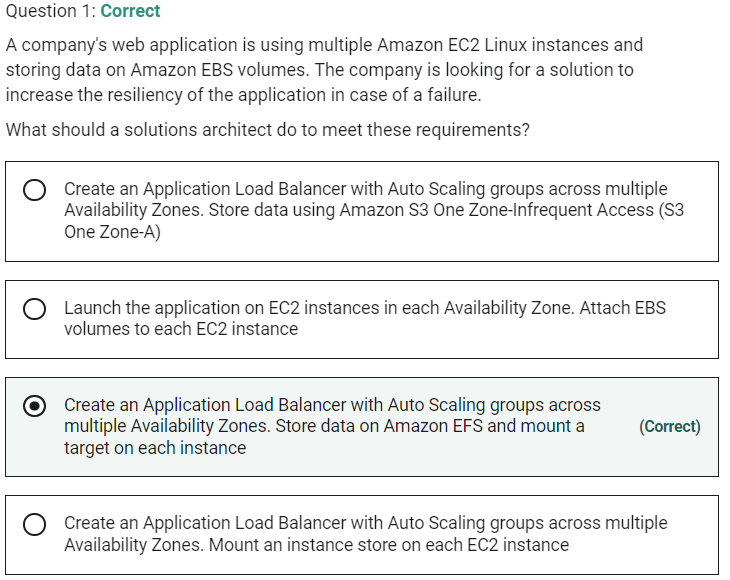
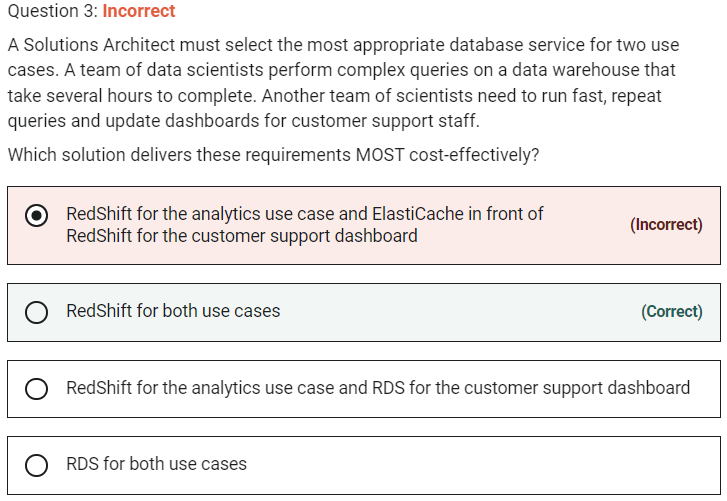
SAA-CO2

Practice Test-8







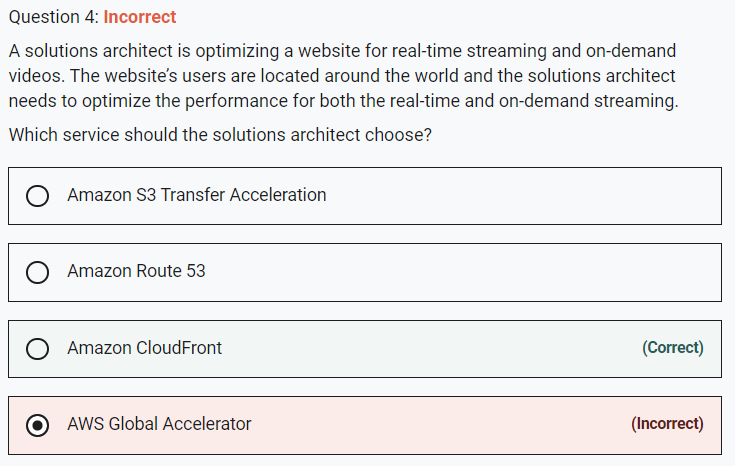
RedShift is a columnar data warehouse DB that is ideal for running long complex queries. RedShift can also improve performance for repeat queries by caching the result and returning the cached result when queries are re-run. Dashboard, visualization, and business intelligence (BI) tools that execute repeat queries see a significant boost in performance due to result caching.

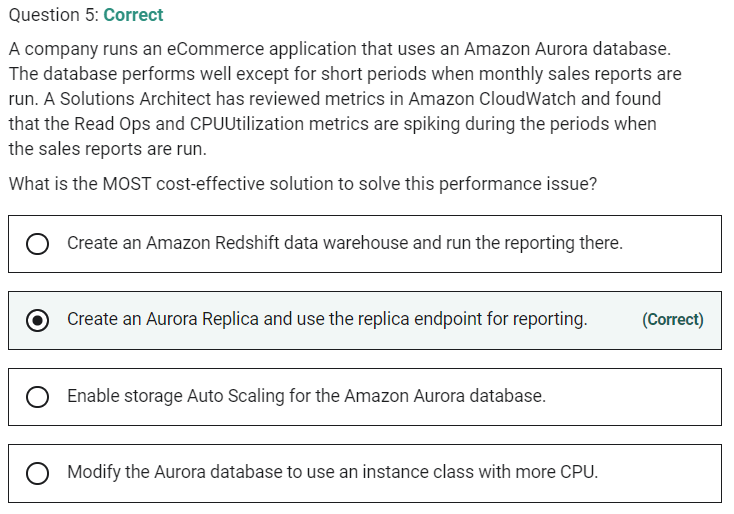
**CORRECT:**"RedShift for both use cases" is the correct answer.

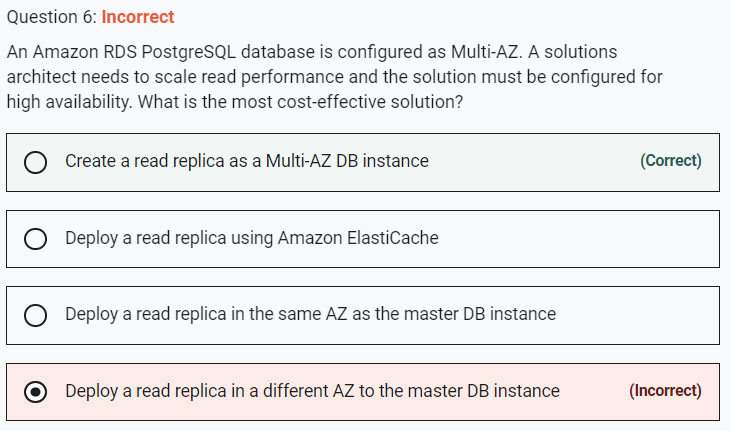
**INCORRECT:** "RDS for both use cases" is incorrect. RDS may be a good fit for the fast queries (not for the complex queries) but you now have multiple DBs to manage and multiple sets of data which is not going to be cost-effective.

**INCORRECT:** "RedShift for the analytics use case and ElastiCache in front of RedShift for the customer support dashboard" is incorrect. You could put ElastiCache in front of the RedShift DB and this would provide good performance for the fast, repeat queries. However, it is not essential and would add cost to the solution so is not the most cost-effective option available.

**INCORRECT:** "RedShift for the analytics use case and RDS for the customer support dashboard" is incorrect as RedShift is a better fit for both use cases.







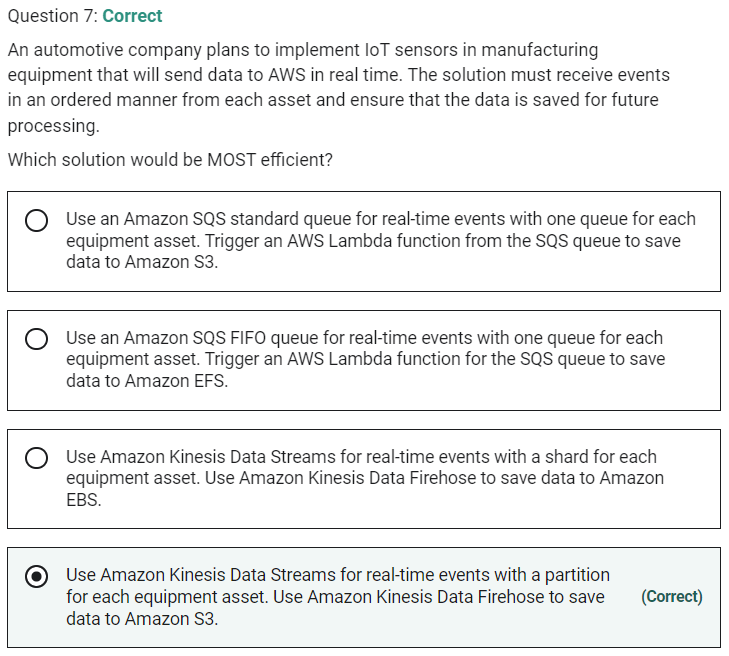
You can create a read replica as a Multi-AZ DB instance. Amazon RDS creates a standby of your replica in another Availability Zone for failover support for the replica. Creating your read replica as a Multi-AZ DB instance is independent of whether the source database is a Multi-AZ DB instance.

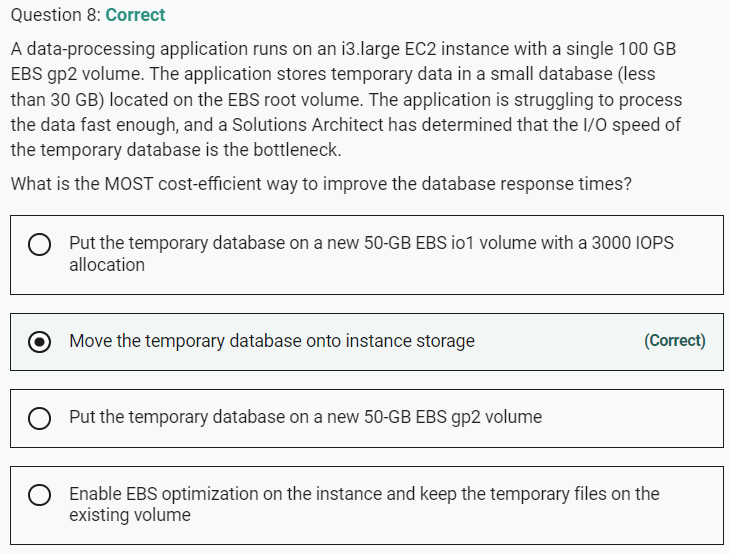
**CORRECT:**"Create a read replica as a Multi-AZ DB instance" is the correct answer.

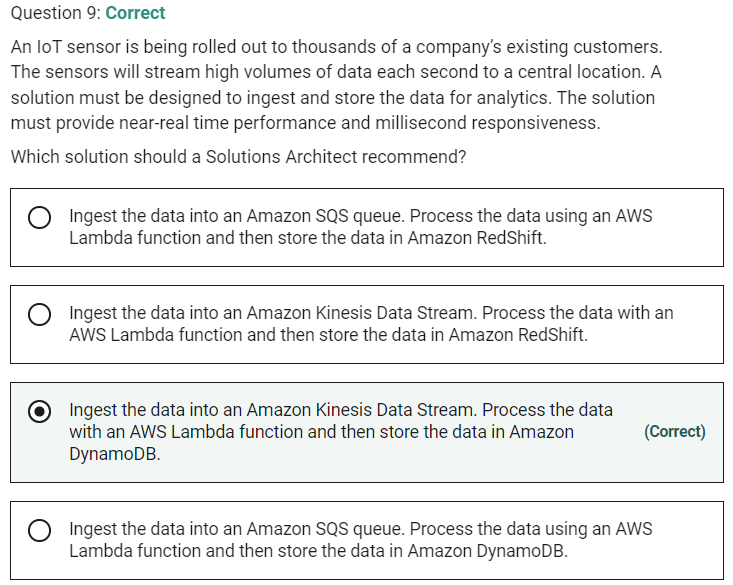
**INCORRECT:** "Deploy a read replica in a different AZ to the master DB instance" is incorrect as this does not provide high availability for the read replica

**INCORRECT:** "Deploy a read replica using Amazon ElastiCache" is incorrect as ElastiCache is not used to create read replicas of RDS database.

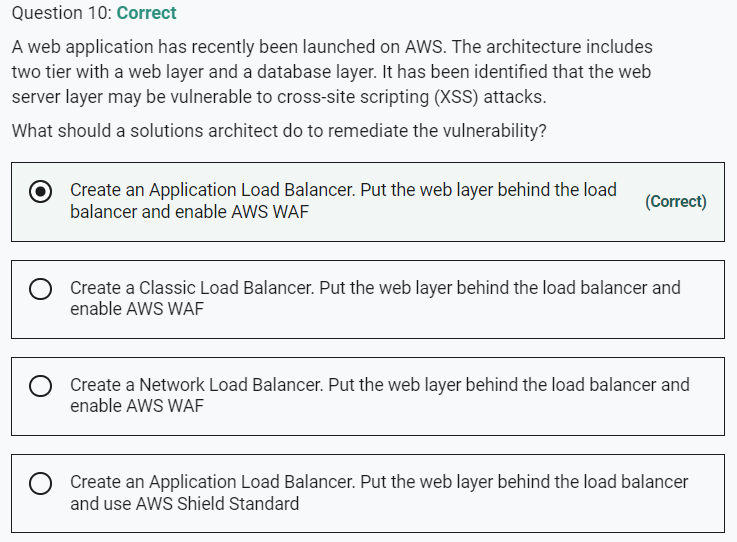
**INCORRECT:** "Deploy a read replica in the same AZ as the master DB instance" is incorrect as this solution does not include HA for the read replica.

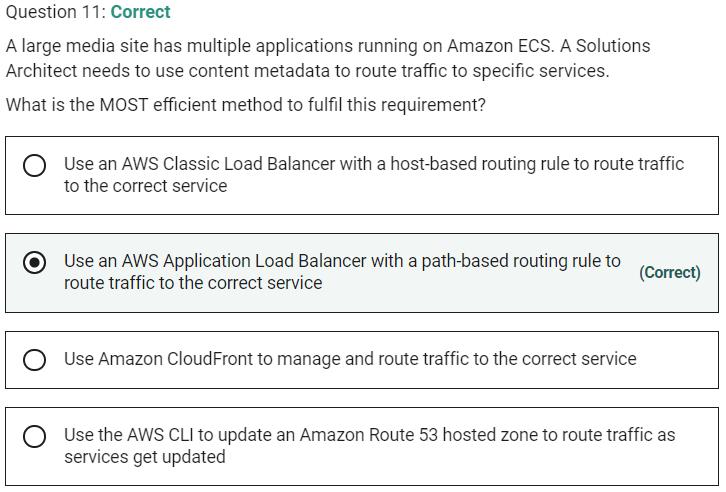


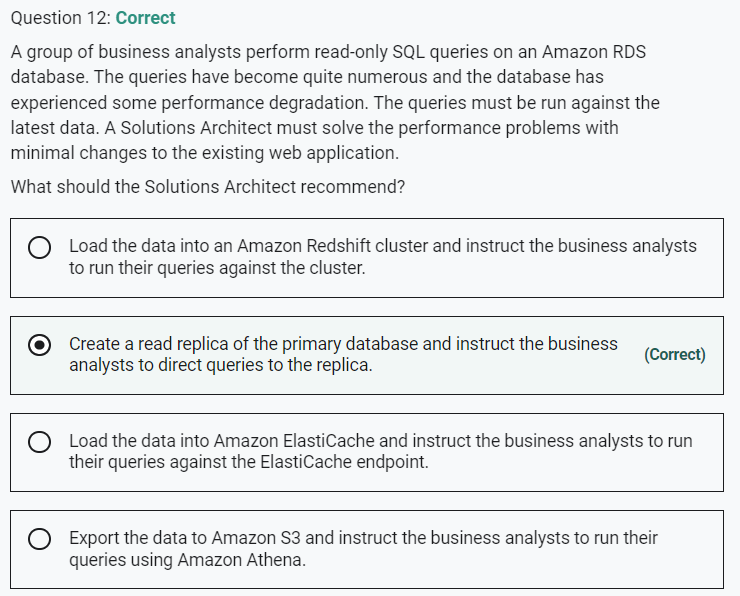


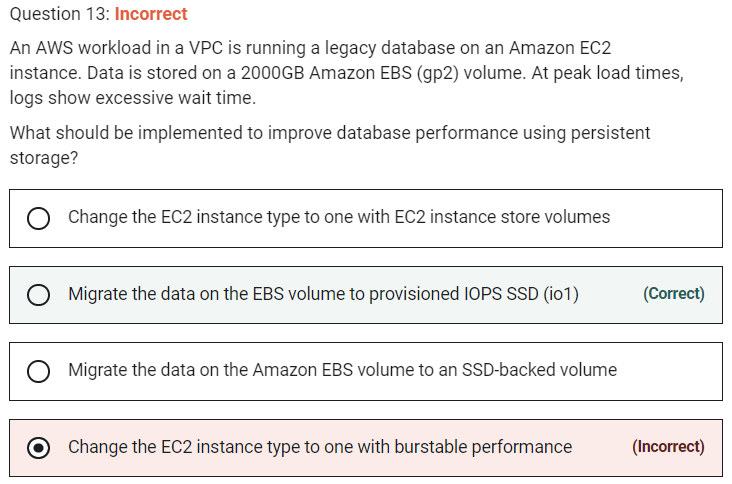


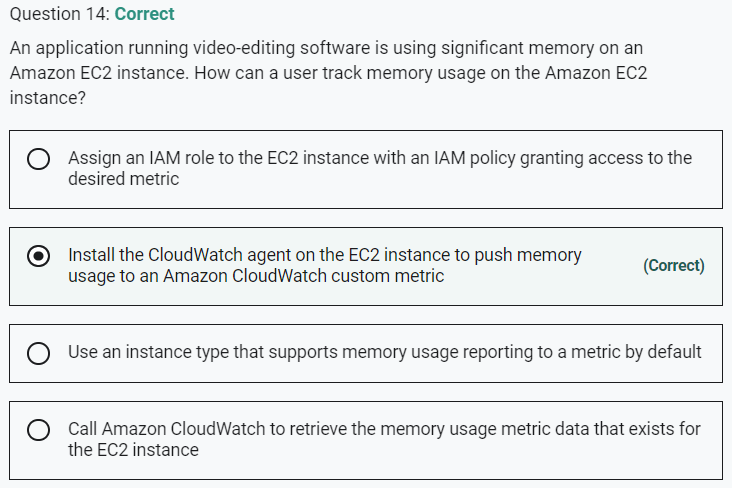
Amazon DynamoDB is the best database for this use case as it supports near-real time performance and millisecond responsiveness.

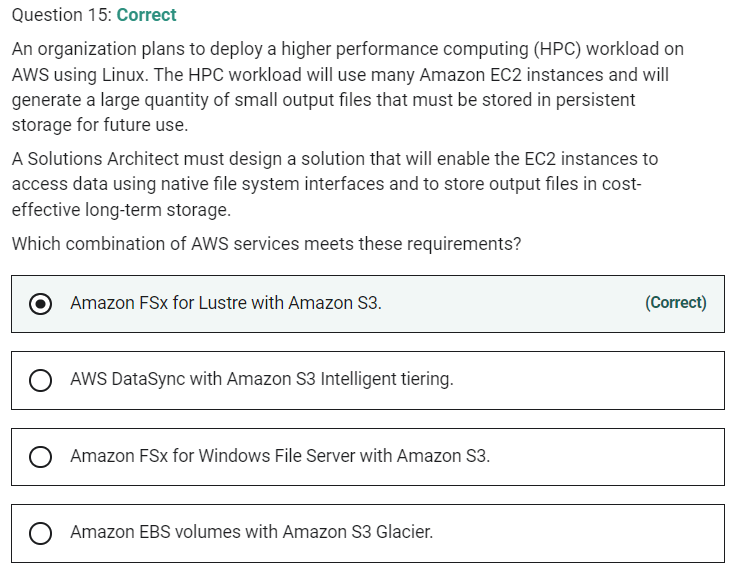


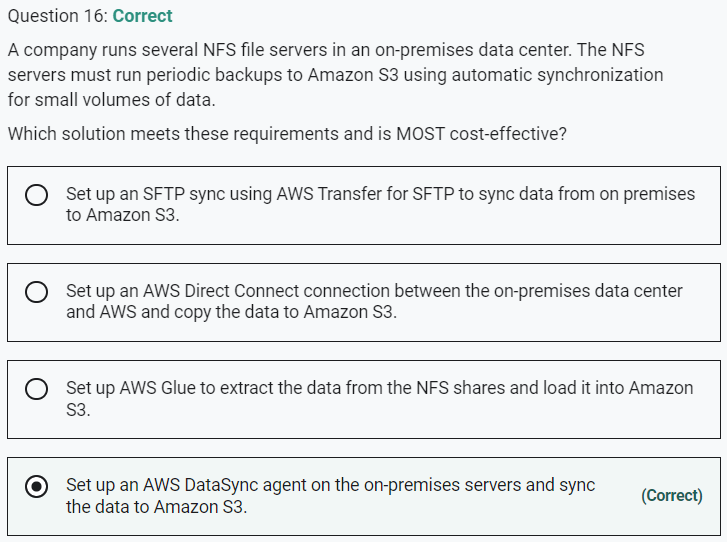


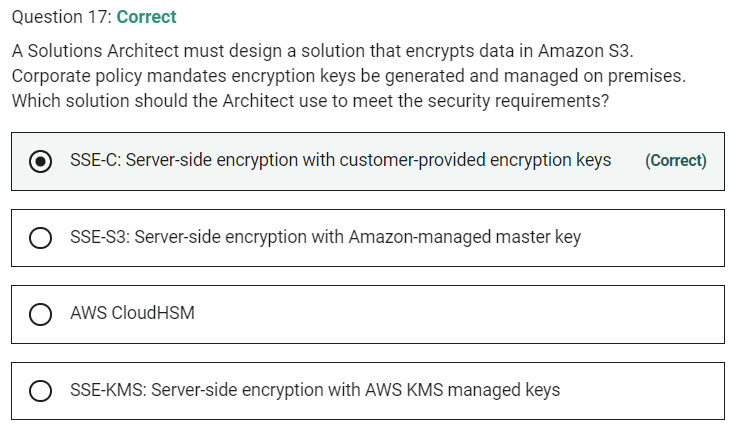










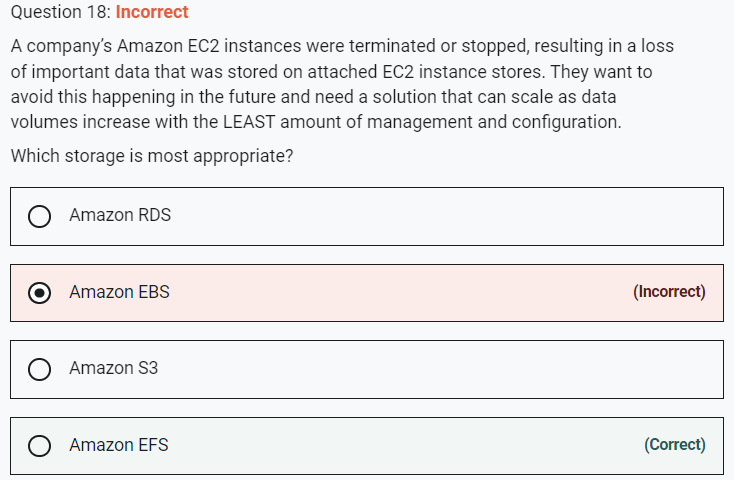


Server-side encryption is about protecting data at rest. Server-side encryption encrypts only the object data, not object metadata. Using server-side encryption with customer-provided encryption keys (SSE-C) allows you to set your own encryption keys. With the encryption key you provide as part of your request, Amazon S3 manages the encryption as it writes to disks and decryption when you access your objects. Therefore, you don't need to maintain any code to perform data encryption and decryption. The only thing you do is manage the encryption keys you provide.

**INCORRECT:** "SSE-S3: Server-side encryption with Amazon-managed master key" is incorrect. With SSE-S3, Amazon manage the keys for you, so this is incorrect.

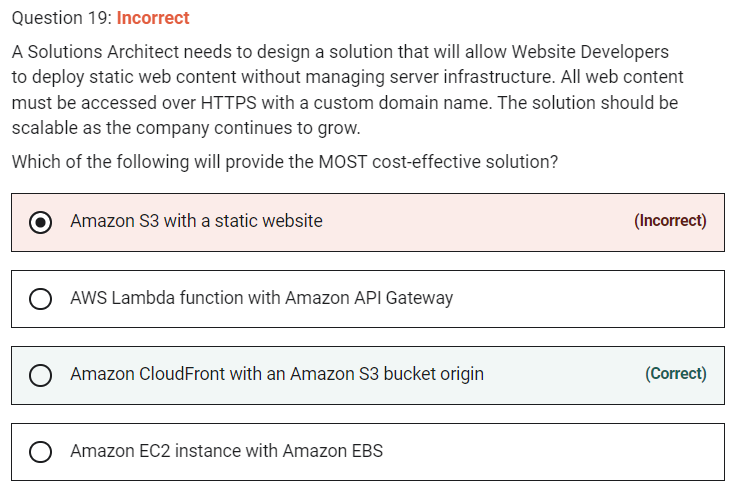
**INCORRECT:** "SSE-KMS: Server-side encryption with AWS KMS managed keys" is incorrect. With SSE-KMS the keys are managed in the Amazon Key Management Service, so this is incorrect.

**INCORRECT:** "AWS CloudHSM" is incorrect. With AWS CloudHSM your keys are held in AWS in a hardware security module. Again, the keys are not on-premises they are in AWS, so this is incorrect.



Amazon EFS is a fully managed service that requires no changes to your existing applications and tools, providing access through a standard file system interface for seamless integration. It is built to scale on demand to petabytes without disrupting applications, growing and shrinking automatically as you add and remove files. This is an easy solution to implement and the option that requires the least management and configuration.

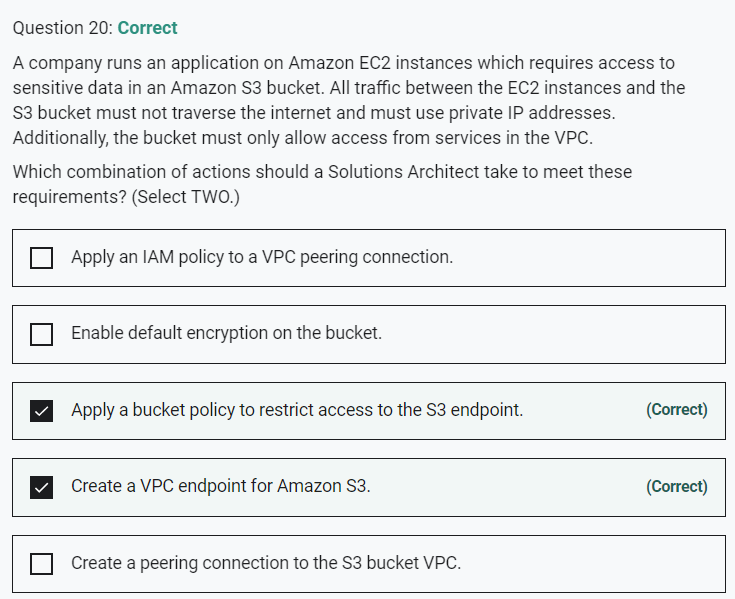
An instance store provides temporary block-level storage for an EC2 instance. If you terminate the instance you lose all data. The alternative is to use Elastic Block Store volumes which are also block-level storage devices but the data is persistent. However, EBS is not a fully managed solution and doesn’t grow automatically as your data requirements increase – you would need to increase the volume size and then extend your filesystem.

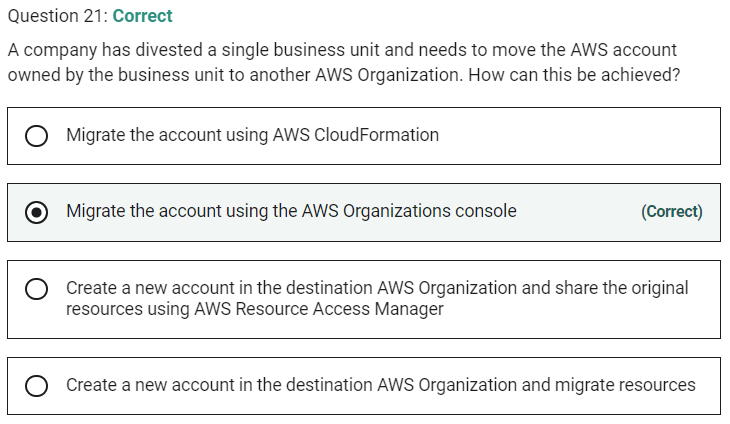


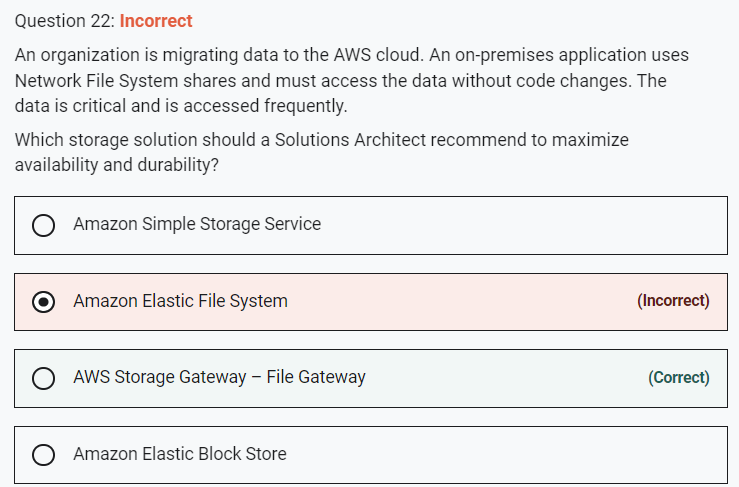
You can create an Amazon CloudFront distribution that uses an S3 bucket as the origin. This will allow you to serve the static content using the HTTPS protocol.

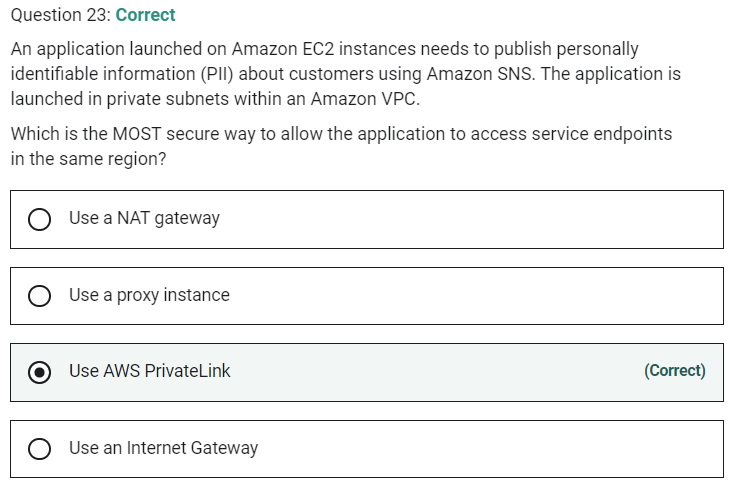
**CORRECT:**"Amazon CloudFront with an Amazon S3 bucket origin" is the correct answer.

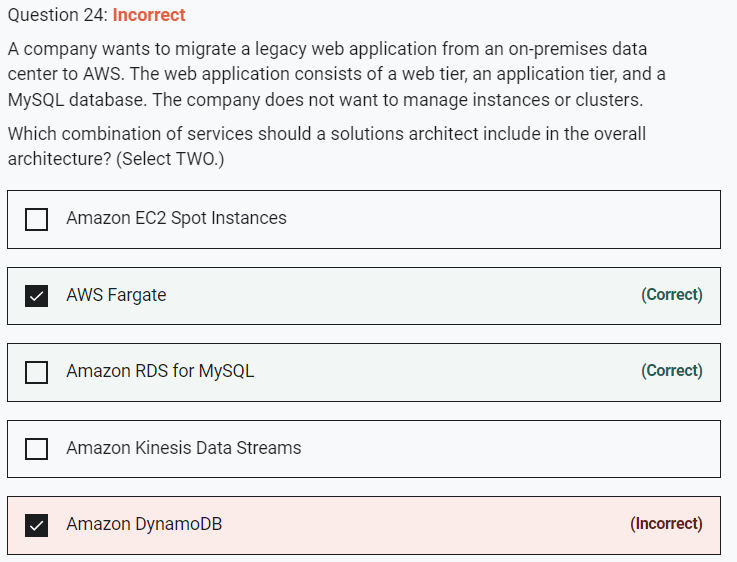
**INCORRECT:** "Amazon S3 with a static website" is incorrect. You can create a static website using Amazon S3 with a custom domain name. However, you cannot connect to an Amazon S3 static website using HTTPS (only HTTP) so this solution does not work.



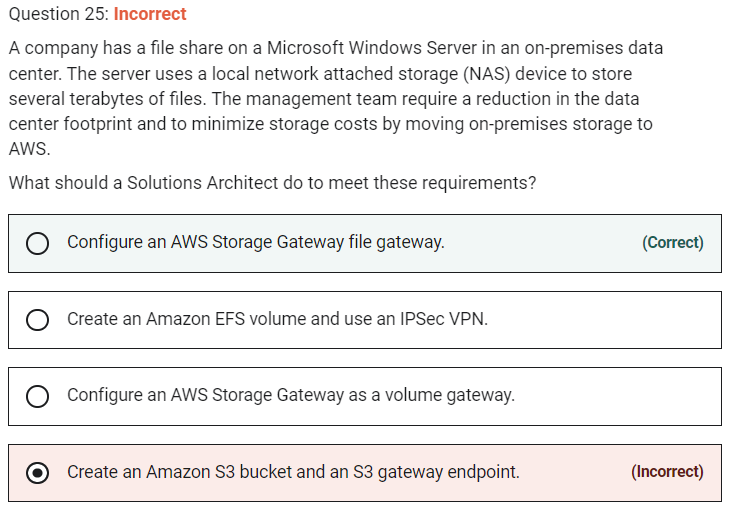




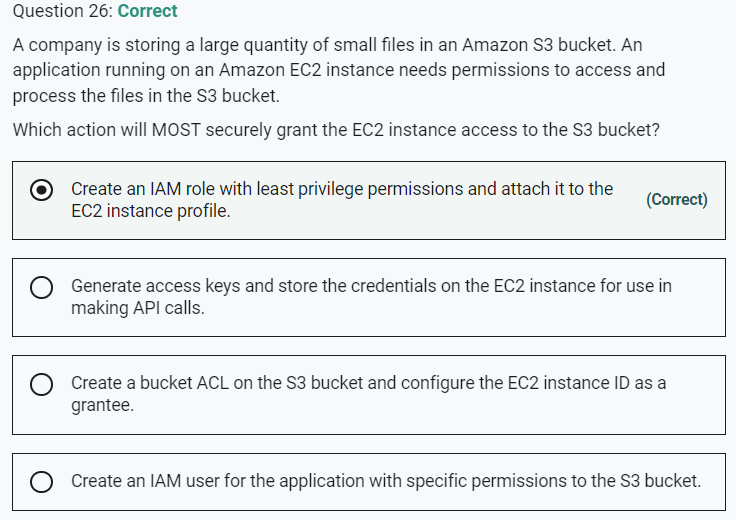


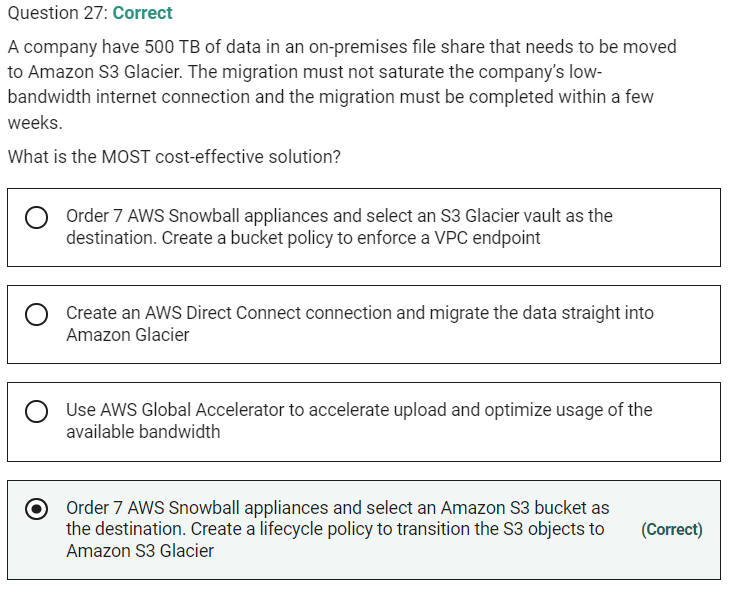


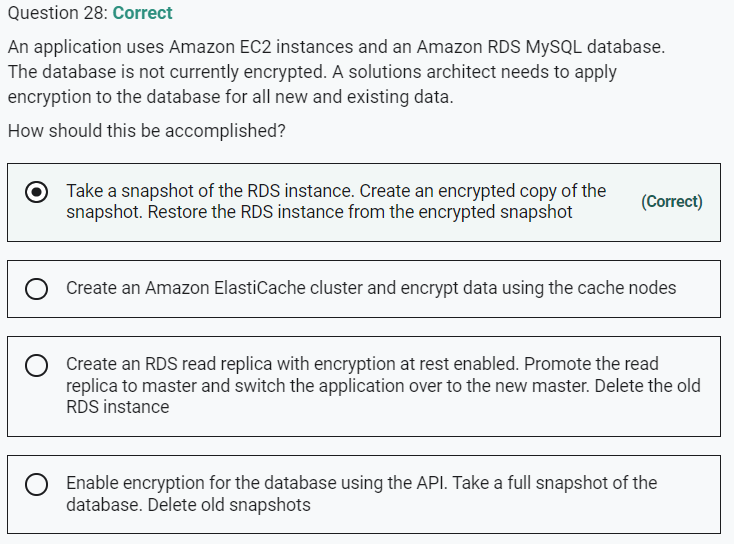
Amazon RDS is a managed service and you do not need to manage the instances. This is an ideal backend for the application and you can run a MySQL database on RDS without any refactoring. For the application components these can run on Docker containers with AWS Fargate. Fargate is a serverless service for running containers on AWS.

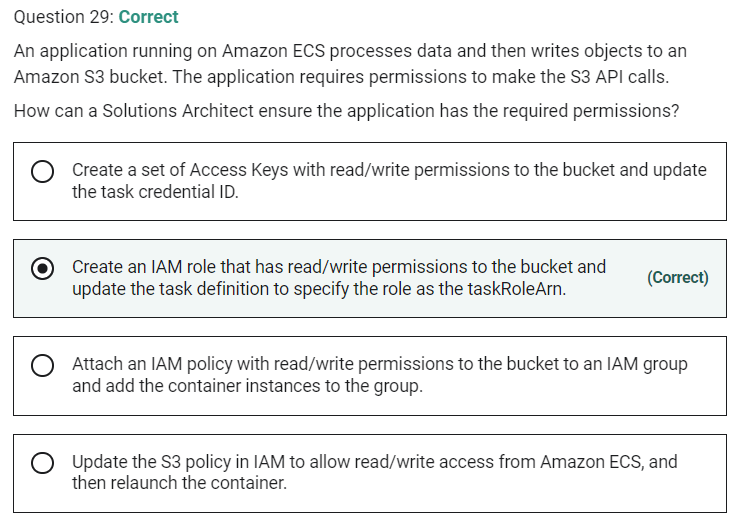


**INCORRECT:** "Create an Amazon S3 bucket and an S3 gateway endpoint" is incorrect. S3 is an object-level storage system so is not suitable for this use case. A gateway endpoint is a method of accessing S3 using private addresses from your VPC, not from your data center.

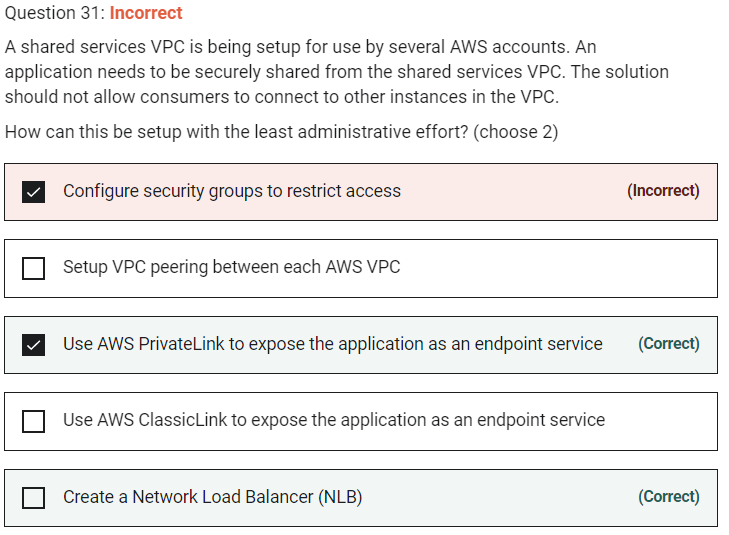


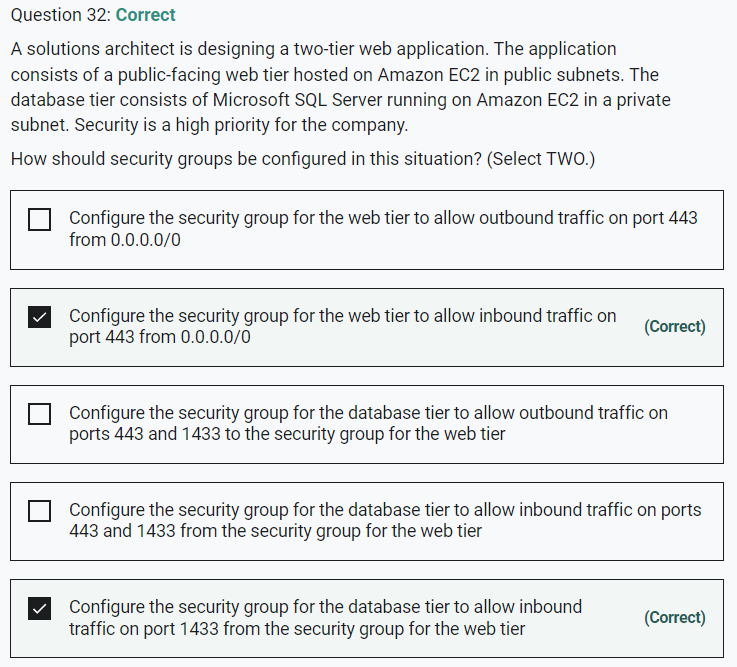


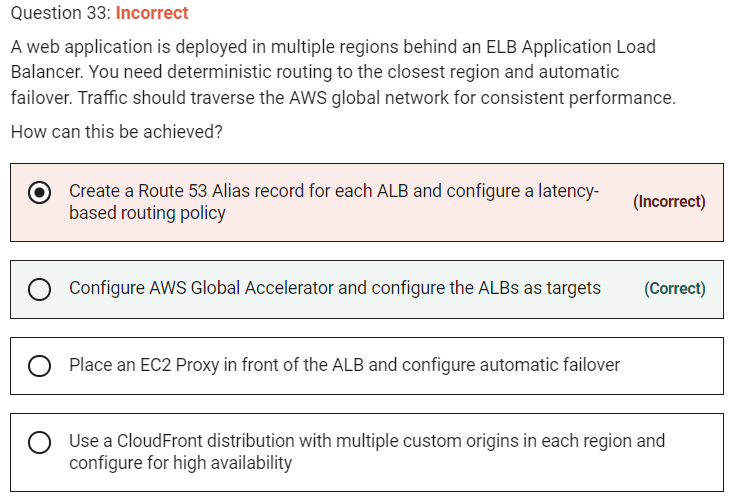


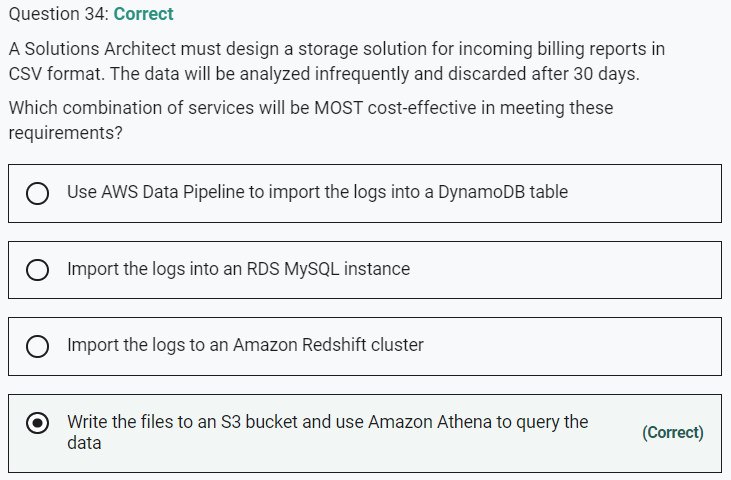


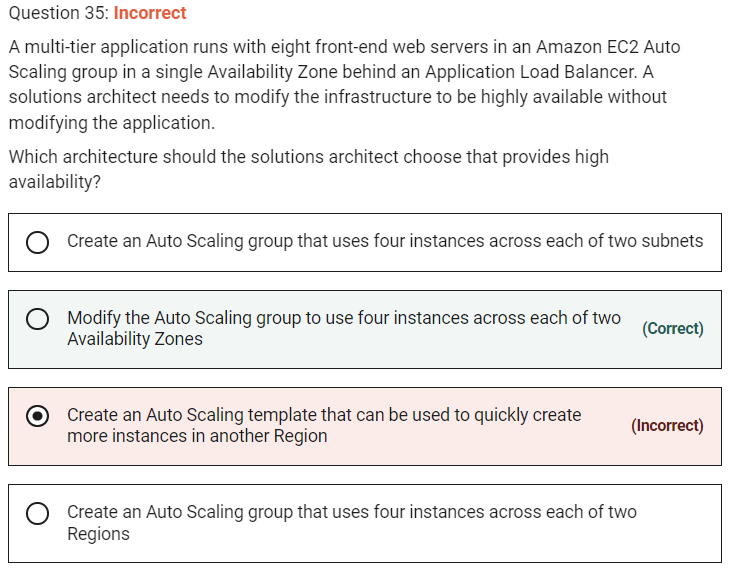










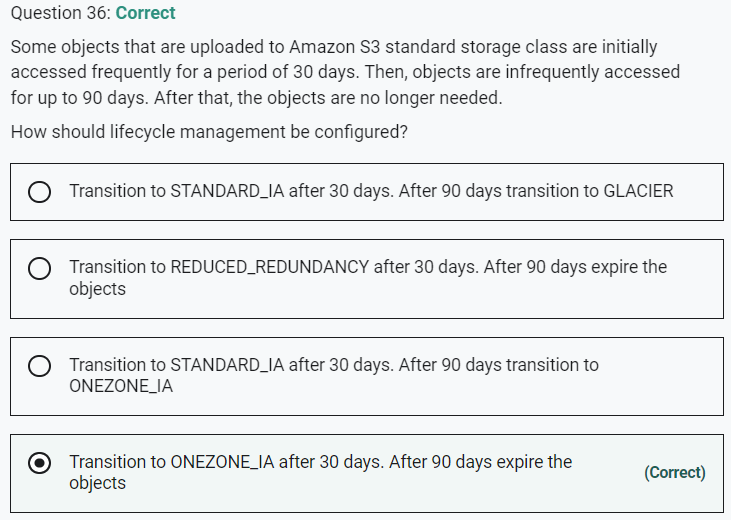


High availability can be enabled for this architecture quite simply by modifying the existing Auto Scaling group to use multiple availability zones.

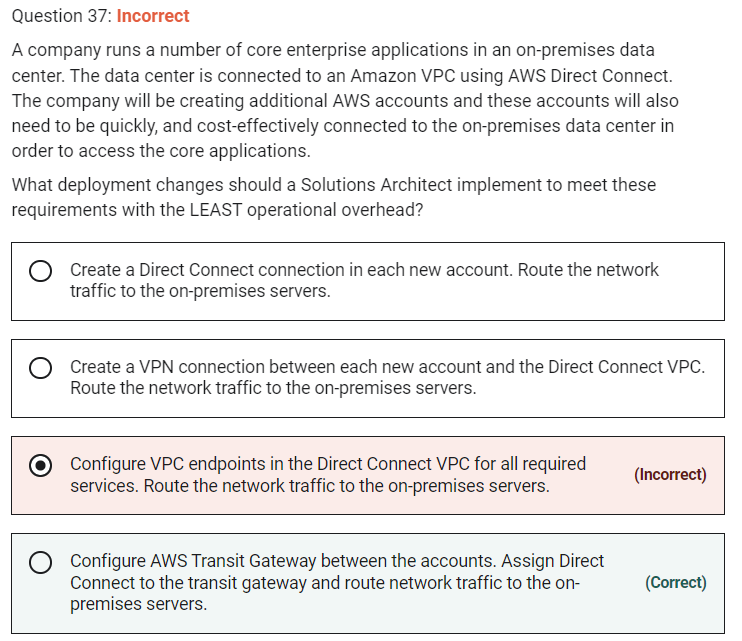
**INCORRECT:** "Create an Auto Scaling group that uses four instances across each of two Regions" is incorrect as EC2 Auto Scaling does not support multiple regions.

**INCORRECT:** "Create an Auto Scaling template that can be used to quickly create more instances in another Region" is incorrect as EC2 Auto Scaling does not support multiple regions.

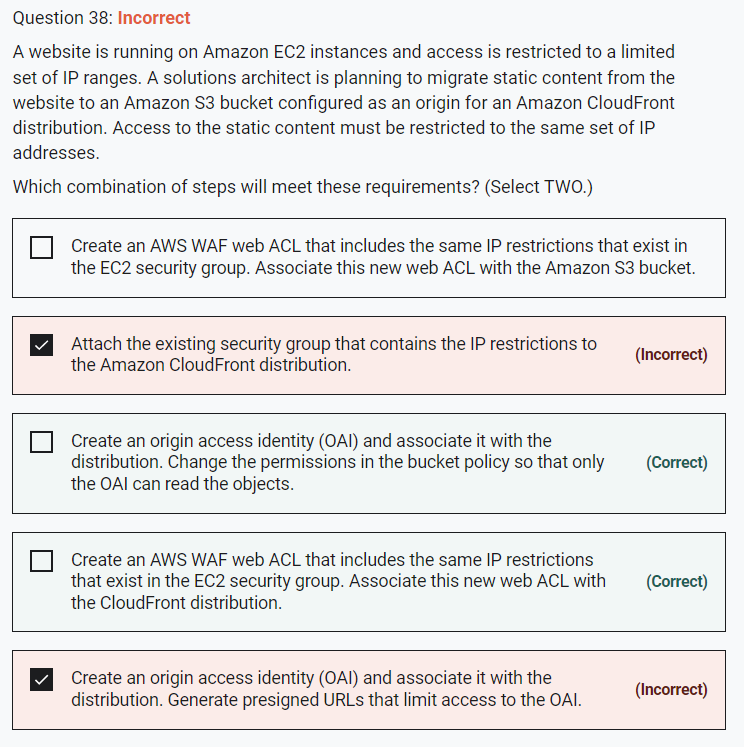
**INCORRECT:** "Create an Auto Scaling group that uses four instances across each of two subnets" is incorrect as the subnets could be in the same AZ.

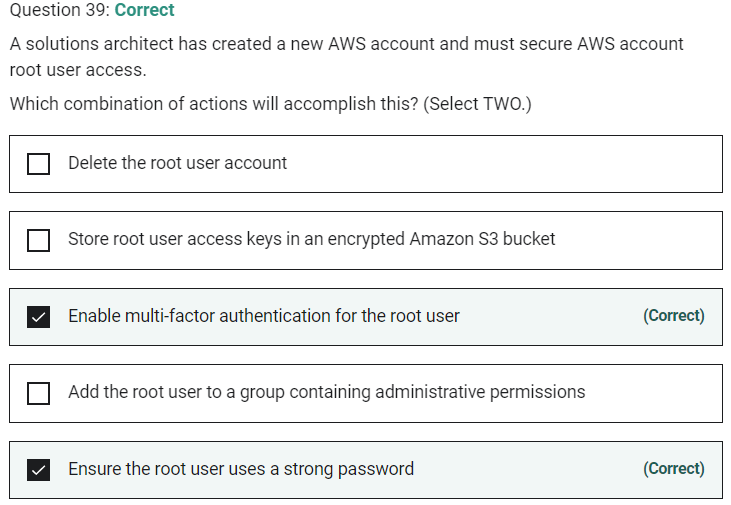


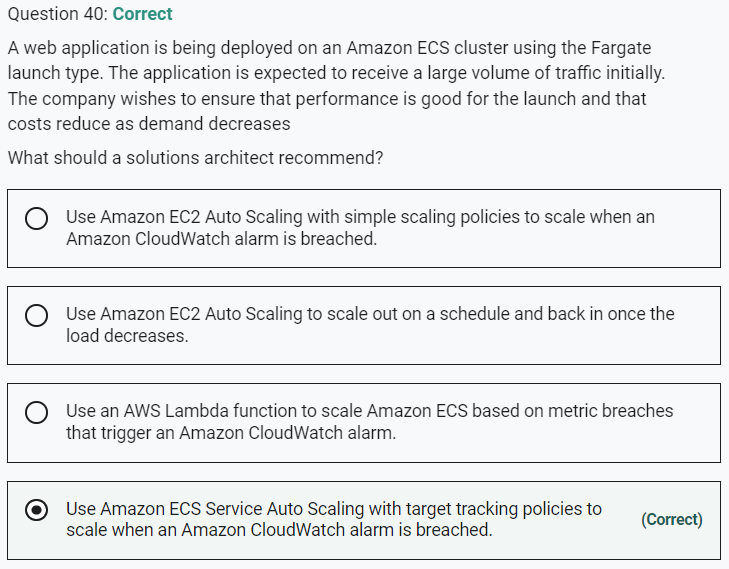
· You cannot transition to REDUCED\_REDUNDANCY from any storage class.

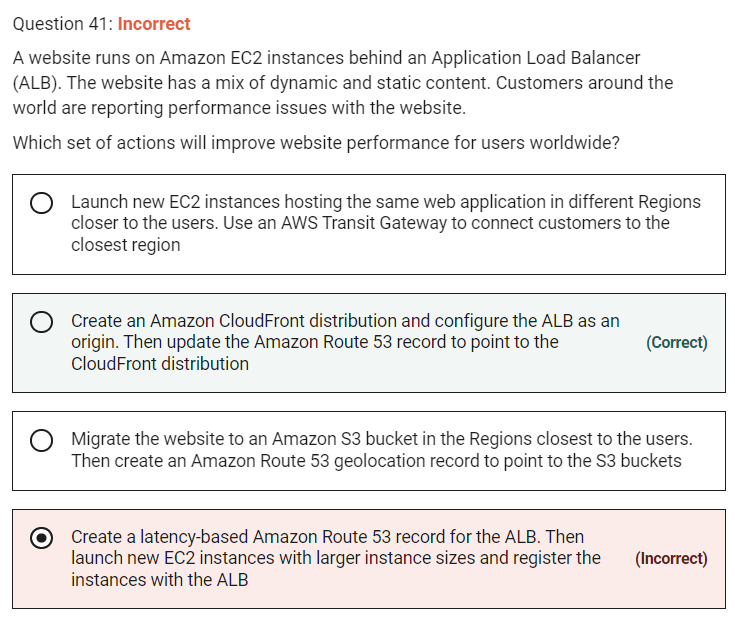


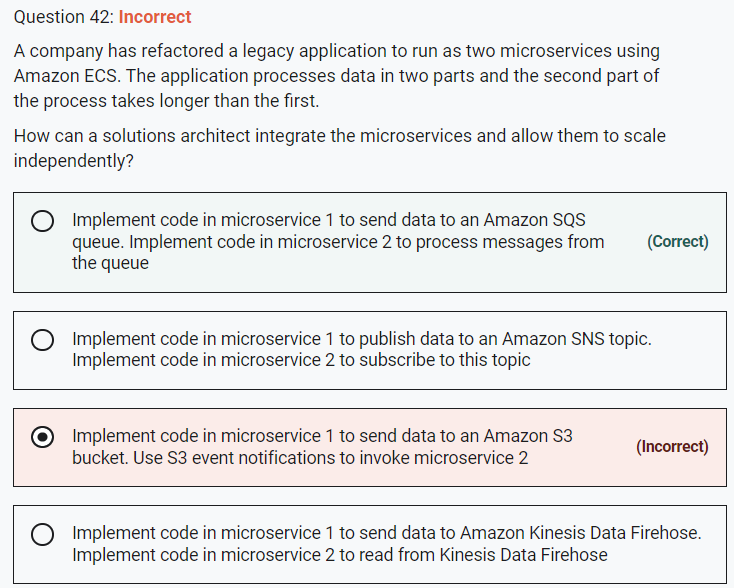
AWS Transit Gateway connects VPCs and on-premises networks through a central hub. With AWS Transit Gateway, you can quickly add Amazon VPCs, AWS accounts, VPN capacity, or AWS Direct Connect gateways to meet unexpected demand, without having to wrestle with complex connections or massive routing tables.



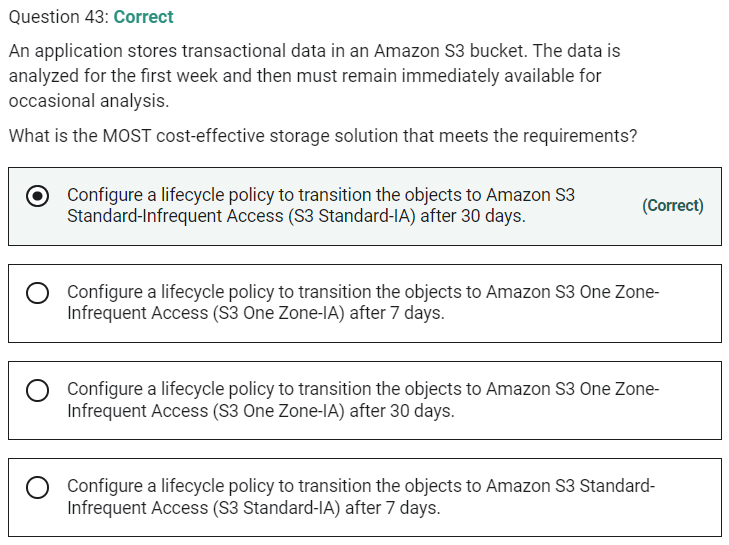


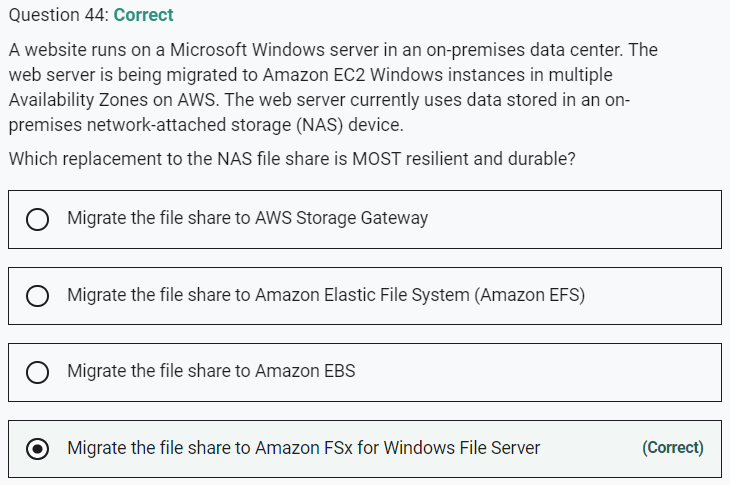


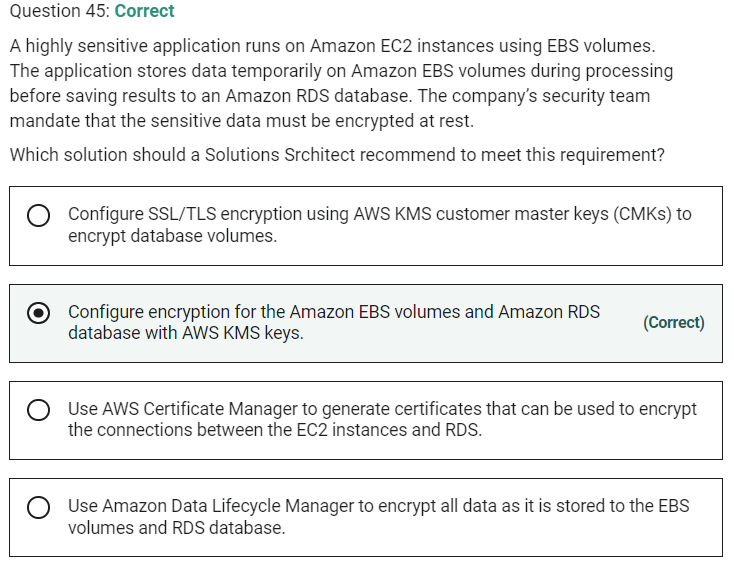


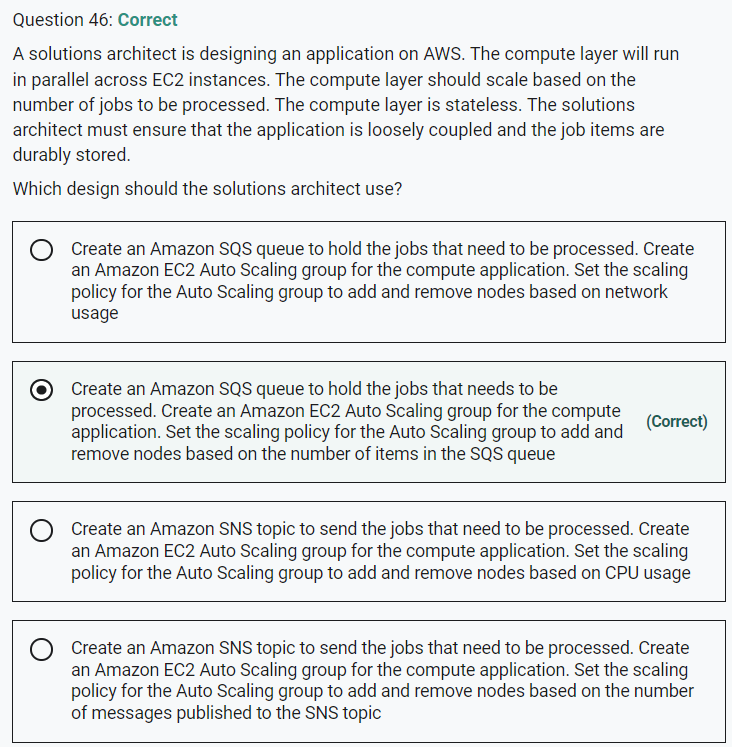


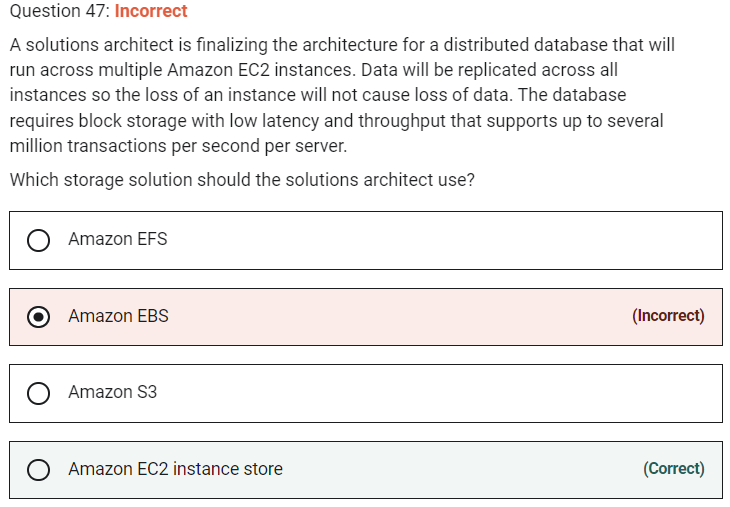
This is a good use case for Amazon SQS. The microservices must be decoupled so they can scale independently. An Amazon SQS queue will enable microservice 1 to add messages to the queue. Microservice 2 can then pick up the messages and process them. This ensures that if there’s a spike in traffic on the frontend, messages do not get lost due to the backend process not being ready to process them.

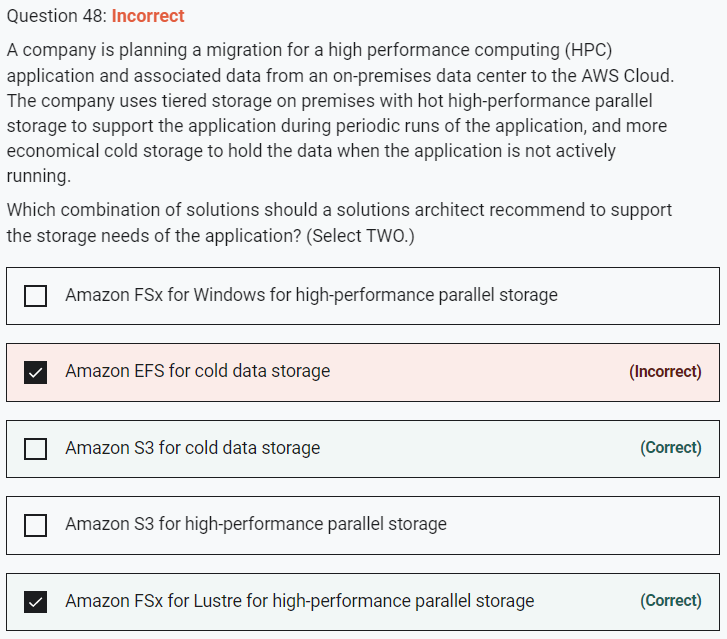


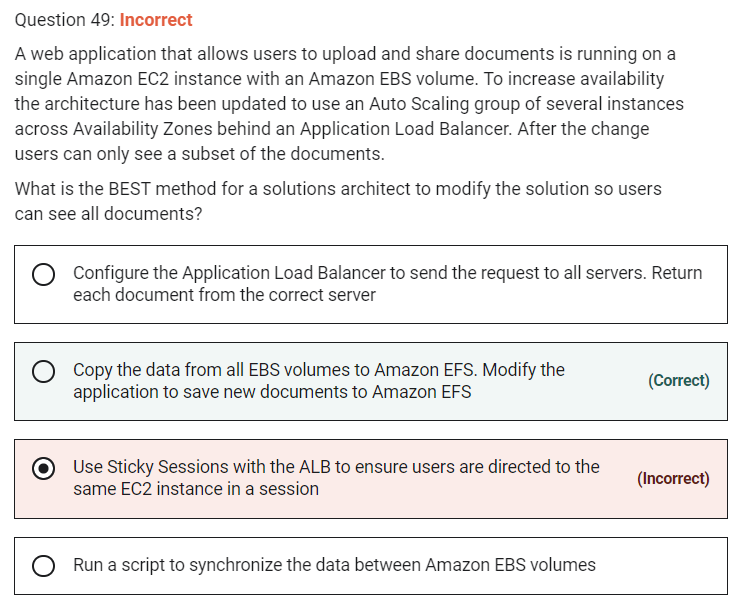






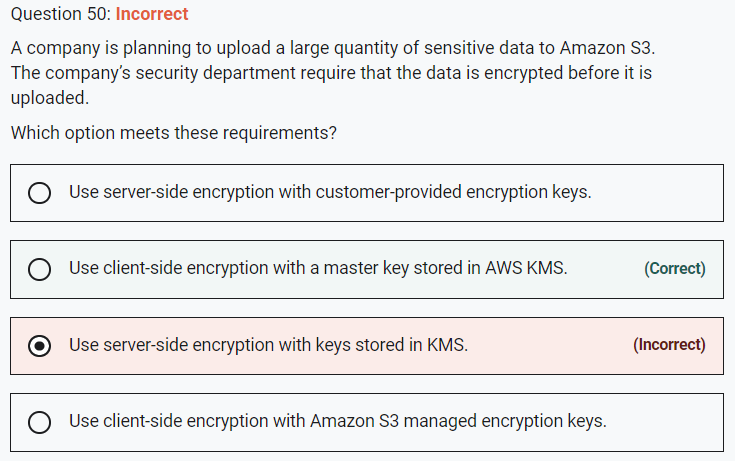






The problem that is being described is that the users are uploading the documents to an individual EC2 instance with a local EBS volume. Therefore, as EBS volumes cannot be shared across AZs, the data is stored separately and the ALB will be distributing incoming connections to different instances / data sets.

**INCORRECT:** "Use Sticky Sessions with the ALB to ensure users are directed to the same EC2 instance in a session" is incorrect as this will just “stick” a user to the same instance. They won’t see documents uploaded to other instances / EBS volumes.

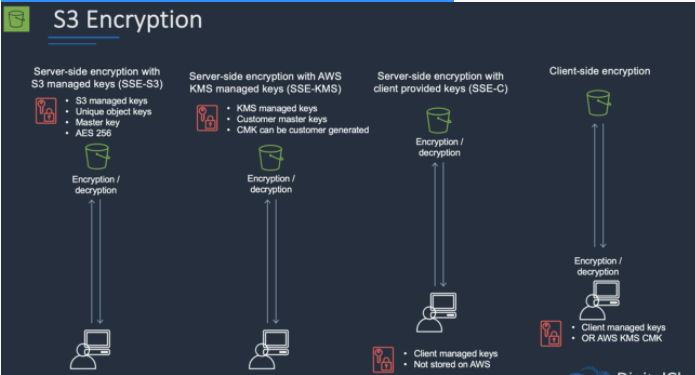


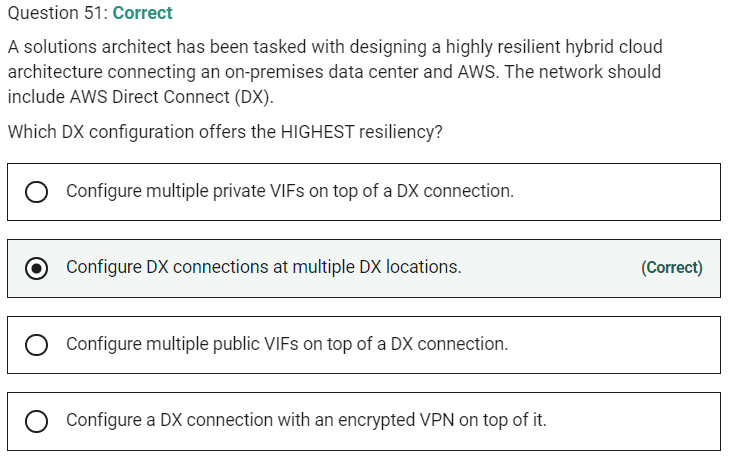
The requirement is that the objects must be encrypted before they are uploaded. The only option presented that meets this requirement is to use client-side encryption. You then have two options for the keys you use to perform the encryption:

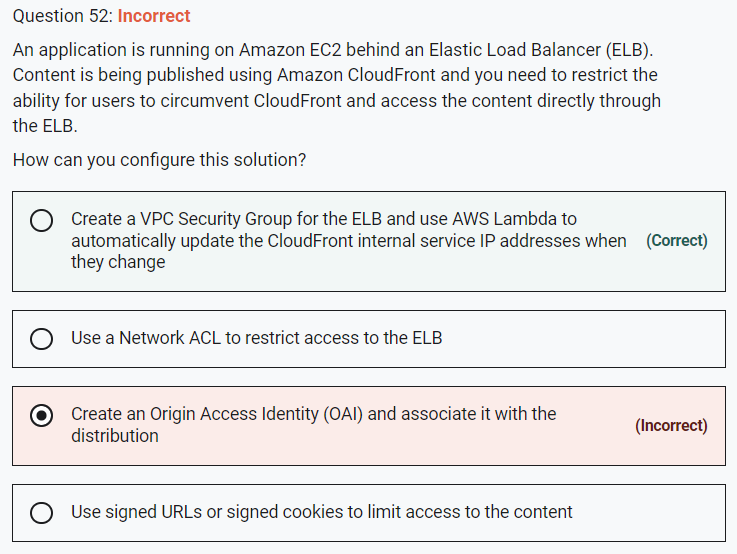
• Use a customer master key (CMK) stored in AWS Key Management Service (AWS KMS).

• Use a master key that you store within your application.

In this case the correct answer is to use an AWS KMS key. Note that you cannot use client-side encryption with keys managed by Amazon S3.







**Explanation**

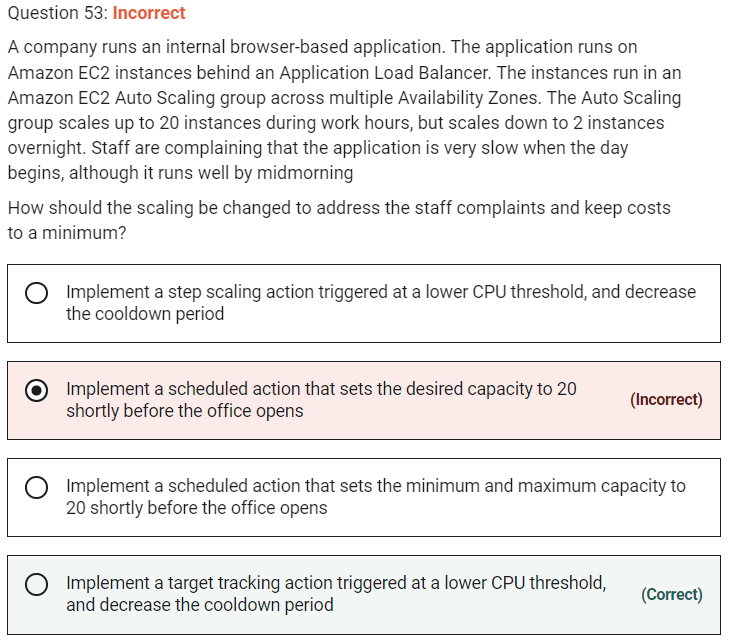
The only way to get this working is by using a VPC Security Group for the ELB that is configured to allow only the internal service IP ranges associated with CloudFront. As these are updated from time to time, you can use AWS Lambda to automatically update the addresses. This is done using a trigger that is triggered when AWS issues an SNS topic update when the addresses are changed.

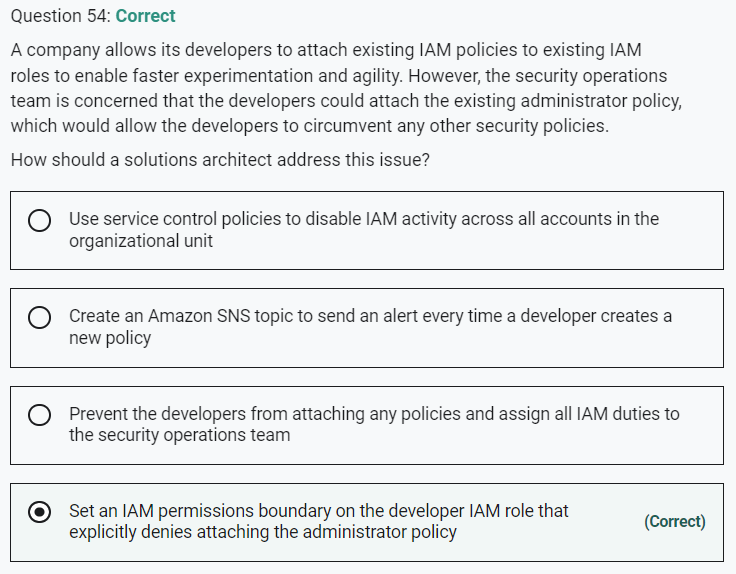
**CORRECT:**"Create a VPC Security Group for the ELB and use AWS Lambda to automatically update the CloudFront internal service IP addresses when they change" is the correct answer.

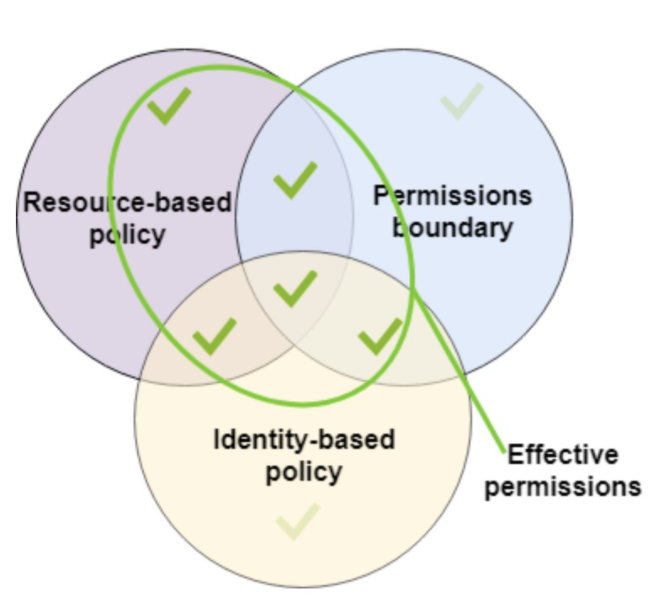
**INCORRECT:** "Create an Origin Access Identity (OAI) and associate it with the distribution" is incorrect. You can use an OAI to restrict access to content in Amazon S3 but not on EC2 or ELB.

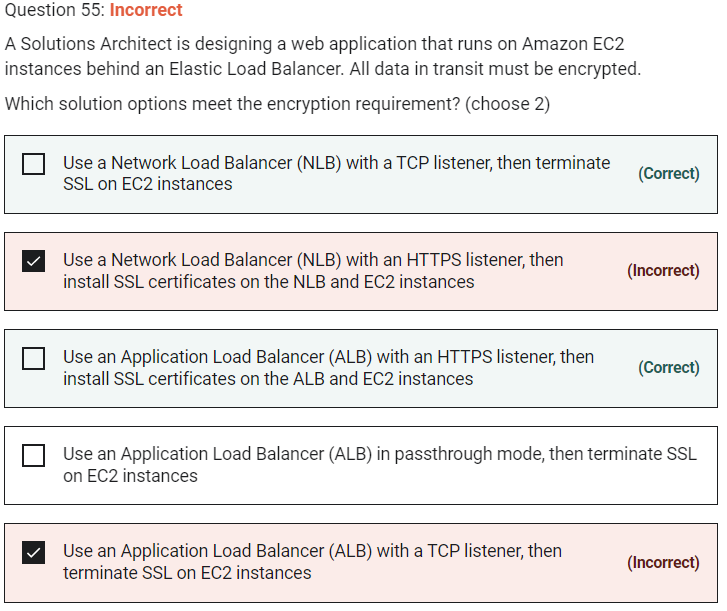
**INCORRECT:** "Use signed URLs or signed cookies to limit access to the content" is incorrect. Signed cookies and URLs are used to limit access to files but this does not stop people from circumventing CloudFront and accessing the ELB directly.

**INCORRECT:** "Use a Network ACL to restrict access to the ELB" is incorrect. A Network ACL can be used to restrict access to an ELB but it is recommended to use security groups and this solution is incomplete as it does not account for the fact that the internal service IP ranges change over time.









You can passthrough encrypted traffic with an NLB and terminate the SSL on the EC2 instances, so this is a valid answer.

You can use a HTTPS listener with an ALB and install certificates on both the ALB and EC2 instances. This does not use passthrough, instead it will terminate the first SSL connection on the ALB and then re-encrypt the traffic and connect to the EC2 instances.

**CORRECT:**"Use a Network Load Balancer (NLB) with a TCP listener, then terminate SSL on EC2 instances" is the correct answer.

**CORRECT:**"Use an Application Load Balancer (ALB) with an HTTPS listener, then install SSL certificates on the ALB and EC2 instances" is the correct answer.

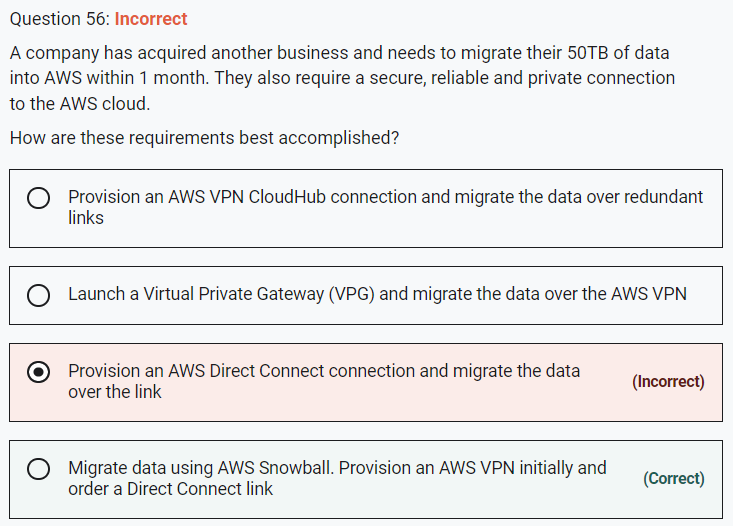
**INCORRECT:** "Use an Application Load Balancer (ALB) in passthrough mode, then terminate SSL on EC2 instances" is incorrect. You cannot use passthrough mode with an ALB and terminate SSL on the EC2 instances.

**INCORRECT:** "Use a Network Load Balancer (NLB) with an HTTPS listener, then install SSL certificates on the NLB and EC2 instances" is incorrect. You cannot use a HTTPS listener with an NLB.

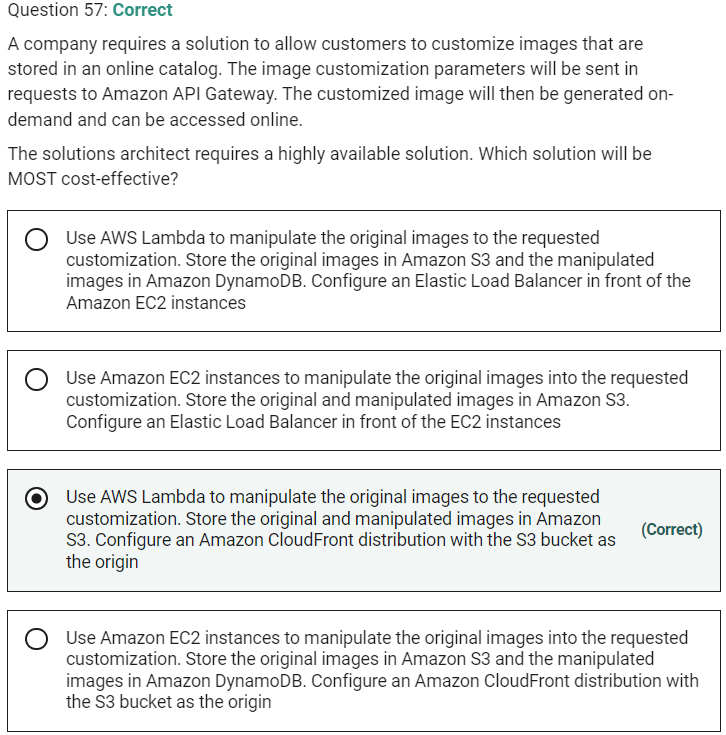
**INCORRECT:** "Use an Application Load Balancer (ALB) with a TCP listener, then terminate SSL on EC2 instances" is incorrect. You cannot use a TCP listener with an ALB.

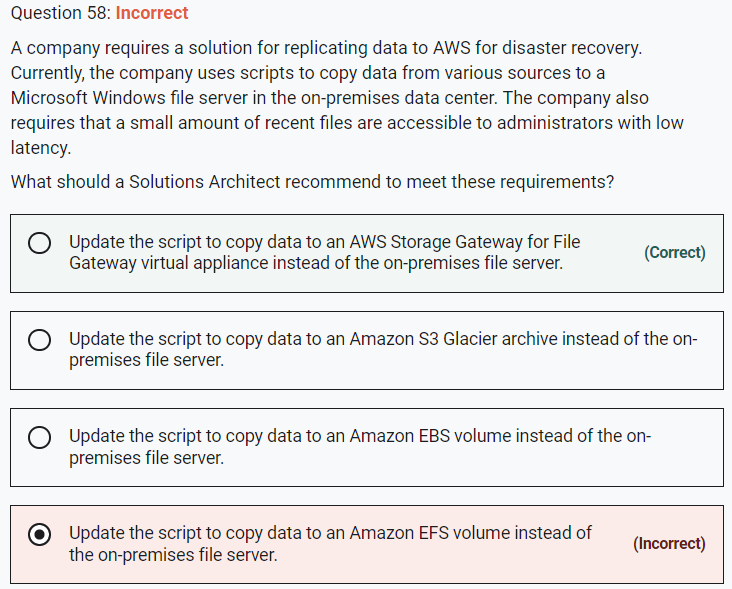
ALB – HTTPS

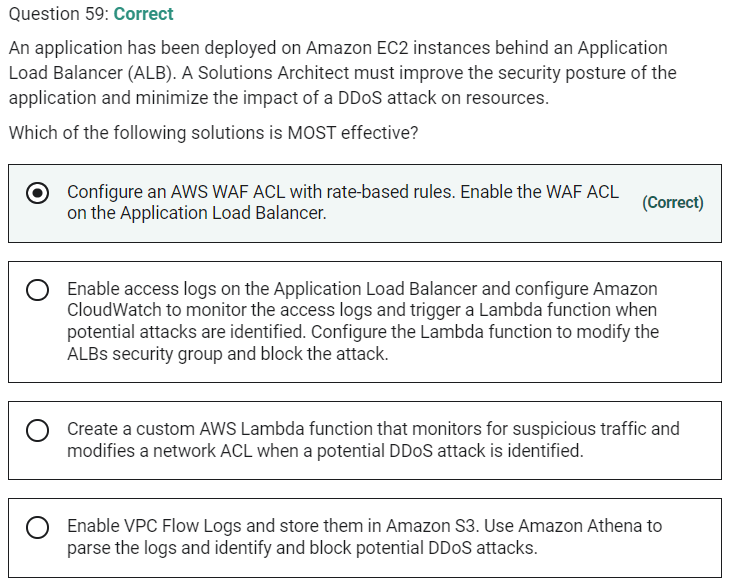
NLB – TCP, passthrough

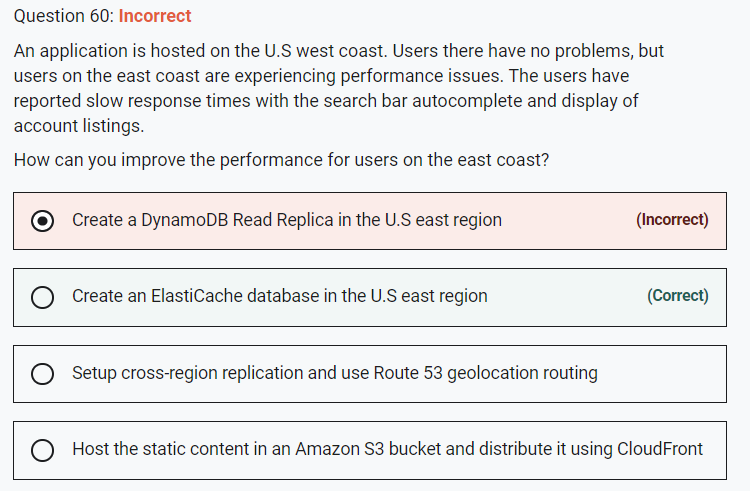


AWS Direct Connect provides a secure, reliable and private connection. However, lead times are often longer than 1 month so it cannot be used to migrate data within the timeframes. Therefore, it is better to use AWS Snowball to move the data and order a Direct Connect connection to satisfy the other requirement later on. In the meantime the organization can use an AWS VPN for secure, private access to their VPC.









#### Explanation

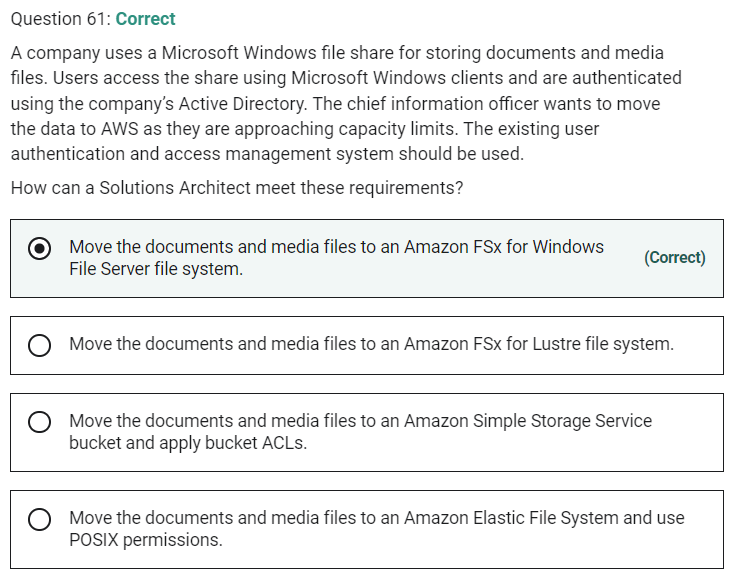
ElastiCache can be deployed in the U.S east region to provide high-speed access to the content. ElastiCache Redis has a good use case for autocompletion (see links below).

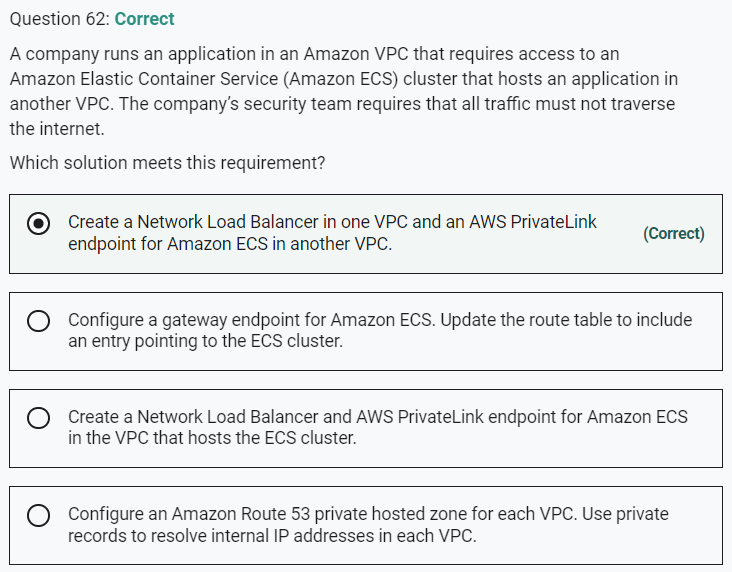
**CORRECT:**"Create an ElastiCache database in the U.S east region" is the correct answer.

**INCORRECT:** "Host the static content in an Amazon S3 bucket and distribute it using CloudFront" is incorrect. This is not static content that can be hosted in an Amazon S3 bucket and distributed using CloudFront.

**INCORRECT:** "Setup cross-region replication and use Route 53 geolocation routing" is incorrect. Cross-region replication is an Amazon S3 concept and the dynamic data that is presented by this application is unlikely to be stored in an S3 bucket.

**INCORRECT:** "Create a DynamoDB Read Replica in the U.S east region" is incorrect. There’s no such thing as a DynamoDB Read Replica (Read Replicas are an RDS concept).





The correct solution is to use AWS PrivateLink in a service provider model. In this configuration a network load balancer will be implemented in the service provider VPC (the one with the ECS cluster in this example), and a PrivateLink endpoint will be created in the consumer VPC (the one with the company’s application).

