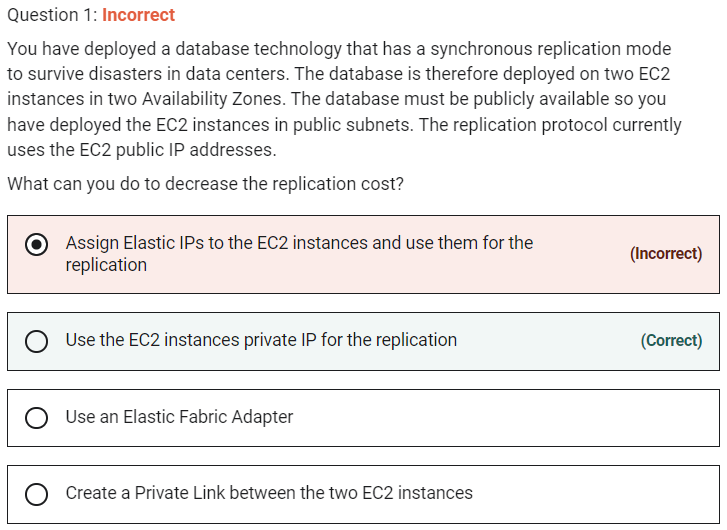
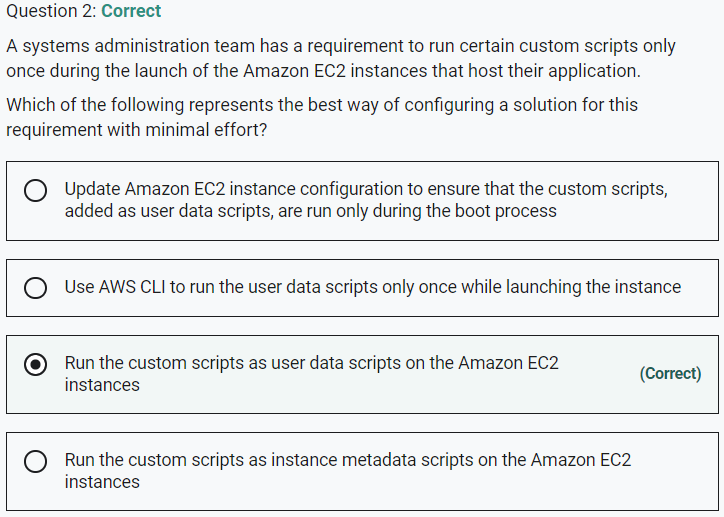
SAA-CO2

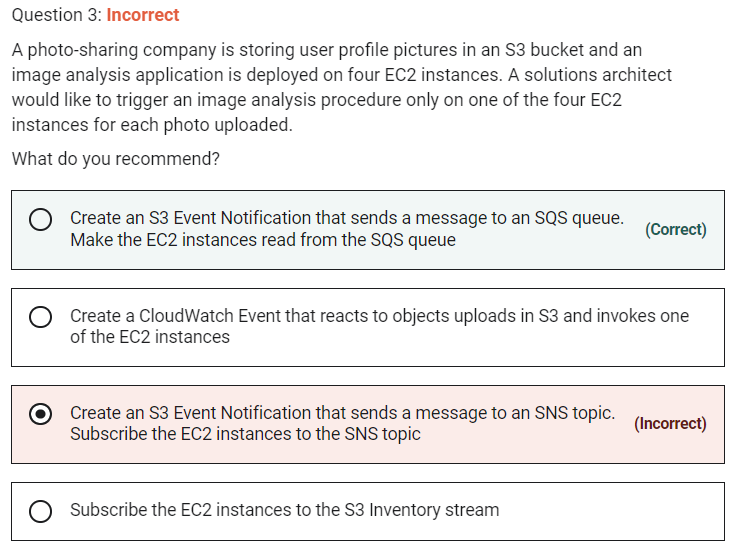
Practice Test-6

Marked





Marked



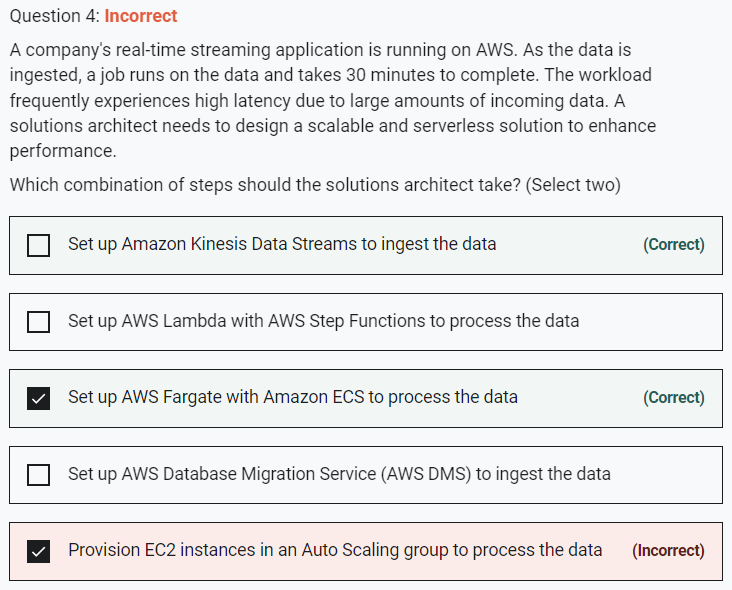
Amazon S3 supports the following destinations where it can publish events:

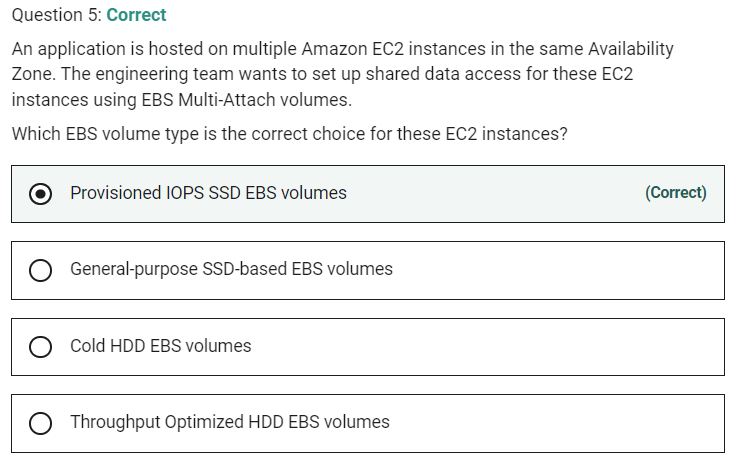
Amazon Simple Notification Service (Amazon SNS) topic

Amazon Simple Queue Service (Amazon SQS) queue

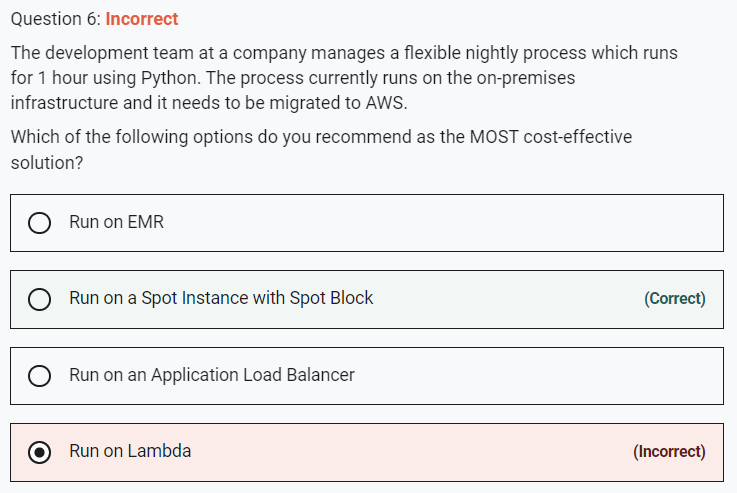
AWS Lambda

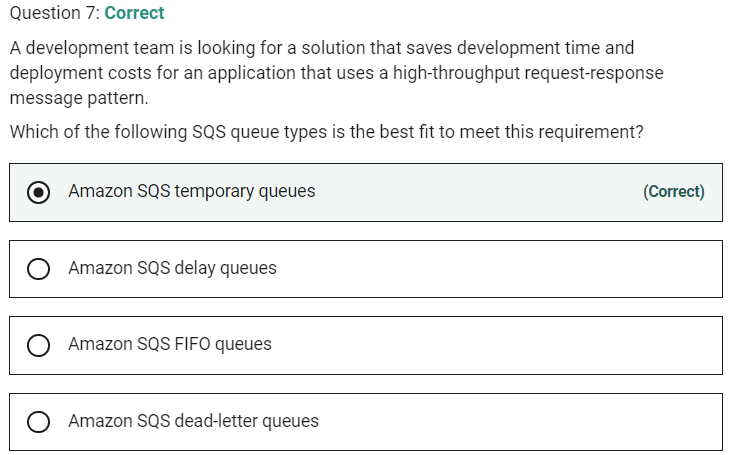
Marked



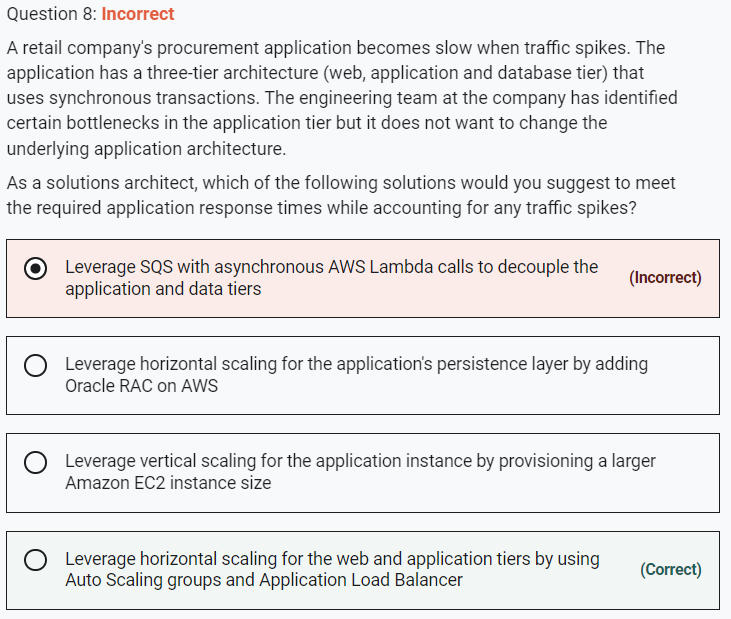


Marked

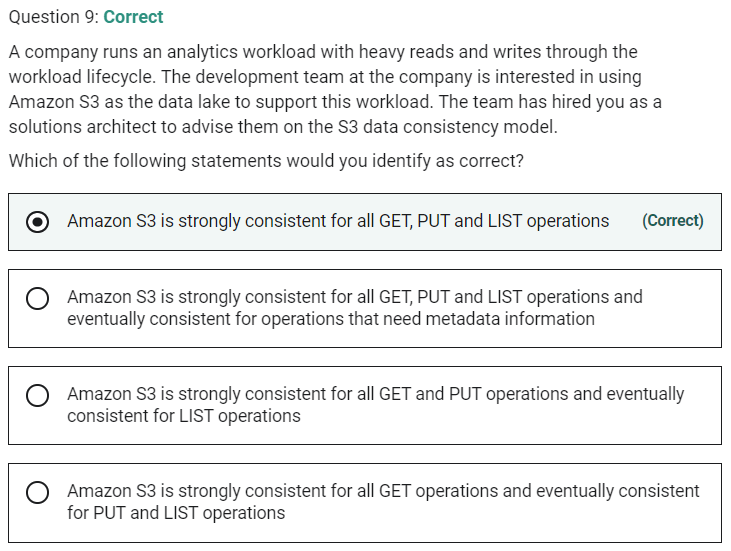




Marked

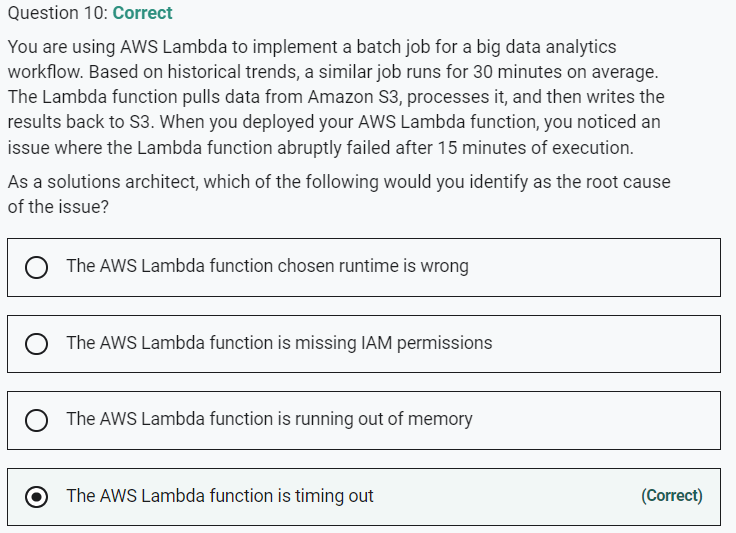


Marked

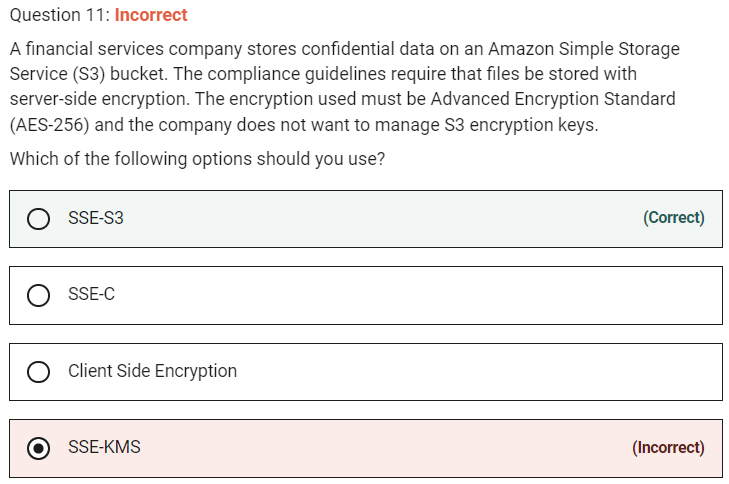


**Amazon S3 is strongly consistent for all GET, PUT and LIST operations** - After a successful write of a new object, or an overwrite or delete of an existing object, any subsequent read request immediately receives the latest version of the object.

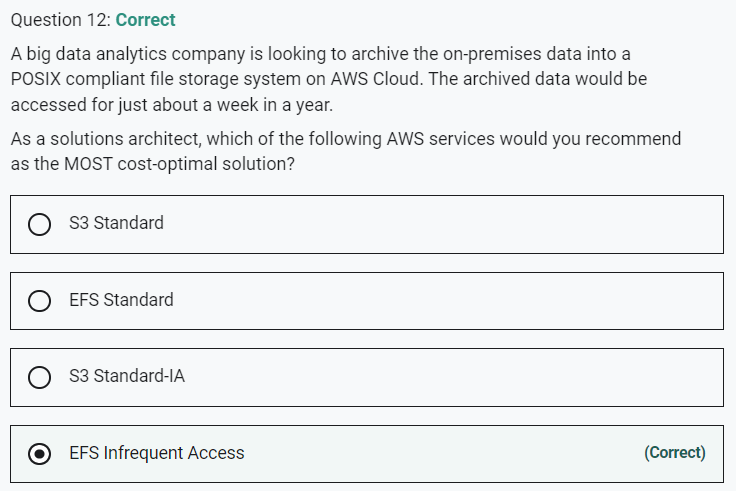
**Amazon S3 is strongly consistent for all GET, PUT and LIST operations and eventually consistent for operations that need metadata information** - For all existing and new objects, and in all regions, all S3 GET, PUT, and LIST operations, as well as operations that change object tags, ACLs, or metadata, are now strongly consistent.

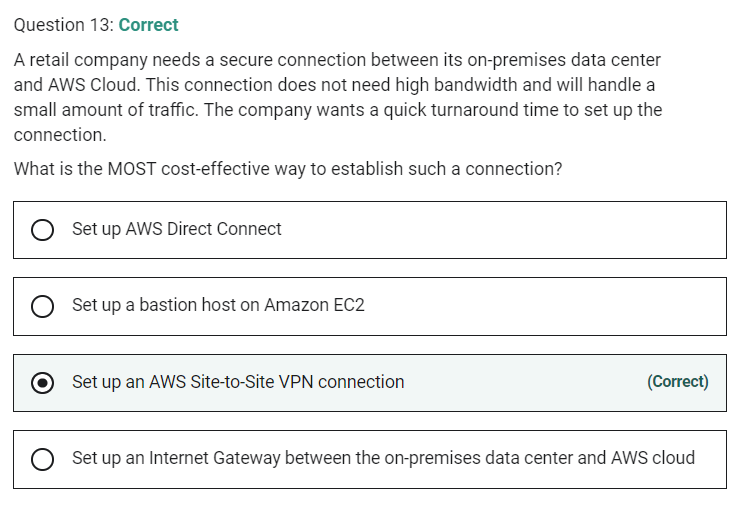


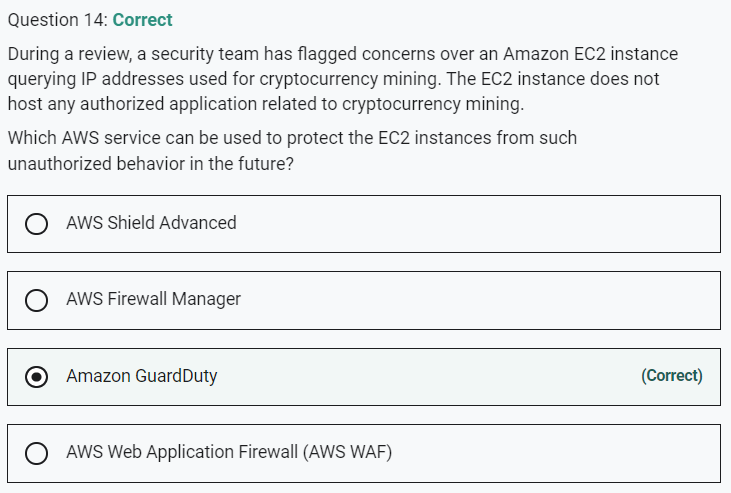
Marked



Amazon S3 server-side encryption uses one of the strongest block ciphers available, 256-bit Advanced Encryption Standard (AES-256),







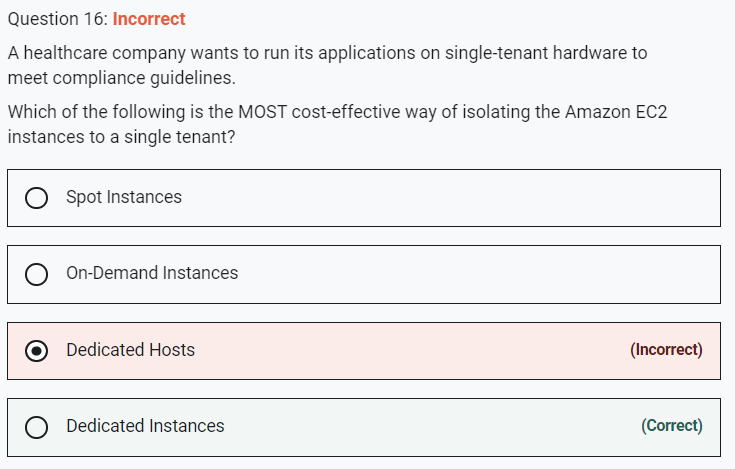
Marked



**Create two SQS standard queues: one for pro and one for lite. Set up EC2 instances to prioritize polling for the pro queue over the lite queue**

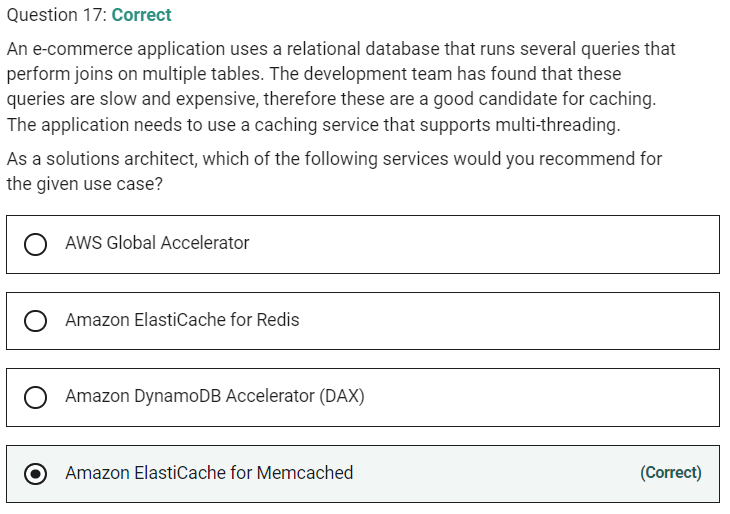
AWS recommends using separate queues to provide prioritization of work. Therefore, for the given use case, you need to create an SQS standard queue for processing pro users' photos and another SQS standard queue for processing lite users' photos. Then you can configure EC2 instances to prioritize polling for the pro queue over the lite queue.

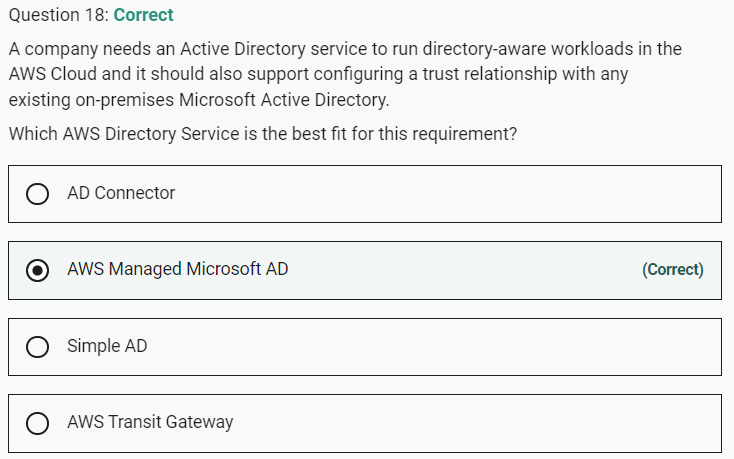
Marked

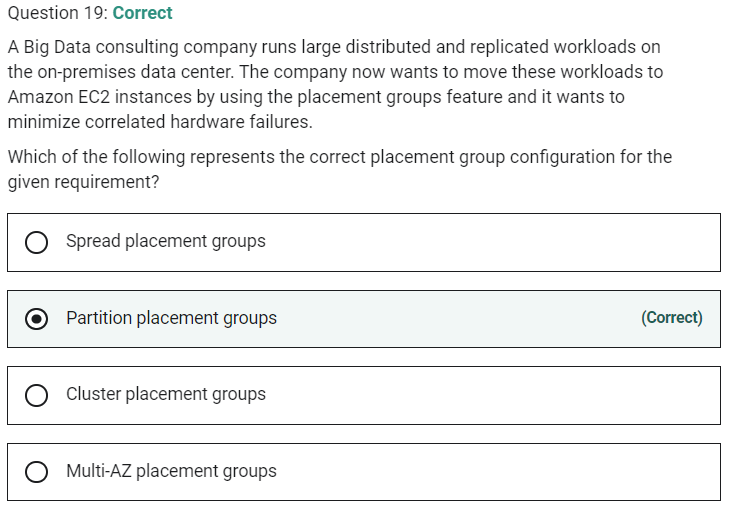


requirement.

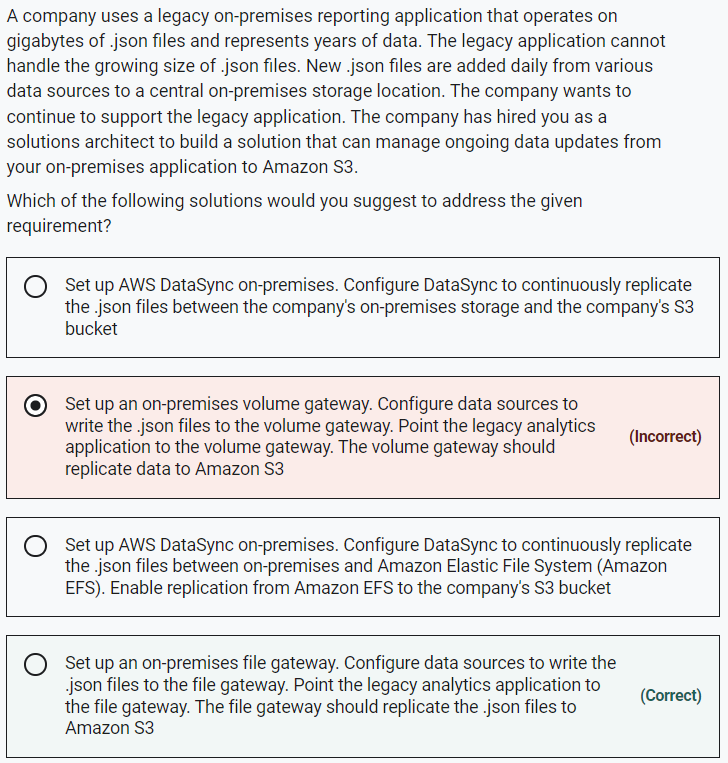
**On-Demand Instances** - With On-Demand Instances, you pay for the compute capacity by the second with no long-term commitments. You have full control over its lifecycle—you decide when to launch, stop, hibernate, start, reboot, or terminate it. Hardware isolation is not possible and on-demand has one of the costliest instance charges and hence is not the correct answer for current requirements.







Marked



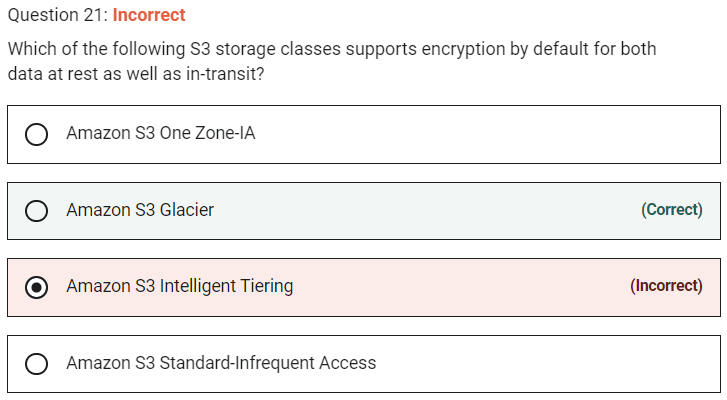
**Set up an on-premises volume gateway. Configure data sources to write the .json files to the volume gateway. Point the legacy analytics application to the volume gateway. The volume gateway should replicate data to Amazon S3** - Volume Gateway is for block storage and not for file storage, so it is not the right option.

**Set up AWS DataSync on-premises. Configure DataSync to continuously replicate the .json files between the company's on-premises storage and the company's S3 bucket**

**Set up AWS DataSync on-premises. Configure DataSync to continuously replicate the .json files between on-premises and Amazon Elastic File System (Amazon EFS). Enable replication from Amazon EFS to the company's S3 bucket**

AWS recommends that you should use AWS DataSync to migrate existing data to Amazon S3, and subsequently use the File Gateway configuration of AWS Storage Gateway to retain access to the migrated data and for ongoing updates from your on-premises file-based applications. Therefore, both these options are incorrect, as they use DataSync for ongoing replication.

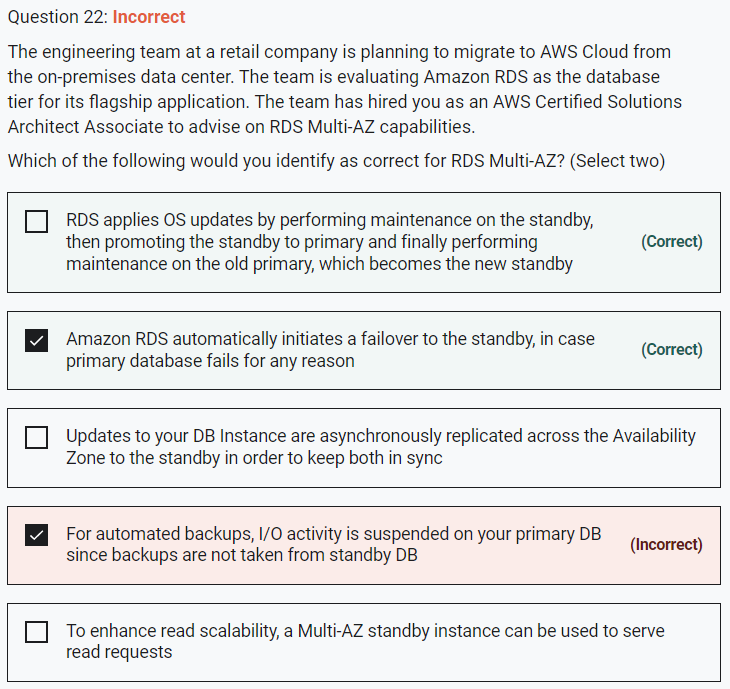
Marked



**Amazon S3 Glacier**

Amazon S3 Glacier is a secure, durable, and low-cost storage class for data archiving. Amazon S3 Glacier automatically encrypts data at rest using Advanced Encryption Standard (AES) 256-bit symmetric keys and supports secure transfer of your data over Secure Sockets Layer (SSL).

Marked



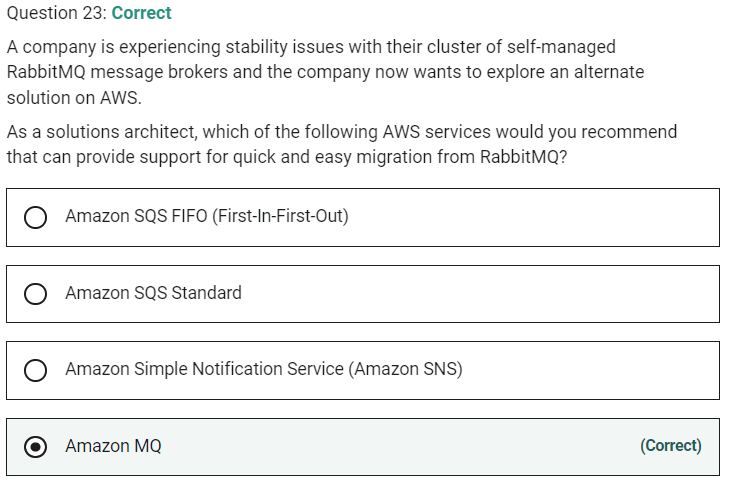
Running a DB instance as a Multi-AZ deployment can further reduce the impact of a maintenance event because Amazon RDS applies operating system updates by following these steps:

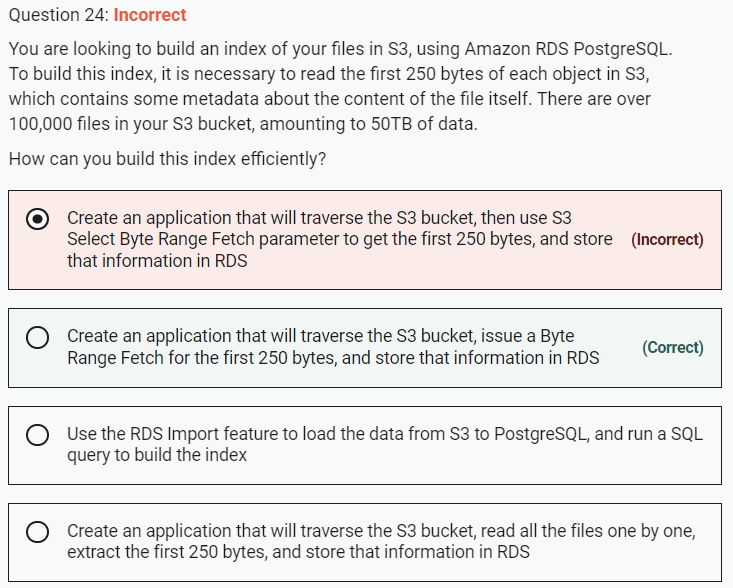
Perform maintenance on the standby.

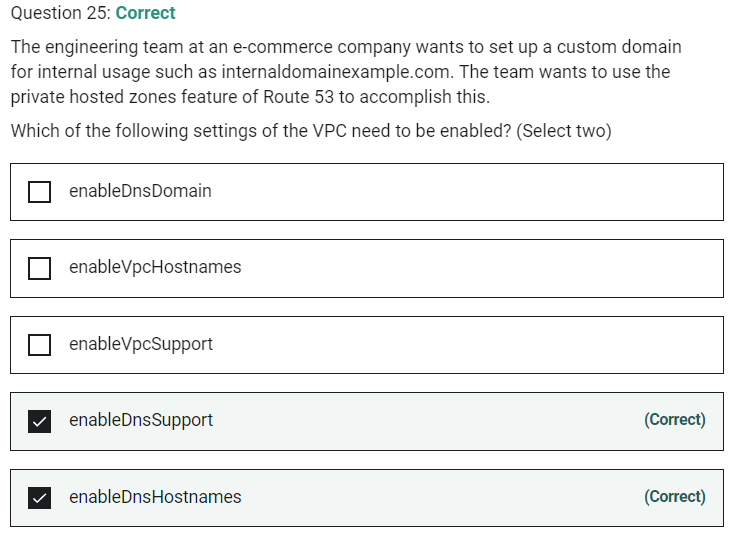
Promote the standby to primary.

Perform maintenance on the old primary, which becomes the new standby.

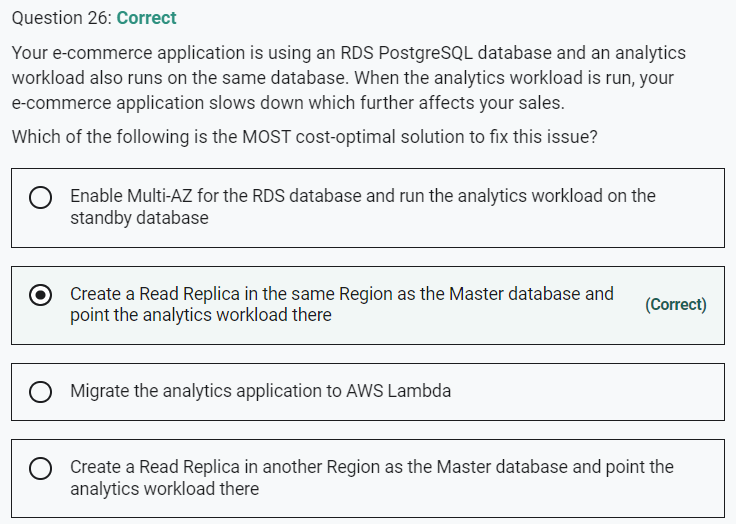
When you modify the database engine for your DB instance in a Multi-AZ deployment, then Amazon RDS upgrades both the primary and secondary DB instances at the same time. In this case, the database engine for the entire Multi-AZ deployment is shut down during the upgrade.



marked

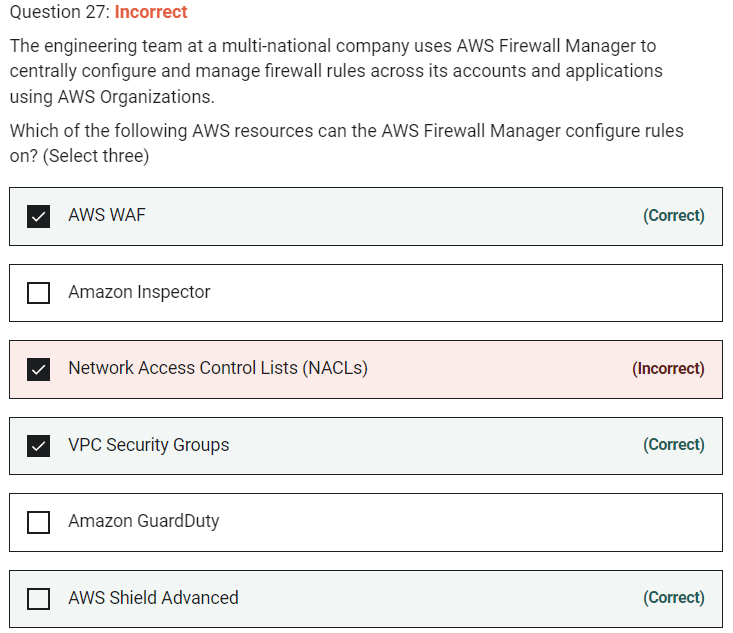


Marked

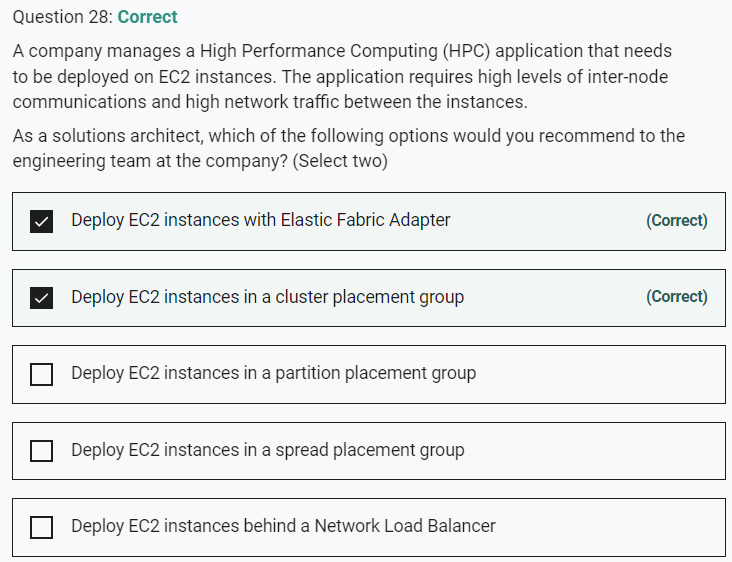


Creating a Read Replica is the answer. As we want to minimize the costs, we need to launch the Read Replica in the same Region as you are not charged for the data transfer incurred in replicating data between your source DB instance and read replica within the same AWS Region.

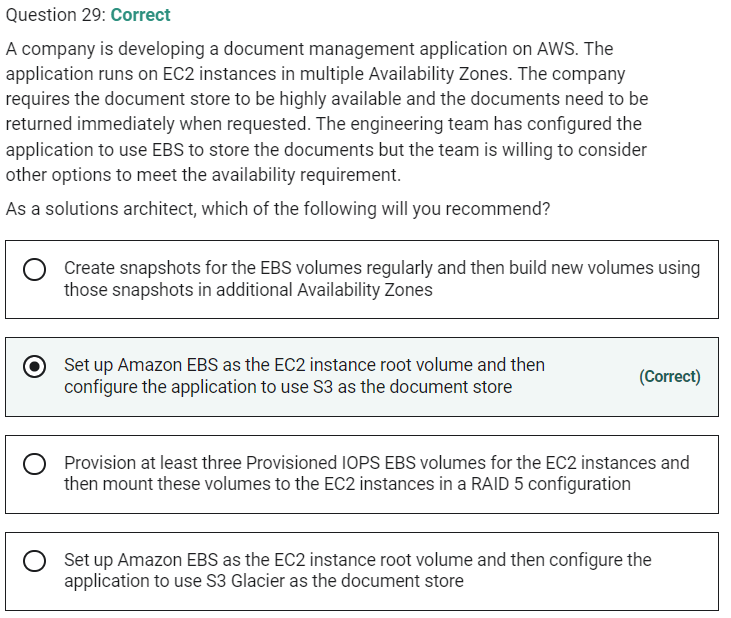
Marked



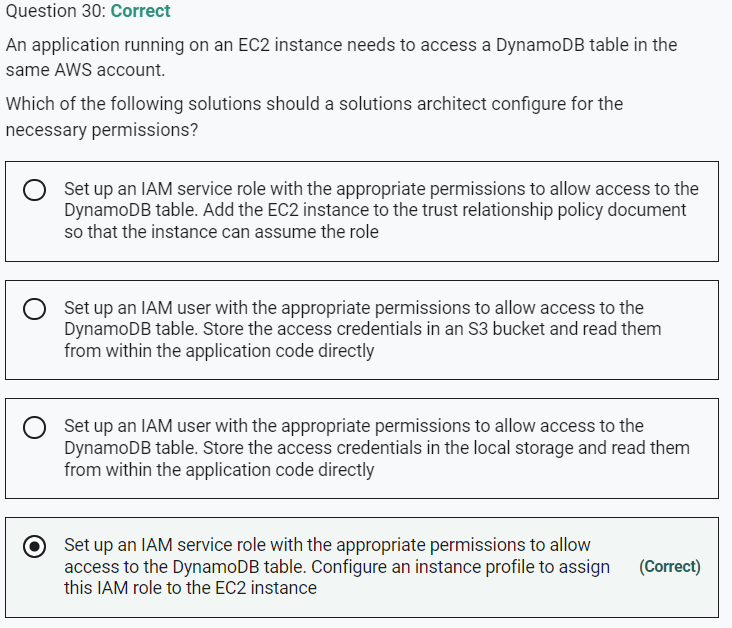
Using AWS Firewall Manager, you can centrally configure AWS WAF rules, AWS Shield Advanced protection, Amazon Virtual Private Cloud (VPC) security groups, AWS Network Firewalls, and Amazon Route 53 Resolver DNS Firewall rules across accounts and resources in your organization.

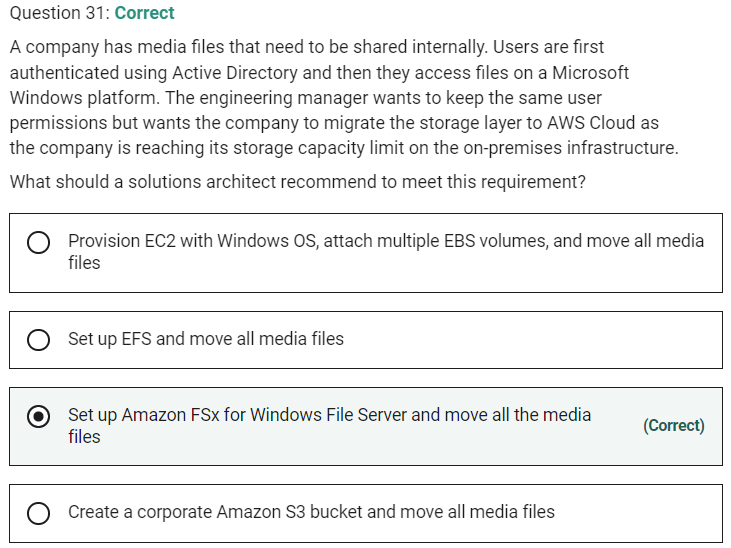


Marked

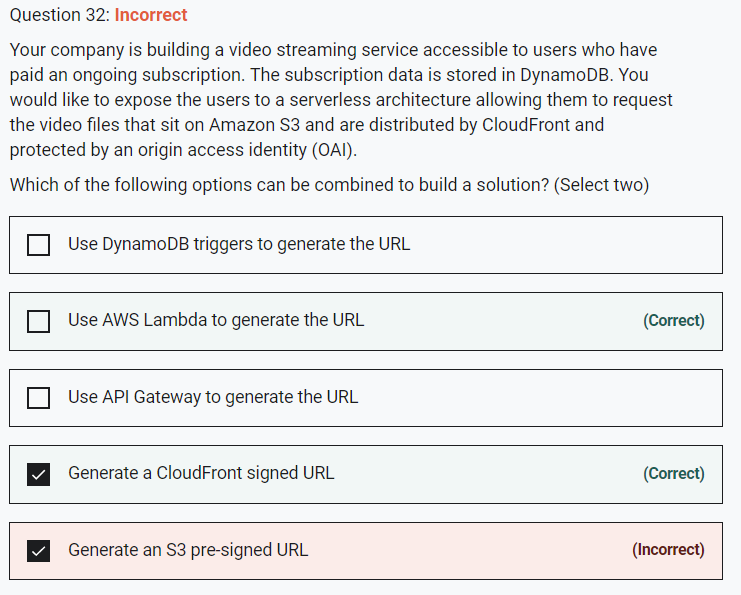


Marked





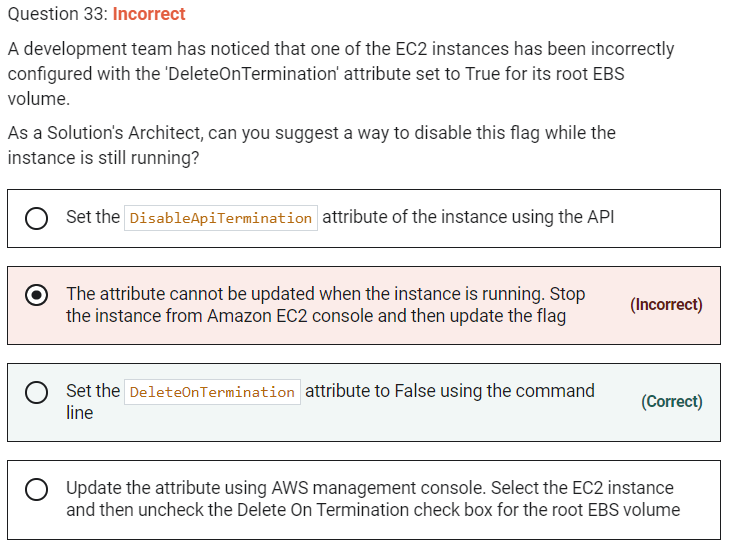
Marked



To generate this URL we must code, and Lambda is the perfect tool for running that code on the fly.

**Generate an S3 pre-signed URL** - Generating S3 pre-signed URLs would bypass CloudFront, therefore we should use CloudFront signed URLs.

Marked

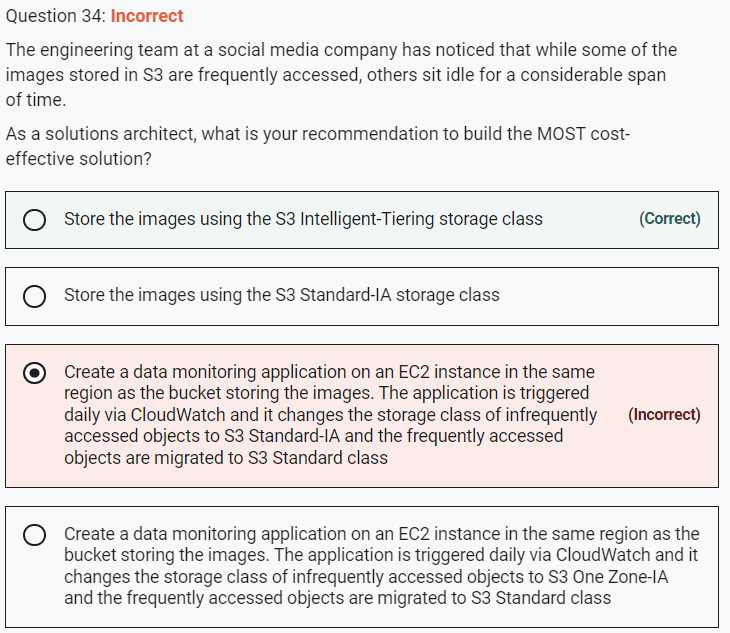


**Set the DeleteOnTermination attribute to False using the command line** - If the instance is already running, you can set DeleteOnTermination to False using the command line.

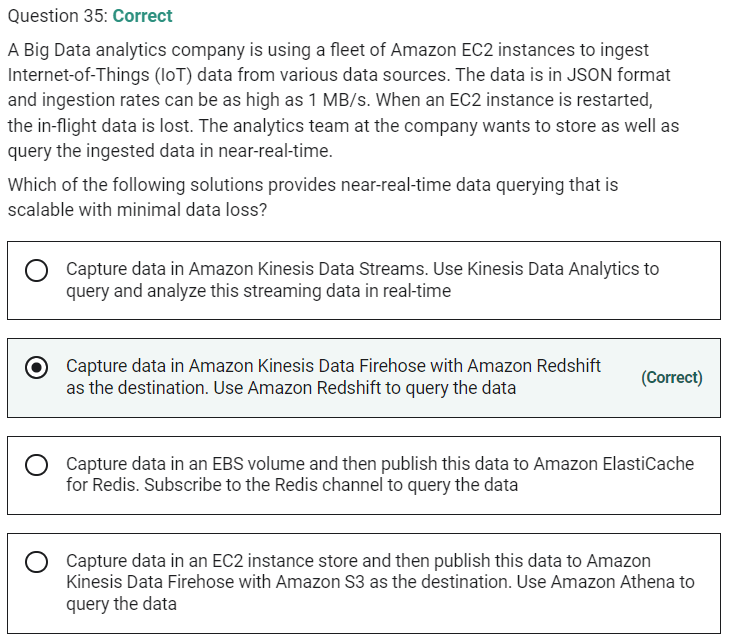
**Update the attribute using AWS management console. Select the EC2 instance and then uncheck the Delete On Termination check box for the root EBS volume** - You can set the DeleteOnTermination attribute to False when you launch a new instance. It is not possible to update this attribute of a running instance from the AWS console.

**Set the DisableApiTermination attribute of the instance using the API** - By default, you can terminate your instance using the Amazon EC2 console, command-line interface, or API. To prevent your instance from being accidentally terminated using Amazon EC2, you can enable termination protection for the instance. The DisableApiTermination attribute controls whether the instance can be terminated using the console, CLI, or API. This option cannot be used to control the delete status for the EBS volume when the instance terminates.

Marked

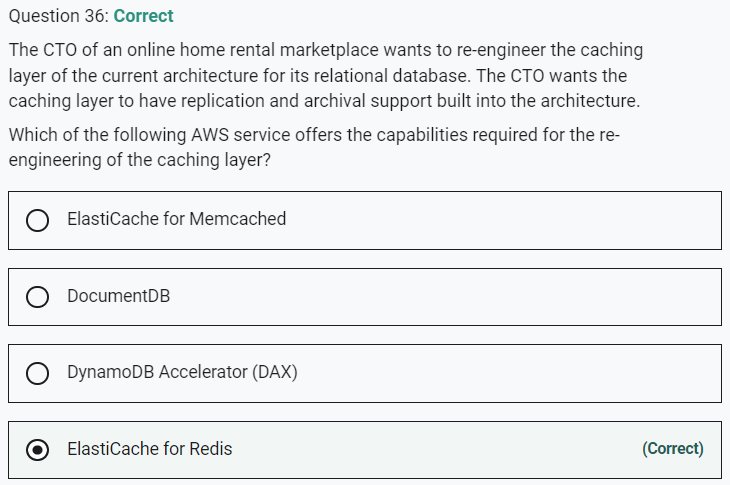


Marked



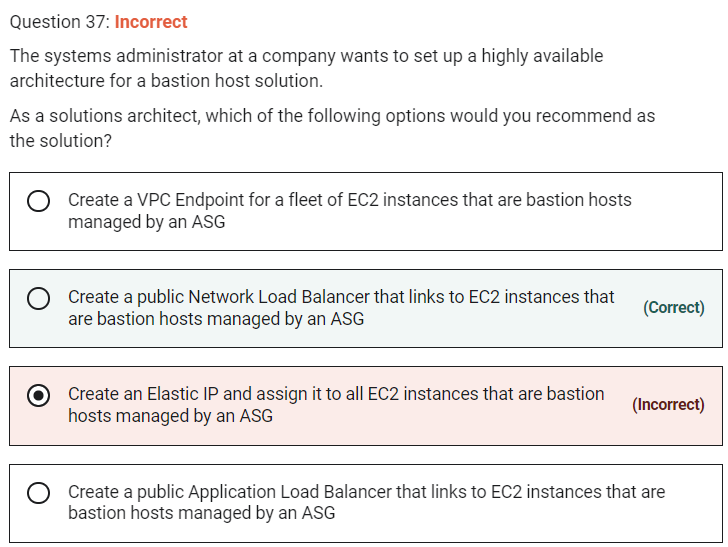
**Capture data in Amazon Kinesis Data Streams. Use Kinesis Data Analytics to query and analyze this streaming data in real-time** - For Kinesis Data Streams, you have to manually allocate the shards for scaling the data ingestion process. Kinesis Data Streams (KDS) and Kinesis Data Analytics are for real-time processing of data and cannot provide long-term storage of data unlike a database or a data warehouse. So, this option is not right for the current use case.

Marked



ElastiCache for Redis supports replication and archival snapshots

Marked



**Create a public Network Load Balancer that links to EC2 instances that are bastion hosts managed by an ASG**

You need to remember that Bastion Hosts are using the SSH protocol, which is a TCP based protocol on port 22. They must be publicly accessible.

Here, the correct answer is to use a Network Load Balancer, which supports TCP traffic, and will automatically allow you to connect to the EC2 instance in the backend.

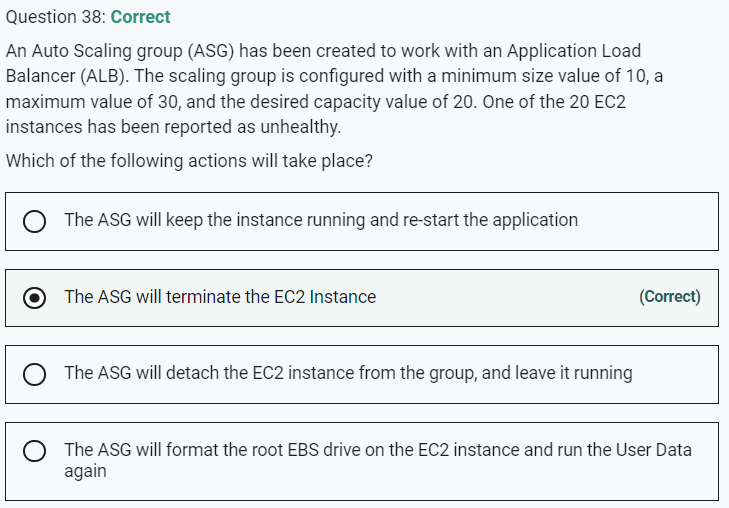
**Create an Elastic IP and assign it to all EC2 instances that are bastion hosts managed by an ASG** - An Elastic IP can only be attached to one EC2 instance at a time, so it won't provide you a highly available setup on its own. Note that if we had two Elastic IPs and two Bastion Hosts, this would work.

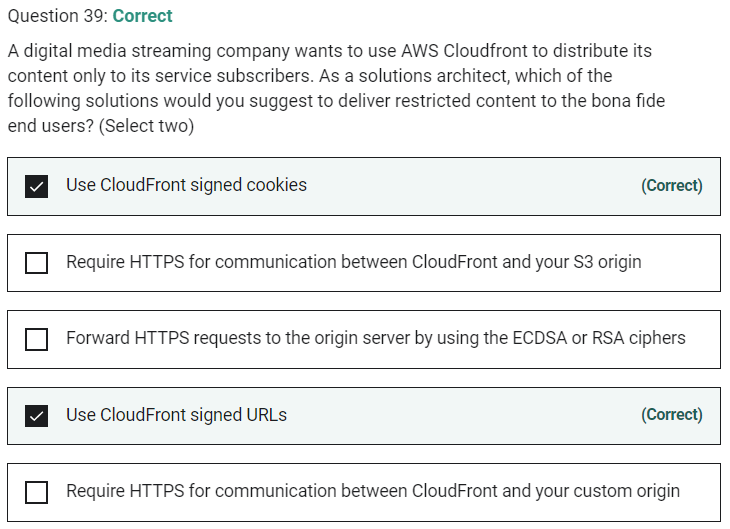
**Create a VPC Endpoint for a fleet of EC2 instances that are bastion hosts managed by an ASG**

VPC Endpoints are not used on top of EC2 instances. They're a way to access AWS services privately within your VPC (without using the public internet). This is a distractor.

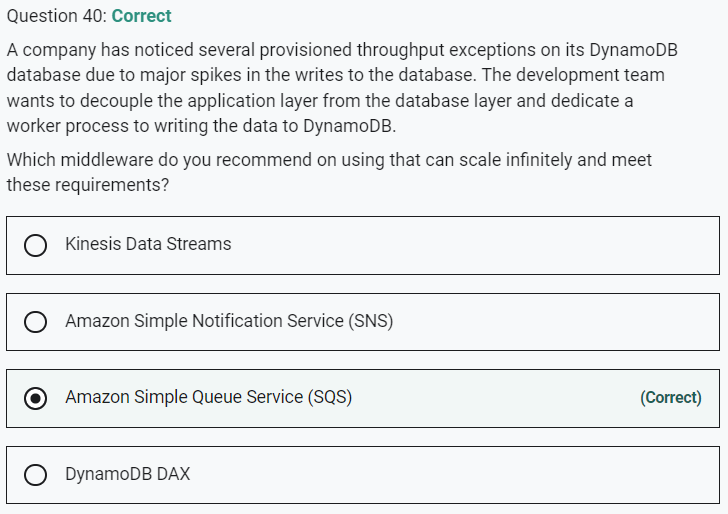
**Create a public Application Load Balancer that links to EC2 instances that are bastion hosts managed by an ASG** -

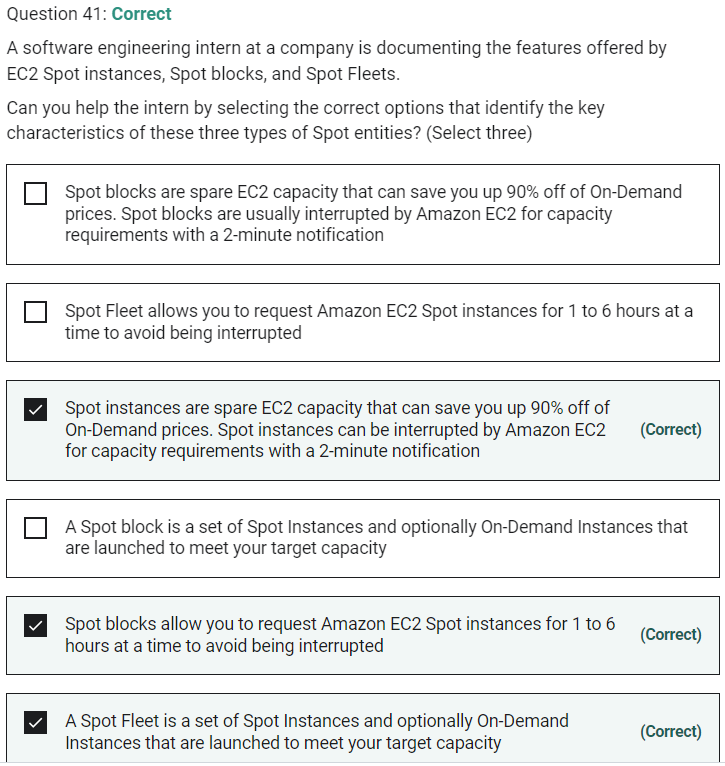
An ALB only supports HTTP traffic, which is layer 7, while the SSH protocol is based on TCP and is layer 4. So, the Application Load Balancer doesn't work.





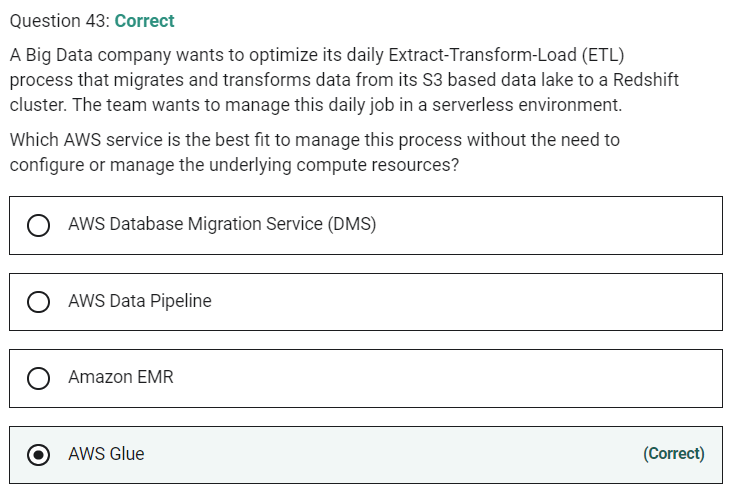
Marked

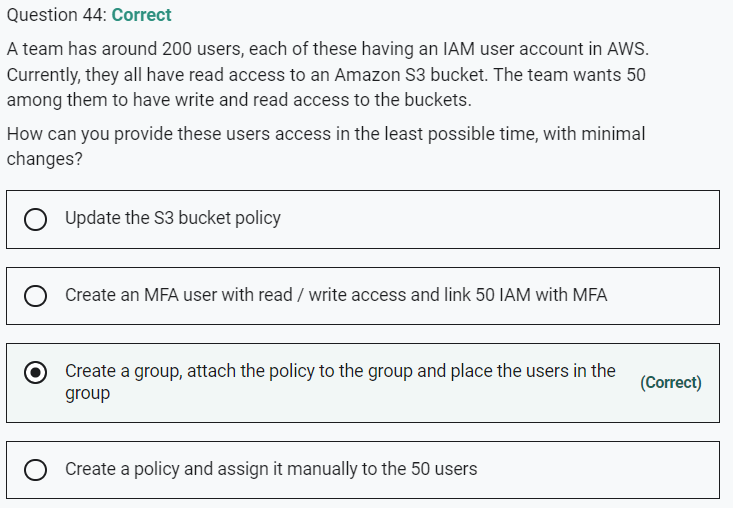




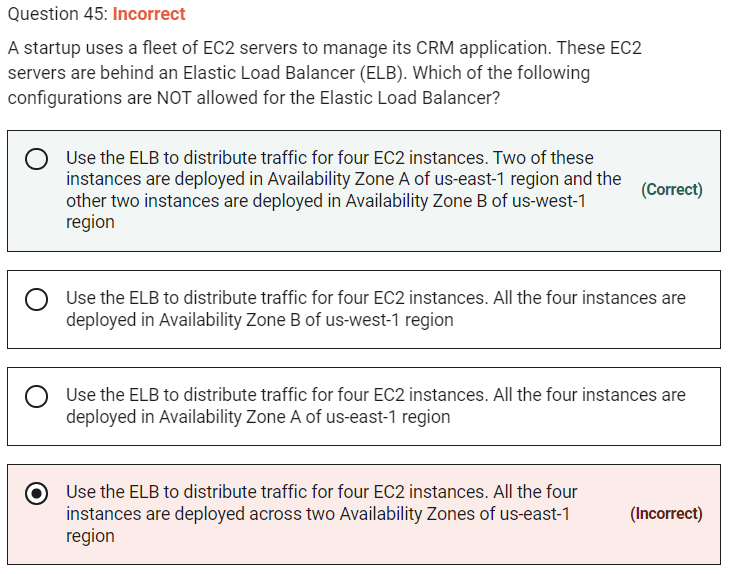
Marked

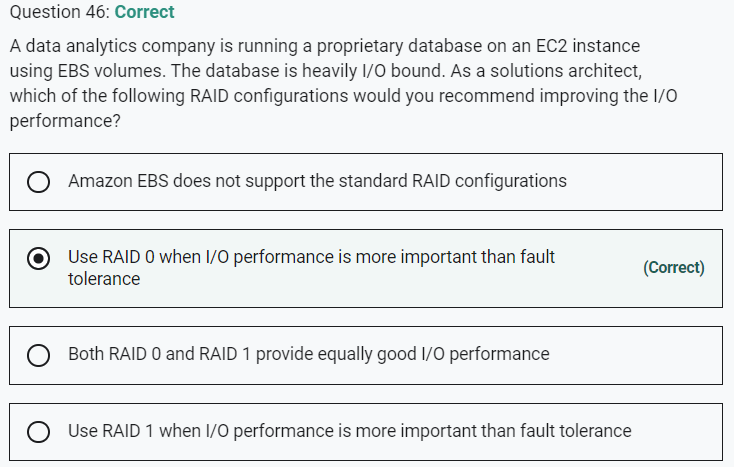


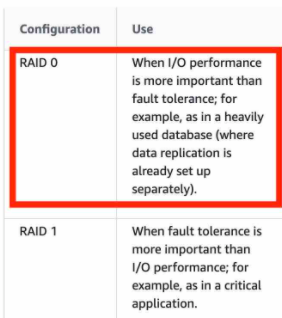




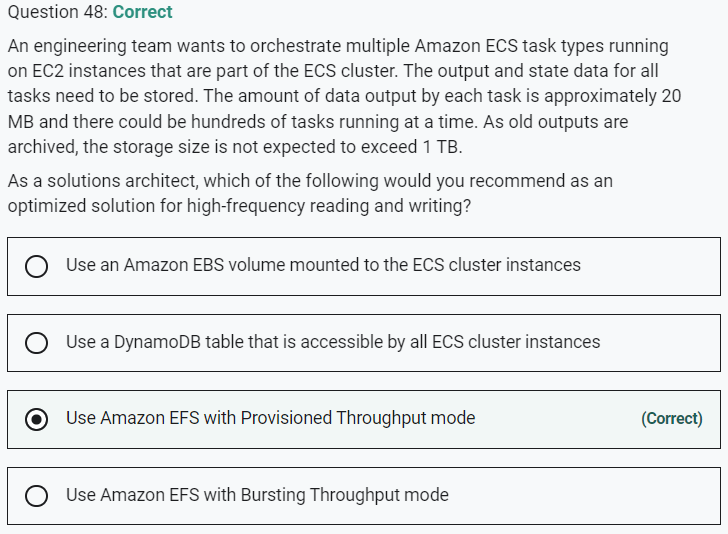
Marked



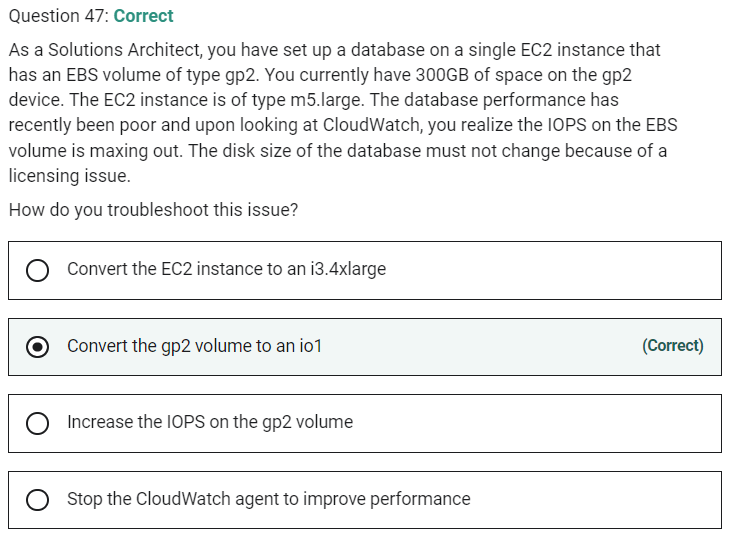


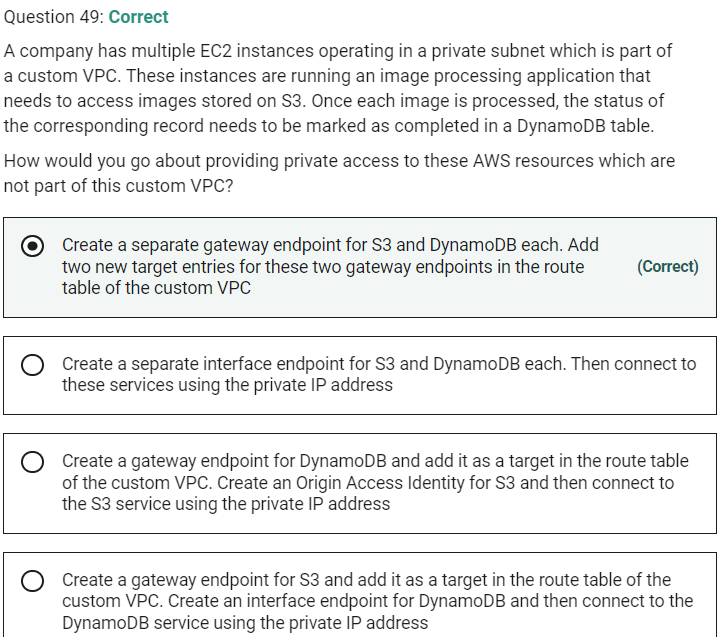


Marked

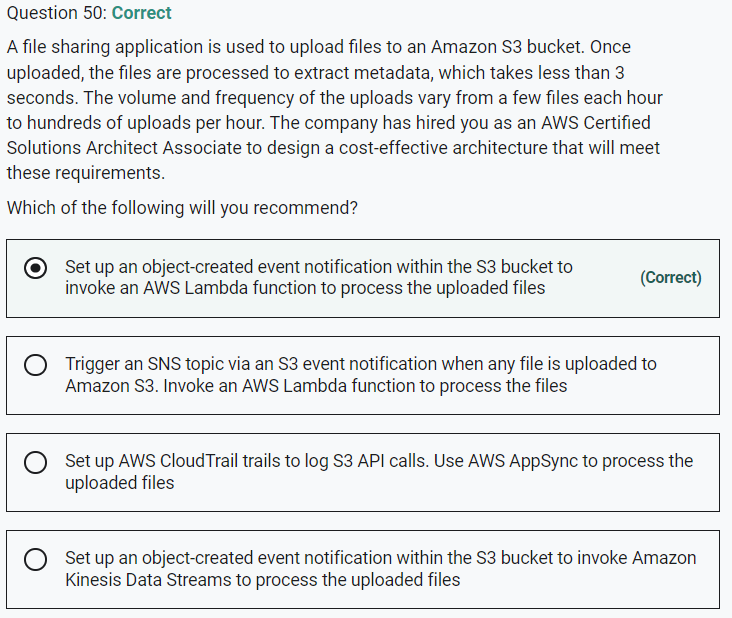


Marked





Marked



Marked

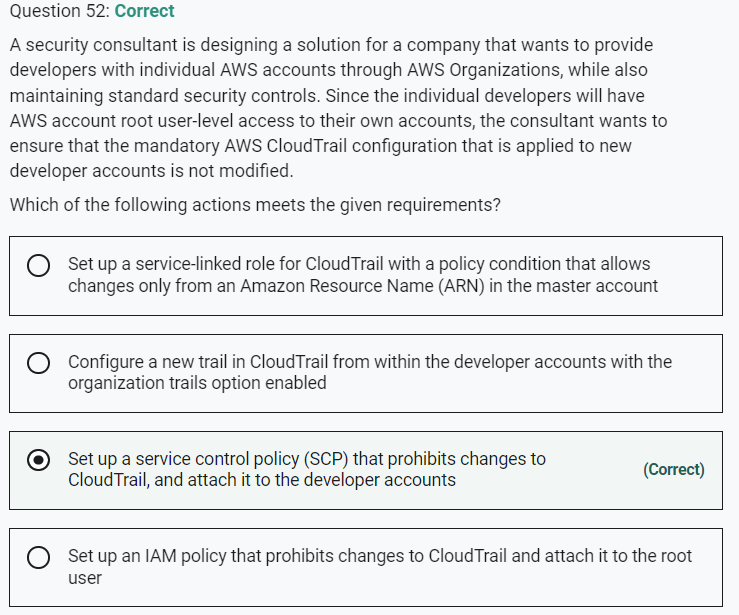


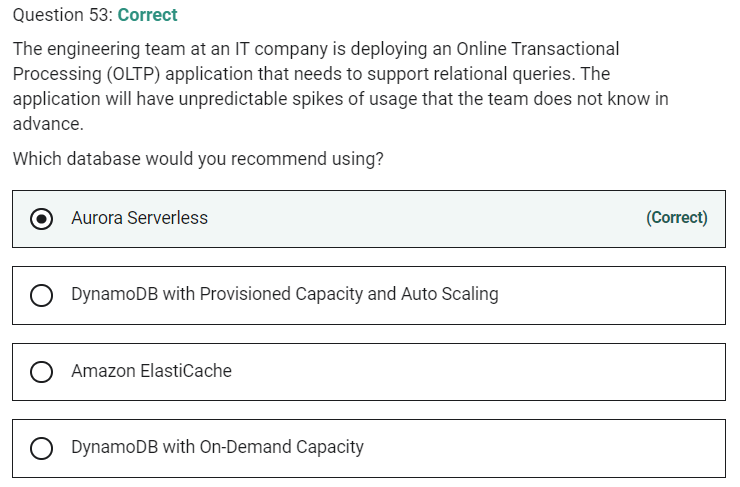
NLB, Application Load Balancer also supports TLS offloading. The Classic Load Balancer supports SSL offloading.

**Set up a Network Load Balancer in the private subnet. Create an Auto Scaling group in the public subnet and associate it with the Network Load Balancer**

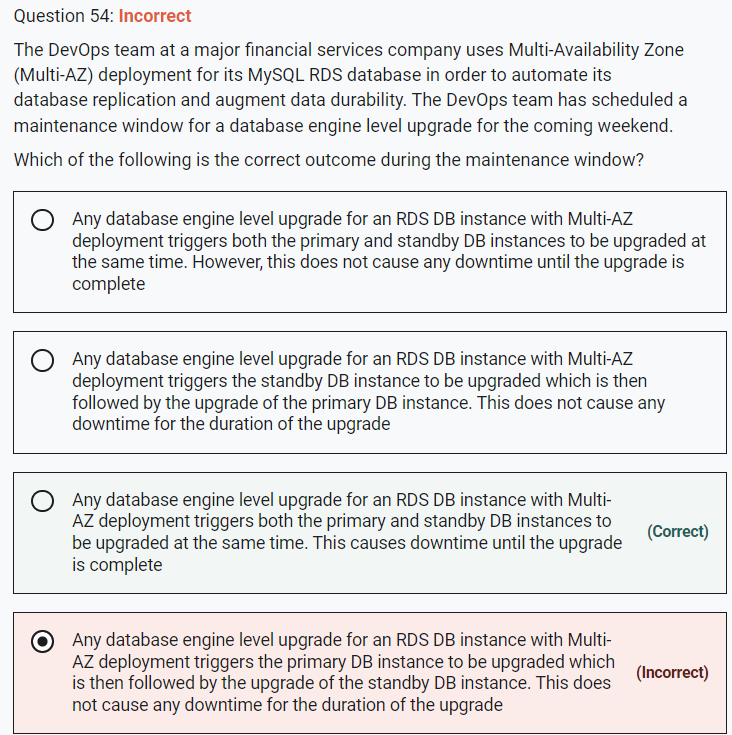
**Set up a Network Load Balancer in the private subnet. Create an Auto Scaling group in the private subnet and associate it with the Network Load Balancer**

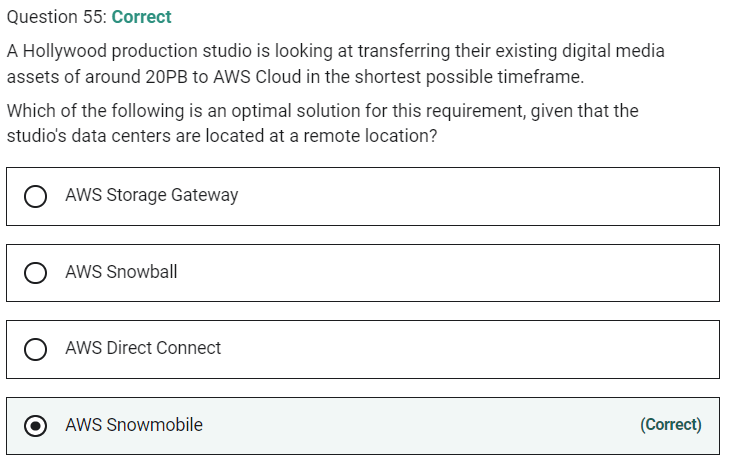
NLB should be in the public subnet as it represents the internet-facing component of the web tier. Therefore, both these options are incorrect.

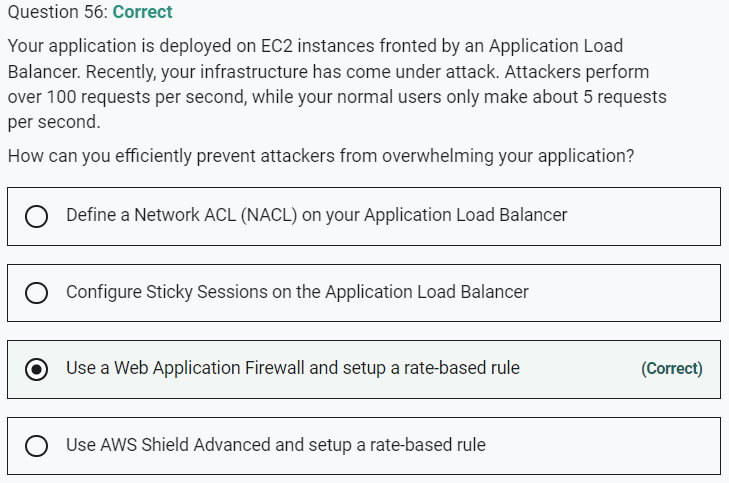




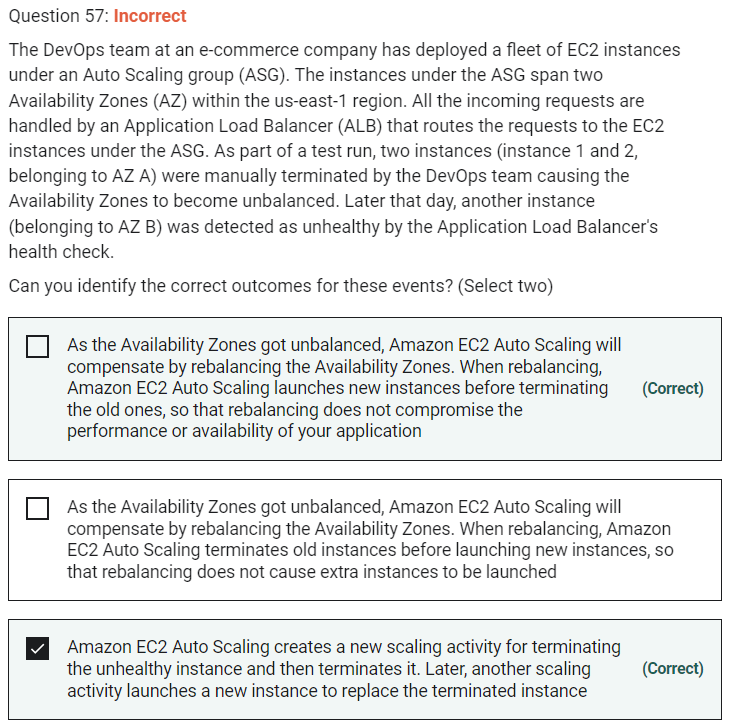
Marked

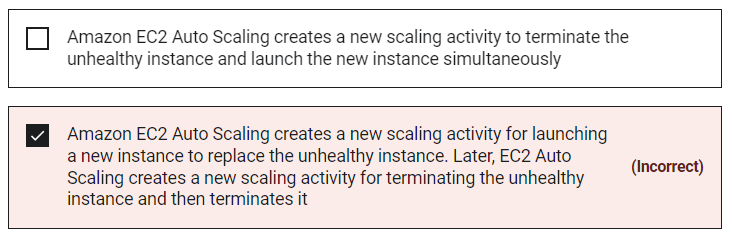






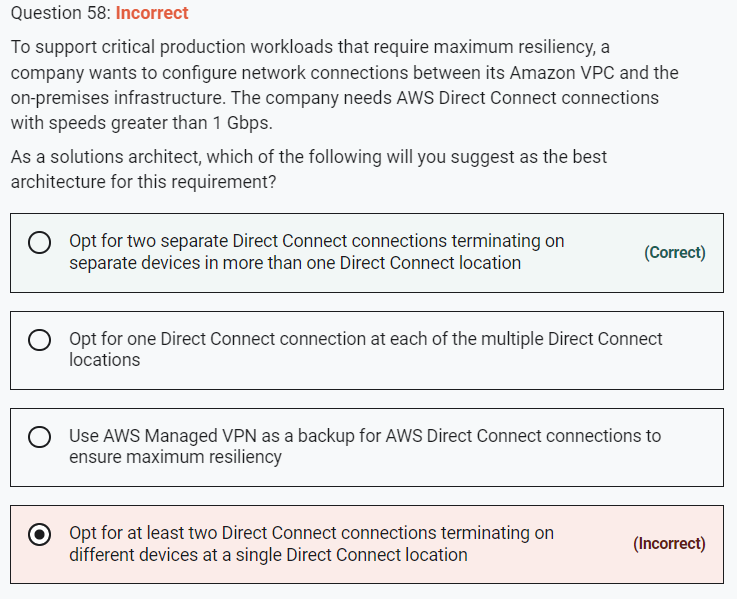
Marked



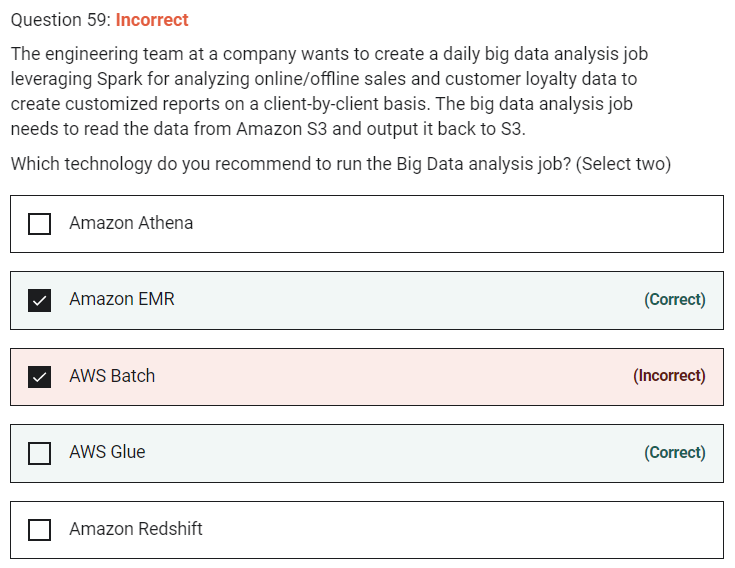


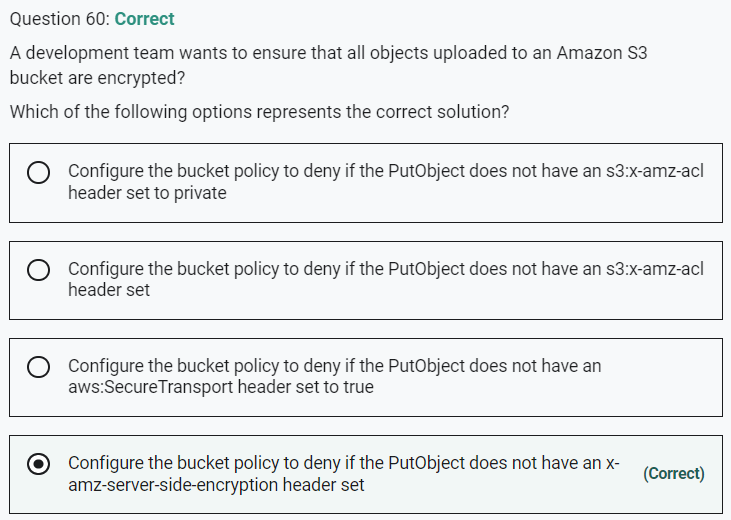
When rebalancing, Amazon EC2 Auto Scaling launches new instances before terminating the old ones, so that rebalancing does not compromise the performance or availability of your application. Therefore, this option is correct.

Marked

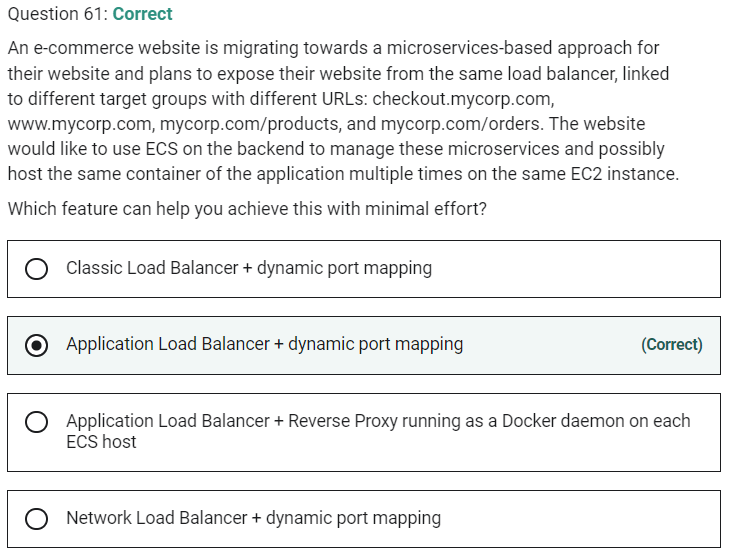


Marked

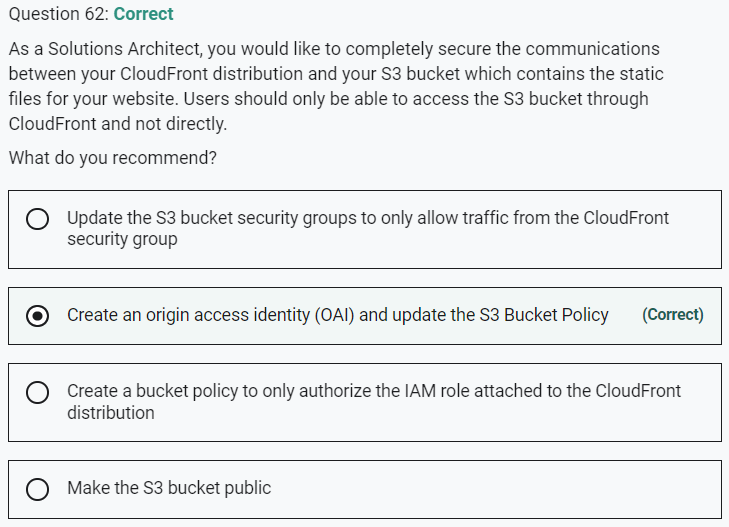




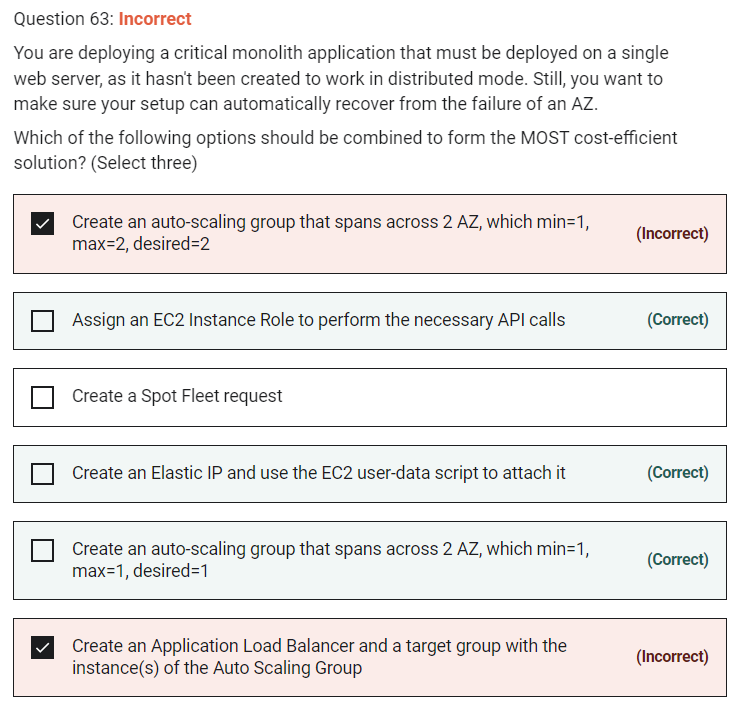
Marked



Dynamic port mapping with an Application Load Balancer makes it easier to run multiple tasks on the same Amazon ECS service on an Amazon ECS cluster.



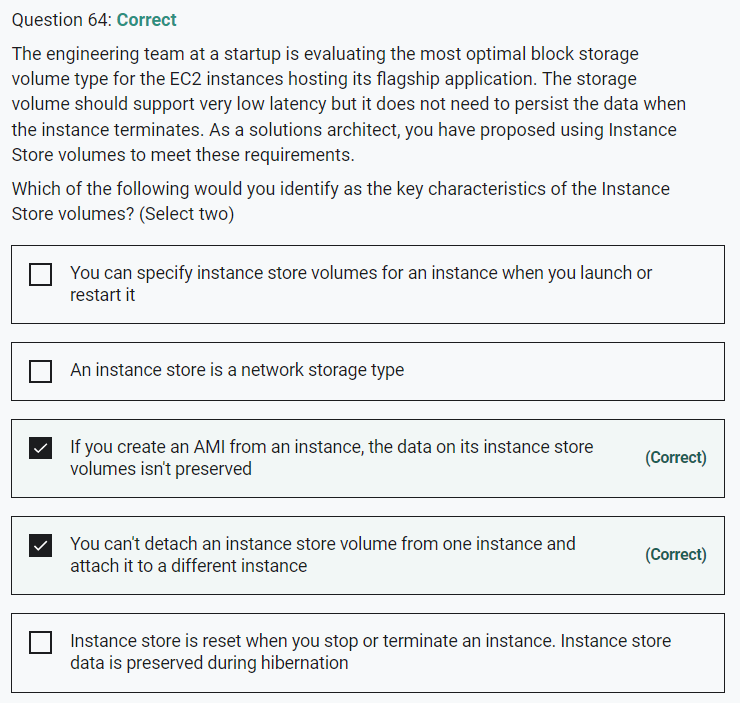
Marked



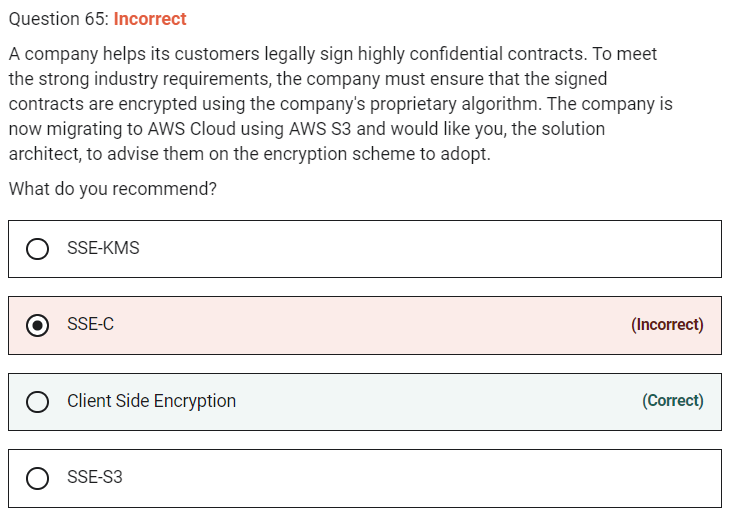
**Assign an EC2 Instance Role to perform the necessary API calls**

For that Elastic IP to be attached to our EC2 instance, we must use an EC2 user data script, and our EC2 instance must have the correct IAM permissions to perform the API call, so we need an EC2 instance role.

Marked



Marked



**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:

Because the company has its proprietary encryption algorithm, you have to leverage client-side encryption.

**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. SSE-KMS provides you with an audit trail that shows when your CMK was used and by whom.

**SSE-S3** - When you use Server-Side Encryption with Amazon S3-Managed Keys (SSE-S3), each object is encrypted with a unique key.

**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.