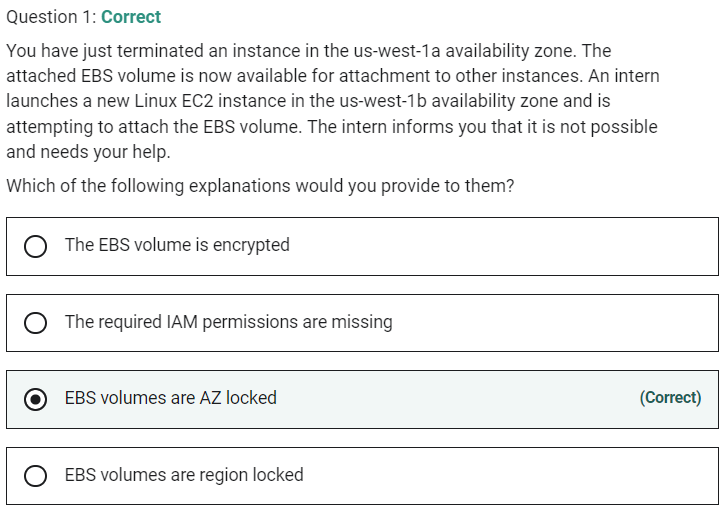
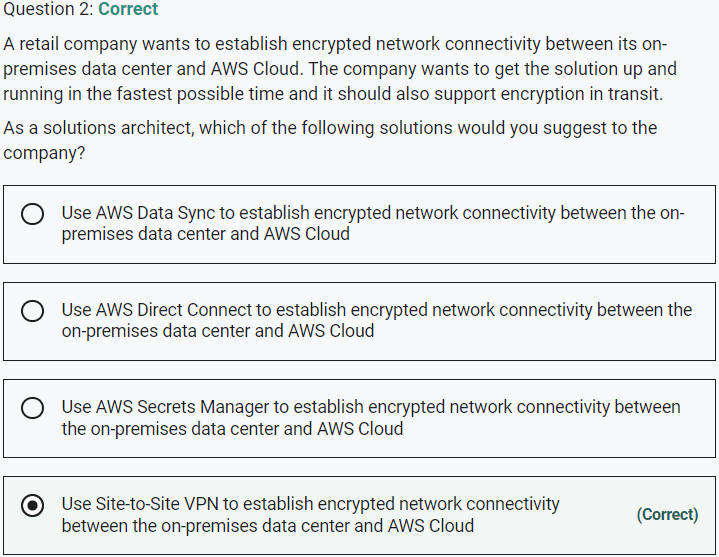
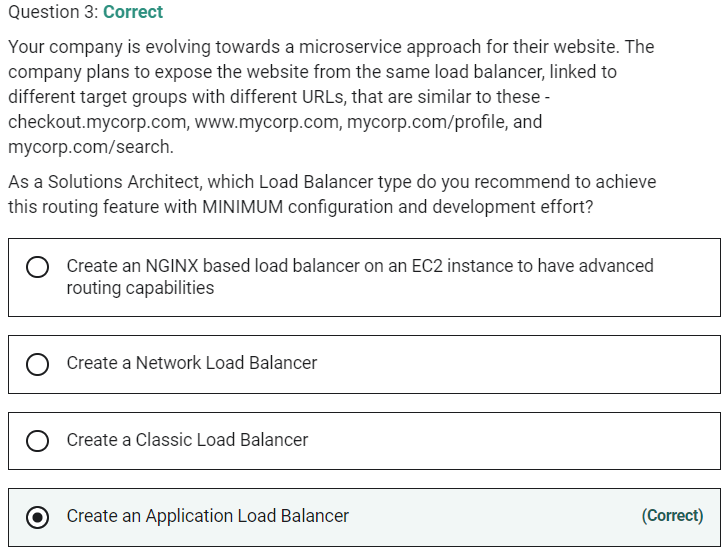
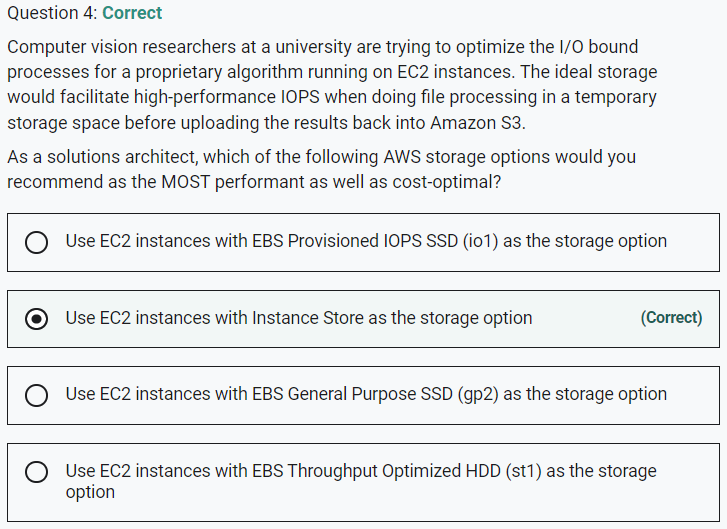
SAA-CO2

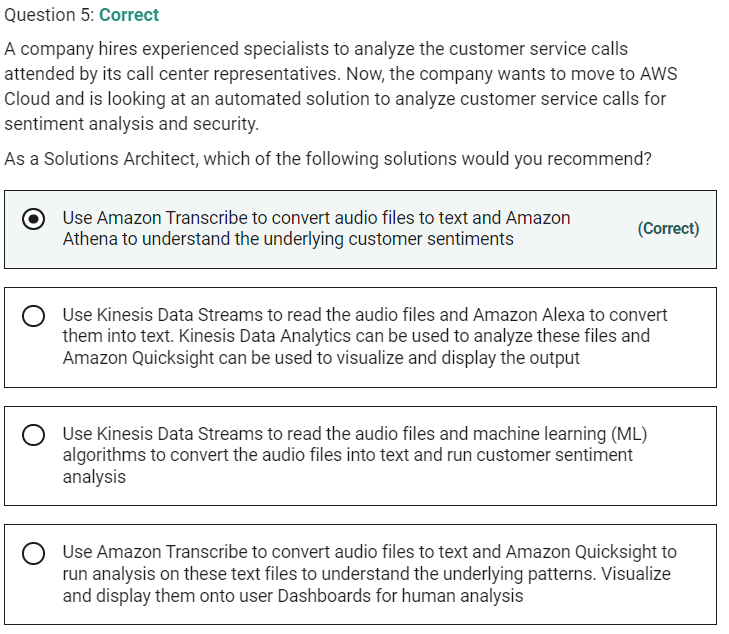
Practice Set-5

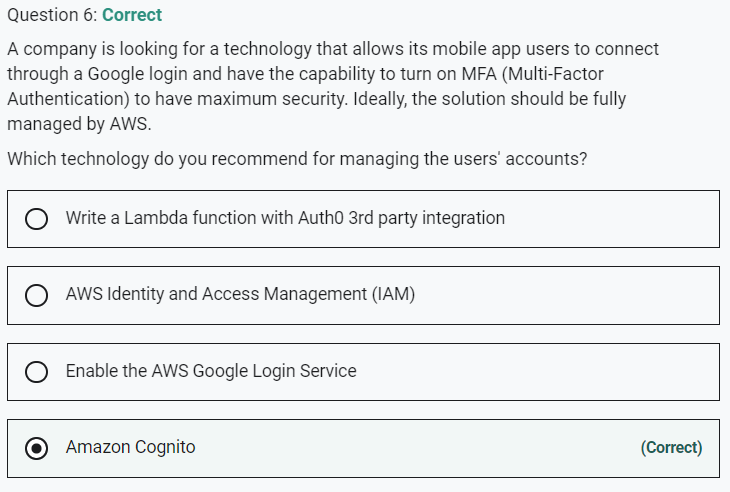


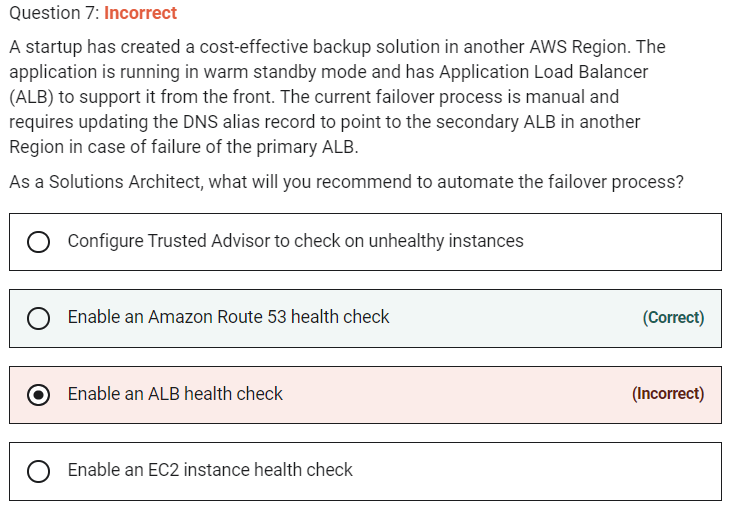




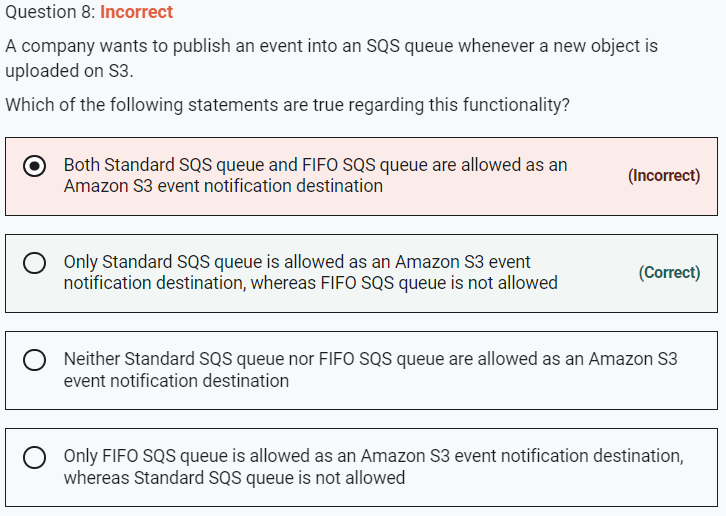




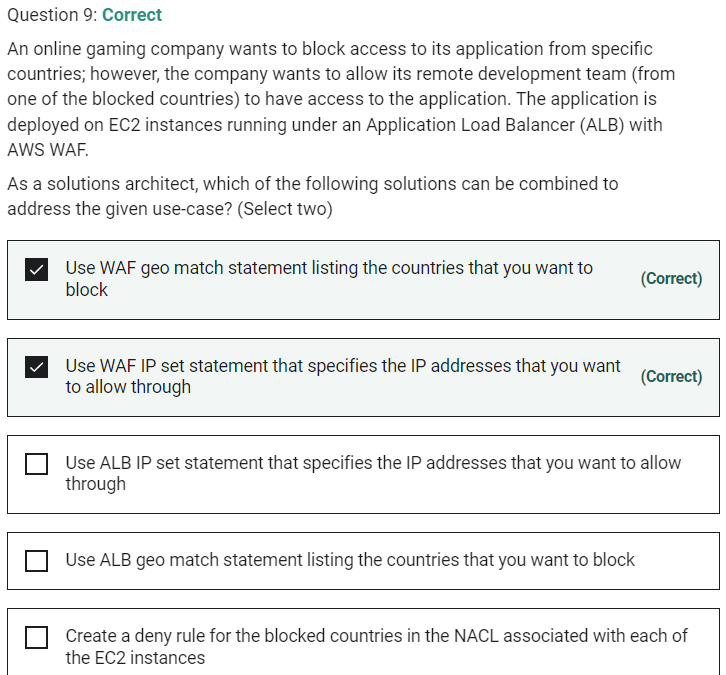


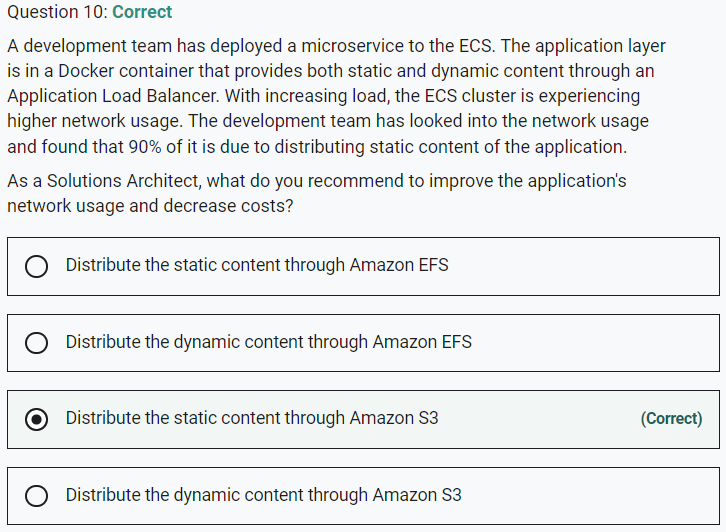


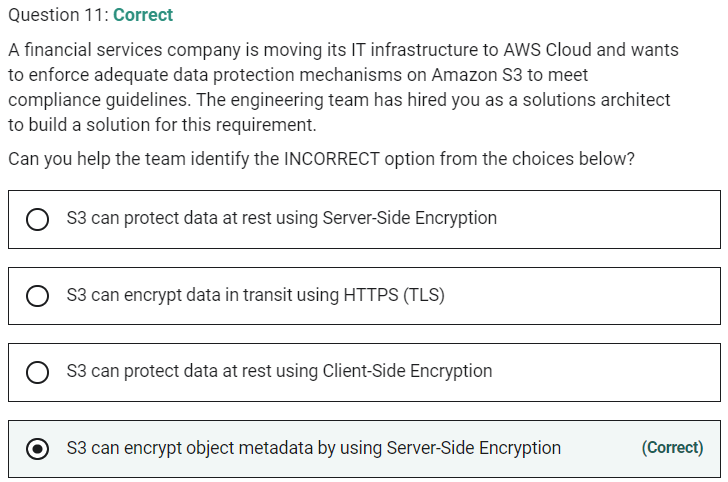
By combining the results of health checks of your EC2 instances and your ELBs, Route 53 DNS Failover can evaluate the health of the load balancer and the health of the application running on the EC2 instances behind it. In other words, if any part of the stack goes down, Route 53 detects the failure and routes traffic away from the failed endpoint.

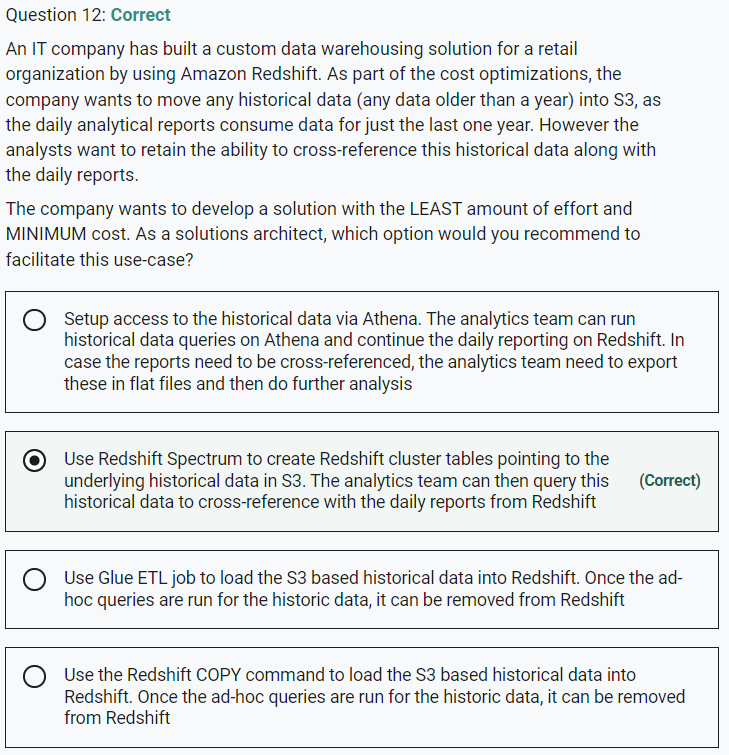


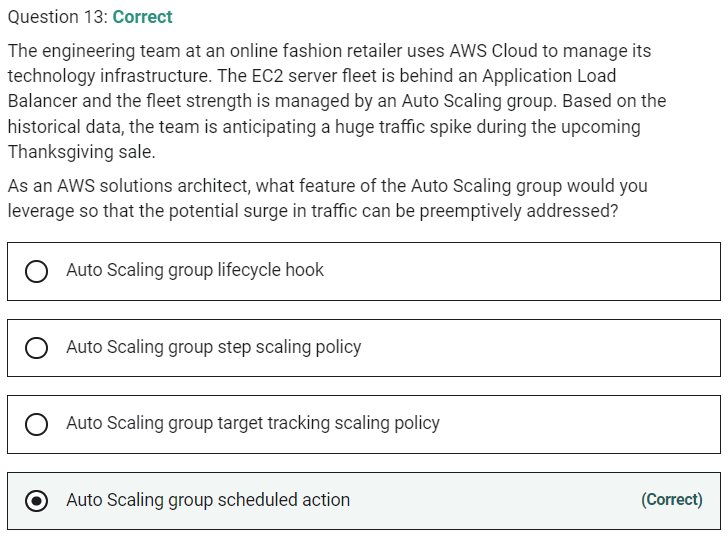
Currently, the Standard SQS queue is only allowed as an Amazon S3 event notification destination, whereas the FIFO SQS queue is not allowed.

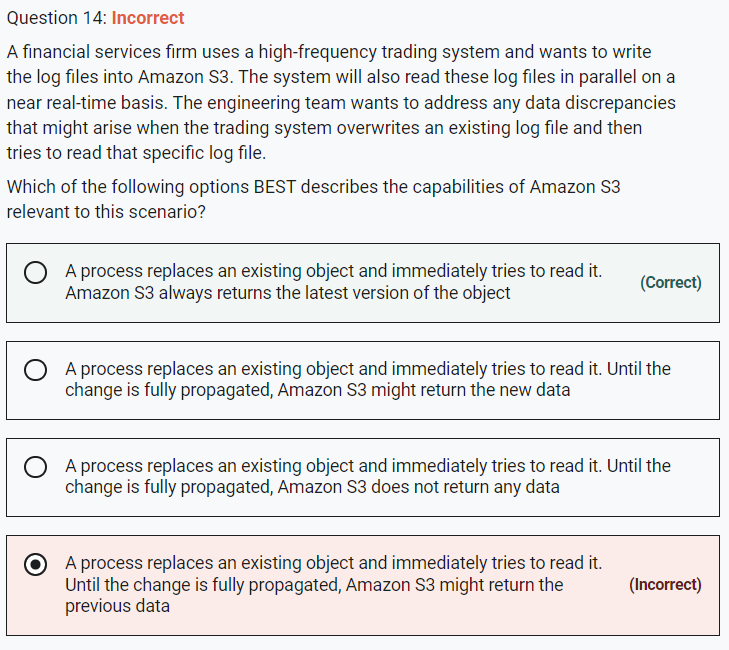












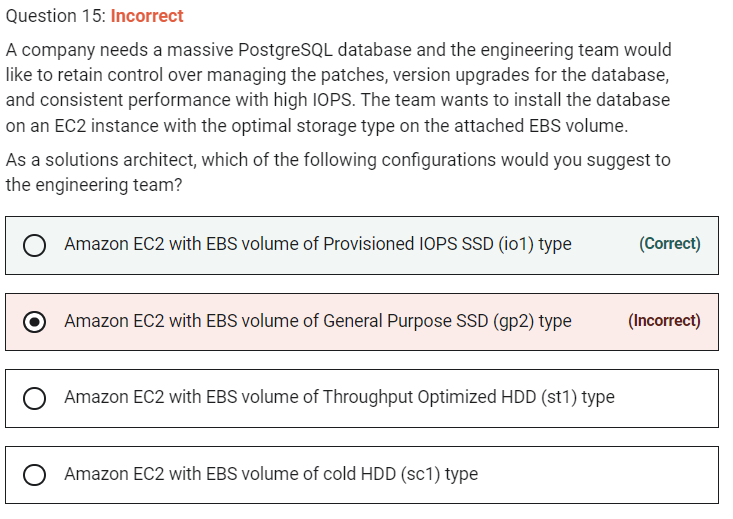
**A process replaces an existing object and immediately tries to read it. Amazon S3 always returns the latest version of the object**

Amazon S3 delivers strong read-after-write consistency automatically, without changes to performance or availability, without sacrificing regional isolation for applications, and at no additional cost.

After a successful write of a new object or an overwrite of an existing object, any subsequent read request immediately receives the latest version of the object. S3 also provides strong consistency for list operations, so after a write, you can immediately perform a listing of the objects in a bucket with any changes reflected.

Strong read-after-write consistency helps when you need to immediately read an object after a write. For example, strong read-after-write consistency when you often read and list immediately after writing objects.

To summarize, all S3 GET, PUT, and LIST operations, as well as operations that change object tags, ACLs, or metadata, are strongly consistent. What you write is what you will read, and the results of a LIST will be an accurate reflection of what’s in the bucket.



Provisioned IOPS is suitable for large database workloads, such as

MySQL

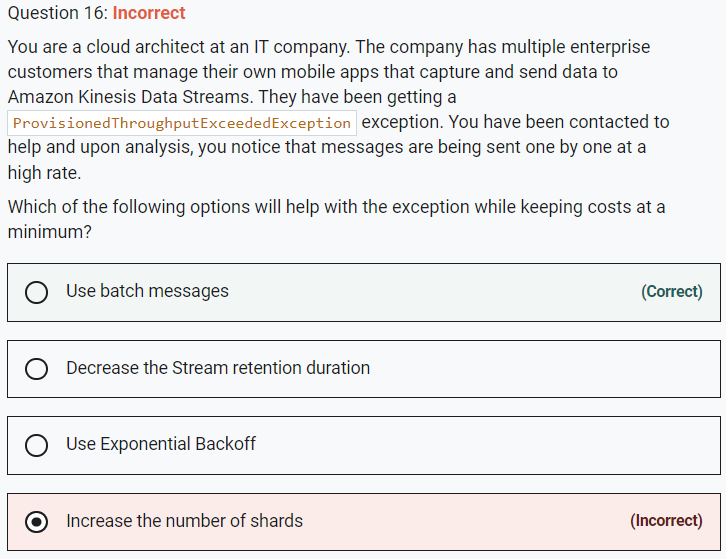
Microsoft SQL Server

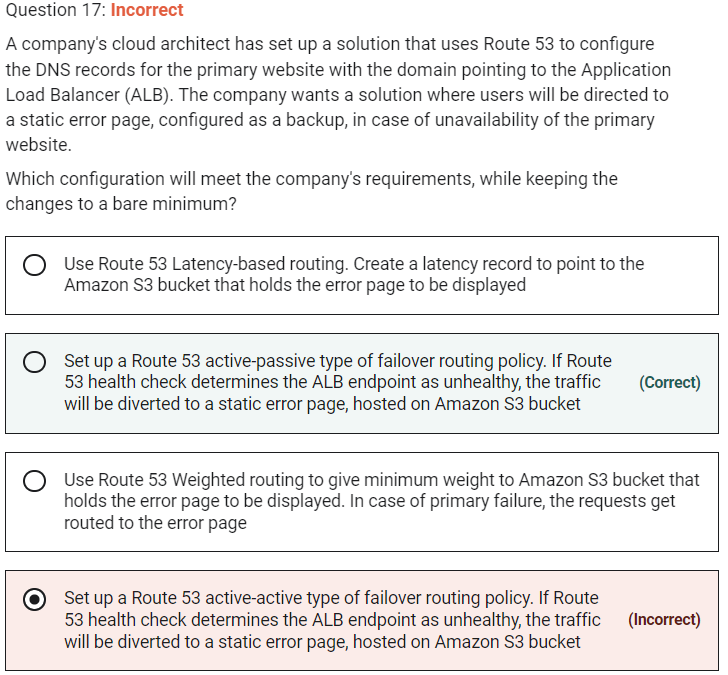
Oracle

MongoDB

PostgreSQL

Cassandra

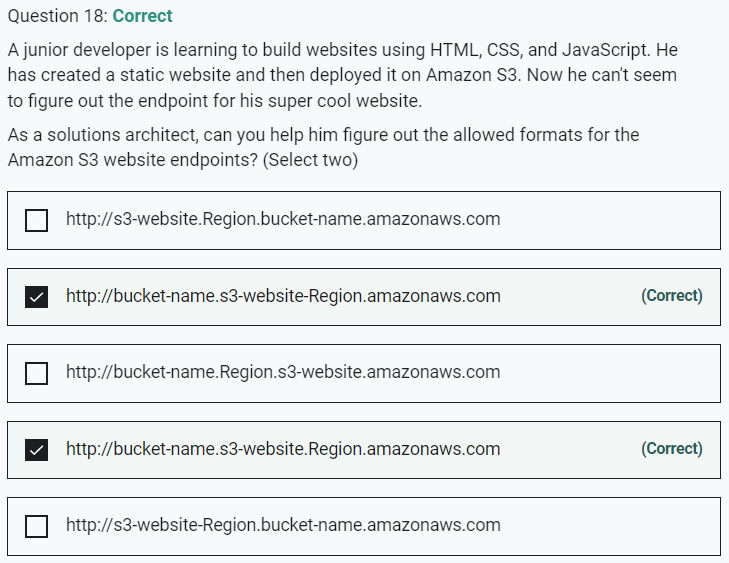


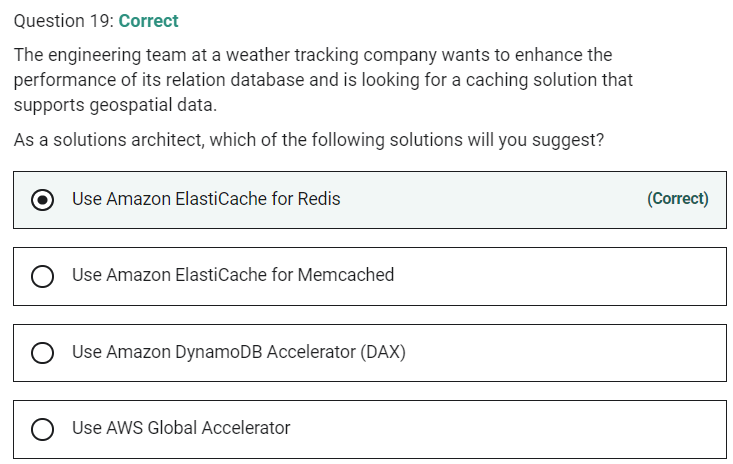


**Set up a Route 53 active-passive type of failover routing policy. If Route 53 health check determines the ALB endpoint as unhealthy, the traffic will be diverted to a static error page, hosted on Amazon S3 bucket**

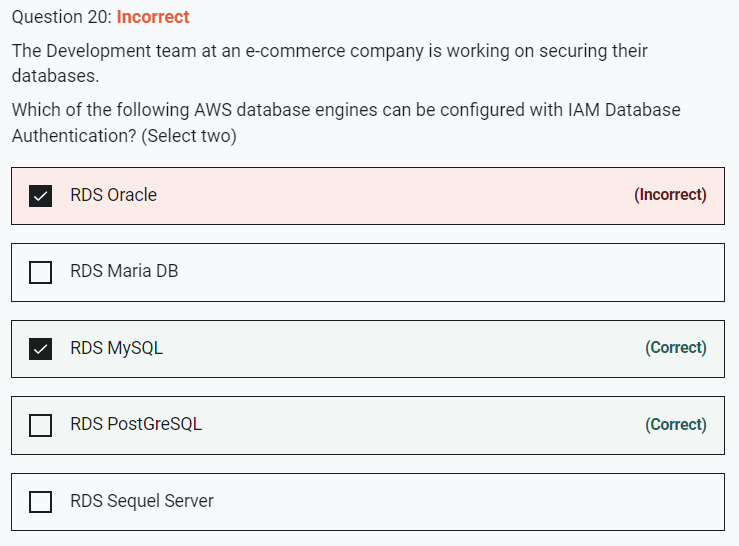
Use an active-passive failover configuration when you want a primary resource or group of resources to be available the majority of the time and you want a secondary resource or group of resources to be on standby in case all the primary resources become unavailable. When responding to queries, Route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

**Set up a Route 53 active-active type of failover routing policy. If Route 53 health check determines the ALB endpoint as unhealthy, the traffic will be diverted to a static error page, hosted on Amazon S3 bucket** - This option has been added as a distractor as there is no such thing as an active-active failover routing policy in Route 53. You can configure active-active failover using any routing policy (or combination of routing policies) other than failover routing policy and you configure active-passive failover only using the failover routing policy. In active-active failover configuration, all the records that have the same name, the same type (such as A or AAAA), and the same routing policy (such as weighted or latency) are active unless Route 53 considers them unhealthy. Route 53 can respond to a DNS query using any healthy record.





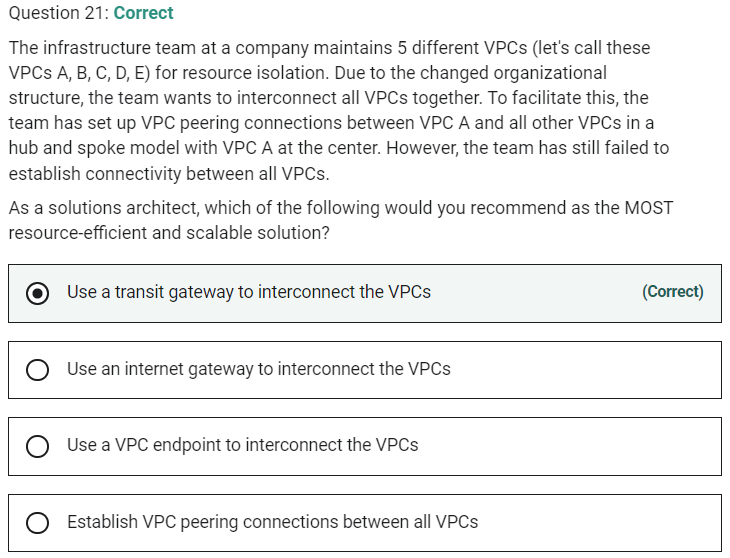
**Use Amazon ElastiCache for Memcached** - Both Redis and MemCached are in-memory, open-source data stores. Memcached, a high-performance distributed memory cache service, is designed for simplicity while Redis offers a rich set of features that make it effective for a wide range of use cases. Memcached does not offer support for geospatial data.

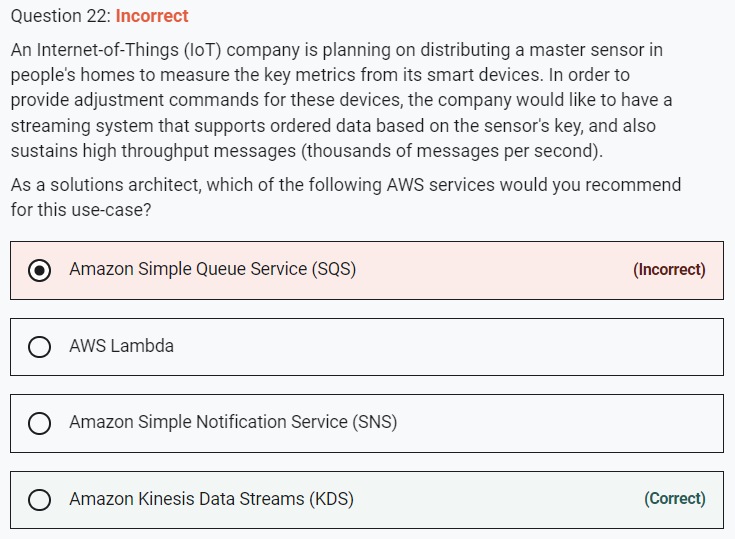


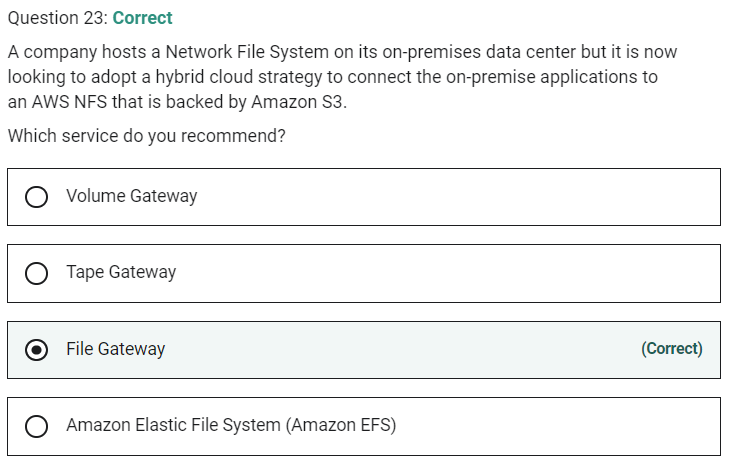
You can authenticate to your DB instance using AWS Identity and Access Management (IAM) database authentication. With this authentication method, you don't need to use a password when you connect to a DB instance. Instead, you use an authentication token. An authentication token is a unique string of characters that Amazon RDS generates on request. Each token has a lifetime of 15 minutes. You don't need to store user credentials in the database, because authentication is managed externally using IAM.

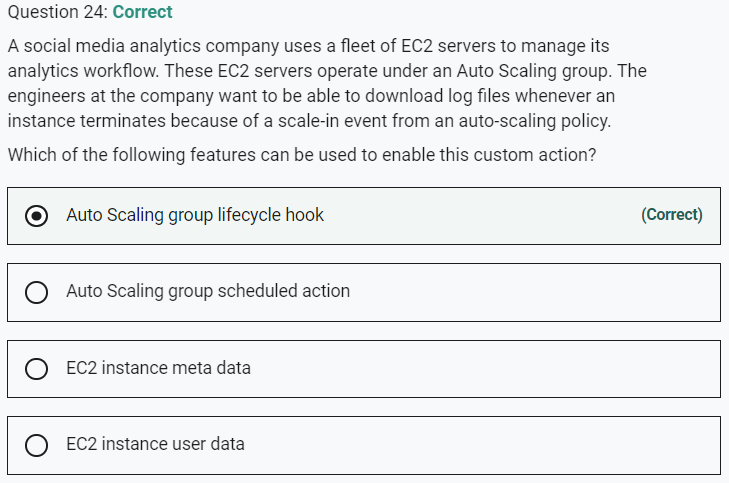
**RDS MySQL** - IAM database authentication works with MySQL and PostgreSQL.

**RDS PostGreSQL** - IAM database authentication works with MySQL and PostgreSQL.

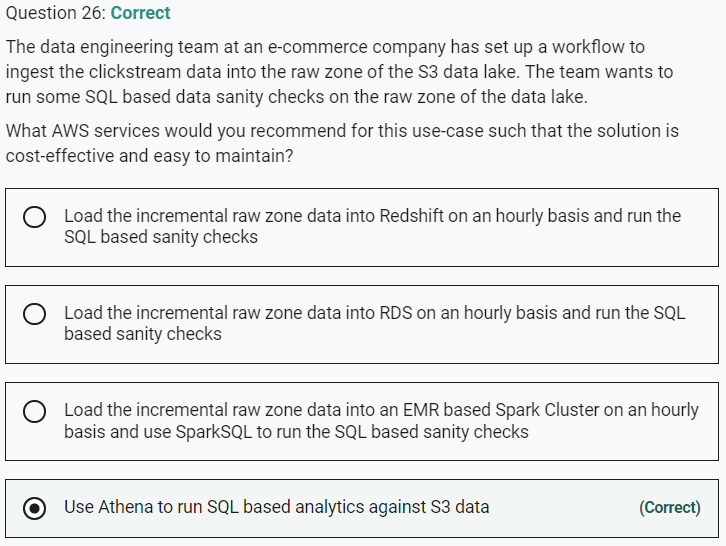


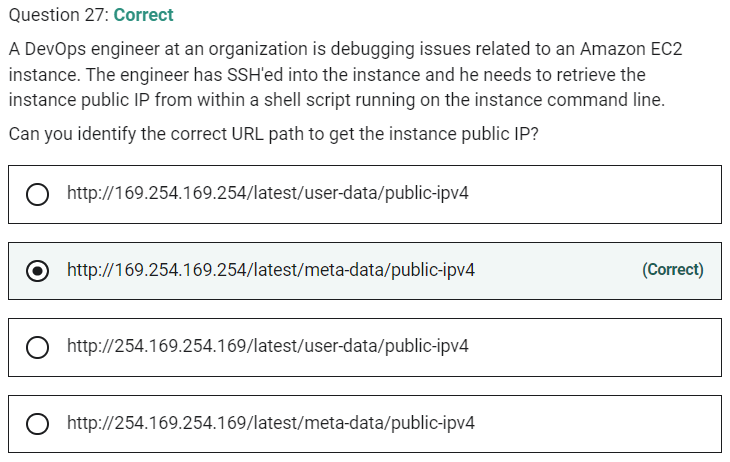


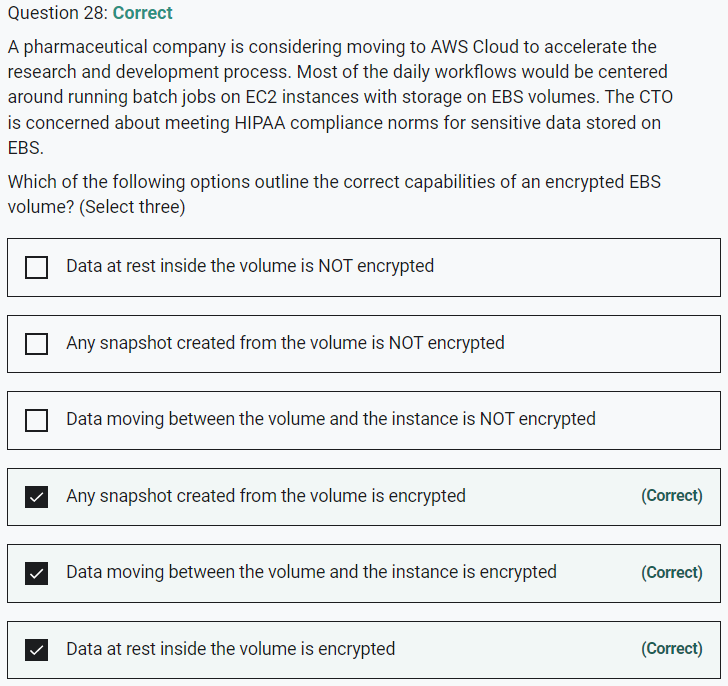


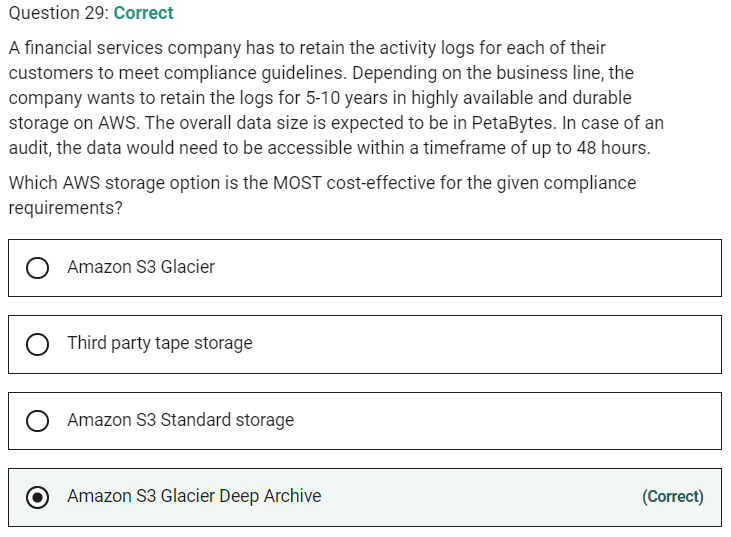


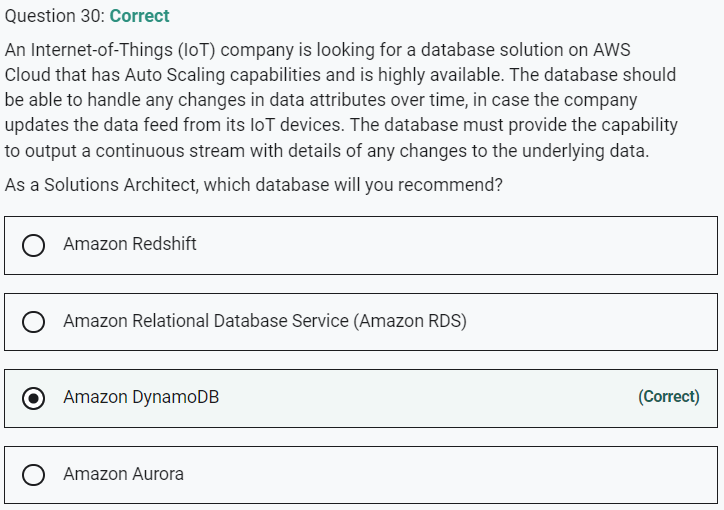


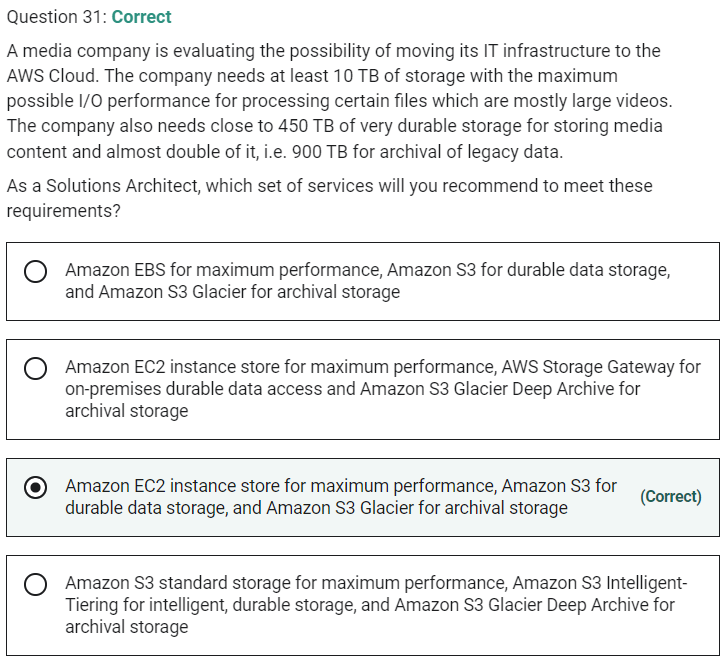


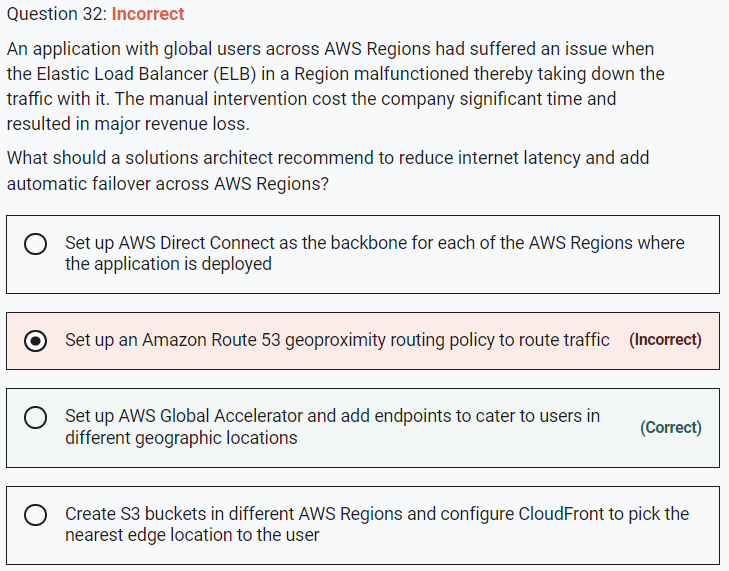










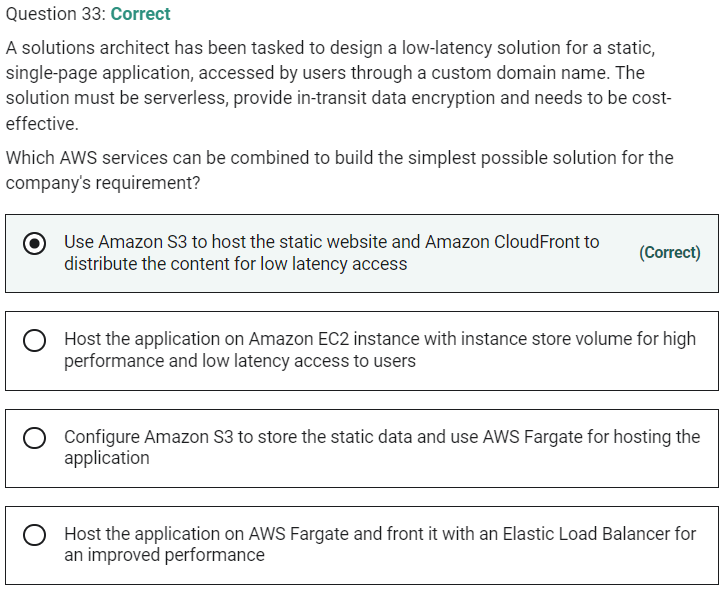


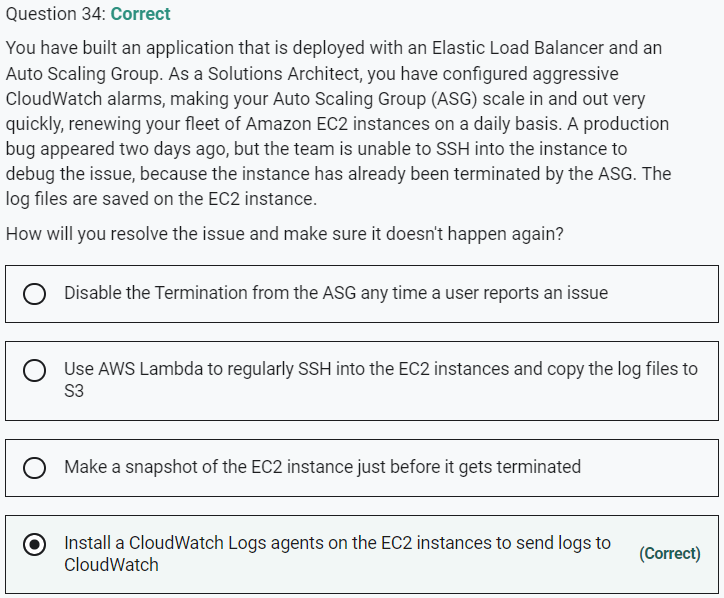
**Set up AWS Global Accelerator and add endpoints to cater to users in different geographic locations**

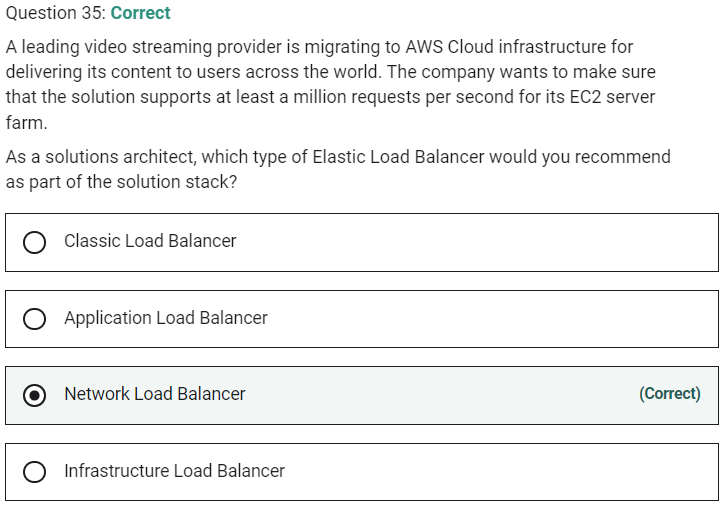
As your application architecture grows, so does the complexity, with longer user-facing IP lists and more nuanced traffic routing logic. AWS Global Accelerator solves this by providing you with two static IPs that are anycast from our globally distributed edge locations, giving you a single entry point to your application, regardless of how many AWS Regions it’s deployed in. This allows you to add or remove origins, Availability Zones or Regions without reducing your application availability. Your traffic routing is managed manually, or in console with endpoint traffic dials and weights. If your application endpoint has a failure or availability issue, AWS Global Accelerator will automatically redirect your new connections to a healthy endpoint within seconds.

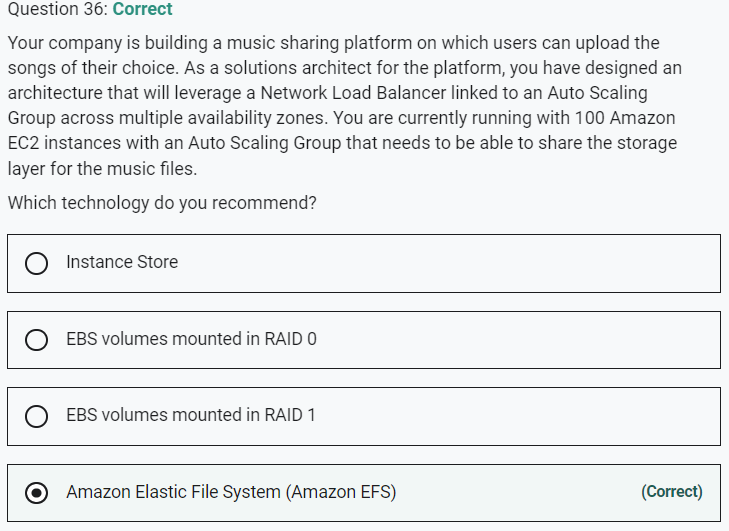
By using AWS Global Accelerator, you can:

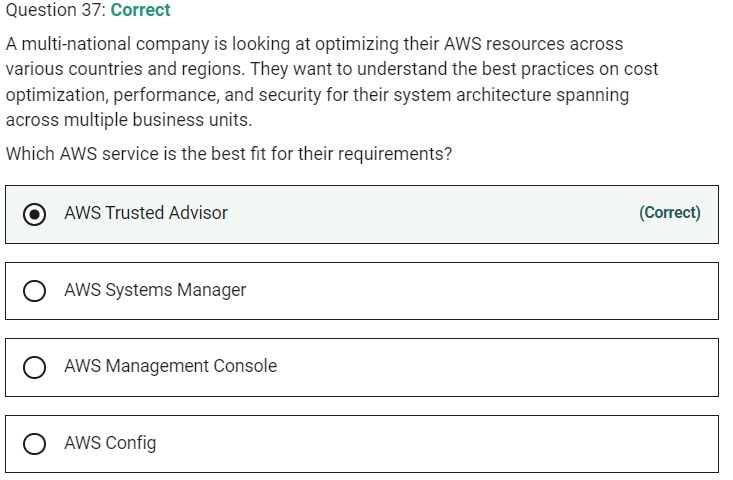
1. Associate the static IP addresses provided by AWS Global Accelerator to regional AWS resources or endpoints, such as Network Load Balancers, Application Load Balancers, EC2 Instances, and Elastic IP addresses. The IP addresses are anycast from AWS edge locations so they provide onboarding to the AWS global network close to your users.

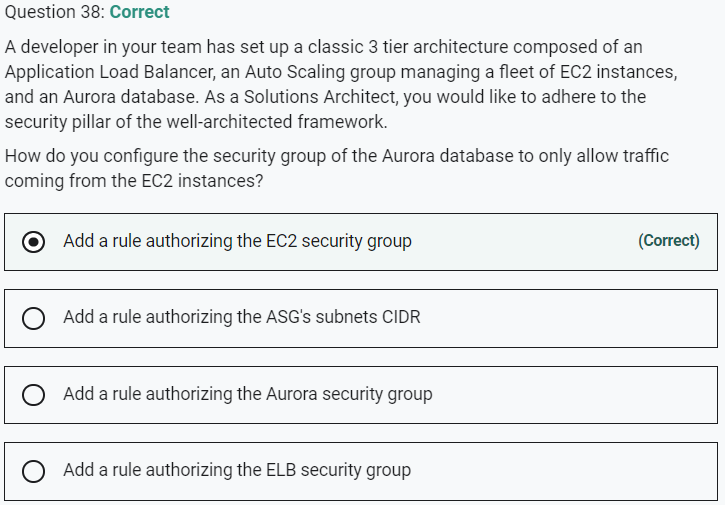


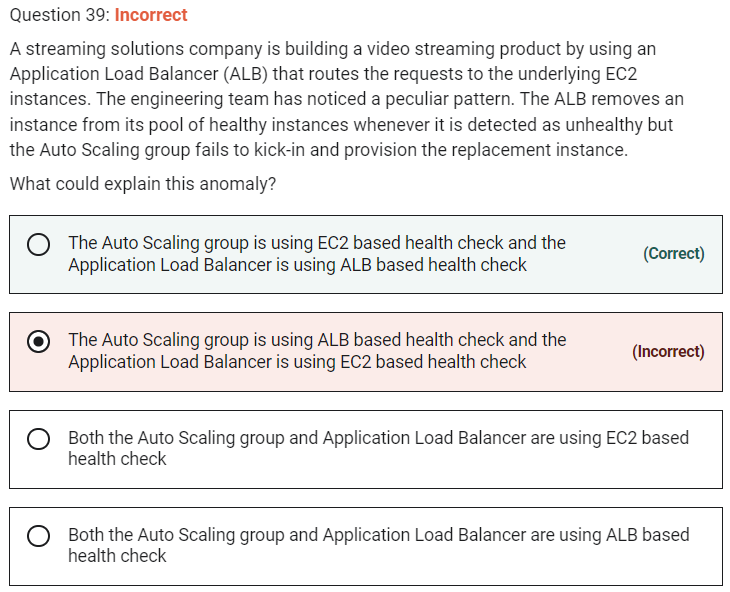










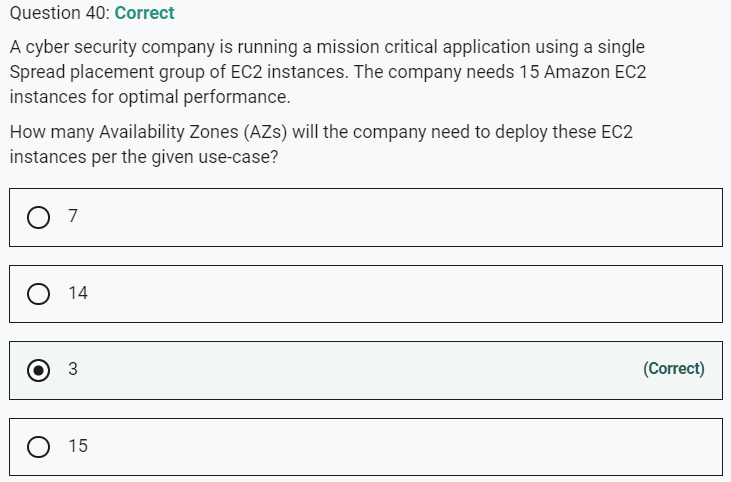


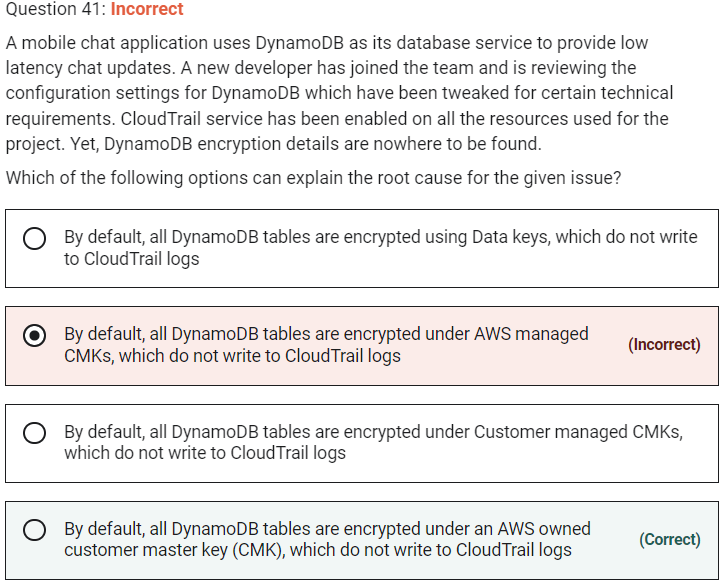
**The Auto Scaling group is using EC2 based health check and the Application Load Balancer is using ALB based health check**.

If the Auto Scaling group (ASG) is using EC2 as the health check type and the Application Load Balancer (ALB) is using its in-built health check, there may be a situation where the ALB health check fails because the health check pings fail to receive a response from the instance. At the same time, ASG health check can come back as successful because it is based on EC2 based health check. Therefore, in this scenario, the ALB will remove the instance from its inventory, however, the ASG will fail to provide the replacement instance. This can lead to the scaling issues mentioned in the problem statement.

ALB cannot use EC2 based health checks, so this option is incorrect.

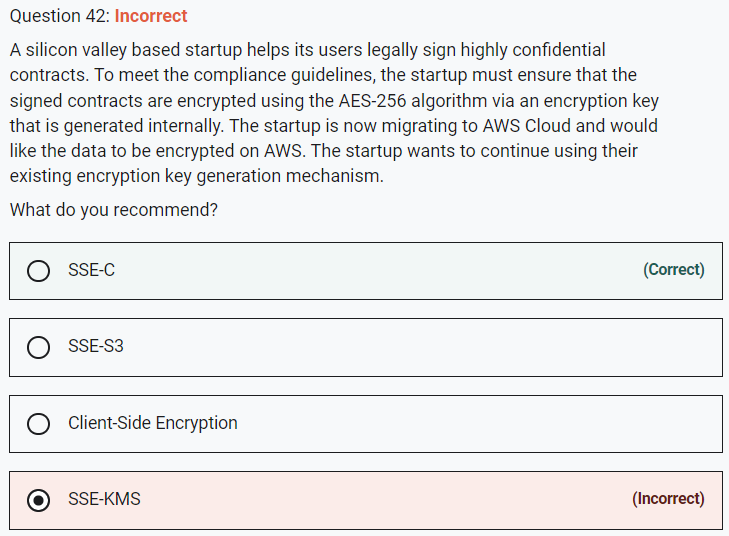
**Both the Auto Scaling group and Application Load Balancer are using ALB based health check** - It is recommended to use ALB based health checks for both Auto Scaling group and Application Load Balancer. If both the Auto Scaling group and Application Load Balancer use ALB based health checks, then you will be able to avoid the scenario mentioned in the question.





**By default, all DynamoDB tables are encrypted under an AWS owned customer master key (CMK), which do not write to CloudTrail logs** - AWS owned CMKs are a collection of CMKs that an AWS service owns and manages for use in multiple AWS accounts. Although AWS owned CMKs are not in your AWS account, an AWS service can use its AWS owned CMKs to protect the resources in your account.

You do not need to create or manage the AWS owned CMKs. However, you cannot view, use, track, or audit them. You are not charged a monthly fee or usage fee for AWS owned CMKs and they do not count against the AWS KMS quotas for your account.

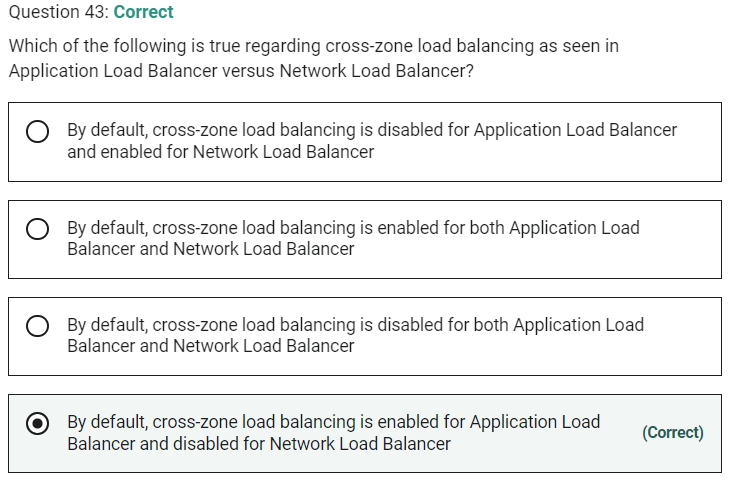


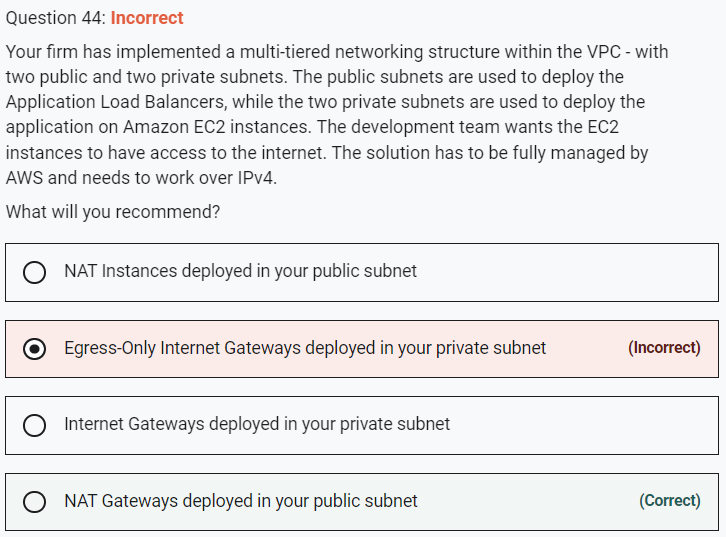
**SSE-C** - With Server-Side Encryption with Customer-Provided Keys (SSE-C), you manage the encryption keys and Amazon S3 manages the encryption, as it writes to disks, and decryption when you access your objects. With SSE-C, the startup can still provide the encryption key but let AWS do the encryption. Therefore, this is the correct option.

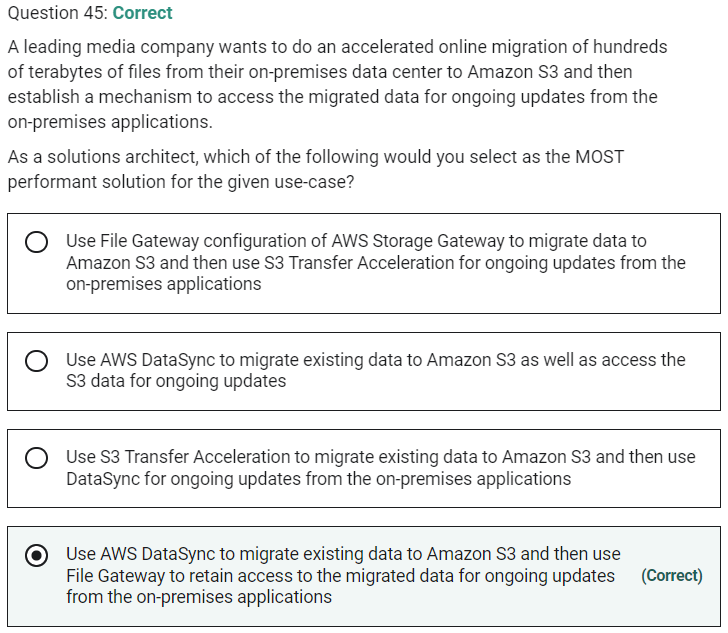
**SSE-KMS** - AWS Key Management Service (AWS KMS) is a service that combines secure, highly available hardware and software to provide a key management system scaled for the cloud. When you use server-side encryption with AWS KMS (SSE-KMS), you can specify a customer-managed CMK that you have already created. But, you never get to know the actual key here.

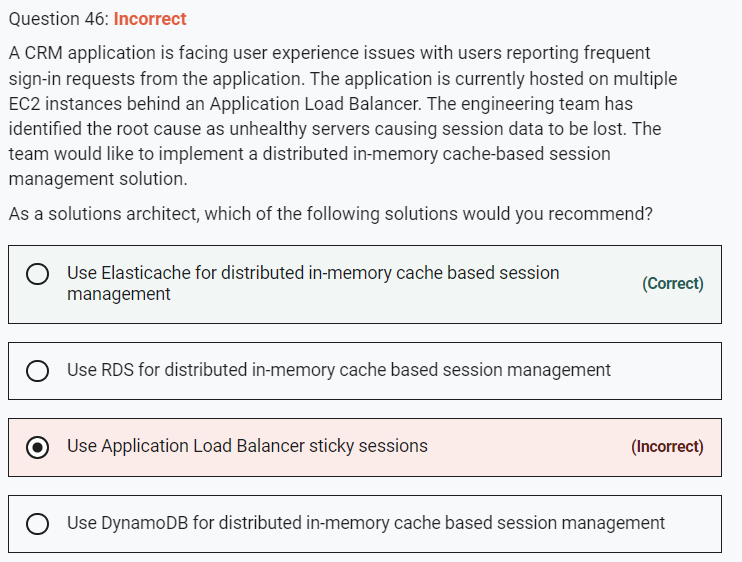
**SSE-S3** - When you use Server-Side Encryption with Amazon S3-Managed Keys (SSE-S3), each object is encrypted with a unique key. However, this option does not provide the ability to audit trail the usage of the encryption keys.

**Client-Side Encryption** - Client-side encryption is the act of encrypting data before sending it to Amazon S3. To enable client-side encryption, you have the following options: Use a customer master key (CMK) stored in AWS Key Management Service (AWS KMS), Use a master key you store within your application. Since the customer wants to use AWS provided facility, this is not an option.

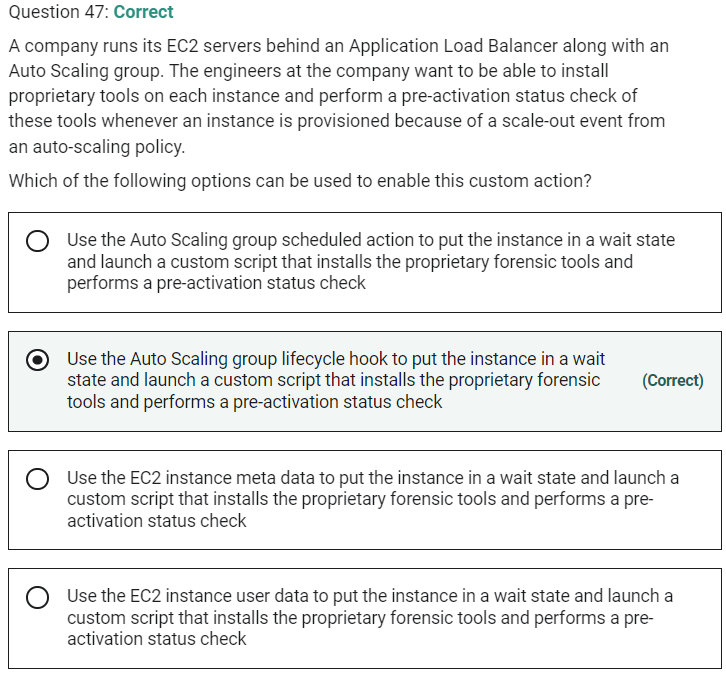


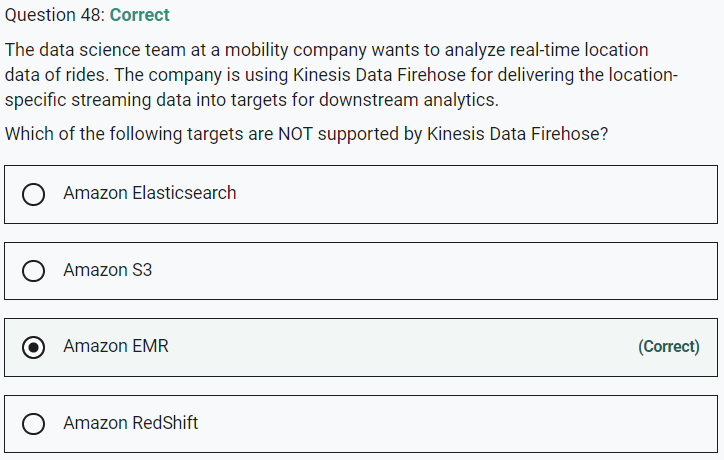






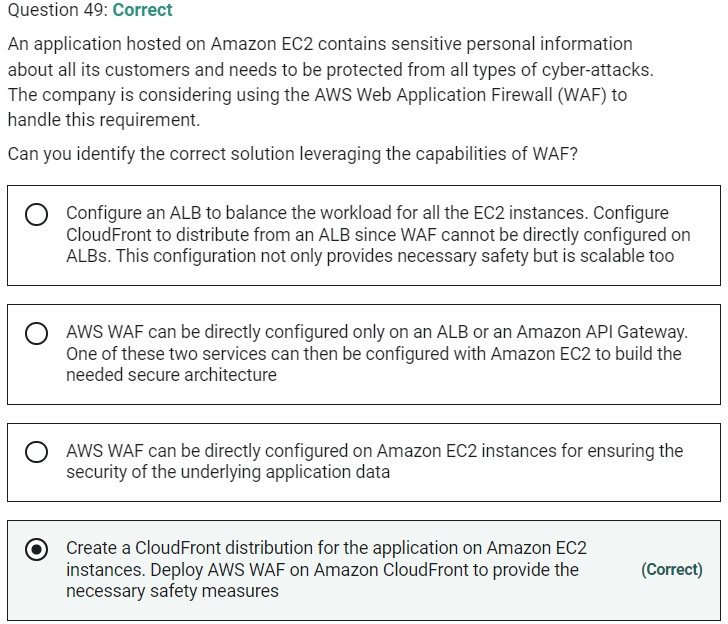
**Use DynamoDB for distributed in-memory cache-based session management** - Amazon DynamoDB is a key-value and document database that delivers single-digit millisecond performance at any scale. DynamoDB is a NoSQL database and is not the right fit for a distributed in-memory cache-based session management solution.





**Amazon EMR**

You can use Amazon Kinesis Data Firehose to load streaming data into data lakes, data stores, and analytics tools. It can capture, transform, and load streaming data into Amazon S3, Amazon Redshift, Amazon Elasticsearch Service, and Splunk.

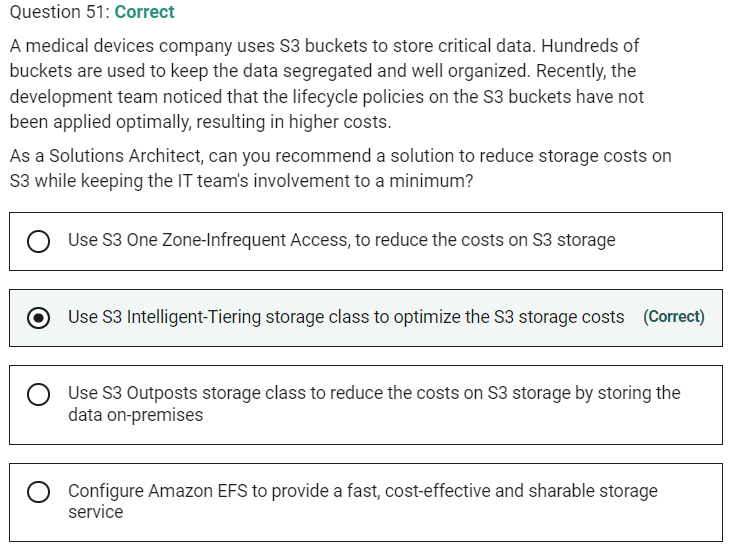


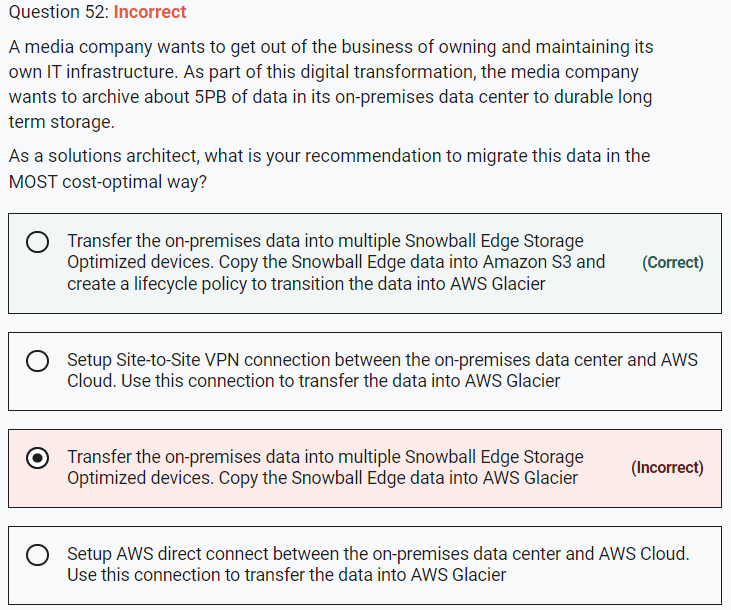
AWS WAF can be deployed on Amazon CloudFront, the Application Load Balancer (ALB), and Amazon API Gateway

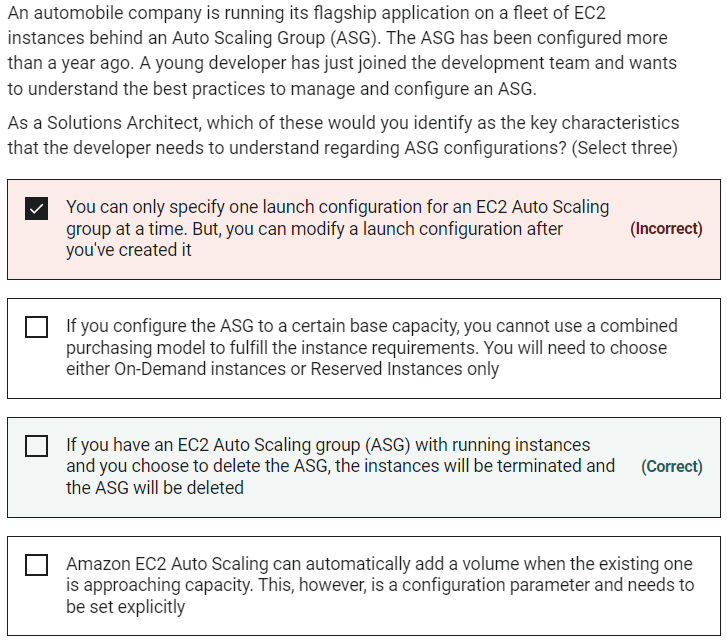


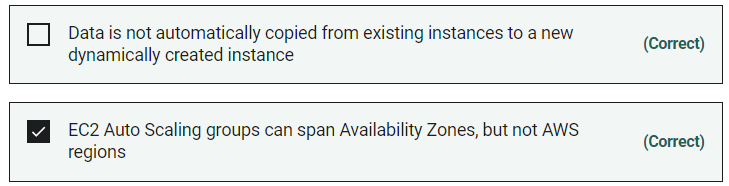
A launch configuration is an instance configuration template that an Auto Scaling group uses to launch EC2 instances. When you create a launch configuration, you specify information for the instances. Include the ID of the Amazon Machine Image (AMI), the instance type, a key pair, one or more security groups, and a block device mapping.

It is not possible to modify a launch configuration once it is created. The correct option is to create a new launch configuration to use the correct instance type. Then modify the Auto Scaling group to use this new launch configuration. Lastly to clean-up, just delete the old launch configuration as it is no longer needed.







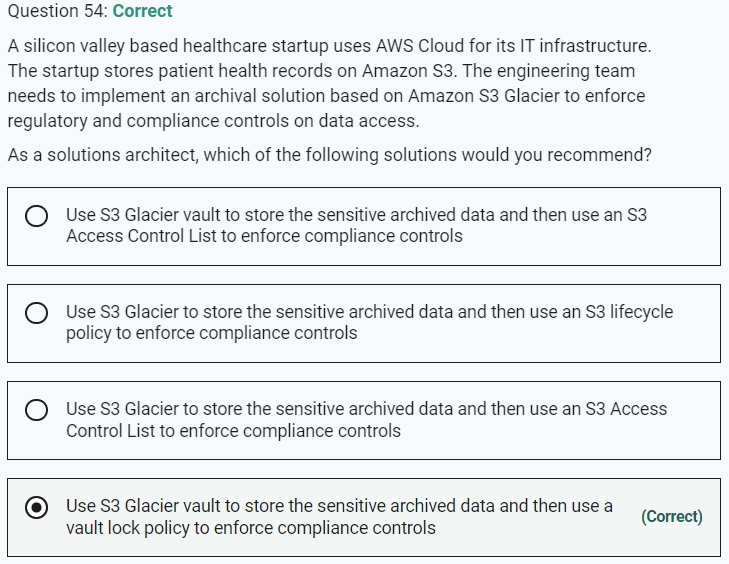


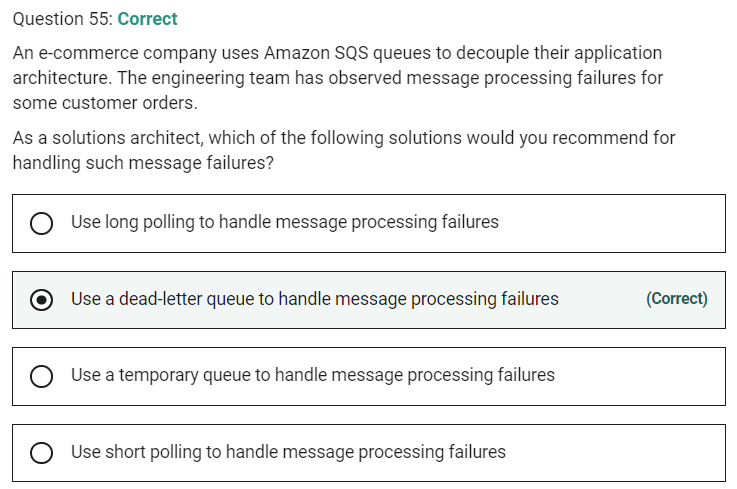
**Data is not automatically copied from existing instances to a new dynamically created instance** - Data is not automatically copied from existing instances to new instances. You can use lifecycle hooks to copy the data.

**If you configure the ASG to a certain base capacity, you cannot use a combined purchasing model to fulfill the instance requirements. You will need to choose either On-Demand instances or Reserved Instances only** - When setting up an ASG with a combined purchasing model, you can specify the base capacity of the group to be fulfilled by On-Demand instances. As the ASG scales in or scales out, EC2 Auto Scaling ensures the base capacity is fulfilled using On-Demand instances and anything beyond that be fulfilled with either only Spot instances or a specified percentage mix of On-Demand or Spot instances.

**Amazon EC2 Auto Scaling can automatically add a volume when the existing one is approaching capacity. This, however, is a configuration parameter and needs to be set explicitly** - Amazon EC2 Auto Scaling doesn't automatically add a volume when the existing one is approaching capacity. You can use the EC2 API to add a volume to an existing instance.

**You can only specify one launch configuration for an EC2 Auto Scaling group at a time. But, you can modify a launch configuration after you've created it** - You can only specify one launch configuration for an EC2 Auto Scaling group at a time, and you can't modify a launch configuration after you've created it.

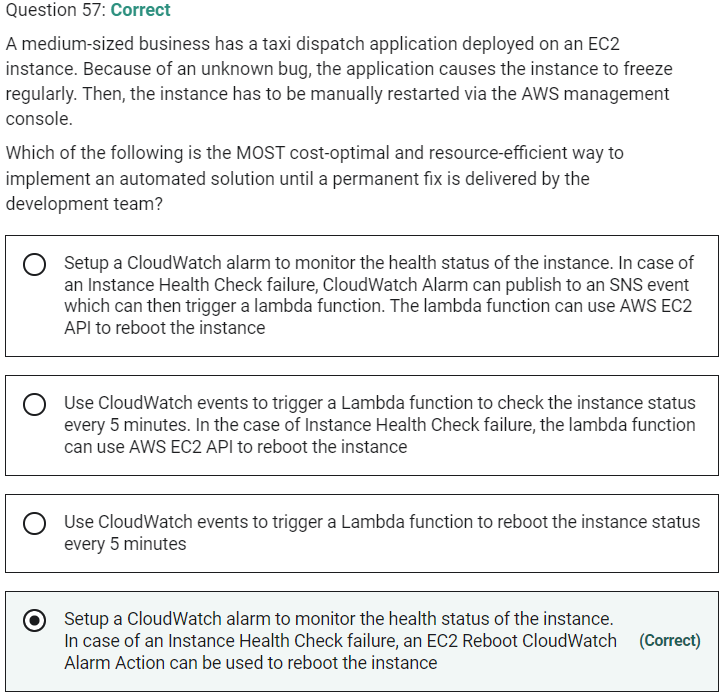


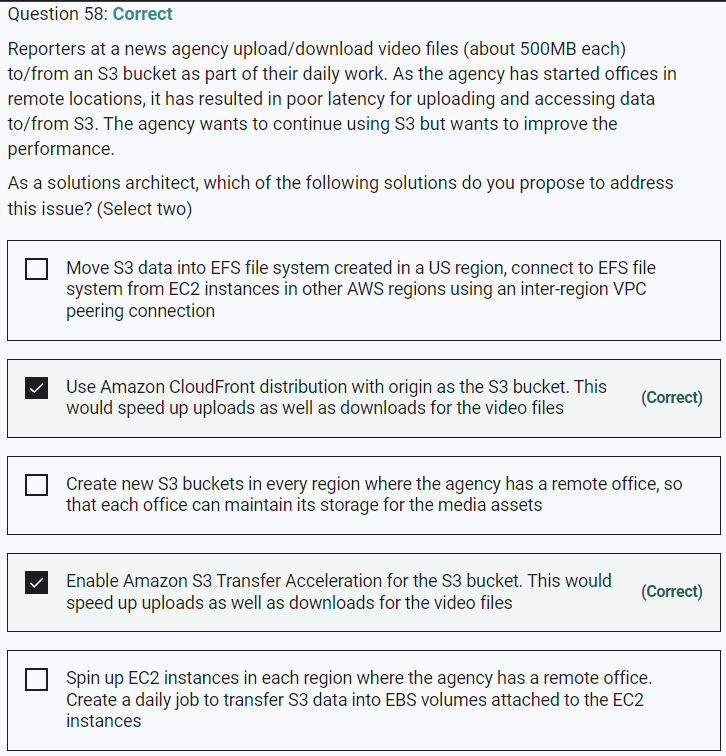


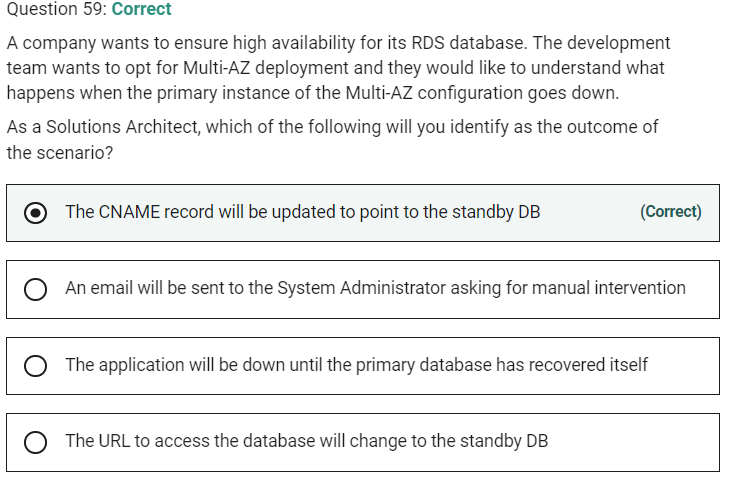


**ElastiCache for Memcached** - Amazon ElastiCache for Memcached is a Memcached-compatible in-memory key-value store service that can be used as a cache or a data store. Amazon ElastiCache for Memcached is a great choice for implementing an in-memory cache to decrease access latency, increase throughput, and ease the load off your relational or NoSQL database. Session stores are easy to create with Amazon ElastiCache for Memcached. Elasticache for Memcached is not HIPAA eligible, so this option is incorrect.

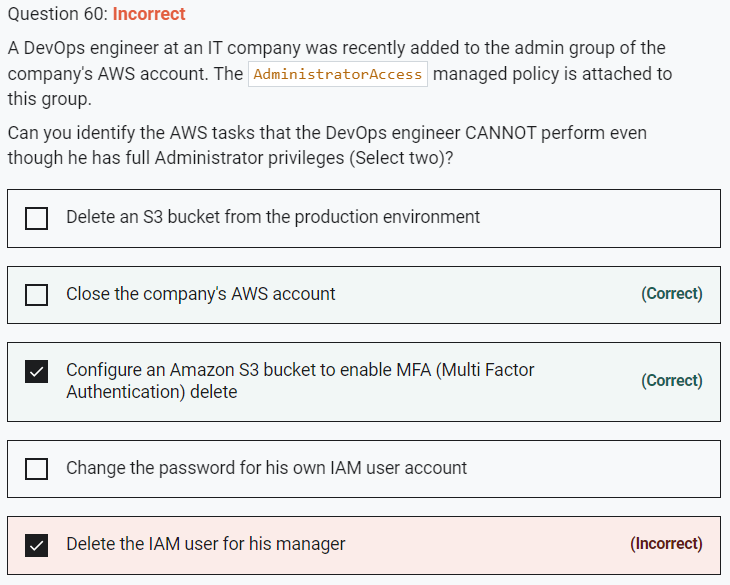
**DynamoDB** - Amazon DynamoDB is a key-value and document database that delivers single-digit millisecond performance at any scale. It's a fully managed, multi-region, multi-master, durable database with built-in security, backup and restore, and in-memory caching (via DAX) for internet-scale applications. DynamoDB is not an in-memory database, so this option is incorrect.

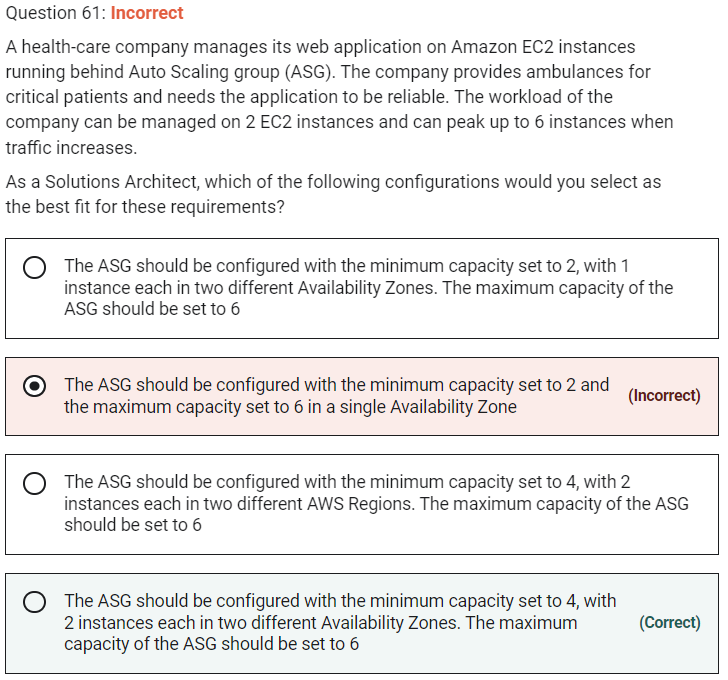


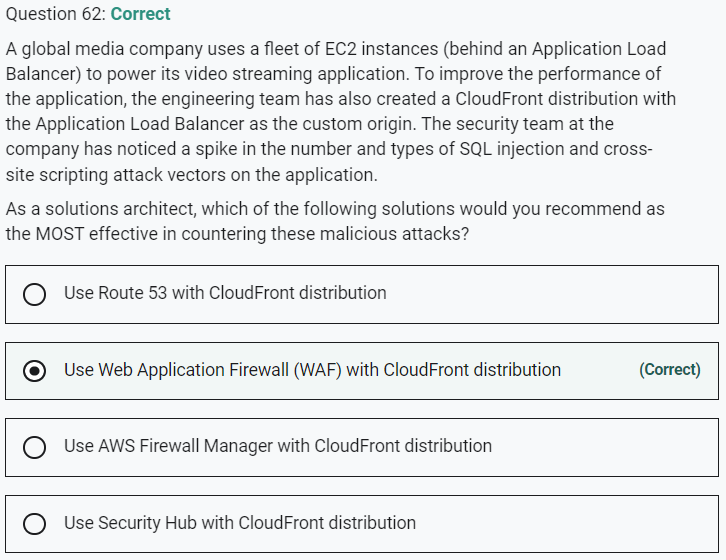


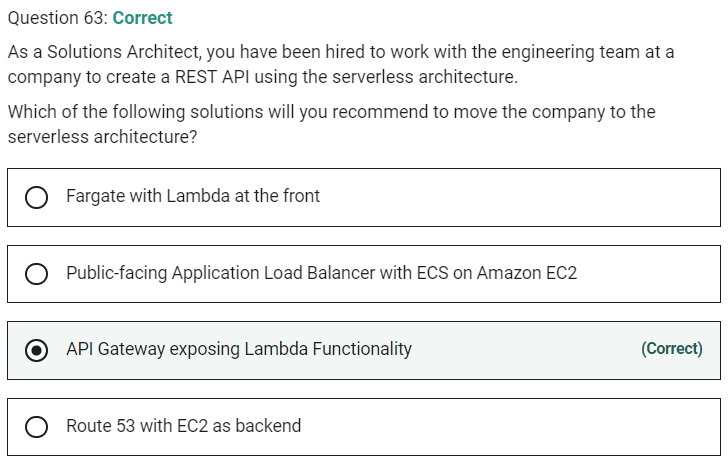


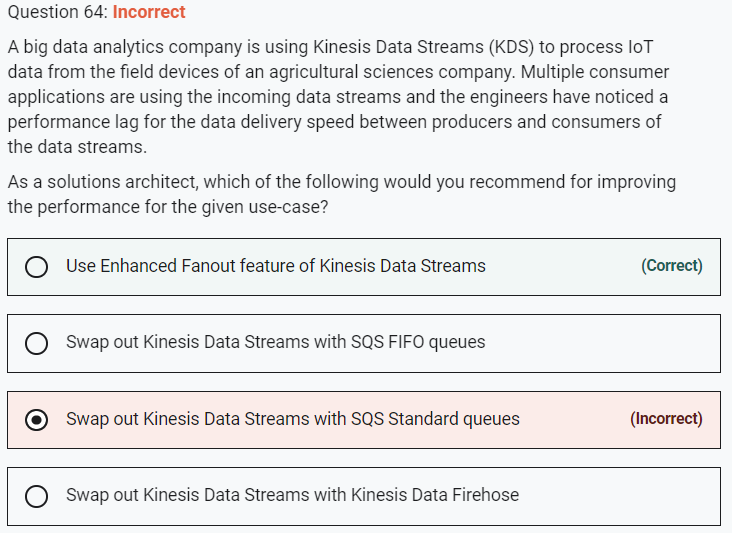
Multi-AZ means the URL is the same, the failover is automated, and the CNAME will automatically be updated to point to the standby database.











**Use Enhanced Fanout feature of Kinesis Data Streams**

Amazon Kinesis Data Streams (KDS) is a massively scalable and durable real-time data streaming service. KDS can continuously capture gigabytes of data per second from hundreds of thousands of sources such as website clickstreams, database event streams, financial transactions, social media feeds, IT logs, and location-tracking events.

By default, the 2MB/second/shard output is shared between all of the applications consuming data from the stream. You should use enhanced fan-out if you have multiple consumers retrieving data from a stream in parallel.

