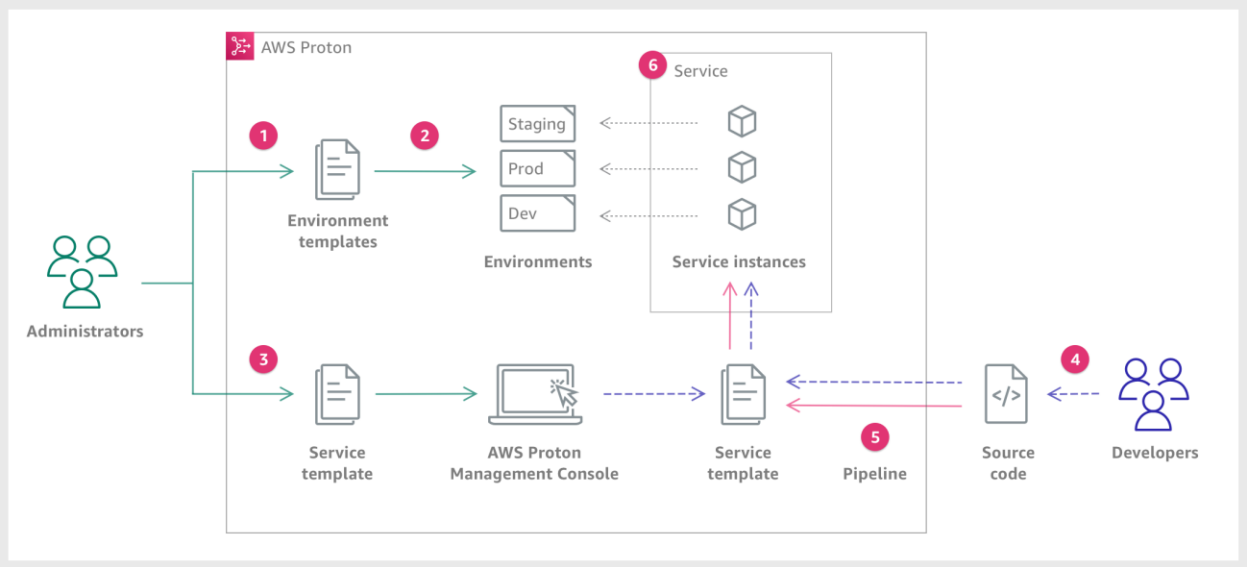
-

Use the Amazon Elastic Container Service (ECS) Anywhere service for deploying container applications and serverless solutions. Configure Prometheus metrics collection on the ECS cluster and use Amazon Managed Service for Prometheus for monitoring frequently-changing resources

Explanation

AWS Proton allows you to deploy any serverless or container-based application with increased efficiency, consistency, and control. You can define infrastructure standards and effective continuous delivery pipelines for your organization. Proton breaks down the infrastructure into environment and service ("infrastructure as code" templates).

As a developer, you select a standardized service template that AWS Proton uses to create a service that deploys and manages your application in a service instance. An AWS Proton service is an instantiation of a service template, which normally includes several service instances and a pipeline.



The diagram above displays the high-level overview of a simple AWS Proton workflow.

In AWS Proton administrators define standard infrastructure that is used across development teams and applications. However, development teams might need to include additional resources for their specific use cases, like Amazon Simple Queue Service (Amazon SQS) queues or Amazon DynamoDB tables. These application-specific resources might change frequently, particularly during early application development. Maintaining these frequent changes in administrator-authored templates might be hard to manage and scale—administrators would need to maintain many more templates without real administrator added value. The alternative—letting application developers author templates for their applications—isn't ideal either, because it takes away

administrators' ability to standardize the main architecture components, like AWS Fargate tasks. This is where components come in.

With a component, a developer can add supplemental resources to their application, above and beyond what administrators defined in environment and service templates. The developer then attaches the component to a service instance. AWS Proton provisions infrastructure resources defined by the component just like it provisions resources for environments and service instances.

Hence, the correct answer is: **Set up AWS Proton for deploying container applications and serverless solutions. Create components from the AWS Proton console and attach them to their respective service instance.**

The option that says: **Use the Amazon EKS Anywhere service for deploying container applications and serverless solutions. Create a service instance for each application-specific resource** is incorrect. Amazon EKS Anywhere just allows you to manage a Kubernetes cluster on external environments that are supported by AWS. It is better to use AWS Proton with custom Components that can be attached to the different service instances of the company's application suite.

The option that says: **Set up AWS Control Tower to automate container-based application deployments. Use AWS Config for application-specific resources that change frequently** is incorrect. AWS Control Tower is used to simplify the creation of new accounts with preconfigured constraints. It isn't used to automate application deployments. Moreover, AWS Config is commonly used for monitoring the changes of AWS resources and not the custom resources for serverless or container-based applications in AWS. A combination of AWS Proton and Components is the most suitable solution for this scenario.

The option that says: **Use the Amazon Elastic Container Service (ECS) Anywhere service for deploying container applications and serverless solutions. Configure Prometheus metrics collection on the ECS cluster and use Amazon Managed Service for Prometheus for monitoring frequently-changing resources** is incorrect. The Amazon Managed Service for Prometheus is only a Prometheus-compatible monitoring and alerting service that makes it easy to monitor containerized applications and infrastructure at scale. It is not capable of tracking or maintaining your application-specific resources that change frequently.

References:

https://docs.aws.amazon.com/proton/latest/userguide/Welcome.html

https://aws.amazon.com/blogs/architecture/simplifying-multi-account-ci-cd-deployments-using-aws-proton/

Question 32: **Incorrect**
A company wants to streamline the process of creating multiple AWS accounts within an AWS Organization. Each organization unit (OU) must be able to launch new accounts with preapproved configurations from the security team which will standardize the baselines and network configurations for all accounts in the organization.

Which solution entails the least amount of effort to implement?

- **Set up an AWS Config aggregator on the root account of the organization to enable multi-account, multi-region data aggregation. Deploy conformance packs to standardize the baselines and network configurations for all accounts.**

- **Set up an AWS Control Tower Landing Zone. Enable pre-packaged guardrails to enforce policies or detect violations.**

  (Correct)

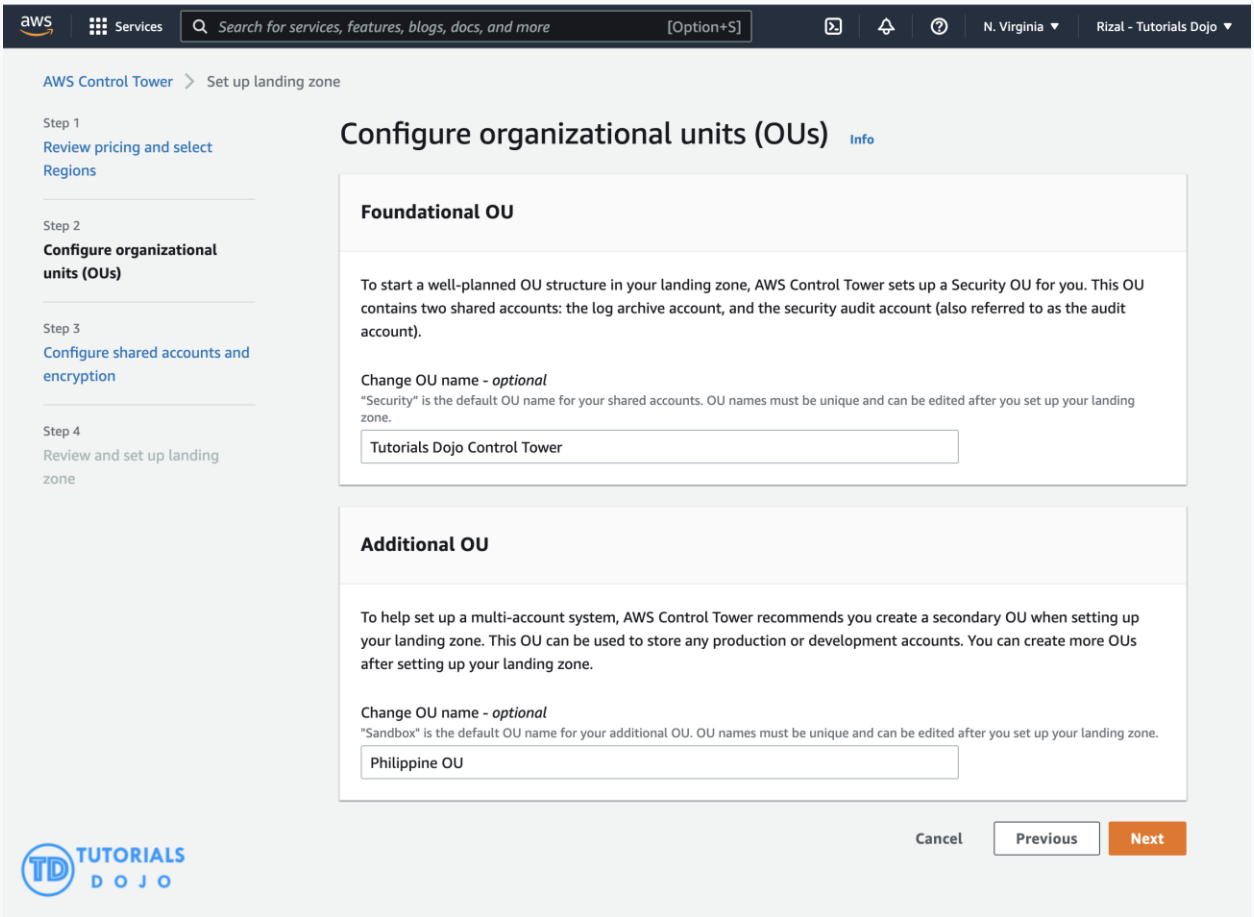
- **Configure AWS Resource Access Manager (AWS RAM) to launch new AWS accounts as well as standardize the baselines and network configurations for each organization unit**

  (Incorrect)

- **Centralized the creation of AWS accounts using AWS Systems Manager OpsCenter. Enforce policies and detect violations to all AWS accounts using AWS Security Hub.**

Explanation
AWS Control Tower provides a single location to easily set up your new well-architected multi-account environment and govern your AWS workloads with rules for security, operations, and internal compliance. You can automate the setup of your AWS environment with best-practices blueprints for multi-account structure, identity, access management, and account provisioning workflow. For ongoing governance, you can

select and apply pre-packaged policies enterprise-wide or to specific groups of accounts.



AWS Control Tower provides three methods for creating member accounts:

- Through the Account Factory console that is part of AWS Service Catalog.

- Through the Enroll account feature within AWS Control Tower.

- From your AWS Control Tower landing zone's management account, using Lambda code and appropriate IAM roles.

AWS Control Tower offers "guardrails" for ongoing governance of your AWS environment. Guardrails provide governance controls by preventing the deployment of resources that don't conform to selected policies or detecting non-conformance of provisioned resources. AWS Control Tower automatically implements guardrails using multiple building blocks such as AWS CloudFormation to establish a baseline, AWS Organizations service control policies (SCPs) to prevent configuration changes, and AWS Config rules to continuously detect non-conformance.

In this scenario, the requirement is to simplify the creation of AWS accounts that have governance guardrails and a defined baseline in place. To save time and resources, you can use AWS Control Tower to automate account creation. With the appropriate user group permissions, you can specify standardized baselines and network configurations for all accounts in the organization.

Hence, the correct answer is: **Set up an AWS Control Tower Landing Zone. Enable pre-packaged guardrails to enforce policies or detect violations.**

The option that says: **Configure AWS Resource Access Manager (AWS RAM) to launch new AWS accounts as well as standardize the baselines and network configurations for each organization unit** is incorrect. The AWS Resource Access Manager (RAM) service simply helps you to securely share your resources across AWS accounts or within your organization or organizational units (OUs) in AWS Organizations. It is not capable of launching new AWS accounts with preapproved configurations.

The option that says: **Set up an AWS Config aggregator on the root account of the organization to enable multi-account, multi-region data aggregation. Deploy conformance packs to standardize the baselines and network configurations for all accounts** is incorrect. AWS Config cannot provision accounts. A conformance pack is only a collection of AWS Config rules and remediation actions that can be easily deployed as a single entity in an account and a Region or across an organization in AWS Organizations.

The option that says: **Centralized the creation of AWS accounts using AWS Systems Manager OpsCenter. Enforce policies and detect violations to all AWS accounts using AWS Security Hub** is incorrect. AWS Systems Manager is just a collection of services used to manage applications and infrastructure running in AWS that is usually in a single AWS account. The AWS Systems Manager OpsCenter service is just one of the capabilities of AWS Systems Manager, provides a central location where operations engineers and IT professionals can view, investigate, and resolve operational work items (OpsItems) related to AWS resources.

References:

https://docs.aws.amazon.com/controltower/latest/userguide/account-factory.html

https://aws.amazon.com/blogs/mt/how-to-automate-the-creation-of-multiple-accounts-in-aws-control-tower/

https://aws.amazon.com/blogs/aws/aws-control-tower-set-up-govern-a-multi-account-aws-environment/

Question 33: **Correct**
A DevOps Engineer is required to design a cloud architecture in AWS. The Engineer is planning to develop a highly available and fault-tolerant architecture consisting of an Elastic Load Balancer and an Auto Scaling group of EC2 instances deployed across multiple Availability Zones. This will be used by an online accounting application that requires path-based routing, host-based routing, and bi-directional streaming using Remote Procedure Call (gRPC).

Which configuration will satisfy the given requirement?

- **Configure an Application Load Balancer in front of the auto-scaling group. Select gRPC as the protocol version.**

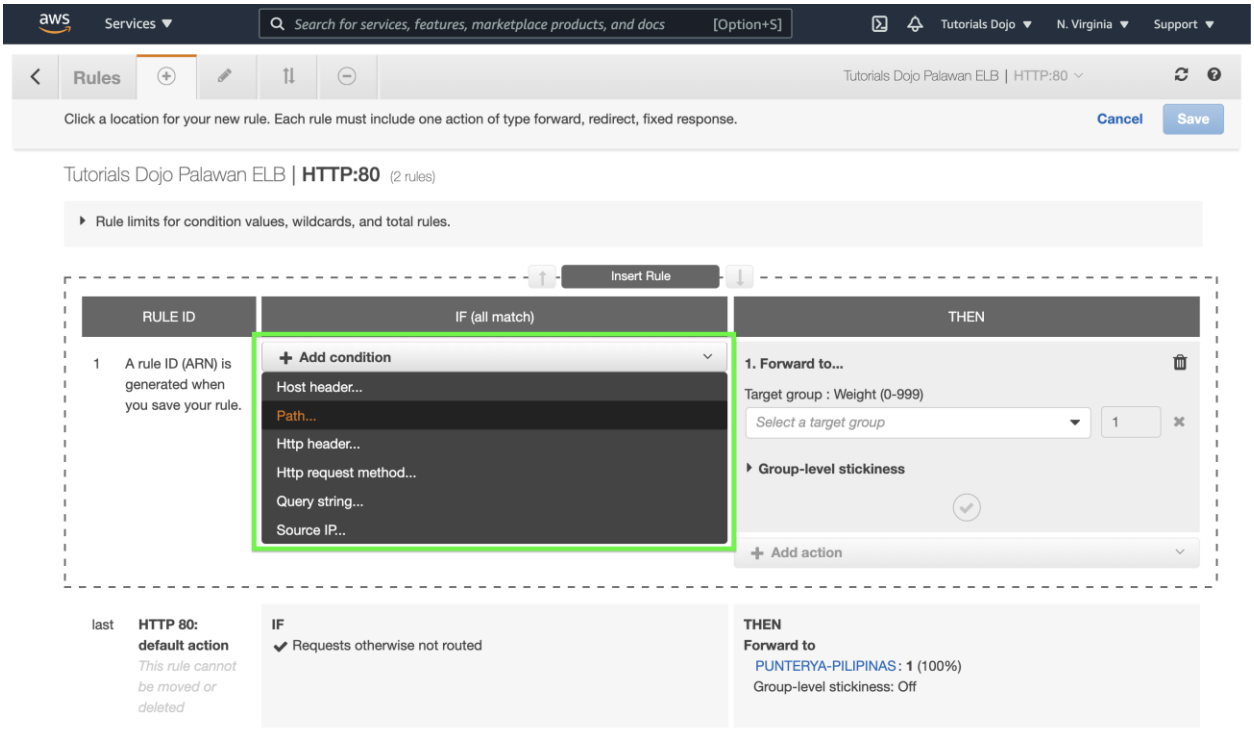
  (Correct)

- **Configure a Network Load Balancer in front of the auto-scaling group. Create an AWS Global Accelerator accelerator and set the load balancer as an endpoint.**

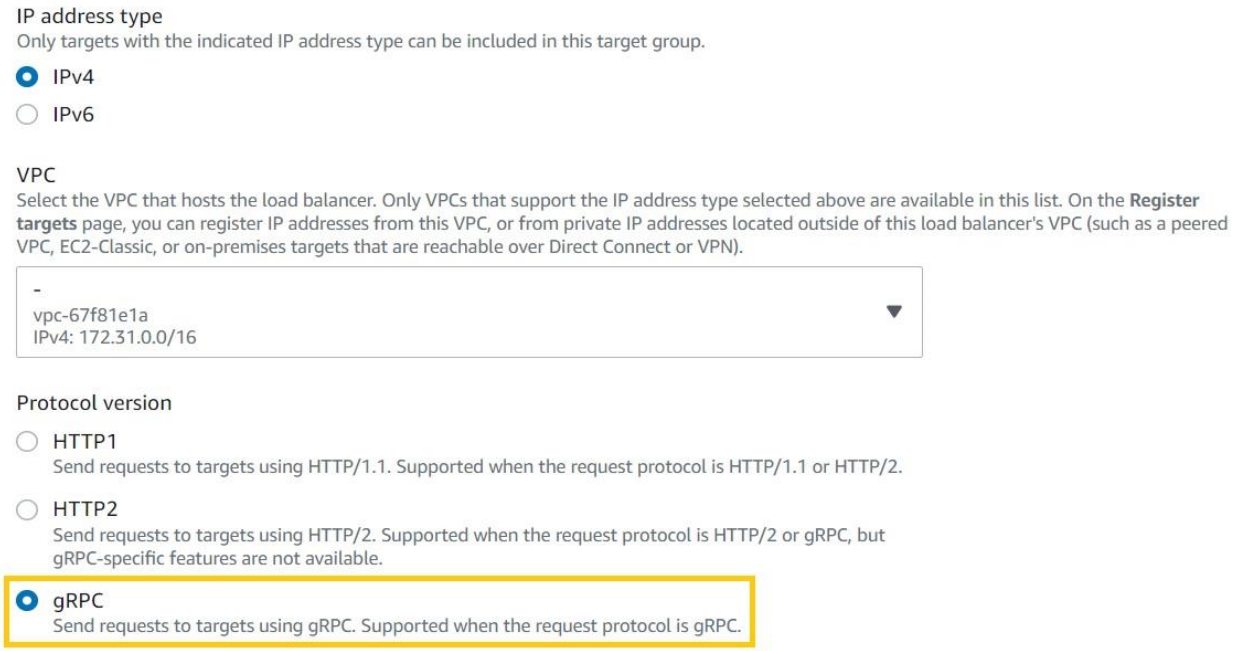
- **Configure a Gateway Load Balancer in front of the auto-scaling group. Ensure that the IP Listener Routing uses the GENEVE protocol on port 6081 to allow gRPC response traffic.**

- **Configure a Network Load Balancer in front of the auto-scaling group. Use a UDP listener for routing.**

Explanation
Application Load Balancer operates at the request level (layer 7), routing traffic to targets (EC2 instances, containers, IP addresses, and Lambda functions) based on the content of the request. Ideal for advanced load balancing of HTTP and HTTPS traffic, Application Load Balancer provides advanced request routing targeted at delivery of modern application architectures, including microservices and container-based applications. Application Load Balancer simplifies and improves the security of your application, by ensuring that the latest SSL/TLS ciphers and protocols are used at all times.



If your application is composed of several individual services, an Application Load Balancer can route a request to a service based on the content of the request such as Host field, Path URL, HTTP header, HTTP method, Query string, or Source IP address.



ALBs can also route and load balance gRPC traffic between microservices or between gRPC-enabled clients and services. This will allow customers to seamlessly introduce gRPC traffic management in their architectures without changing any of the underlying infrastructure on their clients or services.

Therefore, the correct answer is: **Configure an Application Load Balancer in front of the auto-scaling group. Select gRPC as the protocol version.**

The option that says: **Configure a Network Load Balancer in front of the auto-scaling group. Use a UDP listener for routing** is incorrect. Network Load Balancers do not support gRPC.

The option that says: **Configure a Gateway Load Balancer in front of the auto-scaling group. Ensure that the IP Listener Routing uses the GENEVE protocol on port 6081 to allow gRPC response traffic** is incorrect. A Gateway Load Balancer operates as a Layer 3 Gateway and a Layer 4 Load Balancing service. Do take note that the gRPC protocol is at Layer 7 of the OSI Model so this service is not appropriate for this scenario.

The option that says: **Configure a Network Load Balancer in front of the auto-scaling group. Create an AWS Global Accelerator accelerator and set the load balancer as an endpoint** is incorrect. AWS Global Accelerator simply optimizes application performance by routing user traffic to the congestion-free, redundant AWS global network instead of the public internet.

References:

https://aws.amazon.com/elasticloadbalancing/features

https://aws.amazon.com/elasticloadbalancing/faqs/

AWS Elastic Load Balancing Overview:

https://youtu.be/UBl5dw59DO8

Check out this AWS Elastic Load Balancing (ELB) Cheat Sheet:

https://tutorialsdojo.com/aws-elastic-load-balancing-elb/

Application Load Balancer vs Network Load Balancer vs Gateway Load Balancer:

https://tutorialsdojo.com/application-load-balancer-vs-network-load-balancer-vs-classic-load-balancer/

Question 34: **Incorrect**
A Solutions Architect of a multinational gaming company develops video games for PS4, Xbox One, and Nintendo Switch consoles, plus a number of mobile games for Android and iOS. Due to the wide range of their products and services, the architect proposed that they use API Gateway.

What are the key features of API Gateway that the architect can tell to the client? (Select TWO.)

- ☐ **It automatically provides a query language for your APIs similar to GraphQL.**
- ☑ **Enables you to build RESTful APIs and WebSocket APIs that are optimized for serverless workloads.**

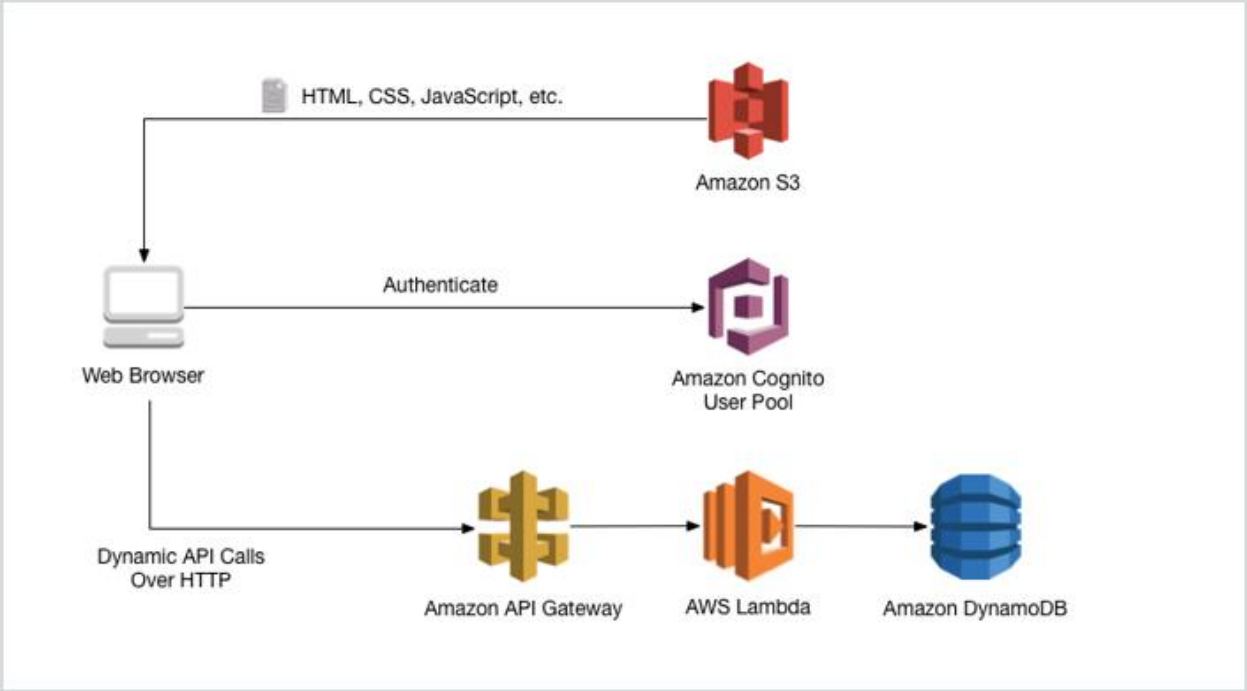
  (Correct)
- ☐ **You pay only for the API calls you receive and the amount of data transferred out.**

  (Correct)
- ☑ **Enables you to run applications requiring high levels of inter-node communications at scale on AWS through its custom-built operating system (OS) bypass hardware interface.**

  (Incorrect)
- ☐ **Provides you with static anycast IP addresses that serve as a fixed entry point to your applications hosted in one or more AWS Regions.**

Explanation
Amazon API Gateway is a fully managed service that makes it easy for developers to create, publish, maintain, monitor, and secure APIs at any scale. With a few clicks in the

AWS Management Console, you can create an API that acts as a “front door” for applications to access data, business logic, or functionality from your back-end services, such as workloads running on Amazon Elastic Compute Cloud (Amazon EC2), code running on AWS Lambda, or any web application. Since it can use AWS Lambda, you can run your APIs without servers.



Amazon API Gateway handles all the tasks involved in accepting and processing up to hundreds of thousands of concurrent API calls, including traffic management, authorization and access control, monitoring, and API version management. Amazon API Gateway has no minimum fees or startup costs. You pay only for the API calls you receive and the amount of data transferred out.

Hence, the correct answers are:

- Enables you to build RESTful APIs and WebSocket APIs that are optimized for serverless workloads

- You pay only for the API calls you receive and the amount of data transferred out.

The option that says: **It automatically provides a query language for your APIs similar to GraphQL** is incorrect because this is not provided by API Gateway.

The option that says: **Provides you with static anycast IP addresses that serve as a fixed entry point to your applications hosted in one or more AWS Regions** is incorrect because this is a capability of AWS Global Accelerator and not API Gateway.

The option that says: **Enables you to run applications requiring high levels of inter-node communications at scale on AWS through its custom-built operating system (OS) bypass hardware interface** is incorrect because this is a capability of Elastic Fabric Adapter and not API Gateway.

References:

https://aws.amazon.com/api-gateway/

https://aws.amazon.com/api-gateway/features/

Check out this Amazon API Gateway Cheat Sheet:

https://tutorialsdojo.com/amazon-api-gateway/

Tutorials Dojo's AWS Certified Solutions Architect Associate Exam Study Guide:

https://tutorialsdojo.com/aws-certified-solutions-architect-associate/

Question 35: **Incorrect**
A GraphQL API hosted is hosted in an Amazon EKS cluster with Fargate launch type and deployed using AWS SAM. The API is connected to an Amazon DynamoDB table with an Amazon DynamoDB Accelerator (DAX) as its data store. Both resources are hosted in the us-east-1 region.

The AWS IAM authenticator for Kubernetes is integrated into the EKS cluster for role-based access control (RBAC) and cluster authentication. A solutions architect must improve network security by preventing database calls from traversing the public internet. An automated cross-account backup for the DynamoDB table is also required for long-term retention.

Which of the following should the solutions architect implement to meet the requirement?

- **Create a DynamoDB interface endpoint. Set up a stateless rule using AWS Network Firewall to control all outbound traffic to only use the** `dynamodb.us-`

`east-1.amazonaws.com` **endpoint. Integrate the DynamoDB table with Amazon Timestream to allow point-in-time recovery from a different AWS account.**

- **Create a DynamoDB gateway endpoint. Set up a Network Access Control List (NACL) rule that allows outbound traffic to the `dynamodb.us-east-1.amazonaws.com` gateway endpoint. Use the built-in on-demand DynamoDB backups for cross-account backup and recovery.**

- **Create a DynamoDB gateway endpoint. Associate the endpoint to the appropriate route table. Use AWS Backup to automatically copy the on-demand DynamoDB backups to another AWS account for disaster recovery.**

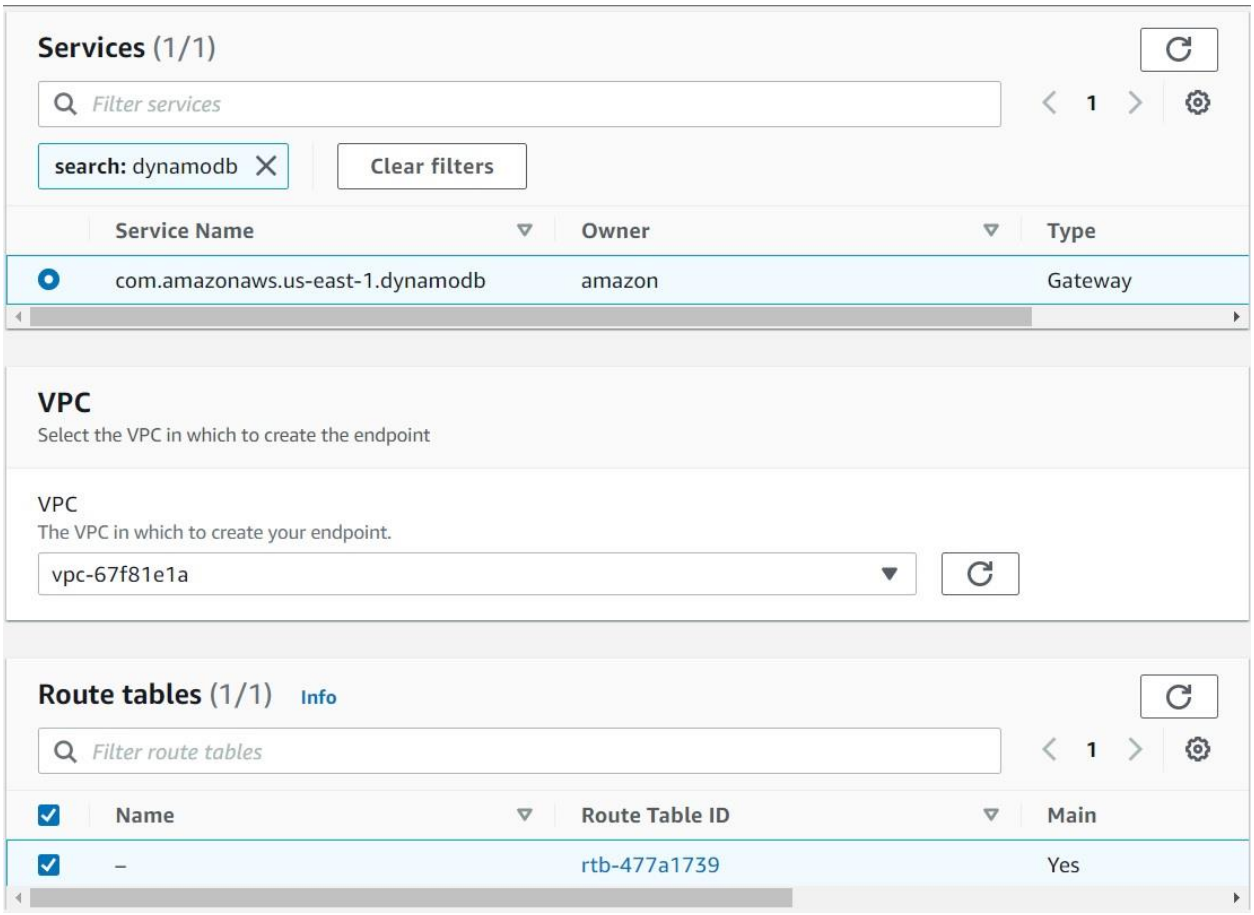
  (Correct)

- **Create a DynamoDB interface endpoint. Associate the endpoint to the appropriate route table. Enable Point-in-Time Recovery (PITR) to restore the DynamoDB table to a particular point in time on the same or a different AWS account.**

  (Incorrect)

Explanation
Since DynamoDB tables are public resources, applications within a VPC rely on an Internet Gateway to route traffic to/from Amazon DynamoDB. You can use a Gateway endpoint if you want to keep the traffic between your VPC and Amazon DynamoDB within the Amazon network. This way, resources residing in your VPC can use their private IP addresses to access DynamoDB with no exposure to the public internet.

When you create a DynamoDB Gateway endpoint, you specify the VPC where it will be deployed as well as the route table that will be associated with the endpoint. The route table will be updated with an Amazon DynamoDB prefix list (list of CIDR blocks) as the destination and the endpoint's ID as the target.



DynamoDB on-demand backups are available at no additional cost beyond the normal pricing that's associated with backup storage size. DynamoDB on-demand backups cannot be copied to a different account or Region. To create backup copies across AWS accounts and Regions and for other advanced features, you should use AWS Backup.

With AWS Backup, you can configure backup policies and monitor activity for your AWS resources and on-premises workloads in one place. Using DynamoDB with AWS Backup, you can copy your on-demand backups across AWS accounts and Regions, add cost allocation tags to on-demand backups, and transition on-demand backups to cold storage for lower costs. To use these advanced features, you must opt into AWS Backup. Opt-in choices apply to the specific account and AWS Region, so you might have to opt into multiple Regions using the same account.

Hence, the correct answer is: **Create a DynamoDB gateway endpoint. Associate the endpoint to the appropriate route table. Use AWS Backup to automatically copy the on-demand DynamoDB backups to another AWS account for disaster recovery.**

The option that says: **Create a DynamoDB interface endpoint. Associate the endpoint to the appropriate route table. Enable Point-in-Time Recovery (PITR) to restore the DynamoDB table to a particular point in time on the same or a different AWS account** is

incorrect because Amazon DynamoDB does not support interface endpoint. You have to create a DynamoDB Gateway endpoint instead. In addition, the Point-in-Time Recovery (PITR) feature is not capable of restoring a DynamoDB table to a particular point in time in a different AWS account. If this functionality is needed, you have to use the AWS Backup service instead.

The option that says: **Create a DynamoDB gateway endpoint. Set up a Network Access Control List (NACL) rule that allows outbound traffic to the `dynamodb.us-east-1.amazonaws.com` gateway endpoint. Use the built-in on-demand DynamoDB backups for cross-account backup and recovery** is incorrect because using a Network Access Control List alone is not enough to prevent traffic traversing to the public Internet. Moreover, you cannot copy DynamoDB on-demand backups to a different account or Region.

The option that says: **Create a DynamoDB interface endpoint. Set up a stateless rule using AWS Network Firewall to control all outbound traffic to only use the `dynamodb.us-east-1.amazonaws.com` endpoint. Integrate the DynamoDB table with Amazon Timestream to allow point-in-time recovery from a different AWS account** is incorrect. Keep in mind that the `dynamodb.us-east-1.amazonaws.com` is a public service endpoint for Amazon DynamoDB. Since the application is able to communicate with Amazon DynamoDB prior to the required architectural change, it's implied that no firewalls (security group, NACL, etc.) are blocking traffic to/from Amazon DynamoDB, hence, adding an NACL rule to allow outbound traffic to DynamoDB is unnecessary. Furthermore, the use of the AWS Network Firewall in this solution is simply incorrect as you have to integrate this with your Amazon VPC. The use of Amazon Timestream is also wrong since this is a time series database service in AWS for IoT and operational applications. You cannot directly integrate DynamoDB and Amazon Timestream for the purpose of point-in-time data recovery.

References:

https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/vpc-endpoints-dynamodb.html

https://aws.amazon.com/blogs/database/how-to-configure-a-private-network-environment-for-amazon-dynamodb-using-vpc-endpoints/

https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/BackupRestore.html

Check out this Amazon DynamoDB Cheat sheet:

https://tutorialsdojo.com/amazon-dynamodb

Question 36: **Incorrect**
Both historical records and frequently accessed data are stored on an on-premises storage system. The amount of current data is growing at an exponential rate. As the storage's capacity is nearing its limit, the company's Solutions Architect has decided to move the historical records to AWS to free up space for the active data.

Which of the following architectures deliver the best solution in terms of cost and operational management?

- **Use AWS Storage Gateway to move the historical records from on-premises to AWS. Choose Amazon S3 Glacier to be the destination for the data. Modify the S3 lifecycle configuration to move the data from the Standard tier to Amazon S3 Glacier Deep Archive after 30 days.**

  (Incorrect)

- **Use AWS DataSync to move the historical records from on-premises to AWS. Choose Amazon S3 Standard to be the destination for the data. Modify the S3 lifecycle configuration to move the data from the Standard tier to Amazon S3 Glacier Deep Archive after 30 days.**

- **Use AWS DataSync to move the historical records from on-premises to AWS. Choose Amazon S3 Glacier Deep Archive to be the destination for the data.**

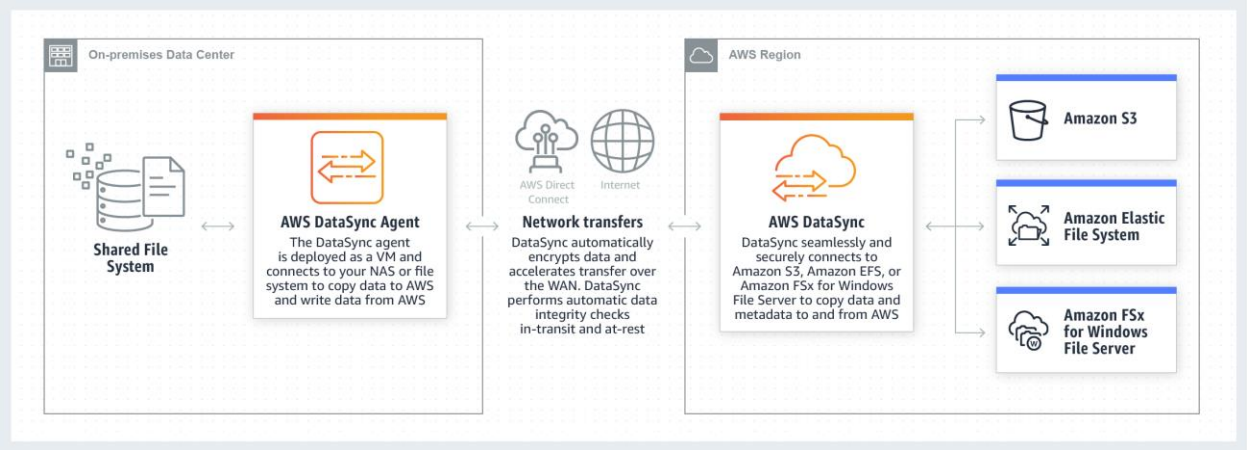
  (Correct)

- **Use AWS Storage Gateway to move the historical records from on-premises to AWS. Choose Amazon S3 Glacier Deep Archive to be the destination for the data.**

Explanation
AWS DataSync makes it simple and fast to move large amounts of data online between on-premises storage and Amazon S3, Amazon Elastic File System (Amazon EFS), or Amazon FSx for Windows File Server. Manual tasks related to data transfers can slow down migrations and burden IT operations. DataSync eliminates or automatically handles many of these tasks, including scripting copy jobs, scheduling, and monitoring

transfers, validating data, and optimizing network utilization. The DataSync software agent connects to your Network File System (NFS), Server Message Block (SMB) storage, and your self-managed object storage, so you don't have to modify your applications.

DataSync can transfer hundreds of terabytes and millions of files at speeds up to 10 times faster than open-source tools, over the Internet or AWS Direct Connect links. You can use DataSync to migrate active data sets or archives to AWS, transfer data to the cloud for timely analysis and processing, or replicate data to AWS for business continuity. Getting started with DataSync is easy: deploy the DataSync agent, connect it to your file system, select your AWS storage resources, and start moving data between them. You pay only for the data you move.



Since the problem is mainly about moving historical records from on-premises to AWS, using AWS DataSync is a more suitable solution. You can use DataSync to move cold data from expensive on-premises storage systems directly to durable and secure long-term storage, such as Amazon S3 Glacier or Amazon S3 Glacier Deep Archive.

Hence, the correct answer is the option that says: **Use AWS DataSync to move the historical records from on-premises to AWS. Choose Amazon S3 Glacier Deep Archive to be the destination for the data.**

The following options are both incorrect:

- Use AWS Storage Gateway to move the historical records from on-premises to AWS. Choose Amazon S3 Glacier Deep Archive to be the destination for the data.

- Use AWS Storage Gateway to move the historical records from on-premises to AWS. Choose Amazon S3 Glacier to be the destination for the data. Modify the S3 lifecycle configuration to move the data from the Standard tier to Amazon S3 Glacier Deep Archive after 30 days.

Although you can copy data from on-premises to AWS with Storage Gateway, it is not suitable for transferring large sets of data to AWS. Storage Gateway is mainly used in providing low-latency access to data by caching frequently accessed data on-premises while storing archive data securely and durably in Amazon cloud storage services. Storage Gateway optimizes data transfer to AWS by sending only changed data and compressing data.

The option that says: **Use AWS DataSync to move the historical records from on-premises to AWS. Choose Amazon S3 Standard to be the destination for the data. Modify the S3 lifecycle configuration to move the data from the Standard tier to Amazon S3 Glacier Deep Archive after 30 days** is incorrect because, with AWS DataSync, you can transfer data from on-premises directly to Amazon S3 Glacier Deep Archive. You don't have to configure the S3 lifecycle policy and wait for 30 days to move the data to Glacier Deep Archive.

References:

https://aws.amazon.com/datasync/faqs/

https://aws.amazon.com/storagegateway/faqs/

Check out these AWS DataSync and Storage Gateway Cheat Sheets:

https://tutorialsdojo.com/aws-datasync/

https://tutorialsdojo.com/aws-storage-gateway/

AWS Storage Gateway vs DataSync:

https://youtu.be/tmfe1rO-AUs

Question 37: **Correct**
A music publishing company is building a multitier web application that requires a key-value store which will save the document models. Each model is composed of band ID, album ID, song ID, composer ID, lyrics, and other data. The web tier will be hosted in an Amazon ECS cluster with AWS Fargate launch type.

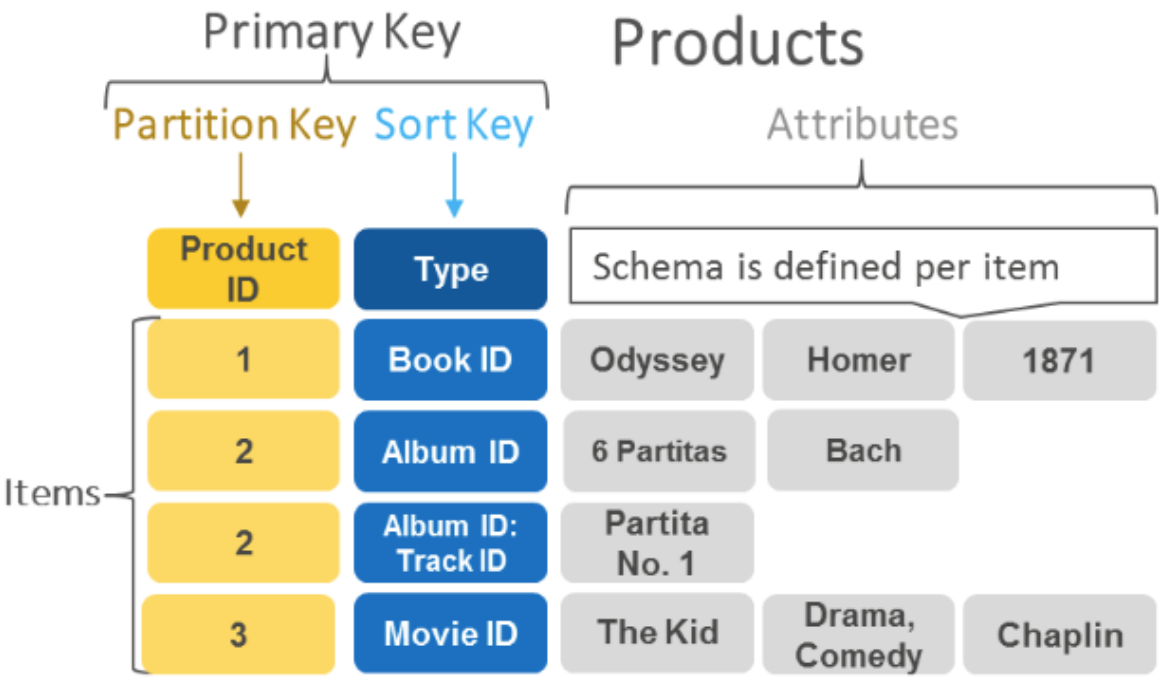
Which of the following is the MOST suitable setup for the database-tier?

- **Use Amazon WorkDocs to store the document models.**
- **Launch a DynamoDB table.**

  (Correct)
- **Launch an Amazon RDS database with Read Replicas.**
- **Launch an Amazon Aurora Serverless database.**

Explanation

Amazon DynamoDB is a fast and flexible NoSQL database service for all applications that need consistent, single-digit millisecond latency at any scale. It is a fully managed cloud database and supports both document and key-value store models. Its flexible data model, reliable performance, and automatic scaling of throughput capacity makes it a great fit for mobile, web, gaming, ad tech, IoT, and many other applications.



Hence, the correct answer is: **Launch a DynamoDB table.**

The option that says: **Launch an Amazon RDS database with Read Replicas** is incorrect because this is a relational database. This is not suitable to be used as a key-value store. A better option is to use DynamoDB as it supports both document and key-value store models.

The option that says: **Use Amazon WorkDocs to store the document models** is incorrect because Amazon WorkDocs simply enables you to share content, provide rich feedback, and collaboratively edit documents. It is not a key-value store like DynamoDB.

The option that says: **Launch an Amazon Aurora Serverless database** is incorrect because this type of database is not suitable to be used as a key-value store. Amazon Aurora Serverless is an on-demand, auto-scaling configuration for Amazon Aurora where the database will automatically start-up, shut down, and scale capacity up or down based on your application's needs. It enables you to run your database in the cloud without managing any database instances. It's a simple, cost-effective option for infrequent, intermittent, or unpredictable workloads and not as a key-value store.

References:

https://aws.amazon.com/dynamodb/

https://aws.amazon.com/nosql/key-value/

Check out this Amazon DynamoDB Cheat Sheet:

https://tutorialsdojo.com/amazon-dynamodb/

Amazon DynamoDB Overview:

https://youtu.be/3ZOyUNIeorU

Question 38: **Correct**
A company launched a website that accepts high-quality photos and turns them into a downloadable video montage. The website offers a free and a premium account that guarantees faster processing. All requests by both free and premium members go through a single SQS queue and then processed by a group of EC2 instances that generate the videos. The company needs to ensure that the premium users who paid for the service have higher priority than the free members.

How should the company re-design its architecture to address this requirement?

- **Use Amazon S3 to store and process the photos and then generate the video montage afterward.**

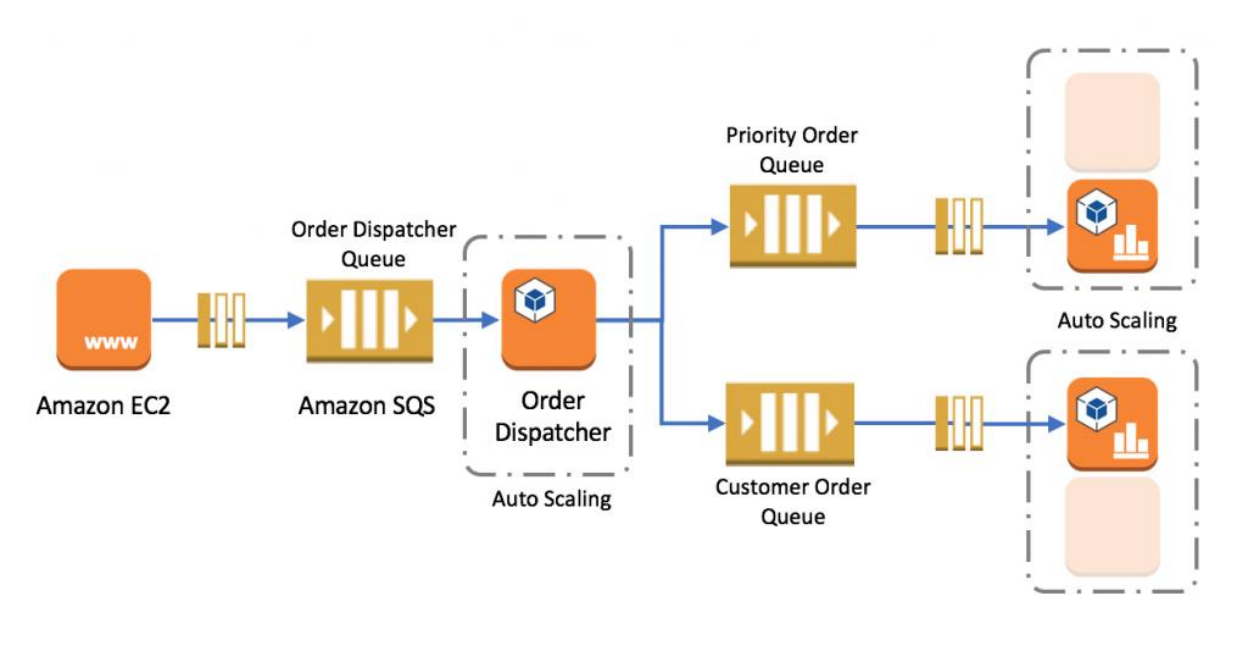
- **For the requests made by premium members, set a higher priority in the SQS queue so it will be processed first compared to the requests made by free members.**

- **Use Amazon Kinesis to process the photos and generate the video montage in real-time.**

- **Create an SQS queue for free members and another one for premium members. Configure your EC2 instances to consume messages from the premium queue first and if it is empty, poll from the free members' SQS queue.**

  (Correct)

Explanation
Amazon Simple Queue Service (SQS) is a fully managed message queuing service that enables you to decouple and scale microservices, distributed systems, and serverless applications. SQS eliminates the complexity and overhead associated with managing and operating message-oriented middleware and empowers developers to focus on differentiating work. Using SQS, you can send, store, and receive messages between software components at any volume without losing messages or requiring other services to be available.



In this scenario, it is best to create 2 separate SQS queues for each type of member. The SQS queues for the premium members can be polled first by the EC2 Instances and once completed, the messages from the free members can be processed next.

Hence, the correct answer is: **Create an SQS queue for free members and another one for premium members. Configure your EC2 instances to consume messages from the premium queue first and if it is empty, poll from the free members' SQS queue.**

The option that says: **For the requests made by premium members, set a higher priority in the SQS queue so it will be processed first compared to the requests made by free members** is incorrect as you cannot set a priority to individual items in the SQS queue.

The option that says: **Using Amazon Kinesis to process the photos and generate the video montage in real time** is incorrect as Amazon Kinesis is used to process streaming data and it is not applicable in this scenario.

The option that says: **Using Amazon S3 to store and process the photos and then generating the video montage afterward** is incorrect as Amazon S3 is used for durable storage and not for processing data.

Reference:

https://docs.aws.amazon.com/AWSSimpleQueueService/latest/SQSDeveloperGuide/sqs-best-practices.html

Check out this Amazon SQS Cheat Sheet:

https://tutorialsdojo.com/amazon-sqs/

Question 39: **Incorrect**
A software company has resources hosted in AWS and on-premises servers. You have been requested to create a decoupled architecture for applications which make use of both resources.

Which of the following options are valid? (Select TWO.)

- ☑ **Use SQS to utilize both on-premises servers and EC2 instances for your decoupled application**

  (Correct)

- ☐ **Use SWF to utilize both on-premises servers and EC2 instances for your decoupled application**

  (Correct)

- ☑ **Use VPC peering to connect both on-premises servers and EC2 instances for your decoupled application**

  (Incorrect)

- ☐ **Use DynamoDB to utilize both on-premises servers and EC2 instances for your decoupled application**

- ☐ **Use RDS to utilize both on-premises servers and EC2 instances for your decoupled application**

Explanation

Amazon Simple Queue Service (SQS) and **Amazon Simple Workflow Service (SWF)** are the services that you can use for creating a decoupled architecture in AWS. Decoupled architecture is a type of computing architecture that enables computing components or layers to execute independently while still interfacing with each other.

Amazon SQS offers reliable, highly-scalable hosted queues for storing messages while they travel between applications or microservices. Amazon SQS lets you move data between distributed application components and helps you decouple these components. Amazon SWF is a web service that makes it easy to coordinate work across distributed application components.

Using RDS to utilize both on-premises servers and EC2 instances for your decoupled application and **using DynamoDB to utilize both on-premises servers and EC2 instances for your decoupled application** are incorrect as RDS and DynamoDB are database services.

Using VPC peering to connect both on-premises servers and EC2 instances for your decoupled application is incorrect because you can't create a VPC peering for your on-premises network and AWS VPC.

References:

https://aws.amazon.com/sqs/

http://docs.aws.amazon.com/amazonswf/latest/developerguide/swf-welcome.html

Check out this Amazon SQS Cheat Sheet:

https://tutorialsdojo.com/amazon-sqs/

Amazon Simple Workflow (SWF) vs AWS Step Functions vs Amazon SQS:

https://tutorialsdojo.com/amazon-simple-workflow-swf-vs-aws-step-functions-vs-amazon-sqs/

Comparison of AWS Services Cheat Sheets:

https://tutorialsdojo.com/comparison-of-aws-services/

Question 40: **Correct**
A digital media company shares static content to its premium users around the world and also to their partners who syndicate their media files. The company is looking for ways to reduce its server costs and securely deliver their data to their customers globally with low latency.

Which combination of services should be used to provide the MOST suitable and cost-effective architecture? (Select TWO.)

- [x] **Amazon S3**

  (Correct)

- [x] **Amazon CloudFront**

  (Correct)

- [ ] **AWS Fargate**

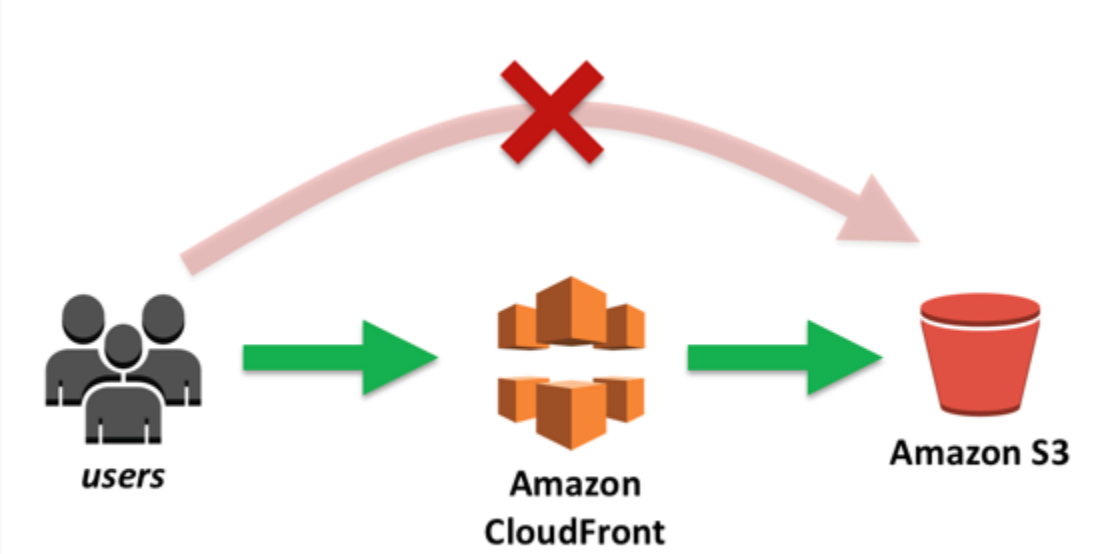
- [ ] **AWS Global Accelerator**

- [ ] **AWS Lambda**

Explanation
Amazon CloudFront is a fast content delivery network (CDN) service that securely delivers data, videos, applications, and APIs to customers globally with low latency, high transfer speeds, all within a developer-friendly environment.

CloudFront is integrated with AWS – both physical locations that are directly connected to the AWS global infrastructure, as well as other AWS services. CloudFront works seamlessly with services, including AWS Shield for DDoS mitigation, Amazon S3, Elastic Load Balancing or Amazon EC2 as origins for your applications, and Lambda@Edge to run custom code closer to customers' users and to customize the user experience. Lastly, if you use AWS origins such as Amazon S3, Amazon EC2 or Elastic Load

Balancing, you don't pay for any data transferred between these services and CloudFront.



Amazon S3 is object storage built to store and retrieve any amount of data from anywhere on the Internet. It's a simple storage service that offers an extremely durable, highly available, and infinitely scalable data storage infrastructure at very low costs.

AWS Global Accelerator and Amazon CloudFront are separate services that use the AWS global network and its edge locations around the world. CloudFront improves performance for both cacheable content (such as images and videos) and dynamic content (such as API acceleration and dynamic site delivery). Global Accelerator improves performance for a wide range of applications over TCP or UDP by proxying packets at the edge to applications running in one or more AWS Regions. Global Accelerator is a good fit for non-HTTP use cases, such as gaming (UDP), IoT (MQTT), or Voice over IP, as well as for HTTP use cases that specifically require static IP addresses or deterministic, fast regional failover. Both services integrate with AWS Shield for DDoS protection.

Hence, the correct options are **Amazon CloudFront** and **Amazon S3.**

AWS Fargate is incorrect because this service is just a serverless compute engine for containers that work with both Amazon Elastic Container Service (ECS) and Amazon Elastic Kubernetes Service (EKS). Although this service is more cost-effective than its server-based counterpart, Amazon S3 still costs way less than Fargate, especially for storing static content.

AWS Lambda is incorrect because this simply lets you run your code serverless without provisioning or managing servers. Although this is also a cost-effective service since you have to pay only for the compute time you consume, you can't use this to store static content or as a Content Delivery Network (CDN). A better combination is Amazon CloudFront and Amazon S3.

AWS Global Accelerator is incorrect because this service is more suitable for non-HTTP use cases, such as gaming (UDP), IoT (MQTT), or Voice over IP, as well as for HTTP use cases that specifically require static IP addresses or deterministic, fast regional failover. Moreover, there is no direct way that you can integrate AWS Global Accelerator with Amazon S3. It's more suitable to use Amazon CloudFront instead in this scenario.

References:

https://aws.amazon.com/premiumsupport/knowledge-center/cloudfront-serve-static-website/

https://aws.amazon.com/blogs/networking-and-content-delivery/amazon-s3-amazon-cloudfront-a-match-made-in-the-cloud/

https://aws.amazon.com/global-accelerator/faqs/

Question 41: **Incorrect**
A company wants to organize the way it tracks its spending on AWS resources. A report that summarizes the total billing accrued by each department must be generated at the end of the month.

Which solution will meet the requirements?

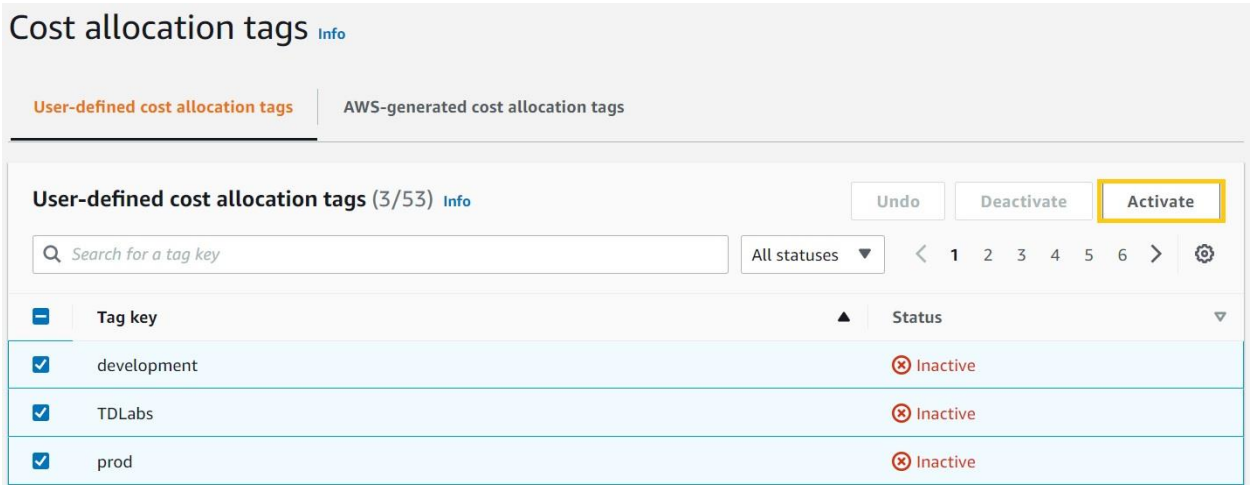
- **Use AWS Cost Explorer to view spending and filter usage data by `Resource`.**
- **Create a Cost and Usage report for AWS services that each department is using.**
- **Tag resources with the department name and configure a budget action in AWS Budget.**

(Incorrect)

- **Tag resources with the department name and enable cost allocation tags.**

  (Correct)

Explanation
A tag is a label that you or AWS assigns to an AWS resource. Each tag consists of a key and a value. For each resource, each tag key must be unique, and each tag key can have only one value. You can use tags to organize your resources and cost allocation tags to track your AWS costs on a detailed level.



After you or AWS applies tags to your AWS resources (such as Amazon EC2 instances or Amazon S3 buckets) and you activate the tags in the Billing and Cost Management console, AWS generates a cost allocation report as a comma-separated value (CSV file) with your usage and costs grouped by your active tags. You can apply tags that represent business categories (such as cost centers, application names, or owners) to organize your costs across multiple services.

Hence, the correct answer is: **Tag resources with the department name and enable cost allocation tags.**

The option that says: **Tag resources with the department name and configure a budget action in AWS Budget** is incorrect. AWS Budgets only allows you to be alerted and run custom actions if your budget thresholds are exceeded.

The option that says: **Use AWS Cost Explorer to view spending and filter usage data by** `Resource` is incorrect. The `Resource` filter just lets you track costs on EC2 instances. This is quite limited compared with using the Cost Allocation Tags method.

The option that says: **Create a Cost and Usage report for AWS services that each department is using** is incorrect. The report must contain a breakdown of costs incurred by each department based on tags and not based on AWS services, which is what the Cost and Usage Report (CUR) contains.

References:

https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/cost-alloc-tags.html

https://aws.amazon.com/blogs/aws-cloud-financial-management/cost-allocation-blog-series-2-aws-generated-vs-user-defined-cost-allocation-tag/

Check out this AWS Billing and Cost Management Cheat sheet:

https://tutorialsdojo.com/aws-billing-and-cost-management/

Question 42: **Incorrect**
A company has an on-premises MySQL database that needs to be replicated in Amazon S3 as CSV files. The database will eventually be launched to an Amazon Aurora Serverless cluster and be integrated with an RDS Proxy to allow the web applications to pool and share database connections. Once data has been fully copied, the ongoing changes to the on-premises database should be continually streamed into the S3 bucket. The company wants a solution that can be implemented with little management overhead yet still highly secure.

Which ingestion pattern should a solutions architect take?

- **Set up a full load replication task using AWS Database Migration Service (AWS DMS). Launch an AWS DMS endpoint with SSL using the AWS Network Firewall service.**

  (Incorrect)

- **Create a full load and change data capture (CDC) replication task using AWS Database Migration Service (AWS DMS). Add a new Certificate Authority (CA) certificate and create an AWS DMS endpoint with SSL.**

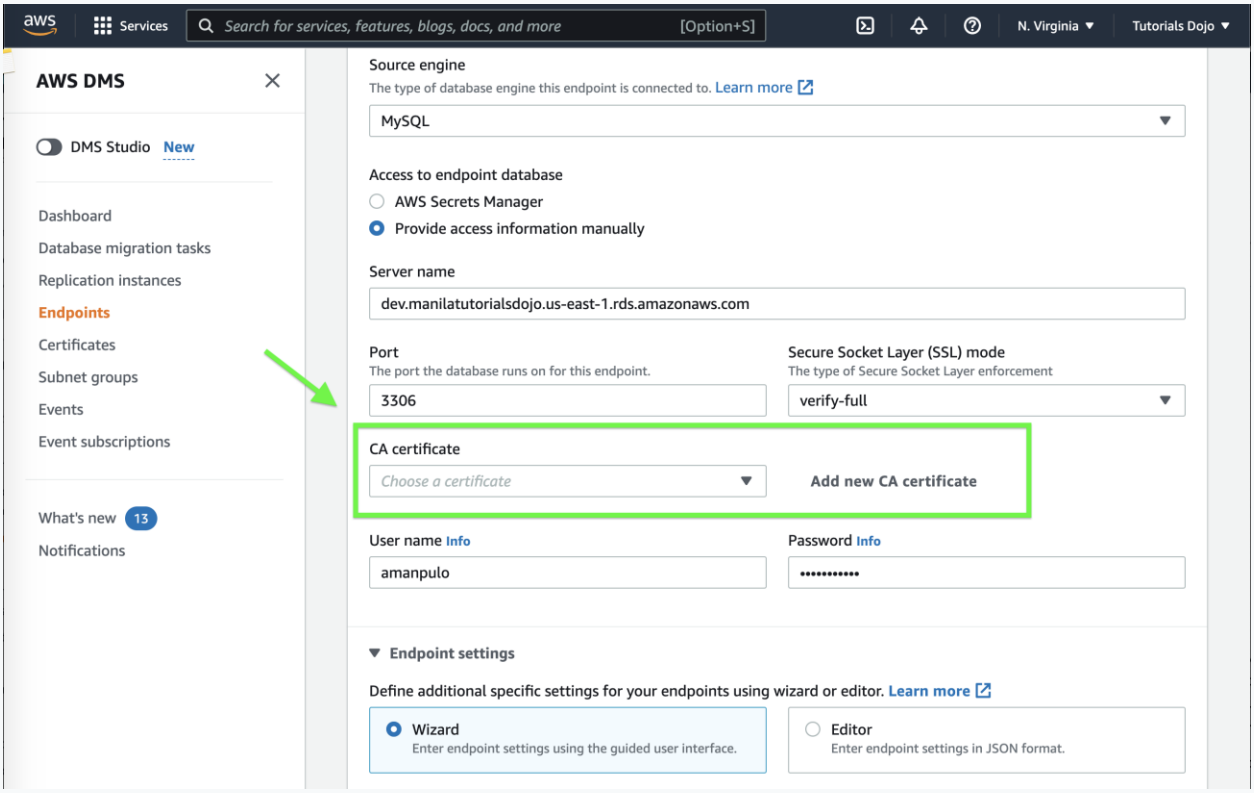
(Correct)

- **Use AWS Schema Conversion Tool (AWS SCT) to convert MySQL data to CSV files. Set up the AWS Server Migration Service (AWS MGN) to capture ongoing changes from the on-premises MySQL database and send them to Amazon S3.**

- **Use an AWS Snowball Edge cluster to migrate data to Amazon S3 and AWS DataSync to capture ongoing changes. Create your own custom AWS KMS envelope encryption key for the associated AWS Snowball Edge job.**

Explanation

AWS Database Migration Service (AWS DMS) is a cloud service that makes it easy to migrate relational databases, data warehouses, NoSQL databases, and other types of data stores. You can use AWS DMS to migrate your data into the AWS Cloud, between on-premises instances (through an AWS Cloud setup) or between combinations of cloud and on-premises setups. With AWS DMS, you can perform one-time migrations, and you can replicate ongoing changes to keep sources and targets in sync.



You can migrate data to Amazon S3 using AWS DMS from any of the supported database sources. When using Amazon S3 as a target in an AWS DMS task, both full

load and change data capture (CDC) data is written to comma-separated value (.csv) format by default.

The comma-separated value (.csv) format is the default storage format for Amazon S3 target objects. For more compact storage and faster queries, you can instead use Apache Parquet (.parquet) as the storage format.

You can encrypt connections for source and target endpoints by using Secure Sockets Layer (SSL). To do so, you can use the AWS DMS Management Console or AWS DMS API to assign a certificate to an endpoint. You can also use the AWS DMS console to manage your certificates.

Not all databases use SSL in the same way. Amazon Aurora MySQL-Compatible Edition uses the server name, the endpoint of the primary instance in the cluster, as the endpoint for SSL. An Amazon Redshift endpoint already uses an SSL connection and does not require an SSL connection set up by AWS DMS.

Hence, the correct answer is: **Create a full load and change data capture (CDC) replication task using AWS Database Migration Service (AWS DMS). Add a new Certificate Authority (CA) certificate and create an AWS DMS endpoint with SSL.**

The option that says: **Set up a full load replication task using AWS Database Migration Service (AWS DMS). Launch an AWS DMS endpoint with SSL using the AWS Network Firewall service** is incorrect because a full load replication task alone won't capture ongoing changes to the database. You still need to implement a change data capture (CDC) replication to copy the recent changes after the migration. Moreover, the AWS Network Firewall service is not capable of creating an AWS DMS endpoint with SSL. The Certificate Authority (CA) certificate can be directly uploaded to the AWS DMS console without the AWS Network Firewall at all.

The option that says: **Use an AWS Snowball Edge cluster to migrate data to Amazon S3 and AWS DataSync to capture ongoing changes** is incorrect. While this is doable, it's more suited to the migration of large databases which require the use of two or more Snowball Edge appliances. Also, the usage of AWS DataSync for replicating ongoing changes to Amazon S3 requires extra steps that can be simplified with AWS DMS.

The option that says: **Use AWS Schema Conversion Tool (AWS SCT) to convert MySQL data to CSV files. Set up the AWS Application Migration Service (AWS MGN) to capture ongoing changes from the on-premises MySQL database and send them to Amazon S3** is incorrect. AWS SCT is not used for data replication; it just eases up the conversion of source databases to a format compatible with the target database when migrating. In addition, using the AWS Application Migration Service (AWS MGN) for this scenario is inappropriate. This service is primarily used for lift-and-shift migrations of applications from physical infrastructure, VMware vSphere, Microsoft Hyper-V, Amazon

Elastic Compute Cloud (AmazonEC2), Amazon Virtual Private Cloud (Amazon VPC), and other clouds to AWS.

References:

https://aws.amazon.com/blogs/big-data/loading-ongoing-data-lake-changes-with-aws-dms-and-aws-glue/

https://docs.aws.amazon.com/dms/latest/userguide/Welcome.html

https://docs.aws.amazon.com/dms/latest/userguide/CHAP_Security.html#CHAP_Security.SSL.Limitations

Check out this AWS Database Migration Service Cheat Sheet:

https://tutorialsdojo.com/aws-database-migration-service/

AWS Migration Services Overview:

https://youtu.be/yqNBkFMnsL8

Question 43: **Correct**
A company has a static corporate website hosted in a standard S3 bucket and a new web domain name that was registered using Route 53. You are instructed by your manager to integrate these two services in order to successfully launch their corporate website.

What are the prerequisites when routing traffic using Amazon Route 53 to a website that is hosted in an Amazon S3 Bucket? (Select TWO.)

- **The Cross-Origin Resource Sharing (CORS) option should be enabled in the S3 bucket**
- **The record set must be of type "MX"**

The S3 bucket must be in the same region as the hosted zone

- ☑

  A registered domain name

  (Correct)

- ☑

  The S3 bucket name must be the same as the domain name

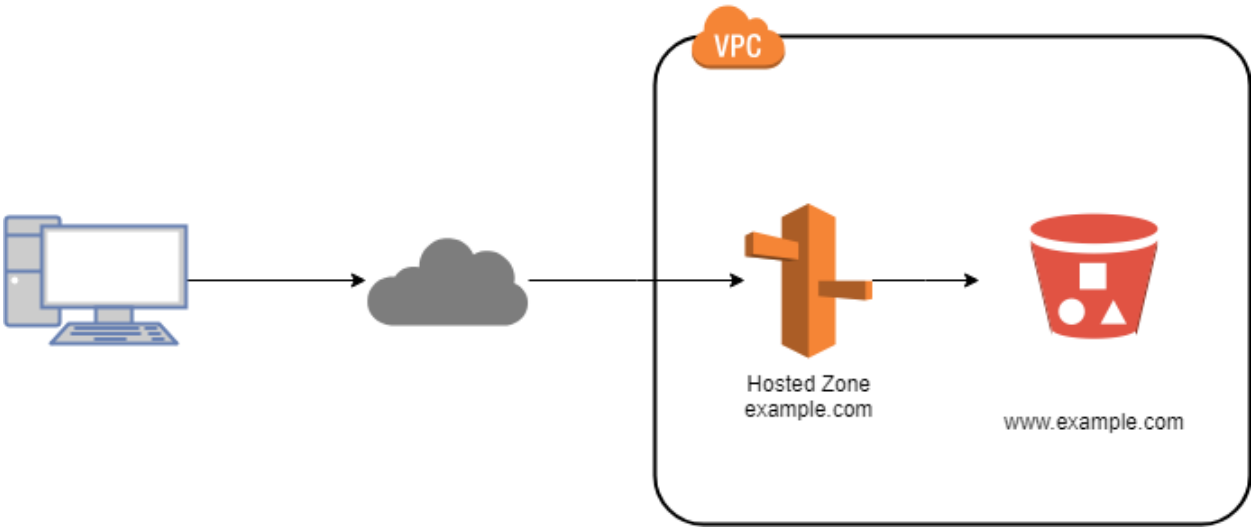
  (Correct)

Explanation
Here are the prerequisites for routing traffic to a website that is hosted in an Amazon S3 Bucket:

- An S3 bucket that is configured to host a static website. The bucket must have the same name as your domain or subdomain. For example, if you want to use the subdomain portal.tutorialsdojo.com, the name of the bucket must be portal.tutorialsdojo.com.

- A registered domain name. You can use Route 53 as your domain registrar, or you can use a different registrar.

- Route 53 as the DNS service for the domain. If you register your domain name by using Route 53, we automatically configure Route 53 as the DNS service for the domain.



The option that says: **The record set must be of type "MX"** is incorrect since an MX record specifies the mail server responsible for accepting email messages on behalf of a domain name. This is not what is being asked by the question.

The option that says: **The S3 bucket must be in the same region as the hosted zone** is incorrect. There is no constraint that the S3 bucket must be in the same region as the hosted zone in order for the Route 53 service to route traffic into it.

The option that says: **The Cross-Origin Resource Sharing (CORS) option should be enabled in the S3 bucket** is incorrect because you only need to enable Cross-Origin Resource Sharing (CORS) when your client web application on one domain interacts with the resources in a different domain.

Reference:

https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/RoutingToS3Bucket.html

Amazon Route 53 Overview:

https://youtu.be/Su308t19ubY

Check out this Amazon Route 53 Cheat Sheet:

https://tutorialsdojo.com/amazon-route-53/

Question 44: **Correct**
A business has a network of surveillance cameras installed within the premises of its data center. Management wants to leverage Artificial Intelligence to monitor and detect unauthorized personnel entering restricted areas. Should an unauthorized person be detected, the security team must be alerted via SMS.

Which solution satisfies the requirement?

- **Use Amazon Kinesis Video to stream live feeds from the cameras. Use Amazon Rekognition to detect authorized personnel. Set the phone numbers of the security as subscribers to an SNS topic.**

  (Correct)

-

Set up Amazon Managed Service for Prometheus to stream live feeds from the cameras. Use Amazon Fraud Detector to detect unauthorized personnel. Set the phone numbers of the security as subscribers to an SNS topic.

-

Configure Amazon Elastic Transcoder to stream live feeds from the cameras. Use Amazon Kendra to detect authorized personnel. Set the phone numbers of the security as subscribers to an SNS topic.

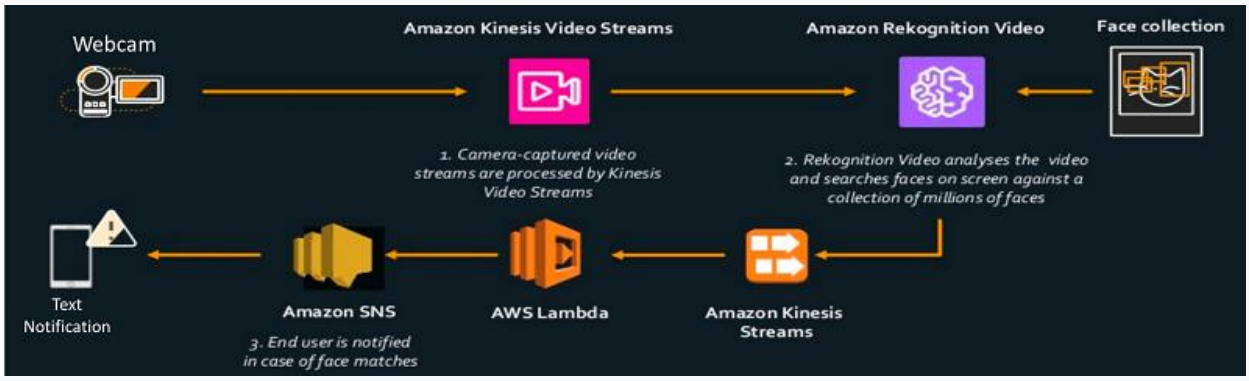
-

Replace the existing cameras with AWS IoT. Upload a face detection model to the AWS IoT devices and send them over to AWS Control Tower for checking and notification

Explanation

Amazon Kinesis Video Streams makes it easy to securely stream video from connected devices to AWS for analytics, machine learning (ML), playback, and other processing. Kinesis Video Streams automatically provisions and elastically scales all the infrastructure needed to ingest streaming video data from millions of devices.

Amazon Rekognition Video can detect objects, scenes, faces, celebrities, text, and inappropriate content in videos. You can also search for faces appearing in a video using your own repository or collection of face images.



The image above illustrates how we can combine these two services to create an intruder alert system using face recognition.

Hence, the correct answer is: **Use Amazon Kinesis Video to stream live feeds from the cameras. Use Amazon Rekognition to detect unauthorized personnel. Set the phone numbers of the security as subscribers to an SNS topic.**

The option that says: **Configure Amazon Elastic Transcoder to stream live feeds from the cameras. Use Amazon Kendra to detect unauthorized personnel. Set the phone**

numbers of the security as subscribers to an SNS topic is incorrect. Amazon Elastic Transcoder just allows you to convert media files from one format to another. Also, Amazon Kendra can't be used for face detection as it's just an intelligent search service.

The option that says: **Replace the existing cameras with AWS IoT. Upload a face detection model to the AWS IoT devices and send them over to AWS Control Tower for checking and notification** is incorrect. AWS IoT simply provides the cloud services that connect your IoT devices to other devices and AWS cloud services. This is basically a device software that can help you integrate your IoT devices into AWS IoT-based solutions and is not used as a physical camera. AWS Control Tower is primarily used to set up and govern a secure multi-account AWS environment and not for receiving video feeds.

The option that says: **Set up Amazon Managed Service for Prometheus to stream live feeds from the cameras. Use Amazon Fraud Detector to detect unauthorized personnel. Set the phone numbers of the security as subscribers to an SNS topic** is incorrect. The Amazon Managed Service for Prometheus is a Prometheus-compatible monitoring and alerting service, which is not used to stream live video feeds. This service makes it easy for you to monitor containerized applications and infrastructure at scale but not stream live feeds. Amazon Fraud Detector is a fully managed service that identifies potentially fraudulent online activities such as online payment fraud and fake account creation. Take note that the Amazon Fraud Detector service is not capable of detecting unauthorized personnel through live streaming feeds alone.

References:

https://docs.aws.amazon.com/kinesisvideostreams/latest/dg/what-is-kinesis-video.html

https://aws.amazon.com/blogs/aws/launch-welcoming-amazon-rekognition-video-service/

Check out these Amazon Kinesis Video Streams and Amazon Rekognition Cheat Sheets:

https://tutorialsdojo.com/amazon-kinesis/

https://tutorialsdojo.com/amazon-rekognition/

Question 45: **Correct**

A company has a requirement to move 80 TB data warehouse to the cloud. It would take 2 months to transfer the data given their current bandwidth allocation.

Which is the most cost-effective service that would allow you to quickly upload their data into AWS?

- **Amazon S3 Multipart Upload**
- **AWS Snowmobile**
- **AWS Snowball Edge**

  (Correct)
- **AWS Direct Connect**

Explanation

AWS Snowball Edge is a type of Snowball device with on-board storage and compute power for select AWS capabilities. Snowball Edge can undertake local processing and edge-computing workloads in addition to transferring data between your local environment and the AWS Cloud.

Each Snowball Edge device can transport data at speeds faster than the internet. This transport is done by shipping the data in the appliances through a regional carrier. The appliances are rugged shipping containers, complete with E Ink shipping labels. The AWS Snowball Edge device differs from the standard Snowball because it can bring the power of the AWS Cloud to your on-premises location, with local storage and compute functionality.

Snowball Edge devices have three options for device configurations – storage optimized, compute optimized, and with GPU.

Hence, the correct answer is: **AWS Snowball Edge.**

AWS Snowmobile is incorrect because this is an Exabyte-scale data transfer service used to move extremely large amounts of data to AWS. It is not suitable for transferring a small amount of data, like 80 TB in this scenario. You can transfer up to 100PB per

Snowmobile, a 45-foot long ruggedized shipping container, pulled by a semi-trailer truck. A more cost-effective solution here is to order a Snowball Edge device instead.

AWS Direct Connect is incorrect because it is primarily used to establish a dedicated network connection from your premises network to AWS. This is not suitable for one-time data transfer tasks, like what is depicted in the scenario.

Amazon S3 Multipart Upload is incorrect because this feature simply enables you to upload large objects in multiple parts. It still uses the same Internet connection of the company, which means that the transfer will still take time due to its current bandwidth allocation.

References:

https://docs.aws.amazon.com/snowball/latest/ug/whatissnowball.html

https://docs.aws.amazon.com/snowball/latest/ug/device-differences.html

Check out this AWS Snowball Edge Cheat Sheet:

https://tutorialsdojo.com/aws-snowball-edge/

AWS Snow Family Overview:

https://youtu.be/9Ar-51Ip53Q

Question 46: **Correct**
A company is receiving semi-structured and structured data from different sources every day. The Solutions Architect plans to use big data processing frameworks to analyze vast amounts of data and access it using various business intelligence tools and standard SQL queries.

Which of the following provides the MOST high-performing solution that fulfills this requirement?

- **Use Amazon Kinesis Data Analytics and store the processed data in Amazon DynamoDB.**

- **Create an Amazon EMR cluster and store the processed data in Amazon Redshift.**

  (Correct)

- **Use AWS Glue and store the processed data in Amazon S3.**

- **Create an Amazon EC2 instance and store the processed data in Amazon EBS.**

Explanation

Amazon EMR is a managed cluster platform that simplifies running big data frameworks, such as Apache Hadoop and Apache Spark, on AWS to process and analyze vast amounts of data. By using these frameworks and related open-source projects, such as Apache Hive and Apache Pig, you can process data for analytics purposes and business intelligence workloads. Additionally, you can use Amazon EMR to transform and move large amounts of data into and out of other AWS data stores and databases.

Amazon Redshift is the most widely used cloud data warehouse. It makes it fast, simple and cost-effective to analyze all your data using standard SQL and your existing Business Intelligence (BI) tools. It allows you to run complex analytic queries against terabytes to petabytes of structured and semi-structured data, using sophisticated query optimization, columnar storage on high-performance storage, and massively parallel query execution.

The key phrases in the scenario are "big data processing frameworks" and "various business intelligence tools and standard SQL queries" to analyze the data. To leverage

big data processing frameworks, you need to use Amazon EMR. The cluster will perform data transformations (ETL) and load the processed data into Amazon Redshift for analytic and business intelligence applications.

Hence, the correct answer is: **Create an Amazon EMR cluster and store the processed data in Amazon Redshift.**

The option that says: **Use AWS Glue and store the processed data in Amazon S3** is incorrect because AWS Glue is just a serverless ETL service that crawls your data, builds a data catalog, performs data preparation, data transformation, and data ingestion. It won't allow you to utilize different big data frameworks effectively, unlike Amazon EMR. In addition, the S3 Select feature in Amazon S3 can only run simple SQL queries against a subset of data from a specific S3 object. To perform queries in the S3 bucket, you need to use Amazon Athena.

The option that says: **Use Amazon Kinesis Data Analytics and store the processed data in Amazon DynamoDB** is incorrect because Amazon DynamoDB doesn't fully support the use of standard SQL and Business Intelligence (BI) tools, unlike Amazon Redshift. It also doesn't allow you to run complex analytic queries against terabytes to petabytes of structured and semi-structured data.

The option that says: **Create an Amazon EC2 instance and store the processed data in Amazon EBS** is incorrect because a single EBS-backed EC2 instance is quite limited in its computing capability. Moreover, it also entails an administrative overhead since you have to manually install and maintain the big data frameworks for the EC2 instance yourself. The most suitable solution to leverage big data frameworks is to use EMR clusters.

References:

https://docs.aws.amazon.com/emr/latest/ManagementGuide/emr-what-is-emr.html

https://docs.aws.amazon.com/redshift/latest/dg/loading-data-from-emr.html

Check out this Amazon EMR Cheat Sheet:

https://tutorialsdojo.com/amazon-emr/

Question 47: **Incorrect**
A commercial bank has a forex trading application. They created an Auto Scaling group of EC2 instances that allow the bank to cope with the current traffic and achieve cost-

efficiency. They want the Auto Scaling group to behave in such a way that it will follow a predefined set of parameters before it scales down the number of EC2 instances, which protects the system from unintended slowdown or unavailability.

Which of the following statements are true regarding the cooldown period? (Select TWO.)

- ☑ **It ensures that the Auto Scaling group does not launch or terminate additional EC2 instances before the previous scaling activity takes effect.**

  (Correct)

- ☐ **Its default value is 300 seconds.**

  (Correct)

- ☐ **It ensures that the Auto Scaling group launches or terminates additional EC2 instances without any downtime.**

- ☑ **It ensures that before the Auto Scaling group scales out, the EC2 instances have ample time to cooldown.**

  (Incorrect)

- ☐ **Its default value is 600 seconds.**

Explanation
In Auto Scaling, the following statements are correct regarding the cooldown period:

It ensures that the Auto Scaling group does not launch or terminate additional EC2 instances before the previous scaling activity takes effect.

Its default value is 300 seconds.

It is a configurable setting for your Auto Scaling group.

The following options are incorrect:

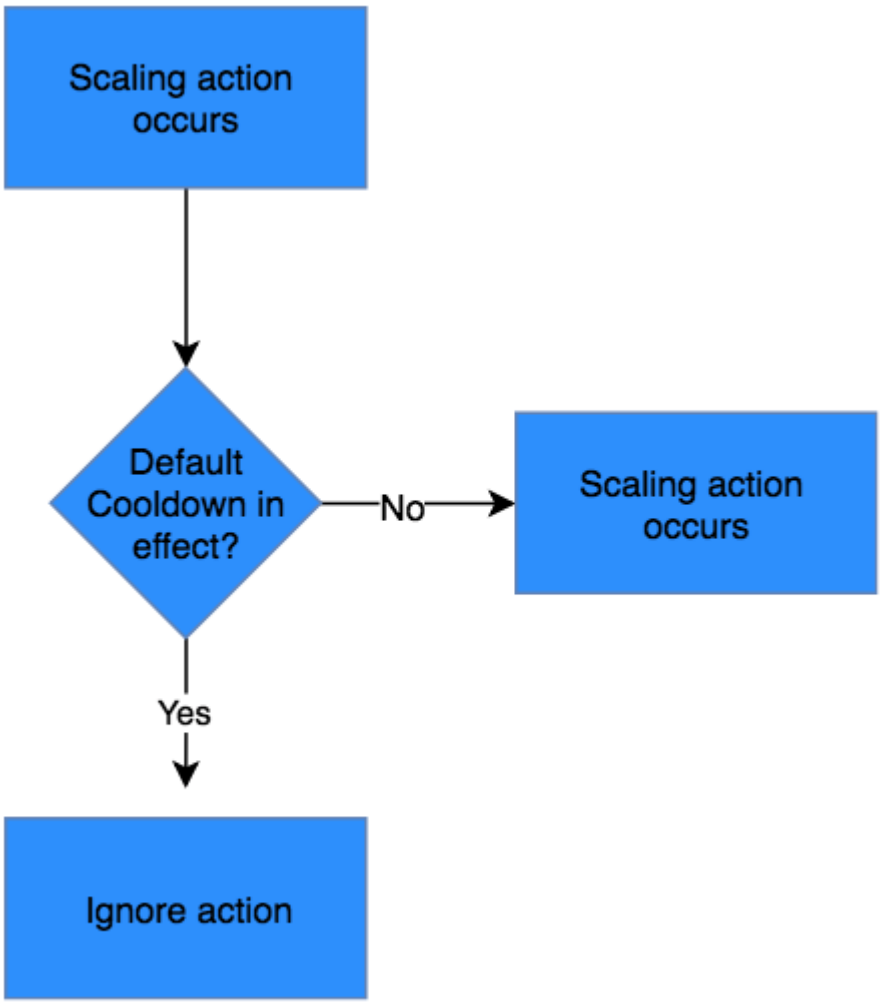
- It ensures that before the Auto Scaling group scales out, the EC2 instances have ample time to cooldown.

- It ensures that the Auto Scaling group launches or terminates additional EC2 instances without any downtime.

- Its default value is 600 seconds.

These statements are inaccurate and don't depict what the word "cooldown" actually means for Auto Scaling. The cooldown period is a configurable setting for your Auto Scaling group that helps to ensure that it doesn't launch or terminate additional instances before the previous scaling activity takes effect. After the Auto Scaling group dynamically scales using a simple scaling policy, it waits for the cooldown period to complete before resuming scaling activities.

The figure below demonstrates the scaling cooldown:



Reference:

http://docs.aws.amazon.com/autoscaling/latest/userguide/as-instance-termination.html

Check out this AWS Auto Scaling Cheat Sheet:

https://tutorialsdojo.com/aws-auto-scaling/

Tutorials Dojo's AWS Certified Solutions Architect Associate Exam Study Guide:

https://tutorialsdojo.com/aws-certified-solutions-architect-associate/

Question 48: **Correct**
A Solutions Architect created a new Standard-class S3 bucket to store financial reports that are not frequently accessed but should immediately be available when an auditor requests them. To save costs, the Architect changed the storage class of the S3 bucket from Standard to Infrequent Access storage class.

In Amazon S3 Standard - Infrequent Access storage class, which of the following statements are true? (Select TWO.)

- ☐ **It automatically moves data to the most cost-effective access tier without any operational overhead.**
- ☑ **It is designed for data that requires rapid access when needed.**

  (Correct)
- ☐ **Ideal to use for data archiving.**
- ☐ **It provides high latency and low throughput performance**
- ☑ **It is designed for data that is accessed less frequently.**

  (Correct)

Explanation
Amazon S3 Standard - Infrequent Access (Standard - IA) is an Amazon S3 storage class for data that is accessed less frequently, but requires rapid access when needed. Standard - IA offers the high durability, throughput, and low latency of Amazon S3 Standard, with a low per GB storage price and per GB retrieval fee.

	S3 Standard	S3 Standard-Infrequent Access (IA)	S3 One Zone-Infrequent Access (IA)	S3 Intelligent Tiering
Features	General-purpose storage of frequently accessed data	For long-lived, rapid but less frequently accessed data; data is stored redundantly in multiple AZs	For long-lived, rapid but less frequently accessed data; data is stored redundantly in only one AZ of your choice	For long-lived data that have unpredictable access patterns
Durability	99.999999999% (11 9's)	99.999999999% (11 9's)	99.999999999% (11 9's)	99.999999999% (11 9's)
Availability	99.99%	99.9%	99.5%	99.9%
Availability SLA	99.9%	99%	99%	99%
Number of Availability Zones	At least 3	At least 3	Only 1	At least 3
Minimum capacity charge per object	N/A	128KB	128KB	N/A
Minimum storage duration charge	N/A	30 days	30 days	30 days
Inserting data	Directly PUT into S3 Standard	Directly PUT into S3 Standard-IA or set Lifecycle policies to transition objects from the S3 Standard to the S3 Standard-IA storage class.	Directly PUT into S3 One Zone-IA or set Lifecycle policies to transition objects from the S3 Standard to the S3 One Zone-IA storage class.	Directly PUT into S3 Intelligent-Tiering or set Lifecycle policies to transition objects from the S3 Standard to the S3 Intelligent-Tiering storage class.
Retrieval fee	N/A	per GB retrieved	per GB retrieved	N/A
First byte latency	milliseconds	milliseconds	milliseconds	milliseconds
Storage transition	S3 Standard to all other S3 storage types including Glacier	S3 Standard-IA to S3 One Zone-IA or S3 Glacier	S3 One Zone-IA to S3 Glacier	S3 Intelligent to S3 One Zone-IA or S3 Glacier
Use Cases	Cloud applications, dynamic websites, content distribution, mobile and gaming applications, and big data analytics.	Ideally suited for long-term file storage, older sync and share storage, and other aging data.	For infrequently-accessed storage, like backup copies, disaster recovery copies, or other easily recreatable data.	Data with unknown or changing access patterns, optimize storage costs automatically, and unpredictable workloads

This combination of low cost and high performance make Standard - IA ideal for long-term storage, backups, and as a data store for disaster recovery. The Standard - IA storage class is set at the object level and can exist in the same bucket as Standard, allowing you to use lifecycle policies to automatically transition objects between storage classes without any application changes.

Key Features:

- Same low latency and high throughput performance of Standard

- Designed for durability of 99.999999999% of objects

- Designed for 99.9% availability over a given year

- Backed with the Amazon S3 Service Level Agreement for availability

- Supports SSL encryption of data in transit and at rest

- Lifecycle management for automatic migration of objects

Hence, the correct answers are:

- **It is designed for data that is accessed less frequently.**

- **It is designed for data that requires rapid access when needed.**

The option that says: **It automatically moves data to the most cost-effective access tier without any operational overhead** is incorrect as it actually refers to Amazon S3 - Intelligent Tiering, which is the only cloud storage class that delivers automatic cost savings by moving objects between different access tiers when access patterns change.

The option that says: **It provides high latency and low throughput performance** is incorrect as it should be "low latency" and "high throughput" instead. S3 automatically scales performance to meet user demands.

The option that says: **Ideal to use for data archiving** is incorrect because this statement refers to Amazon S3 Glacier. Glacier is a secure, durable, and extremely low-cost cloud storage service for data archiving and long-term backup.

References:

https://aws.amazon.com/s3/storage-classes/

https://aws.amazon.com/s3/faqs

Check out this Amazon S3 Cheat Sheet:

https://tutorialsdojo.com/amazon-s3/

Question 49: **Correct**
A Solutions Architect is building a cloud infrastructure where EC2 instances require access to various AWS services such as S3 and Redshift. The Architect will also need to provide access to system administrators so they can deploy and test their changes.

Which configuration should be used to ensure that the access to the resources is secured and not compromised? (Select TWO.)

- ☑ **Assign an IAM role to the Amazon EC2 instance.**

  (Correct)

- ☐ **Store the AWS Access Keys in the EC2 instance.**

- ☐ **Assign an IAM user for each Amazon EC2 Instance.**
- ☐ **Store the AWS Access Keys in ACM.**
- ☑ **Enable Multi-Factor Authentication.**

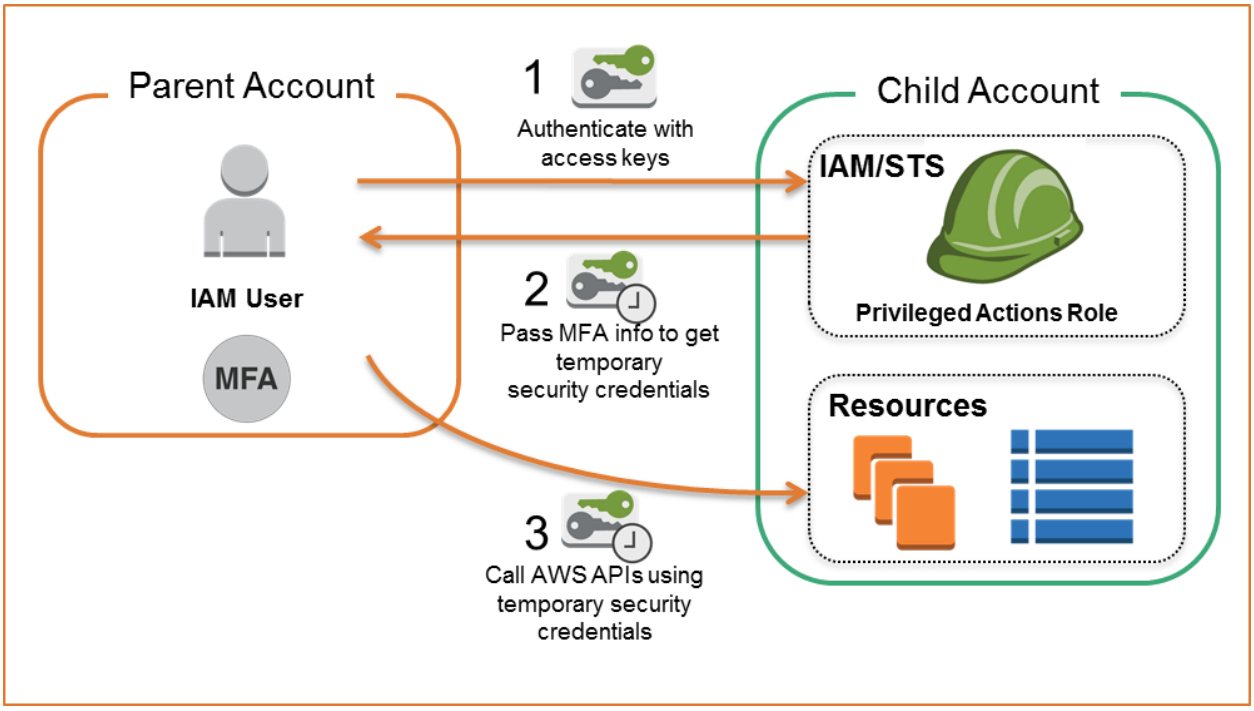
  (Correct)

Explanation
In this scenario, the correct answers are:

- Enable Multi-Factor Authentication

- Assign an IAM role to the Amazon EC2 instance

Always remember that you should associate IAM roles to EC2 instances and not an IAM user, for the purpose of accessing other AWS services. IAM roles are designed so that your applications can securely make API requests from your instances, without requiring you to manage the security credentials that the applications use. Instead of creating and distributing your AWS credentials, you can delegate permission to make API requests using IAM roles.



AWS Multi-Factor Authentication (MFA) is a simple best practice that adds an extra layer of protection on top of your user name and password. With MFA enabled, when a user signs in to an AWS website, they will be prompted for their user name and password (the first factor—what they know), as well as for an authentication code from their AWS MFA device (the second factor—what they have). Taken together, these multiple factors provide increased security for your AWS account settings and resources. You can enable MFA for your AWS account and for individual IAM users you have created under your account. MFA can also be used to control access to AWS service APIs.

Storing the AWS Access Keys in the EC2 instance is incorrect. This is not recommended by AWS as it can be compromised. Instead of storing access keys on an EC2 instance for use by applications that run on the instance and make AWS API requests, you can use an IAM role to provide temporary access keys for these applications.

Assigning an IAM user for each Amazon EC2 Instance is incorrect because there is no need to create an IAM user for this scenario since IAM roles already provide greater flexibility and easier management.

Storing the AWS Access Keys in ACM is incorrect because ACM is just a service that lets you easily provision, manage, and deploy public and private SSL/TLS certificates for use with AWS services and your internal connected resources. It is not used as a secure storage for your access keys.

References:

https://aws.amazon.com/iam/details/mfa/

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/iam-roles-for-amazon-ec2.html

Check out this AWS IAM Cheat Sheet:

https://tutorialsdojo.com/aws-identity-and-access-management-iam/

Question 50: **Correct**
A media company is setting up an ECS batch architecture for its image processing application. It will be hosted in an Amazon ECS Cluster with two ECS tasks that will handle image uploads from the users and image processing. The first ECS task will process the user requests, store the image in an S3 input bucket, and push a message to a queue. The second task reads from the queue, parses the message containing the object name, and then downloads the object. Once the image is processed and transformed, it will upload the objects to the S3 output bucket. To complete the architecture, the Solutions Architect must create a queue and the necessary IAM permissions for the ECS tasks.

Which of the following should the Architect do next?

- **Launch a new Amazon MQ queue and configure the second ECS task to read from it. Create an IAM role that the ECS tasks can assume in order to get access to the S3 buckets and Amazon MQ queue. Set the (`EnableTaskIAMRole`) option to true in the task definition.**

- **Launch a new Amazon Kinesis Data Firehose and configure the second ECS task to read from it. Create an IAM role that the ECS tasks can assume in order to get access to the S3 buckets and Kinesis Data Firehose. Specify the ARN of the IAM Role in the (`taskDefinitionArn`) field of the task definition.**

-

Launch a new Amazon SQS queue and configure the second ECS task to read from it. Create an IAM role that the ECS tasks can assume in order to get access to the S3 buckets and SQS queue. Declare the IAM Role (`taskRoleArn`) in the task definition.

(Correct)

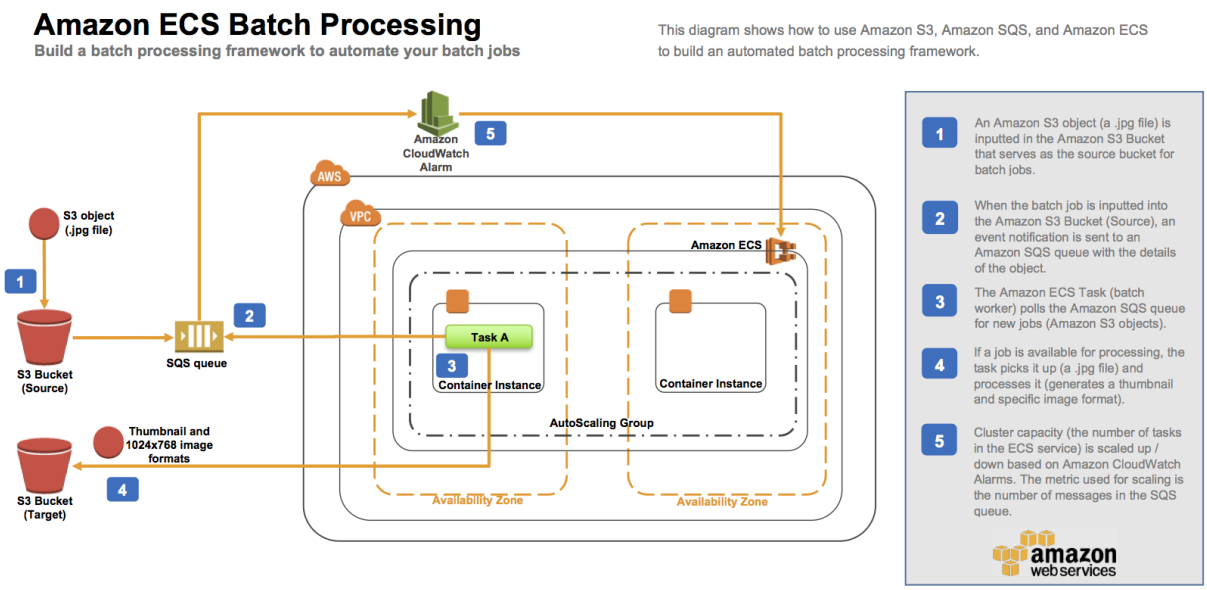
- 

Launch a new Amazon AppStream 2.0 queue and configure the second ECS task to read from it. Create an IAM role that the ECS tasks can assume in order to get access to the S3 buckets and AppStream 2.0 queue. Declare the IAM Role (`taskRoleArn`) in the task definition.

Explanation

Docker containers are particularly suited for batch job workloads. Batch jobs are often short-lived and embarrassingly parallel. You can package your batch processing application into a Docker image so that you can deploy it anywhere, such as in an Amazon ECS task.

Amazon ECS supports batch jobs. You can use Amazon ECS *Run Task* action to run one or more tasks once. The Run Task action starts the ECS task on an instance that meets the task's requirements including CPU, memory, and ports.



AWS Reference Architectures

For example, you can set up an ECS Batch architecture for an image processing application. You can set up an AWS CloudFormation template that creates an Amazon S3 bucket, an Amazon SQS queue, an Amazon CloudWatch alarm, an ECS cluster, and an ECS task definition. Objects uploaded to the input S3 bucket trigger an event that sends object details to the SQS queue. The ECS task deploys a Docker container that reads from that queue, parses the message containing the object name and then downloads the object. Once transformed it will upload the objects to the S3 output bucket.

By using the SQS queue as the location for all object details, you can take advantage of its scalability and reliability as the queue will automatically scale based on the incoming messages and message retention can be configured. The ECS Cluster will then be able to scale services up or down based on the number of messages in the queue.

You have to create an IAM Role that the ECS task assumes in order to get access to the S3 buckets and SQS queue. Note that the permissions of the IAM role don't specify the S3 bucket ARN for the incoming bucket. This is to avoid a circular dependency issue in

the CloudFormation template. You should always make sure to assign the least amount of privileges needed to an IAM role.

Hence, the correct answer is: **Launch a new Amazon SQS queue and configure the second ECS task to read from it. Create an IAM role that the ECS tasks can assume in order to get access to the S3 buckets and SQS queue. Declare the IAM Role (`taskRoleArn`) in the task definition.**

The option that says: **Launch a new Amazon AppStream 2.0 queue and configure the second ECS task to read from it. Create an IAM role that the ECS tasks can assume in order to get access to the S3 buckets and AppStream 2.0 queue. Declare the IAM Role (`taskRoleArn`) in the task definition** is incorrect because Amazon AppStream 2.0 is a fully managed application streaming service and can't be used as a queue. You have to use Amazon SQS instead.

The option that says: **Launch a new Amazon Kinesis Data Firehose and configure the second ECS task to read from it. Create an IAM role that the ECS tasks can assume in order to get access to the S3 buckets and Kinesis Data Firehose. Specify the ARN of the IAM Role in the (`taskDefinitionArn`) field of the task definition** is incorrect because Amazon Kinesis Data Firehose is a fully managed service for delivering real-time streaming data. Although it can stream data to an S3 bucket, it is not suitable to be used as a queue for a batch application in this scenario. In addition, the ARN of the IAM Role should be declared in the `taskRoleArn` and not in the `taskDefinitionArn` field.

The option that says: **Launch a new Amazon MQ queue and configure the second ECS task to read from it. Create an IAM role that the ECS tasks can assume in order to get access to the S3 buckets and Amazon MQ queue. Set the (`EnableTaskIAMRole`) option to true in the task definition** is incorrect because Amazon MQ is primarily used as a managed message broker service and not a queue. The `EnableTaskIAMRole` option is only applicable for Windows-based ECS Tasks that require extra configuration.

References:

https://github.com/aws-samples/ecs-refarch-batch-processing

https://docs.aws.amazon.com/AmazonECS/latest/developerguide/common_use_cases.html

https://aws.amazon.com/ecs/faqs/

Question 51: **Correct**

A company has a serverless application made up of AWS Amplify, Amazon API Gateway

and a Lambda function. The application is connected to an Amazon RDS MySQL database instance inside a private subnet. A Lambda Function URL is also implemented as the dedicated HTTPS endpoint for the function, which has the following value:

```
https://12june1898pil1pinas.lambda-url.us-west-2.on.aws/
```

There are times during peak loads when the database throws a “too many connections” error preventing the users from accessing the application.

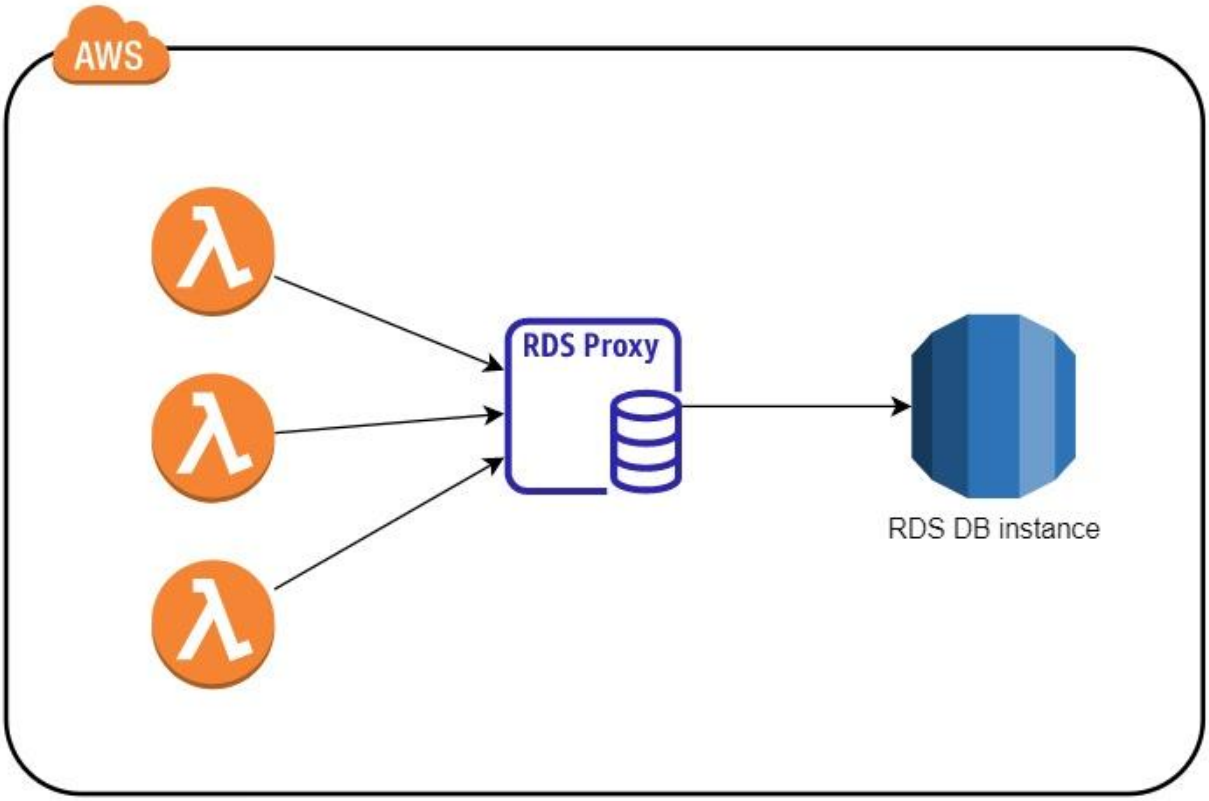
Which solution could the company take to resolve the issue?

- **Increase the memory allocation of the Lambda function**
- **Increase the concurrency limit of the Lambda function**
- **Provision an RDS Proxy between the Lambda function and RDS database instance**

  (Correct)
- **Increase the rate limit of API Gateway**

Explanation

If a "Too Many Connections" error happens to a client connecting to a MySQL database, it means all available connections are in use by other clients. Opening a connection consumes resources on the database server. Since Lambda functions can scale to tens of thousands of concurrent connections, your database needs more resources to open and maintain connections instead of executing queries. The maximum number of connections a database can support is largely determined by the amount of memory allocated to it. Upgrading to a database instance with higher memory is a straightforward way of solving the problem. Another approach would be to maintain a connection pool that clients can reuse. This is where RDS Proxy comes in.



RDS Proxy helps you manage a large number of connections from Lambda to an RDS database by establishing a warm connection pool to the database. Your Lambda functions interact with RDS Proxy instead of your database instance. It handles the connection pooling necessary for scaling many simultaneous connections created by concurrent Lambda functions. This allows your Lambda applications to reuse existing connections, rather than creating new connections for every function invocation.

Thus, the correct answer is: **Provision an RDS Proxy between the Lambda function and RDS database instance format**

The option that says: **Increase the concurrency limit of the Lambda function** is incorrect. The concurrency limit refers to the maximum requests AWS Lambda can handle at the same time. Increasing the limit will allow for more requests to open a database connection, which could potentially worsen the problem.

The option that says: **Increase the rate limit of API Gateway** is incorrect. This won't fix the issue at all as all it does is increase the number of API requests a client can make.

The option that says: **Increase the memory allocation of the Lambda function** is incorrect. Increasing the Lambda function's memory would only make it run processes faster. It can help but it won't likely do any significant effect to get rid of the error. The "too many connections" error is a database-related issue. Solutions that have to do with

databases, like upgrading to a larger database instance or, in this case, creating a database connection pool using RDS Proxy have better chance of resolving the problem.

References:

https://aws.amazon.com/rds/proxy/

https://aws.amazon.com/blogs/compute/using-amazon-rds-proxy-with-aws-lambda/

Check out this Amazon RDS Cheat Sheet:

https://tutorialsdojo.com/amazon-relational-database-service-amazon-rds/

Question 52: **Incorrect**
A company runs a messaging application in the `ap-northeast-1` and `ap-southeast-2` region. A Solutions Architect needs to create a routing policy wherein a larger portion of traffic from the Philippines and North India will be routed to the resource in the `ap-northeast-1` region.

Which Route 53 routing policy should the Solutions Architect use?

- **Latency Routing**
- **Geoproximity Routing**

  (Correct)
- **Weighted Routing**
- **Geolocation Routing**

  (Incorrect)

Explanation

Amazon Route 53 is a highly available and scalable Domain Name System (DNS) web service. You can use Route 53 to perform three main functions in any combination: domain registration, DNS routing, and health checking. After you create a hosted zone for your domain, such as example.com, you create records to tell the Domain Name System (DNS) how you want traffic to be routed for that domain.

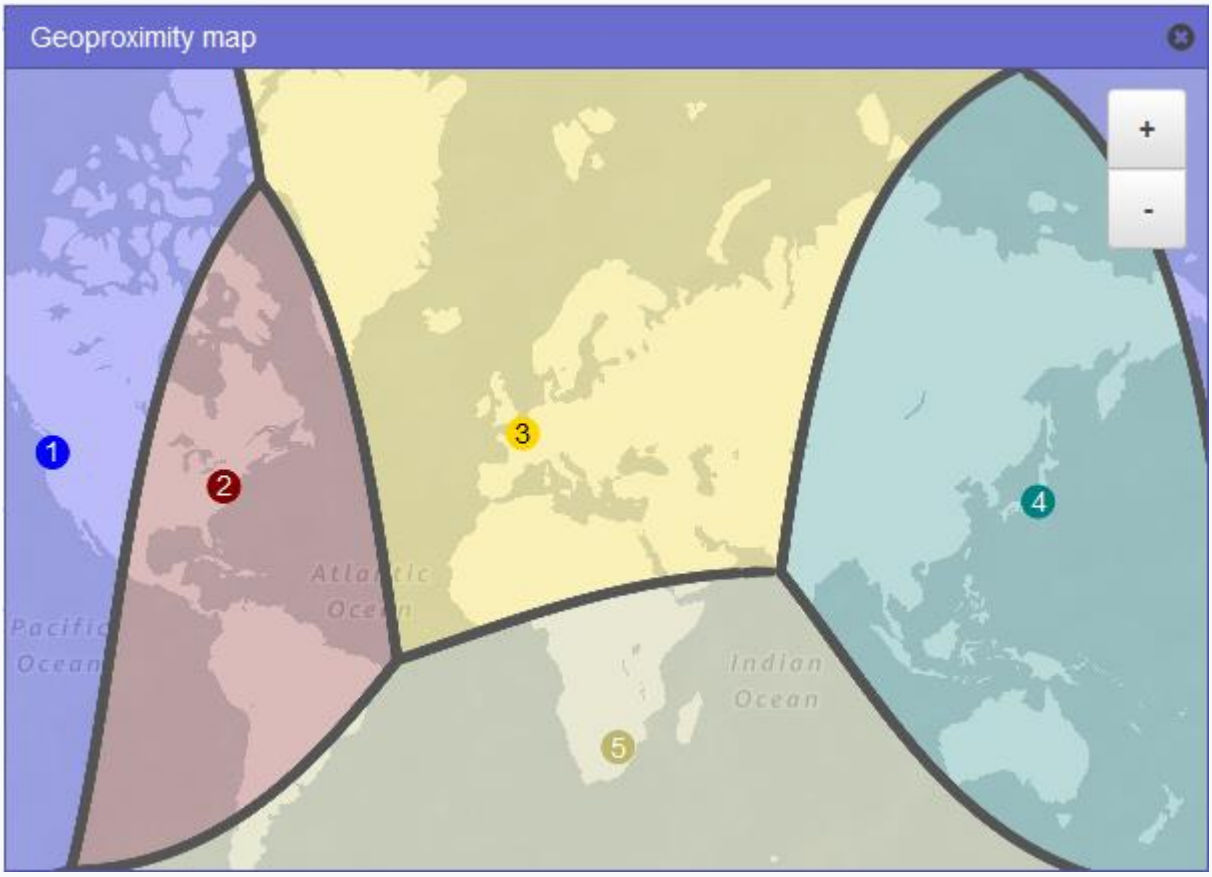
For example, you might create records that cause DNS to do the following:

- Route Internet traffic for example.com to the IP address of a host in your data center.

- Route email for that domain (jose.rizal@tutorialsdojo.com) to a mail server (mail.tutorialsdojo.com).

- Route traffic for a subdomain called operations.manila.tutorialsdojo.com to the IP address of a different host.

Each record includes the name of a domain or a subdomain, a record type (for example, a record with a type of MX routes email), and other information applicable to the record type (for MX records, the hostname of one or more mail servers and a priority for each server).



Route 53 has different routing policies that you can choose from. Below are some of the policies:

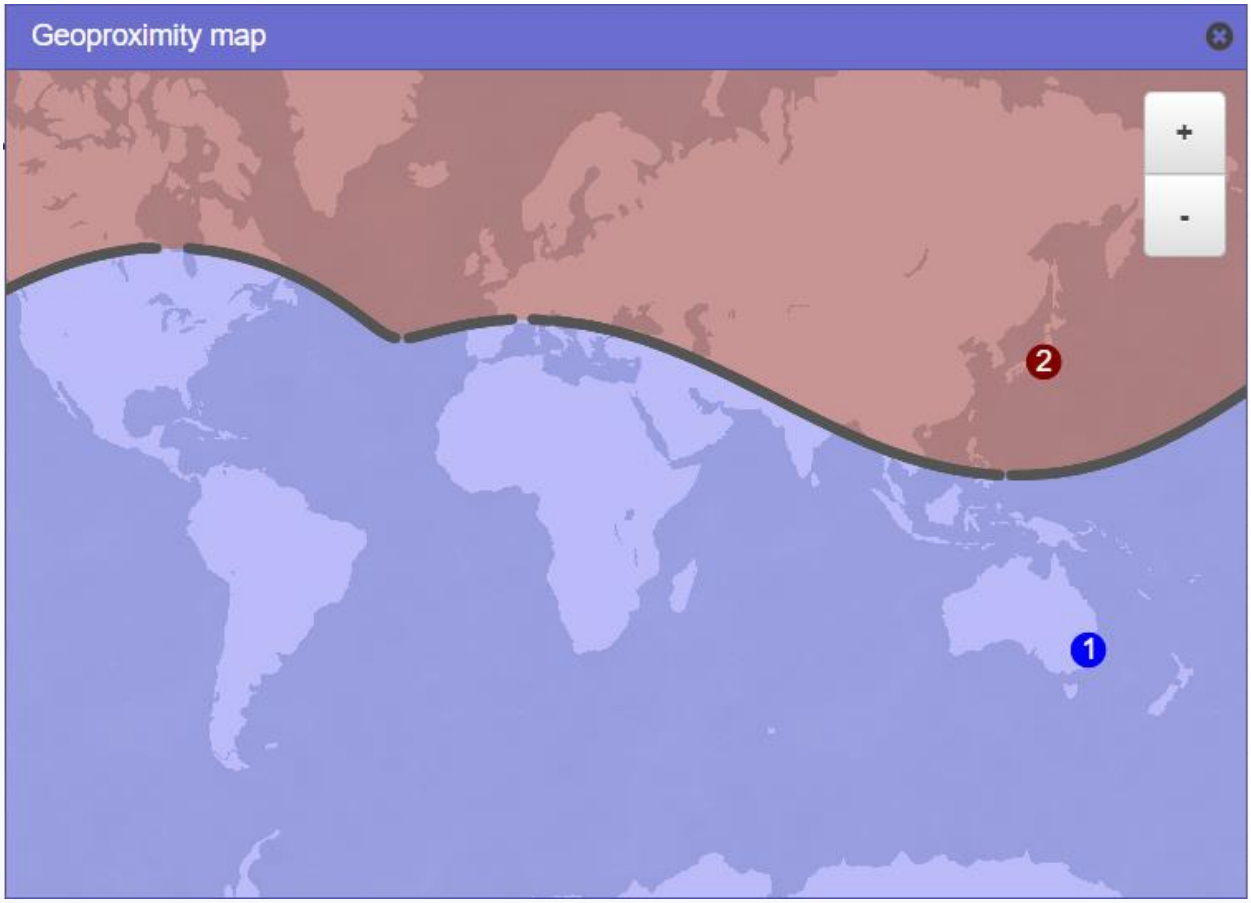
Latency Routing lets Amazon Route 53 serve user requests from the AWS Region that provides the lowest latency. It does not, however, guarantee that users in the same geographic region will be served from the same location.

Geoproximity Routing lets Amazon Route 53 route traffic to your resources based on the geographic location of your users and your resources. You can also optionally choose to route more traffic or less to a given resource by specifying a value, known as a *bias*. A **bias** expands or shrinks the size of the geographic region from which traffic is routed to a resource.

Geolocation Routing lets you choose the resources that serve your traffic based on the geographic location of your users, meaning the location that DNS queries originate from.

Weighted Routing lets you associate multiple resources with a single domain name (tutorialsdojo.com) or subdomain name (subdomain.tutorialsdojo.com) and choose how much traffic is routed to each resource.

In this scenario, the problem requires a routing policy that will let Route 53 route traffic to the resource in the Tokyo region from a larger portion of the Philippines and North India.



You need to use Geoproximity Routing and specify a bias to control the size of the geographic region from which traffic is routed to your resource. The sample image above uses a bias of -40 in the Tokyo region and a bias of 1 in the Sydney Region. Setting up the bias configuration in this manner would cause Route 53 to route traffic coming from the middle and northern part of the Philippines, as well as the northern part of India to the resource in the Tokyo Region.

Hence, the correct answer is **Geoproximity Routing.**

Geolocation Routing is incorrect because you cannot control the coverage size from which traffic is routed to your instance in Geolocation Routing. It just lets you choose the instances that will serve traffic based on the location of your users.

Latency Routing is incorrect because it is mainly used for improving performance by letting Route 53 serve user requests from the AWS Region that provides the lowest latency.

Weighted Routing is incorrect because it is used for routing traffic to multiple resources in proportions that you specify. This can be useful for load balancing and testing new versions of software.

References:

https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/routing-policy.html#routing-policy-geoproximity

https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/rrsets-working-with.html

Latency Routing vs. Geoproximity Routing vs. Geolocation Routing:

https://tutorialsdojo.com/latency-routing-vs-geoproximity-routing-vs-geolocation-routing/

Question 53: **Incorrect**
A company has developed public APIs hosted in Amazon EC2 instances behind an Elastic Load Balancer. The APIs will be used by various clients from their respective on-premises data centers. A Solutions Architect received a report that the web service clients can only access trusted IP addresses whitelisted on their firewalls.

What should you do to accomplish the above requirement?

- **Associate an Elastic IP address to a Network Load Balancer.**

  (Correct)

- **Associate an Elastic IP address to an Application Load Balancer.**

  (Incorrect)

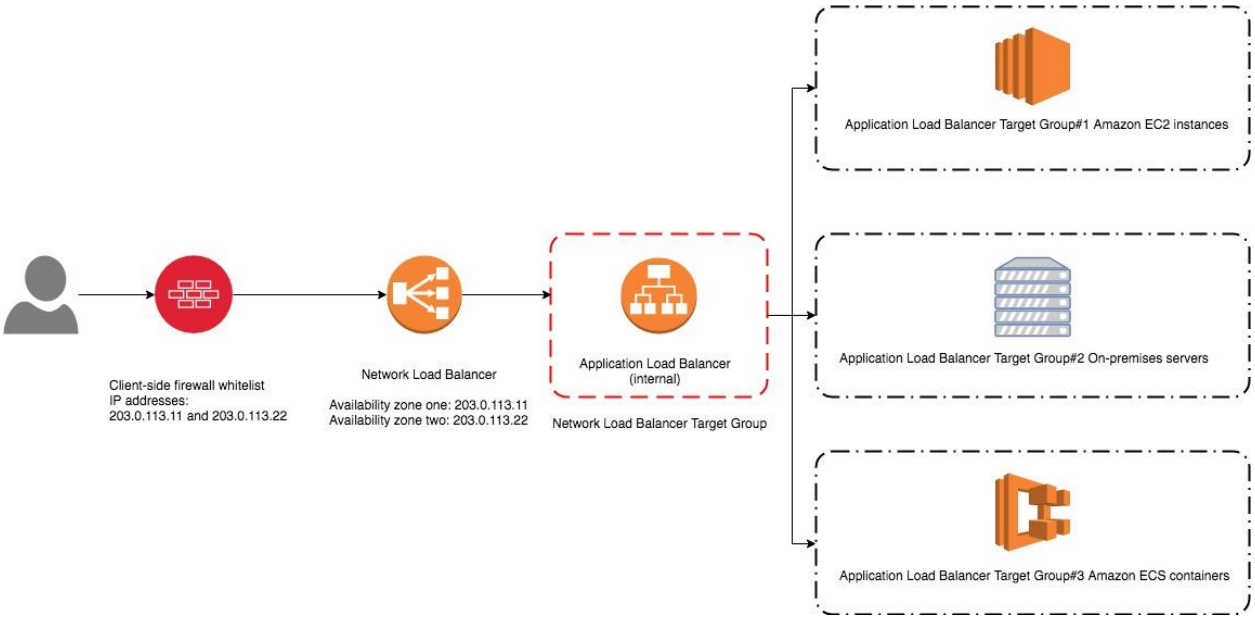
- **Create an Alias Record in Route 53 which maps to the DNS name of the load balancer.**

-

Create a CloudFront distribution whose origin points to the private IP addresses of your web servers.

Explanation

A **Network Load Balancer** functions at the fourth layer of the Open Systems Interconnection (OSI) model. It can handle millions of requests per second. After the load balancer receives a connection request, it selects a target from the default rule's target group. It attempts to open a TCP connection to the selected target on the port specified in the listener configuration.



Based on the given scenario, web service clients can only access trusted IP addresses. To resolve this requirement, you can use the Bring Your Own IP (BYOIP) feature to use the trusted IPs as Elastic IP addresses (EIP) to a Network Load Balancer (NLB). This way, there's no need to re-establish the whitelists with new IP addresses.

Hence, the correct answer is: **Associate an Elastic IP address to a Network Load Balancer.**

The option that says: **Associate an Elastic IP address to an Application Load Balancer** is incorrect because you can't assign an Elastic IP address to an Application Load Balancer. The alternative method you can do is assign an Elastic IP address to a Network Load Balancer in front of the Application Load Balancer.

The option that says: **Create a CloudFront distribution whose origin points to the private IP addresses of your web servers** is incorrect because web service clients can only access trusted IP addresses. The fastest way to resolve this requirement is to attach an Elastic IP address to a Network Load Balancer.

The option that says: **Create an Alias Record in Route 53 which maps to the DNS name of the load balancer** is incorrect. This approach won't still allow them to access the application because of trusted IP addresses on their firewalls.

References:

https://aws.amazon.com/premiumsupport/knowledge-center/elb-attach-elastic-ip-to-public-nlb/

https://aws.amazon.com/blogs/networking-and-content-delivery/using-static-ip-addresses-for-application-load-balancers/

https://docs.aws.amazon.com/elasticloadbalancing/latest/network/introduction.html

Check out this AWS Elastic Load Balancing Cheat Sheet:

https://tutorialsdojo.com/aws-elastic-load-balancing-elb/

Question 54: **Correct**
A company is running a multi-tier web application farm in a virtual private cloud (VPC) that is not connected to their corporate network. They are connecting to the VPC over the Internet to manage the fleet of Amazon EC2 instances running in both the public and private subnets. The Solutions Architect has added a bastion host with Microsoft Remote Desktop Protocol (RDP) access to the application instance security groups, but the company wants to further limit administrative access to all of the instances in the VPC.

Which of the following bastion host deployment options will meet this requirement?

- **Deploy a Windows Bastion host with an Elastic IP address in the public subnet and allow RDP access to bastion only from the corporate IP addresses.**

  (Correct)

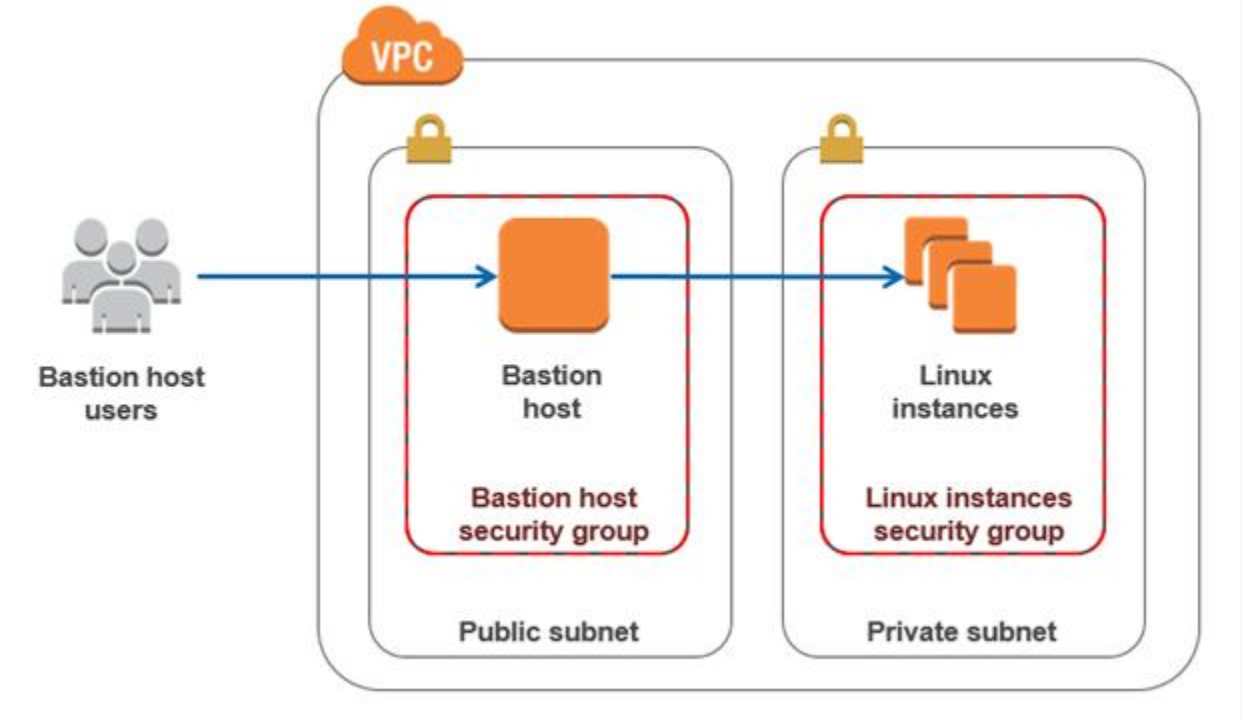
- **Deploy a Windows Bastion host with an Elastic IP address in the private subnet, and restrict RDP access to the bastion from only the corporate public IP addresses.**

- **Deploy a Windows Bastion host with an Elastic IP address in the public subnet and allow SSH access to the bastion from anywhere.**

- **Deploy a Windows Bastion host on the corporate network that has RDP access to all EC2 instances in the VPC.**

Explanation
The correct answer is to deploy a Windows Bastion host with an Elastic IP address in the public subnet and allow RDP access to bastion only from the corporate IP addresses.

A bastion host is a special purpose computer on a network specifically designed and configured to withstand attacks. If you have a bastion host in AWS, it is basically just an EC2 instance. It should be in a public subnet with either a public or Elastic IP address with sufficient RDP or SSH access defined in the security group. Users log on to the bastion host via SSH or RDP and then use that session to manage other hosts in the private subnets.



Deploying a Windows Bastion host on the corporate network that has RDP access to all EC2 instances in the VPC is incorrect since you do not deploy the Bastion host to your corporate network. It should be in the public subnet of a VPC.

Deploying a Windows Bastion host with an Elastic IP address in the private subnet and restricting RDP access to the bastion from only the corporate public IP addresses is incorrect since it should be deployed in a public subnet, not a private subnet.

Deploying a Windows Bastion host with an Elastic IP address in the public subnet and allowing SSH access to the bastion from anywhere is incorrect. Since it is a Windows bastion, you should allow RDP access and not SSH as this is mainly used for Linux-based systems.

Reference:

https://docs.aws.amazon.com/quickstart/latest/linux-bastion/architecture.html

Check out this Amazon VPC Cheat Sheet:

https://tutorialsdojo.com/amazon-vpc/

Question 55: **Correct**
A media company has two VPCs: VPC-1 and VPC-2 with peering connection between each other. VPC-1 only contains private subnets while VPC-2 only contains public subnets. The company uses a single AWS Direct Connect connection and a virtual interface to connect their on-premises network with VPC-1.

Which of the following options increase the fault tolerance of the connection to VPC-1? (Select TWO.)

- ☑ **Establish a hardware VPN over the Internet between VPC-1 and the on-premises network.**

  (Correct)

- ☐ **Establish a hardware VPN over the Internet between VPC-2 and the on-premises network.**

- ☑ **Establish another AWS Direct Connect connection and private virtual interface in the same AWS region as VPC-1.**

  (Correct)

- ☐ **Establish a new AWS Direct Connect connection and private virtual interface in the same region as VPC-2.**

- ☐ **Use the AWS VPN CloudHub to create a new AWS Direct Connect connection and private virtual interface in the same region as VPC-2.**

Explanation

In this scenario, you have two VPCs which have peering connections with each other. Note that a VPC peering connection does not support edge to edge routing. This means that if either VPC in a peering relationship has one of the following connections, you cannot extend the peering relationship to that connection:

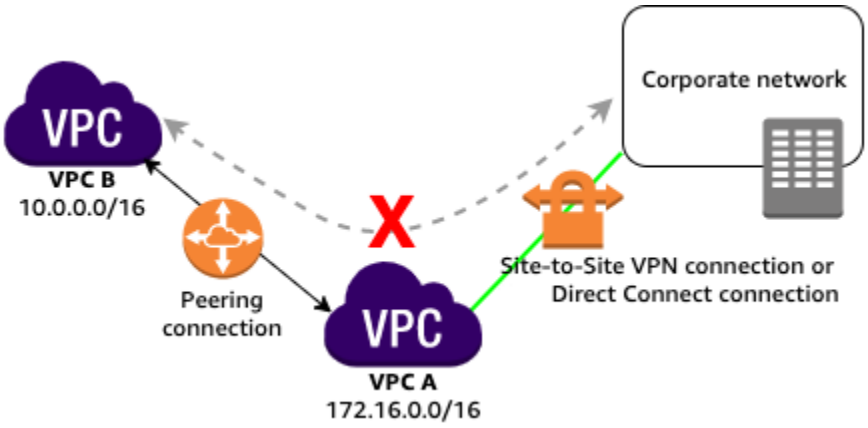
- A VPN connection or an AWS Direct Connect connection to a corporate network

- An Internet connection through an Internet gateway

- An Internet connection in a private subnet through a NAT device

- A gateway VPC endpoint to an AWS service; for example, an endpoint to Amazon S3.

- (IPv6) A ClassicLink connection. You can enable IPv4 communication between a linked EC2-Classic instance and instances in a VPC on the other side of a VPC peering connection. However, IPv6 is not supported in EC2-Classic, so you cannot extend this connection for IPv6 communication.



For example, if VPC A and VPC B are peered, and VPC A has any of these connections, then instances in VPC B cannot use the connection to access resources on the other side of the connection. Similarly, resources on the other side of a connection cannot use the connection to access VPC B.

Hence, this means that you cannot use VPC-2 to extend the peering relationship that exists between VPC-1 and the on-premises network. For example, traffic from the corporate network can't directly access VPC-1 by using the VPN connection or the AWS Direct Connect connection to VPC-2, which is why the following options are incorrect:

- Use the AWS VPN CloudHub to create a new AWS Direct Connect connection and private virtual interface in the same region as VPC-2.

- Establish a hardware VPN over the Internet between VPC-2 and the on-premises network.

- Establish a new AWS Direct Connect connection and private virtual interface in the same region as VPC-2.

You can do the following to provide a highly available, fault-tolerant network connection:

- Establish a hardware VPN over the Internet between the VPC and the on-premises network.

- Establish another AWS Direct Connect connection and private virtual interface in the same AWS region.

References:

https://docs.aws.amazon.com/vpc/latest/peering/invalid-peering-configurations.html#edge-to-edge-vgw

https://aws.amazon.com/premiumsupport/knowledge-center/configure-vpn-backup-dx/

https://aws.amazon.com/answers/networking/aws-multiple-data-center-ha-network-connectivity/

Check out this Amazon VPC Cheat Sheet:

https://tutorialsdojo.com/amazon-vpc/

Question 56: **Correct**
As part of the Business Continuity Plan of your company, your IT Director instructed you to set up an automated backup of all of the EBS Volumes for your EC2 instances as soon as possible.

What is the fastest and most cost-effective solution to automatically back up all of your EBS Volumes?

- **Use an EBS-cycle policy in Amazon S3 to automatically back up the EBS volumes.**
- **Use Amazon Data Lifecycle Manager (Amazon DLM) to automate the creation of EBS snapshots.**

  (Correct)
- **Set your Amazon Storage Gateway with EBS volumes as the data source and store the backups in your on-premises servers through the storage gateway.**
- **For an automated solution, create a scheduled job that calls the "create-snapshot" command via the AWS CLI to take a snapshot of production EBS volumes periodically.**

Explanation

You can use Amazon Data Lifecycle Manager (Amazon DLM) to automate the creation, retention, and deletion of snapshots taken to back up your Amazon EBS volumes. Automating snapshot management helps you to:

- Protect valuable data by enforcing a regular backup schedule.

- Retain backups as required by auditors or internal compliance.

- Reduce storage costs by deleting outdated backups.

Combined with the monitoring features of Amazon CloudWatch Events and AWS CloudTrail, Amazon DLM provides a complete backup solution for EBS volumes at no additional cost.

Hence, **using Amazon Data Lifecycle Manager (Amazon DLM) to automate the creation of EBS snapshots** is the correct answer as it is the fastest and most cost-effective solution that provides an automated way of backing up your EBS volumes.

The option that says: **For an automated solution, create a scheduled job that calls the "create-snapshot" command via the AWS CLI to take a snapshot of production EBS volumes periodically** is incorrect because even though this is a valid solution, you would still need additional time to create a scheduled job that calls the "create-snapshot" command. It would be better to use Amazon Data Lifecycle Manager (Amazon DLM) instead as this provides you the fastest solution which enables you to automate the creation, retention, and deletion of the EBS snapshots without having to write custom shell scripts or creating scheduled jobs.

Setting your Amazon Storage Gateway with EBS volumes as the data source and storing the backups in your on-premises servers through the storage gateway is incorrect as the Amazon Storage Gateway is used only for creating a backup of data from your on-premises server and not from the Amazon Virtual Private Cloud.

Using an EBS-cycle policy in Amazon S3 to automatically back up the EBS volumes is incorrect as there is no such thing as EBS-cycle policy in Amazon S3.

References:

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/snapshot-lifecycle.html

http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ebs-creating-snapshot.html

Check out this Amazon EBS Cheat Sheet:

https://tutorialsdojo.com/amazon-ebs/

Amazon EBS Overview - SSD vs HDD:

https://www.youtube.com/watch?v=LW7x8wyLFvw&t=8s

Question 57: **Correct**
A media company hosts large volumes of archive data that are about 250 TB in size on their internal servers. They have decided to move these data to S3 because of its durability and redundancy. The company currently has a 100 Mbps dedicated line connecting their head office to the Internet.

Which of the following is the FASTEST and the MOST cost-effective way to import all these data to Amazon S3?

- **Use AWS Snowmobile to transfer the data over to S3.**
- **Establish an AWS Direct Connect connection then transfer the data over to S3.**
- **Order multiple AWS Snowball devices to upload the files to Amazon S3.**

  (Correct)
- **Upload it directly to S3**

Explanation
AWS Snowball is a petabyte-scale data transport solution that uses secure appliances to transfer large amounts of data into and out of the AWS cloud. Using Snowball addresses common challenges with large-scale data transfers, including high network costs, long transfer times, and security concerns. Transferring data with Snowball is simple, fast, secure, and can be as little as one-fifth the cost of high-speed Internet.

Snowball is a strong choice for data transfer if you need to more securely and quickly transfer terabytes to many petabytes of data to AWS. Snowball can also be the right choice if you don’t want to make expensive upgrades to your network infrastructure, if you frequently experience large backlogs of data, if you're located in a physically isolated environment, or if you're in an area where high-speed Internet connections are not available or cost-prohibitive.

As a rule of thumb, if it takes more than one week to upload your data to AWS using the spare capacity of your existing Internet connection, then you should consider using Snowball. For example, if you have a 100 Mb connection that you can solely dedicate to transferring your data and need to transfer 100 TB of data, it takes more than 100 days to complete data transfer over that connection. You can make the same transfer by using multiple Snowballs in about a week.

Available Internet Connection	Theoretical Min. Number of Days to Transfer 100TB at 80% Network Utilization	When to Consider AWS Snowball?
T3 (44.736Mbps)	269 days	2TB or more
100Mbps	120 days	5TB or more
1000Mbps	12 days	60TB or more

Hence, **ordering multiple AWS Snowball devices to upload the files to Amazon S3** is the correct answer.

Uploading it directly to S3 is incorrect since this would take too long to finish due to the slow Internet connection of the company.

Establishing an AWS Direct Connect connection then transferring the data over to S3 is incorrect since provisioning a line for Direct Connect would take too much time and might not give you the fastest data transfer solution. In addition, the scenario didn't warrant an establishment of a dedicated connection from your on-premises data center to AWS. The primary goal is to just do a one-time migration of data to AWS which can be accomplished by using AWS Snowball devices.

Using AWS Snowmobile to transfer the data over to S3 is incorrect because Snowmobile is more suitable if you need to move extremely large amounts of data to AWS or need to transfer up to 100PB of data. This will be transported on a 45-foot long ruggedized shipping container, pulled by a semi-trailer truck. Take note that you only need to migrate 250 TB of data, hence, this is not the most suitable and cost-effective solution.

References:

https://aws.amazon.com/snowball/

https://aws.amazon.com/snowball/faqs/

S3 Transfer Acceleration vs Direct Connect vs VPN vs Snowball vs Snowmobile:

https://tutorialsdojo.com/s3-transfer-acceleration-vs-direct-connect-vs-vpn-vs-snowball-vs-snowmobile/

Comparison of AWS Services Cheat Sheets:

https://tutorialsdojo.com/comparison-of-aws-services/

Question 58: **Incorrect**
An advertising company is currently working on a proof of concept project that automatically provides SEO analytics for its clients. Your company has a VPC in AWS that operates in a dual-stack mode in which IPv4 and IPv6 communication is allowed. You deployed the application to an Auto Scaling group of EC2 instances with an Application Load Balancer in front that evenly distributes the incoming traffic. You are ready to go live but you need to point your domain name (tutorialsdojo.com) to the Application Load Balancer.

In Route 53, which record types will you use to point the DNS name of the Application Load Balancer? (Select TWO.)

- ☑ **Alias with a type "CNAME" record set**

  (Incorrect)

- ☐ **Alias with a type of "MX" record set**

- ☐ **Non-Alias with a type "A" record set**

- ☐ **Alias with a type "AAAA" record set**

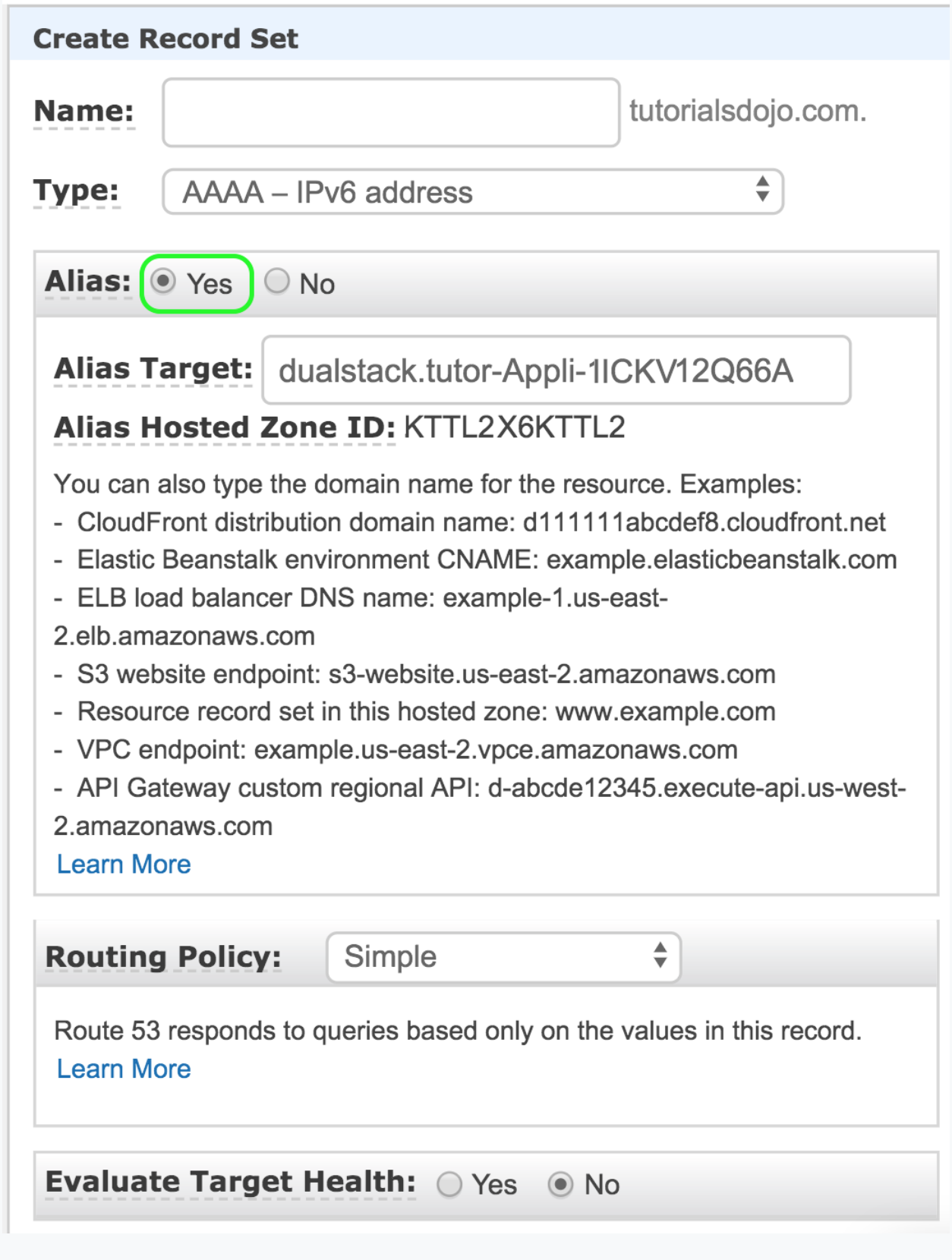
  (Correct)

- ☑ **Alias with a type "A" record set**

  (Correct)

Explanation
The correct answers are: **Alias with a type "AAAA" record set** and **Alias with a type "A" record set.**

To route domain traffic to an ELB load balancer, use Amazon Route 53 to create an alias record that points to your load balancer. An alias record is a Route 53 extension to DNS. It's similar to a CNAME record, but you can create an alias record both for the root domain, such as tutorialsdojo.com and for subdomains, such as portal.tutorialsdojo.com. (You can create CNAME records only for subdomains.) To enable IPv6 resolution, you would need to create a second resource record, tutorialsdojo.com ALIAS AAAA -> myelb.us-west-2.elb.amazonnaws.com, this is assuming your Elastic Load Balancer has IPv6 support.


Create Record Set
Name:
tutorialsdojo.com.
Type:
AAAA – IPv6 address
Alias:
Yes
No
Alias Target:
dualstack.tutor-Appli-1ICKV12Q66A
Alias Hosted Zone ID:
KTTL2X6KTTL2
You can also type the domain name for the resource. Examples:
- CloudFront distribution domain name: d111111abcdef8.cloudfront.net
- Elastic Beanstalk environment CNAME: example.elasticbeanstalk.com
- ELB load balancer DNS name: example-1.us-east-2.elb.amazonaws.com
- S3 website endpoint: s3-website.us-east-2.amazonaws.com
- Resource record set in this hosted zone: www.example.com
- VPC endpoint: example.us-east-2.vpce.amazonaws.com
- API Gateway custom regional API: d-abcde12345.execute-api.us-west-2.amazonaws.com
Learn More
Routing Policy:
Simple
Route 53 responds to queries based only on the values in this record.
Learn More
Evaluate Target Health:
Yes
No

Non-Alias with a type "A" record set is incorrect because you only use Non-Alias with a type “A” record set for IP addresses.

Alias with a type "CNAME" record set is incorrect because you can't create a CNAME record at the zone apex. For example, if you register the DNS name tutorialsdojo.com, the zone apex is tutorialsdojo.com.

Alias with a type of “MX” record set is incorrect because an MX record is primarily used for mail servers. It includes a priority number and a domain name, for example: `10 mailserver.tutorialsdojo.com`.

Reference:

https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/routing-to-elb-load-balancer.html

https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/resource-record-sets-choosing-alias-non-alias.html

Check out this Amazon Route 53 Cheat Sheet:

https://tutorialsdojo.com/amazon-route-53/

Question 59: **Correct**
An organization stores and manages financial records of various companies in its on-premises data center, which is almost out of space. The management decided to move all of their existing records to a cloud storage service. All future financial records will also be stored in the cloud. For additional security, all records must be prevented from being deleted or overwritten.

Which of the following should you do to meet the above requirement?

- **Use AWS Storage Gateway to establish hybrid cloud storage. Store all of your data in Amazon S3 and enable object lock.**
- **Use AWS Storage Gateway to establish hybrid cloud storage. Store all of your data in Amazon EBS and enable object lock.**

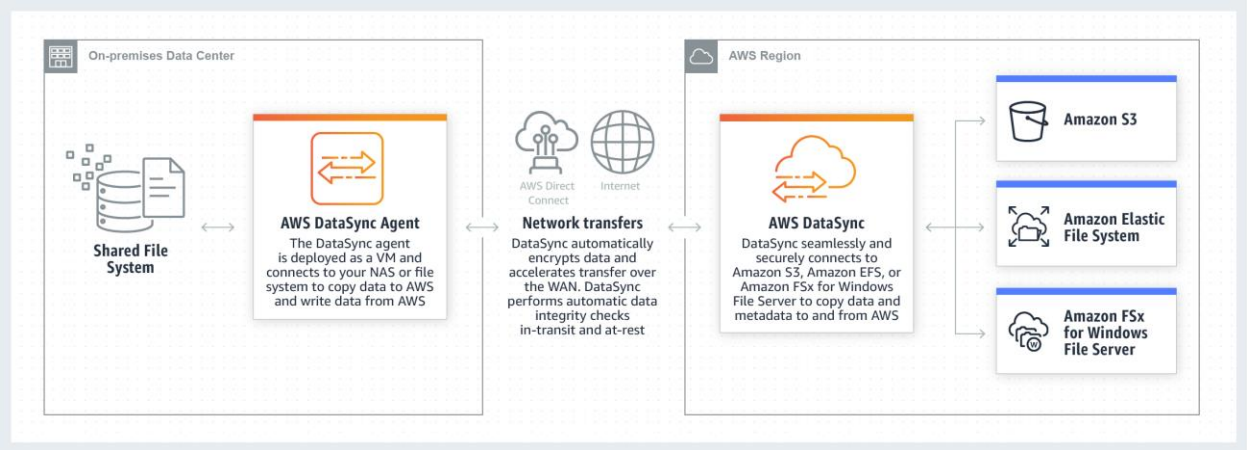
- **Use AWS DataSync to move the data. Store all of your data in Amazon S3 and enable object lock.**

  (Correct)

- **Use AWS DataSync to move the data. Store all of your data in Amazon EFS and enable object lock.**

Explanation

AWS DataSync allows you to copy large datasets with millions of files without having to build custom solutions with open source tools or licenses and manage expensive commercial network acceleration software. You can use DataSync to migrate active data to AWS, transfer data to the cloud for analysis and processing, archive data to free up on-premises storage capacity, or replicate data to AWS for business continuity.



AWS DataSync enables you to migrate your on-premises data to Amazon S3, Amazon EFS, and Amazon FSx for Windows File Server. You can configure DataSync to make an initial copy of your entire dataset and schedule subsequent incremental transfers of changing data towards Amazon S3. Enabling S3 Object Lock prevents your existing and future records from being deleted or overwritten.

AWS DataSync is primarily used to migrate existing data to Amazon S3. On the other hand, AWS Storage Gateway is more suitable if you still want to retain access to the migrated data and for ongoing updates from your on-premises file-based applications.

Hence, the correct answer in this scenario is: **Use AWS DataSync to move the data. Store all of your data in Amazon S3 and enable object lock**.

The option that says: **Use AWS DataSync to move the data. Store all of your data in Amazon EFS and enable object lock** is incorrect because Amazon EFS only supports file locking. Object lock is a feature of Amazon S3 and not Amazon EFS.

The options that says: **Use AWS Storage Gateway to establish hybrid cloud storage. Store all of your data in Amazon S3 and enable object lock** is incorrect because the scenario requires that all of the existing records must be migrated to AWS. The future records will also be stored in AWS and not in the on-premises network. This means that setting up a hybrid cloud storage is not necessary since the on-premises storage will no longer be used.

The option that says: **Use AWS Storage Gateway to establish hybrid cloud storage. Store all of your data in Amazon EBS, and enable object lock** is incorrect because Amazon EBS does not support object lock. Amazon S3 is the only service capable of locking objects to prevent an object from being deleted or overwritten.

References:

https://aws.amazon.com/datasync/faqs/

https://docs.aws.amazon.com/datasync/latest/userguide/what-is-datasync.html

https://docs.aws.amazon.com/AmazonS3/latest/dev/object-lock.html

Check out this AWS DataSync Cheat Sheet:

https://tutorialsdojo.com/aws-datasync/

AWS Storage Gateway vs. DataSync:

https://youtu.be/tmfe1rO-AUs

Amazon S3 vs EBS vs EFS Cheat Sheet:

https://tutorialsdojo.com/amazon-s3-vs-ebs-vs-efs/

Question 60: **Correct**
A startup has multiple AWS accounts that are assigned to its development teams. Since the company is projected to grow rapidly, the management wants to consolidate all of

its AWS accounts into a multi-account setup. To simplify the login process on the AWS accounts, the management wants to utilize its existing directory service for user authentication

Which combination of actions should a solutions architect recommend to meet these requirements? (Select TWO.)

- [x] **On the master account, use AWS Organizations to create a new organization with all features turned on. Invite the child accounts to this new organization.**

  (Correct)

- [ ] **Create Service Control Policies (SCP) in the organization to manage the child accounts. Configure AWS IAM Identity Center (AWS Single Sign-On) to use AWS Directory Service.**

- [x] **Configure AWS IAM Identity Center (AWS Single Sign-On) for the organization and integrate it with the company's directory service using the Active Directory Connector**

  (Correct)

- [ ] **On the master account, use AWS Organizations to create a new organization with all features turned on. Enable the organization's external authentication and point it to use the company's directory service.**

- [ ] **Create an identity pool on Amazon Cognito and configure it to use the company's directory service. Configure AWS IAM Identity Center (AWS Single Sign-On) to accept Cognito authentication.**

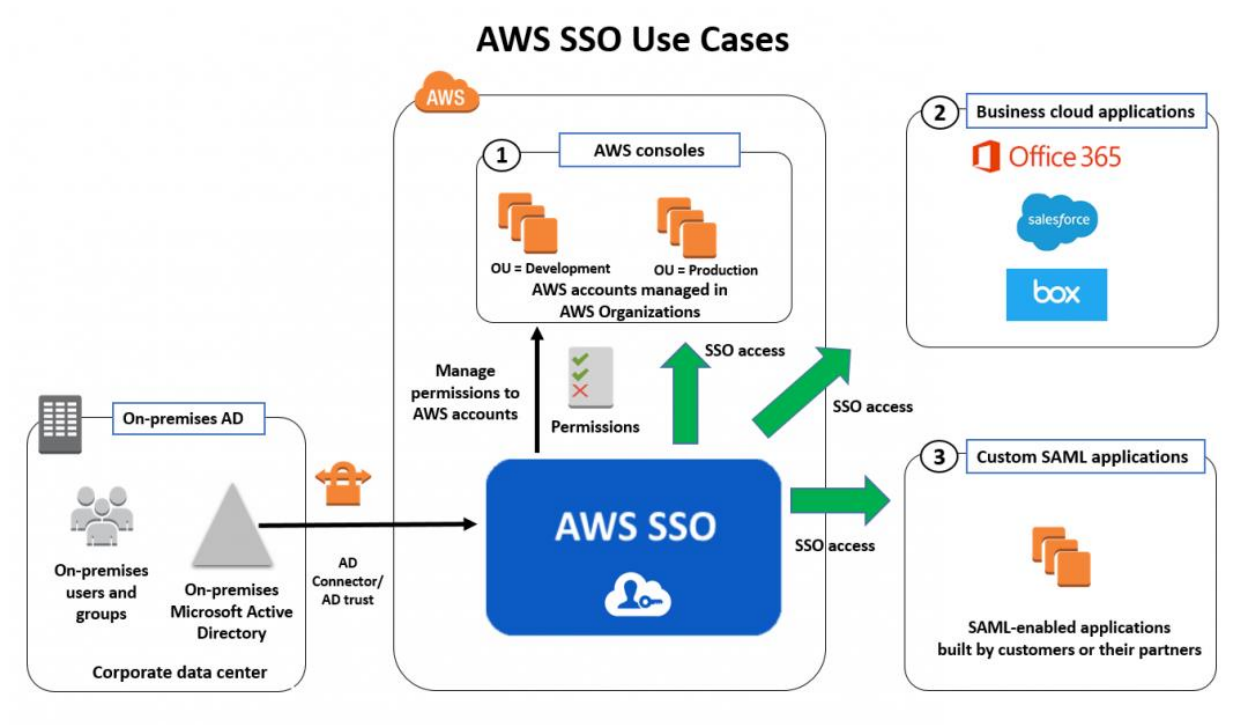
Explanation

AWS Organizations is an account management service that enables you to consolidate multiple AWS accounts into an organization that you create and centrally manage. AWS Organizations includes account management and consolidated billing capabilities that enable you to better meet the budgetary, security, and compliance needs of your

business. As an administrator of an organization, you can create accounts in your organization and invite existing accounts to join the organization.

AWS IAM Identity Center (successor to AWS Single Sign-On) provides single sign-on access for all of your AWS accounts and cloud applications. It connects with Microsoft Active Directory through AWS Directory Service to allow users in that directory to sign in to a personalized AWS access portal using their existing Active Directory user names and passwords. From the AWS access portal, users have access to all the AWS accounts and cloud applications that they have permission for.

Users in your self-managed directory in Active Directory (AD) can also have single sign-on access to AWS accounts and cloud applications in the AWS access portal.



Therefore, the correct answers are:

-On the master account, use AWS Organizations to create a new organization with all features turned on. Invite the child accounts to this new organization.

-Configure AWS IAM Identity Center (AWS Single Sign-On) for the organization and integrate it with the company's directory service using the Active Directory Connector

The option that says: **On the master account, use AWS Organizations to create a new organization with all features turned on. Enable the organization's external authentication and point it to use the company's directory service** is incorrect. There is

no option to use an external authentication on AWS Organizations. You will need to configure AWS SSO if you want to use an existing Directory Service.

The option that says: **Create an identity pool on Amazon Cognito and configure it to use the company's directory service. Configure AWS IAM Identity Center (AWS Single Sign-On) to accept Cognito authentication** is incorrect. Amazon Cognito is used for single sign-on in mobile and web applications. You don't have to use it if you already have an existing Directory Service to be used for authentication.

The option that says: **Create Service Control Policies (SCP) in the organization to manage the child accounts. Configure AWS IAM Identity Center (AWS Single Sign-On) to use AWS Directory Service** is incorrect. SCPs are not necessarily needed for logging in on this scenario. You can use SCP if you want to restrict or implement a policy across several accounts in the organization.

References:

https://docs.aws.amazon.com/organizations/latest/userguide/orgs_introduction.html

https://docs.aws.amazon.com/singlesignon/latest/userguide/what-is.html

https://docs.aws.amazon.com/organizations/latest/userguide/services-that-can-integrate-sso.html

Check out AWS Organizations Cheat Sheets:

https://tutorialsdojo.com/aws-organizations/

Question 61: **Correct**
A company has multiple VPCs with IPv6 enabled for its suite of web applications. The Solutions Architect tried to deploy a new Amazon EC2 instance but she received an error saying that there is no IP address available on the subnet.

How should the Solutions Architect resolve this problem?

- **Set up a new IPv4 subnet with a larger CIDR range. Associate the new subnet with the VPC and then launch the instance.**

  (Correct)

- **Set up a new IPv6-only subnet with a large CIDR range. Associate the new subnet with the VPC then launch the instance.**

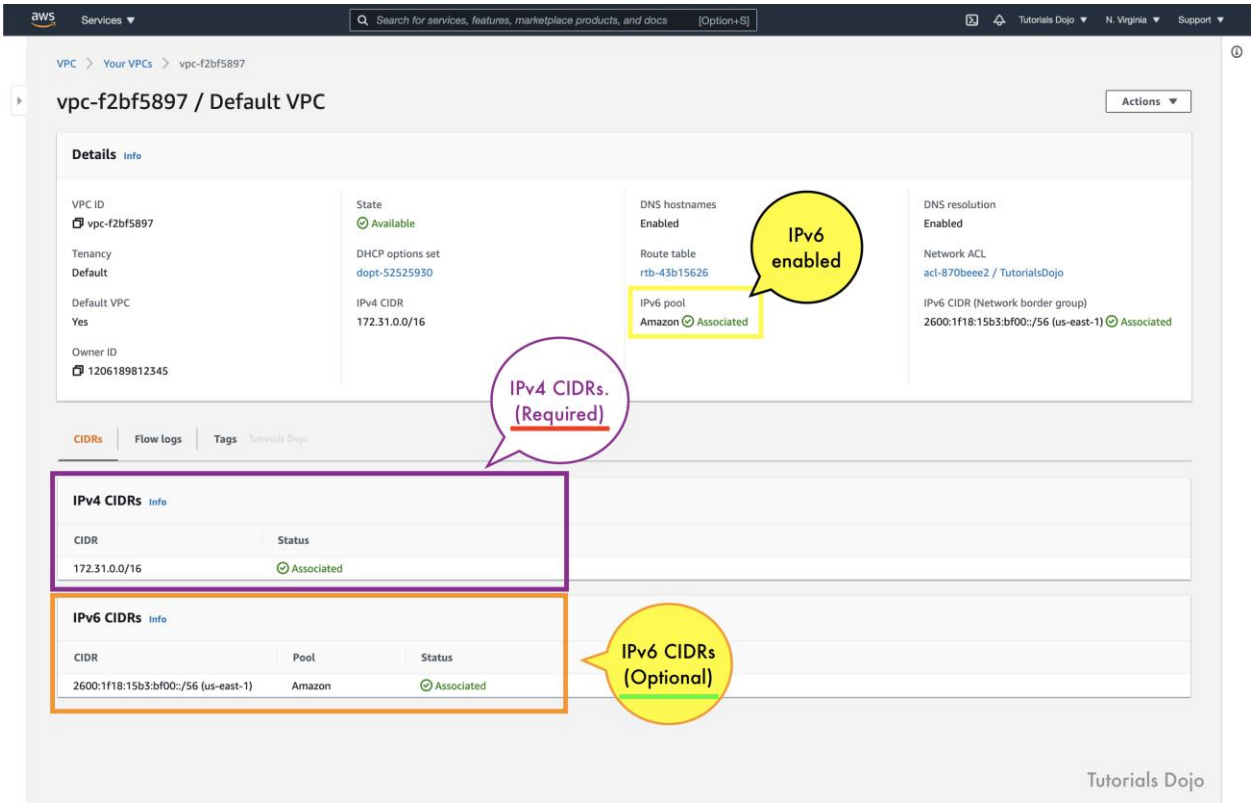
- **Ensure that the VPC has IPv6 CIDRs only. Remove any IPv4 CIDRs associated with the VPC.**

- **Disable the IPv4 support in the VPC and use the available IPv6 addresses.**

Explanation

Amazon Virtual Private Cloud (VPC) is a service that lets you launch AWS resources in a logically isolated virtual network that you define. You have complete control over your virtual networking environment, including selection of your own IP address range, creation of subnets, and configuration of route tables and network gateways. You can use both IPv4 and IPv6 for most resources in your virtual private cloud, helping to ensure secure and easy access to resources and applications.

A subnet is a range of IP addresses in your VPC. You can launch AWS resources into a specified subnet. When you create a VPC, you must specify a range of IPv4 addresses for the VPC in the form of a CIDR block. Each subnet must reside entirely within one Availability Zone and cannot span zones. You can also optionally assign an IPv6 CIDR block to your VPC, and assign IPv6 CIDR blocks to your subnets.



If you have an existing VPC that supports IPv4 only and resources in your subnet that are configured to use IPv4 only, you can enable IPv6 support for your VPC and resources. Your VPC can operate in dual-stack mode — your resources can communicate over IPv4, or IPv6, or both. IPv4 and IPv6 communication are independent of each other. You cannot disable IPv4 support for your VPC and subnets since this is the default IP addressing system for Amazon VPC and Amazon EC2.

By default, a new EC2 instance uses an IPv4 addressing protocol. To fix the problem in the scenario, you need to create a new IPv4 subnet and deploy the EC2 instance in the new subnet.

Hence, the correct answer is: **Set up a new IPv4 subnet with a larger CIDR range. Associate the new subnet with the VPC and then launch the instance.**

The option that says: **Set up a new IPv6-only subnet with a large CIDR range. Associate the new subnet with the VPC then launch the instance** is incorrect because you need to add IPv4 subnet first before you can create an IPv6 subnet.

The option that says: **Ensure that the VPC has IPv6 CIDRs only. Remove any IPv4 CIDRs associated with the VPC** is incorrect because you can't have a VPC with IPv6 CIDRs only. The default IP addressing system in VPC is IPv4. You can only change your VPC to dual-stack mode where your resources can communicate over IPv4, or IPv6, or both, but not exclusively with IPv6 only.

The option that says: **Disable the IPv4 support in the VPC and use the available IPv6 addresses** is incorrect because you cannot disable the IPv4 support for your VPC and subnets since this is the default IP addressing system.

References:

https://docs.aws.amazon.com/vpc/latest/userguide/vpc-migrate-ipv6.html

https://docs.aws.amazon.com/vpc/latest/userguide/vpc-ip-addressing.html

https://aws.amazon.com/vpc/faqs/

Check out this Amazon VPC Cheat Sheet:

https://tutorialsdojo.com/amazon-vpc/

Question 62: **Correct**
A solutions architect is instructed to host a website that consists of HTML, CSS, and some Javascript files. The web pages will display several high-resolution images. The website should have optimal loading times and be able to respond to high request rates.

Which of the following architectures can provide the most cost-effective and fastest loading experience?

- **Upload the HTML, CSS, Javascript, and the images in a single bucket. Then enable website hosting. Create a CloudFront distribution and point the domain on the S3 website endpoint.**

  (Correct)

- **Host the website using an Nginx server in an EC2 instance. Upload the images in an S3 bucket. Use CloudFront as a CDN to deliver the images closer to end-users.**

-

Host the website in an AWS Elastic Beanstalk environment. Upload the images in an S3 bucket. Use CloudFront as a CDN to deliver the images closer to your end-users.

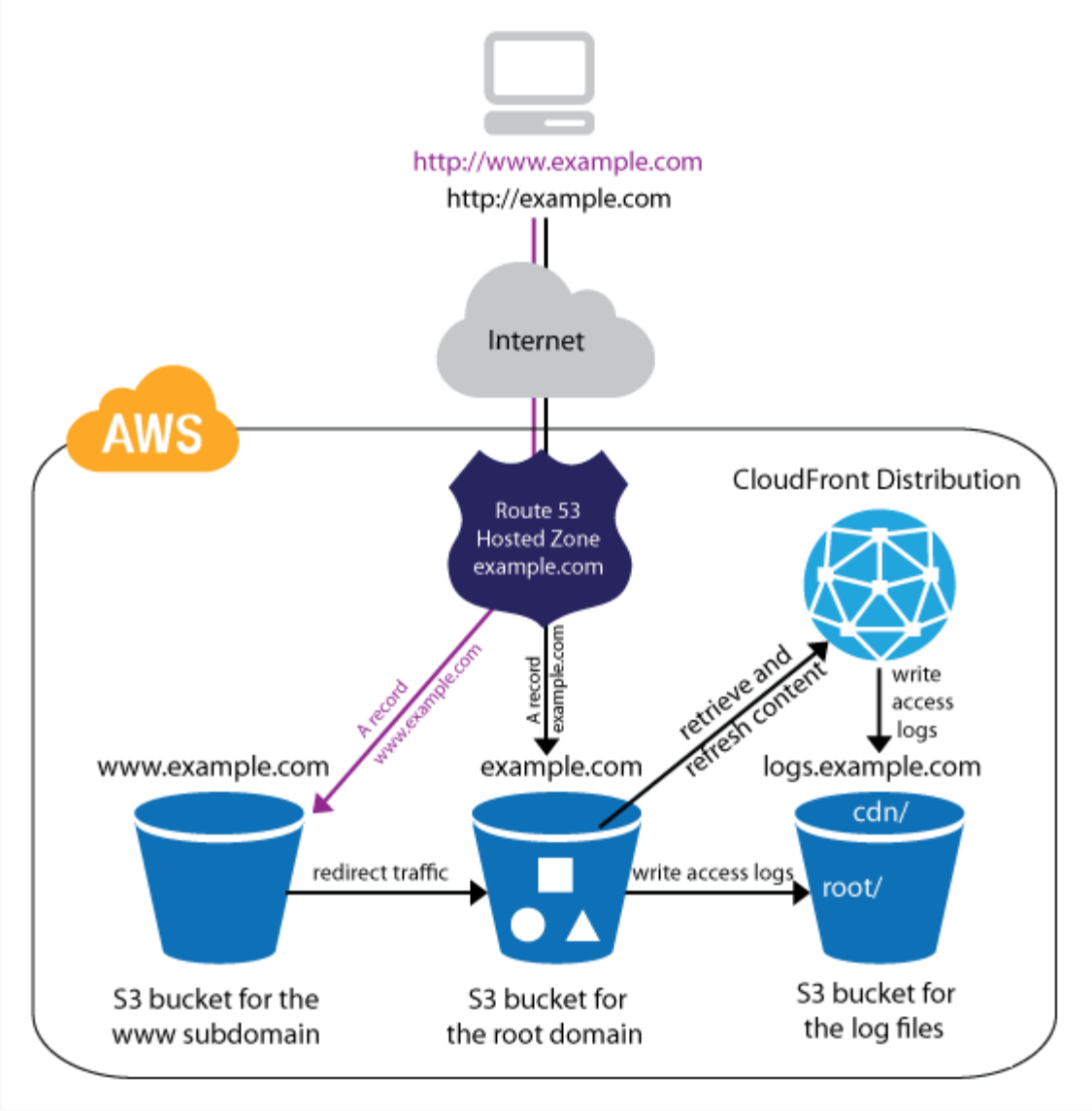
- 

Launch an Auto Scaling Group using an AMI that has a pre-configured Apache web server, then configure the scaling policy accordingly. Store the images in an Elastic Block Store. Then, point your instance's endpoint to AWS Global Accelerator.

Explanation

Amazon S3 is an object storage service that offers industry-leading scalability, data availability, security, and performance. Additionally, You can use Amazon S3 to host a static website. On a static website, individual webpages include static content. Amazon S3 is **highly scalable and you only pay for what you use**, you can start small and grow your application as you wish, with no compromise on performance or reliability.

Amazon CloudFront is a fast content delivery network (CDN) service that securely delivers data, videos, applications, and APIs to customers globally with low latency, high transfer speeds. CloudFront can be integrated with Amazon S3 for fast delivery of data originating from an S3 bucket to your end-users. By design, delivering data out of CloudFront can be more cost-effective than delivering it from S3 directly to your users.



In the scenario, Since we are only dealing with static content, we can leverage the web hosting feature of S3. Then we can improve the architecture further by integrating it with CloudFront. This way, users will be able to load both the web pages and images faster than if we hosted them on a webserver that we built from scratch.

Hence, the correct answer is: **Upload the HTML, CSS, Javascript, and the images in a single bucket. Then enable website hosting. Create a CloudFront distribution and point the domain on the S3 website endpoint.**

The option that says: **Host the website using an Nginx server in an EC2 instance. Upload the images in an S3 bucket. Use CloudFront as a CDN to deliver the images closer to end-users** is incorrect. Creating your own web server to host a static website in AWS is a costly solution. Web Servers on an EC2 instance are usually used for hosting applications that require server-side processing (connecting to a database, data

validation, etc.). Since static websites contain web pages with fixed content, we should use S3 website hosting instead.

The option that says: **Launch an Auto Scaling Group using an AMI that has a pre-configured Apache web server, then configure the scaling policy accordingly. Store the images in an Elastic Block Store. Then, point your instance's endpoint to AWS Global Accelerator** is incorrect. This is how we serve static websites in the old days. Now, with the help of S3 website hosting, we can host our static contents from a durable, high-availability, and highly scalable environment without managing any servers. Hosting static websites in S3 is cheaper than hosting it in an EC2 instance. In addition, Using ASG for scaling instances that host a static website is an over-engineered solution that carries unnecessary costs. S3 automatically scales to high requests and you only pay for what you use.

The option that says: **Host the website in an AWS Elastic Beanstalk environment. Upload the images in an S3 bucket. Use CloudFront as a CDN to deliver the images closer to your end-users** is incorrect. AWS Elastic Beanstalk simply sets up the infrastructure (EC2 instance, load balancer, auto-scaling group) for your application. It's a more expensive and a bit of an overkill solution for hosting a bunch of client-side files.

References:

https://docs.aws.amazon.com/AmazonS3/latest/dev/WebsiteHosting.html

https://aws.amazon.com/blogs/networking-and-content-delivery/amazon-s3-amazon-cloudfront-a-match-made-in-the-cloud/

Check out these Amazon S3 and CloudFront Cheat Sheets:

https://tutorialsdojo.com/amazon-s3/

https://tutorialsdojo.com/amazon-cloudfront/

Question 63: **Incorrect**
A company plans to migrate all of their applications to AWS. The Solutions Architect suggested to store all the data to EBS volumes. The Chief Technical Officer is worried that EBS volumes are not appropriate for the existing workloads due to compliance requirements, downtime scenarios, and IOPS performance.

Which of the following are valid points in proving that EBS is the best service to use for migration? (Select TWO.)

- ☐

  EBS volumes can be attached to any EC2 Instance in any Availability Zone.

- ☐

  When you create an EBS volume in an Availability Zone, it is automatically replicated on a separate AWS region to prevent data loss due to a failure of any single hardware component.

- ☐

  EBS volumes support live configuration changes while in production which means that you can modify the volume type, volume size, and IOPS capacity without service interruptions.

  (Correct)

- ☑

  An EBS volume is off-instance storage that can persist independently from the life of an instance.

  (Correct)

- ☑

  Amazon EBS provides the ability to create snapshots (backups) of any EBS volume and write a copy of the data in the volume to Amazon RDS, where it is stored redundantly in multiple Availability Zones

  (Incorrect)

Explanation
An Amazon EBS volume is a durable, block-level storage device that you can attach to a single EC2 instance. You can use EBS volumes as primary storage for data that requires frequent updates, such as the system drive for an instance or storage for a database application. You can also use them for throughput-intensive applications that perform continuous disk scans. EBS volumes persist independently from the running life of an EC2 instance.

Here is a list of important information about EBS Volumes:

- When you create an EBS volume in an Availability Zone, it is automatically replicated within that zone to prevent data loss due to a failure of any single hardware component.

- An EBS volume can only be attached to one EC2 instance at a time.

- After you create a volume, you can attach it to any EC2 instance in the same Availability Zone

- An EBS volume is off-instance storage that can persist independently from the life of an instance. You can specify not to terminate the EBS volume when you terminate the EC2 instance during instance creation.

- EBS volumes support live configuration changes while in production which means that you can modify the volume type, volume size, and IOPS capacity without service interruptions.

- Amazon EBS encryption uses 256-bit Advanced Encryption Standard algorithms (AES-256)

- EBS Volumes offer 99.999% SLA.

The option that says: **When you create an EBS volume in an Availability Zone, it is automatically replicated on a separate AWS region to prevent data loss due to a failure of any single hardware component** is incorrect because when you create an EBS volume in an Availability Zone, it is automatically replicated within that zone only, and not on a separate AWS region, to prevent data loss due to a failure of any single hardware component.

The option that says: **EBS volumes can be attached to any EC2 Instance in any Availability Zone** is incorrect as EBS volumes can only be attached to an EC2 instance in the same Availability Zone.

The option that says: **Amazon EBS provides the ability to create snapshots (backups) of any EBS volume and write a copy of the data in the volume to Amazon RDS, where it is stored redundantly in multiple Availability Zones** is almost correct. But instead of storing the volume to Amazon RDS, the EBS Volume snapshots are actually sent to Amazon S3.

References:

http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EBSVolumes.html

https://aws.amazon.com/ebs/features/

Check out this Amazon EBS Cheat Sheet:

https://tutorialsdojo.com/amazon-ebs/

Here is a short video tutorial on EBS:

https://youtu.be/ljYH5lHQdxo

Question 64: **Incorrect**
A solutions architect is designing a three-tier website that will be hosted on an Amazon EC2 Auto Scaling group fronted by an Internet-facing Application Load Balancer (ALB). The website will persist data to an Amazon Aurora Serverless DB cluster, which will also be used for generating monthly reports.

The company requires a network topology that follows a layered approach to reduce the impact of misconfigured security groups or network access lists. Web filtering must also be enabled to automatically stop traffic to known malicious URLs and to immediately drop requests coming from blacklisted fully qualified domain names (FQDNs).

Which network topology provides the minimum resources needed for the website to work?

- **Set up an Application Load Balancer deployed in a public subnet, then host the Auto Scaling Group of Amazon EC2 instances and the Aurora Serverless DB cluster in private subnets. Launch an AWS Network Firewall with the appropriate firewall policy to automatically stop traffic to known malicious URLs and drop requests coming from blacklisted FQDNs. Reroute your Amazon VPC network traffic through the firewall endpoints.**

  (Correct)

- **Set up an Auto Scaling group of Amazon EC2 instances behind an Application Load Balancer with an Aurora Serverless DB cluster to store application data. Deploy all resources in a public subnet. Configure host-based routing to the Application Load Balancer to stop traffic to known malicious URLs and drop requests from blacklisted FQDNs.**

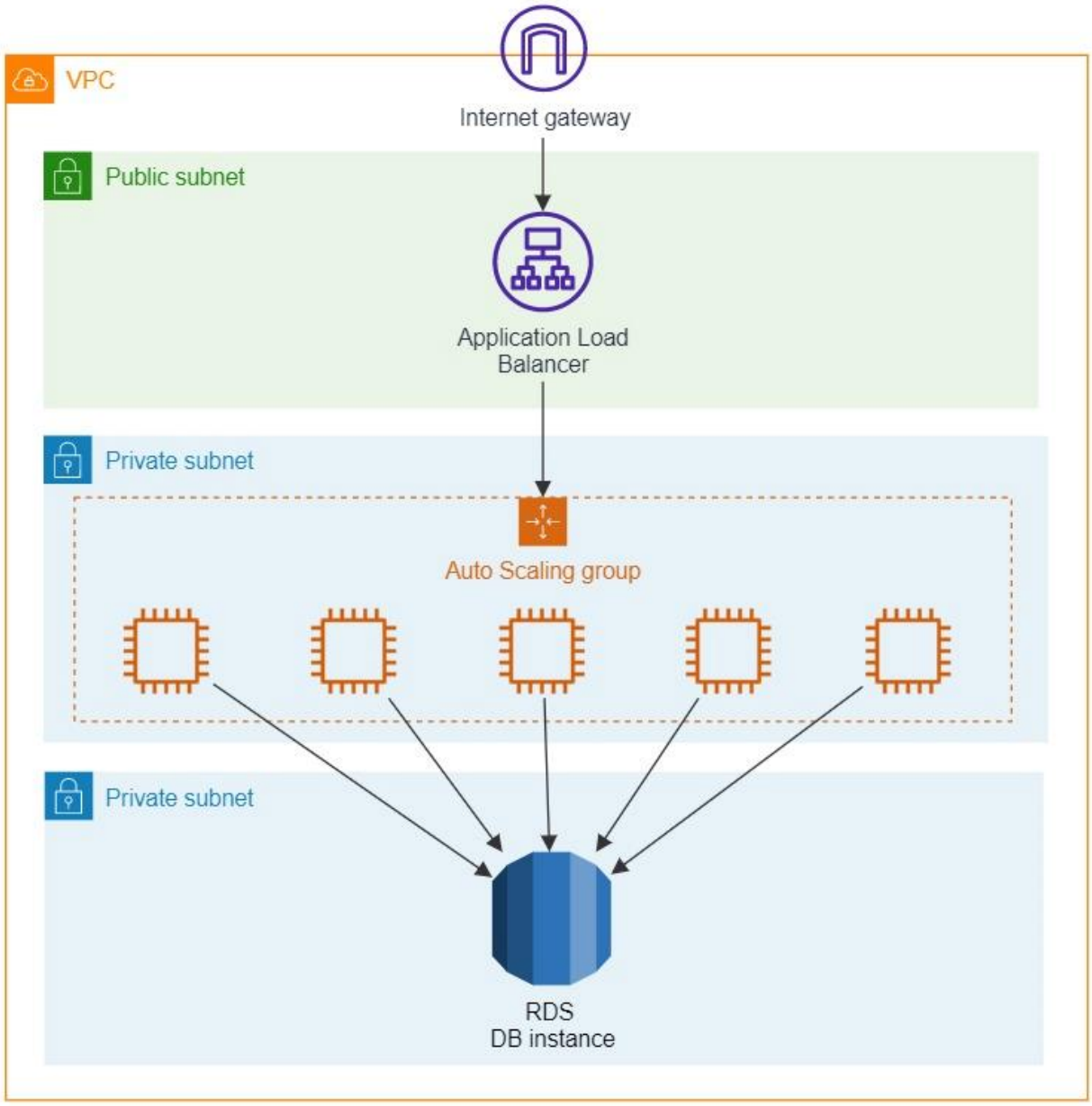
- **Set up an Application Load Balancer in front of an Auto Scaling group of Amazon EC2 instances with an Aurora Serverless DB cluster to persist data. Launch a NAT Gateway in a public subnet to restrict external services from initiating a connection to the EC2 instances and immediately drop requests from unauthorized FQDNs. Deploy all other resources in private subnets.**

- **Set up an Application Load Balancer and a NAT Gateway deployed in public subnets. Launch the Auto Scaling Group of Amazon EC2 instances and Aurora Serverless DB cluster in private subnets. Directly integrate the AWS Network Firewall with the Application Load Balancer to automatically stop traffic to known malicious URLs and drop requests coming from blacklisted FQDNs.**

  (Incorrect)

Explanation
A defense-in-depth strategy is one of the design principles for security in the AWS cloud. This strategy entails implementing security controls at multiple layers (for example, edge of network, VPC, load balancing, every instance and compute service, operating

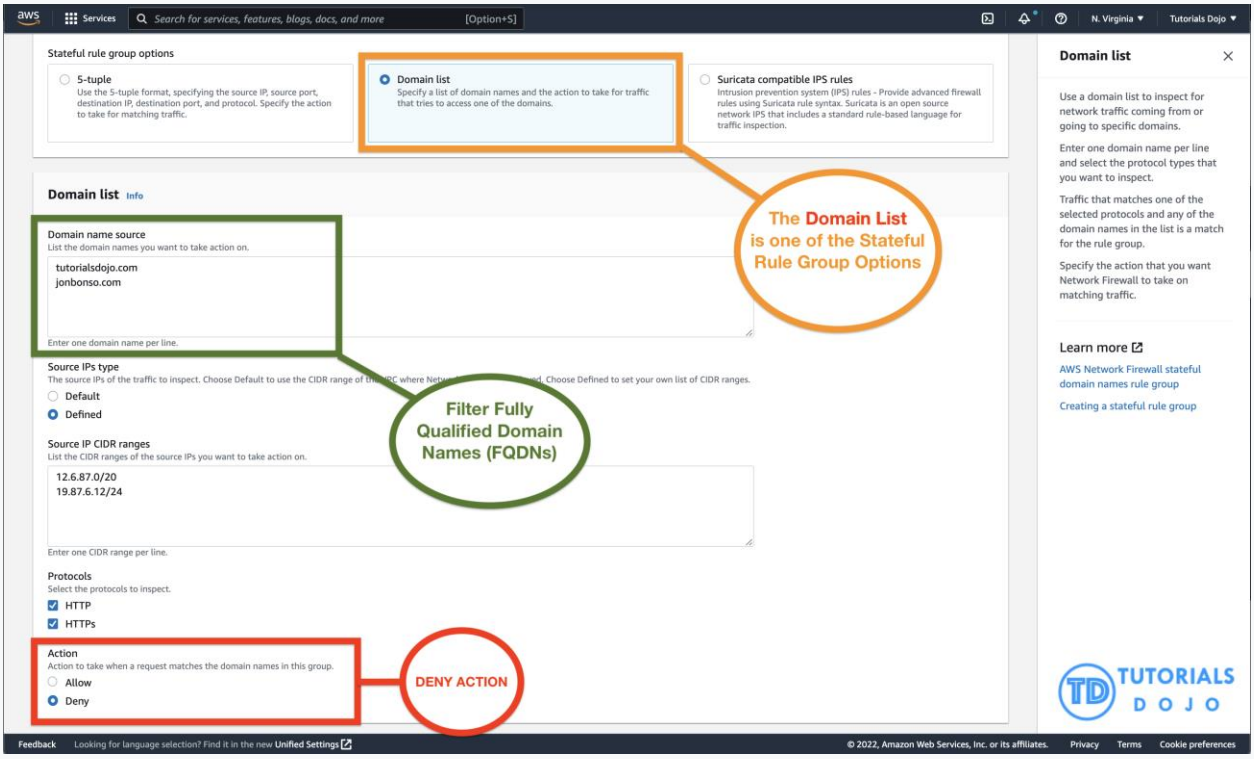
system, application, and code).



Components such as EC2 instances, RDS database clusters, and Lambda functions that share reachability requirements can be segmented into layers formed by subnets. For example, an RDS database cluster in a VPC with no need for internet access should be placed in subnets with no route to or from the internet. This layered approach for the controls mitigates the impact of a single layer misconfiguration, which could allow unintended access.

AWS Network Firewall is a stateful, managed network firewall and intrusion detection and prevention service for your virtual private cloud (VPC) that you created in Amazon Virtual Private Cloud (Amazon VPC). With Network Firewall, you can filter traffic at the

perimeter of your VPC. This includes filtering traffic going to and coming from an internet gateway, NAT gateway, or over VPN or AWS Direct Connect. Network Firewall uses the open source intrusion prevention system (IPS), Suricata, for stateful inspection. Network Firewall supports Suricata compatible rules.



AWS Network Firewall supports domain name stateful network traffic inspection. You can create *Allow* lists and *Deny* lists with domain names that the stateful rules engine looks for in network traffic.

Hence, the correct answer in this scenario is: **Set up an Application Load Balancer deployed in a public subnet, then host the Auto Scaling Group of Amazon EC2 instances and the Aurora Serverless DB cluster in private subnets. Launch an AWS Network Firewall with the appropriate firewall policy to automatically stop traffic to known malicious URLs and drop requests coming from blacklisted FQDNs. Reroute your Amazon VPC network traffic through the firewall endpoints.**

The option that says: **Set up an Application Load Balancer and a NAT Gateway deployed in public subnets. Launch the Auto Scaling Group of Amazon EC2 instances and Aurora Serverless DB cluster in private subnets. Directly integrate the AWS Network Firewall with the Application Load Balancer to automatically stop traffic to known malicious URLs and drop requests coming from blacklisted FQDNs** is incorrect. NAT Gateway is commonly used to provide internet access to EC2 instances in private subnets while preventing external services from initiating connections to the instances. This component is not necessary for the application to work. Take note that you cannot

directly integrate the AWS Network Firewall with the Application Load Balancer. There is a straightforward way of integrating an AWS WAF with an ALB but not an AWS Network Firewall with an ALB.

The option that says: **Set up an Application Load Balancer in front of an Auto Scaling group of Amazon EC2 instances with an Aurora Serverless DB cluster to persist data. Launch a NAT Gateway in a public subnet to restrict external services from initiating a connection to the EC2 instances and immediately drop requests from unauthorized FQDNs. Deploy all other resources in private subnets** is incorrect. You have to place the Application Load Balancer in a public subnet in order for the application to serve requests from the Internet. Furthermore, a NAT Gateway does not have any features to immediately drop requests from unauthorized FQDNs.

The option that says: **Set up an Auto Scaling group of Amazon EC2 instances behind an Application Load Balancer with an Aurora Serverless DB cluster to store application data. Deploy all resources in a public subnet. Configure host-based routing to the Application Load Balancer to stop traffic to known malicious URLs and drop requests from blacklisted FQDNs** is incorrect. While this setup works fine, it does not follow a layered approach since all components are placed in a single public subnet. It is better to place the Aurora database into a private subnet to further protect the application data. In addition, the host-based routing in the Application Load Balancer is not capable of totally stopping the requests coming from, or going to, known malicious URLs and blacklisted FQDNs. You have to use the AWS Network Firewall service for this particular scenario.

References:

https://wa.aws.amazon.com/wellarchitected/2020-07-02T19-33-23/wat.pillar.security.en.html

https://docs.aws.amazon.com/wellarchitected/latest/security-pillar/protecting-networks.html

https://docs.aws.amazon.com/network-firewall/latest/developerguide/stateful-rule-groups-domain-names.html

Check out these Amazon EC2, AWS ELB, and Amazon Aurora cheat sheets:

https://tutorialsdojo.com/amazon-elastic-compute-cloud-amazon-ec2/

https://tutorialsdojo.com/aws-elastic-load-balancing-elb/

https://tutorialsdojo.com/amazon-aurora/

Question 65: **Correct**
A company has a cryptocurrency exchange portal that is hosted in an Auto Scaling group of EC2 instances behind an Application Load Balancer and is deployed across multiple AWS regions. The users can be found all around the globe, but the majority are from Japan and Sweden. Because of the compliance requirements in these two locations, you want the Japanese users to connect to the servers in the `ap-northeast-1` Asia Pacific (Tokyo) region, while the Swedish users should be connected to the servers in the `eu-west-1` EU (Ireland) region.

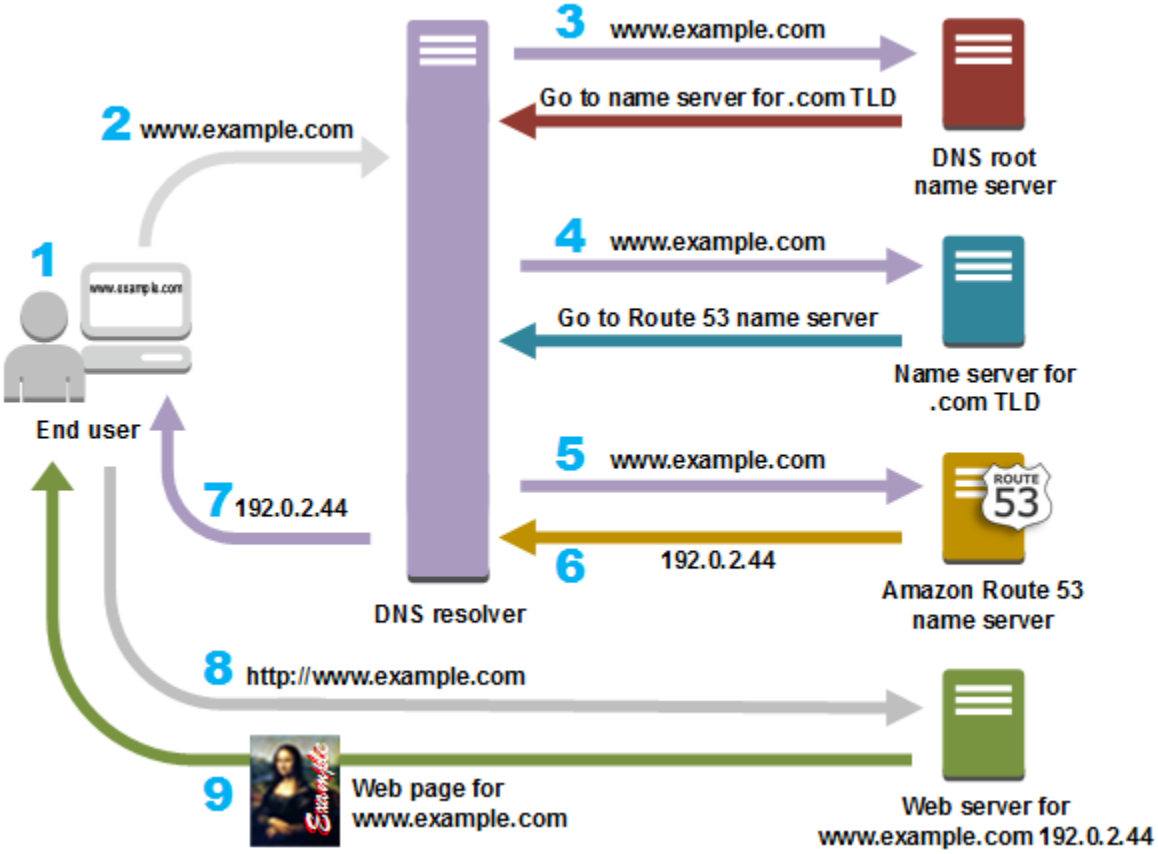
Which of the following services would allow you to easily fulfill this requirement?

- **Set up an Application Load Balancers that will automatically route the traffic to the proper AWS region.**
- **Set up a new CloudFront web distribution with the geo-restriction feature enabled.**
- **Use Route 53 Weighted Routing policy.**
- **Use Route 53 Geolocation Routing policy.**

  (Correct)

Explanation
Geolocation routing lets you choose the resources that serve your traffic based on the geographic location of your users, meaning the location that DNS queries originate from. For example, you might want all queries from Europe to be routed to an ELB load balancer in the Frankfurt region.

When you use geolocation routing, you can localize your content and present some or all of your website in the language of your users. You can also use geolocation routing to restrict the distribution of content to only the locations in which you have distribution rights. Another possible use is for balancing load across endpoints in a predictable, easy-to-manage way so that each user location is consistently routed to the same endpoint.



Setting up an Application Load Balancers that will automatically route the traffic to the proper AWS region is incorrect because Elastic Load Balancers distribute traffic among EC2 instances across multiple Availability Zones but not across AWS regions.

Setting up a new CloudFront web distribution with the geo-restriction feature enabled is incorrect because the CloudFront geo-restriction feature is primarily used to prevent users in specific geographic locations from accessing content that you're distributing through a CloudFront web distribution. It does not let you choose the resources that serve your traffic based on the geographic location of your users, unlike the Geolocation routing policy in Route 53.

Using Route 53 Weighted Routing policy is incorrect because this is not a suitable solution to meet the requirements of this scenario. It just lets you associate multiple resources with a single domain name (tutorialsdojo.com) or subdomain name (forums.tutorialsdojo.com) and choose how much traffic is routed to each resource. You have to use a Geolocation routing policy instead.

References:

https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/routing-policy.html

https://aws.amazon.com/premiumsupport/knowledge-center/geolocation-routing-policy

Check out this Amazon Route 53 Cheat Sheet:

https://tutorialsdojo.com/amazon-route-53/

Latency Routing vs. Geoproximity Routing vs. Geolocation Routing:

https://tutorialsdojo.com/latency-routing-vs-geoproximity-routing-vs-geolocation-routing/

Comparison of AWS Services Cheat Sheets:

https://tutorialsdojo.com/comparison-of-aws-services/