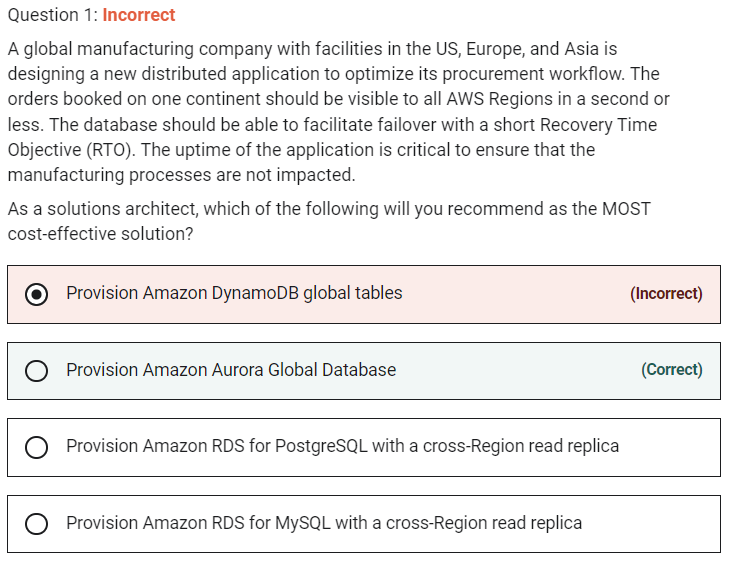
SAA-CO2

Practice Test- 3

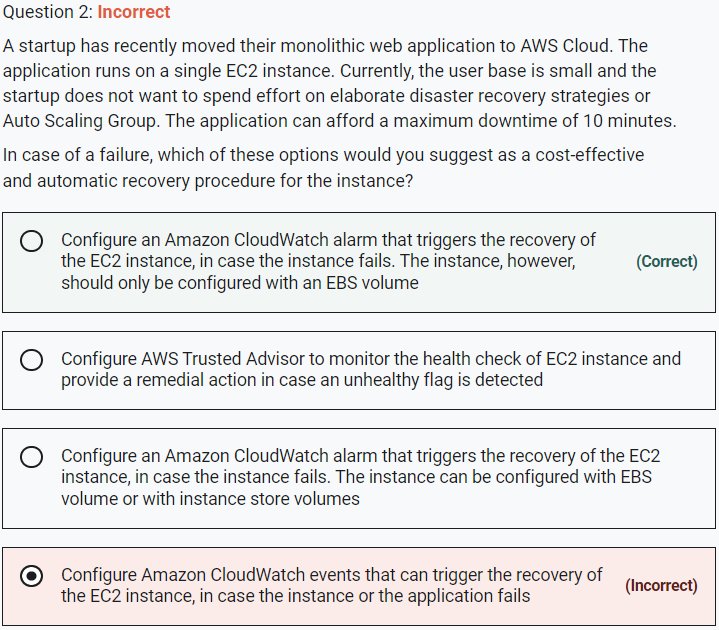
Marked



**Provision Amazon Aurora Global Database**

An Aurora global database provides more comprehensive failover capabilities than the failover provided by a default Aurora DB cluster. By using an Aurora global database, you can plan for and recover from disaster fairly quickly.

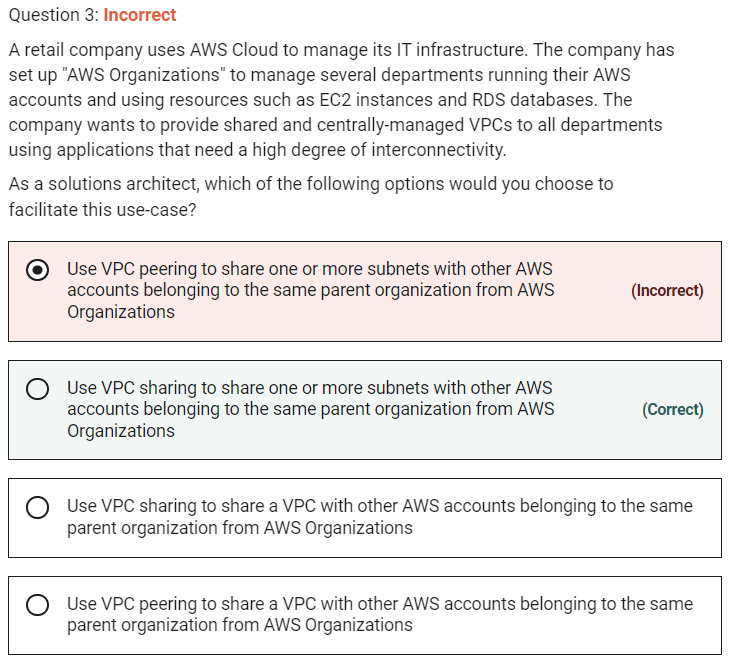
**Provision Amazon DynamoDB global tables** - Aurora Global Database is good for applications that need to support cross-Region reads with low latency updates and the ability to quickly failover between regions. DynamoDB global tables provide cross-region active-active capabilities with high performance, but you lose some of the data access flexibility that comes with SQL-based databases. Due to the active-active configuration of DynamoDB global tables, there is no concept of failover because the application writes to the table in its region, and then the data is replicated to keep the other regions' table in sync. DynamoDB global tables is a much costlier solution than Aurora Global Database for the given requirement.



**Configure Amazon CloudWatch events that can trigger the recovery of the EC2 instance, in case the instance or the application fails** - You cannot use CloudWatch events to directly trigger the recovery of the EC2 instance.

**Configure an Amazon CloudWatch alarm that triggers the recovery of the EC2 instance, in case the instance fails. The instance can be configured with EBS volume or with instance store volumes** - The recover action is supported only on instances that have EBS volumes configured on them, instance store volumes are not supported for automatic recovery by CloudWatch alarms.

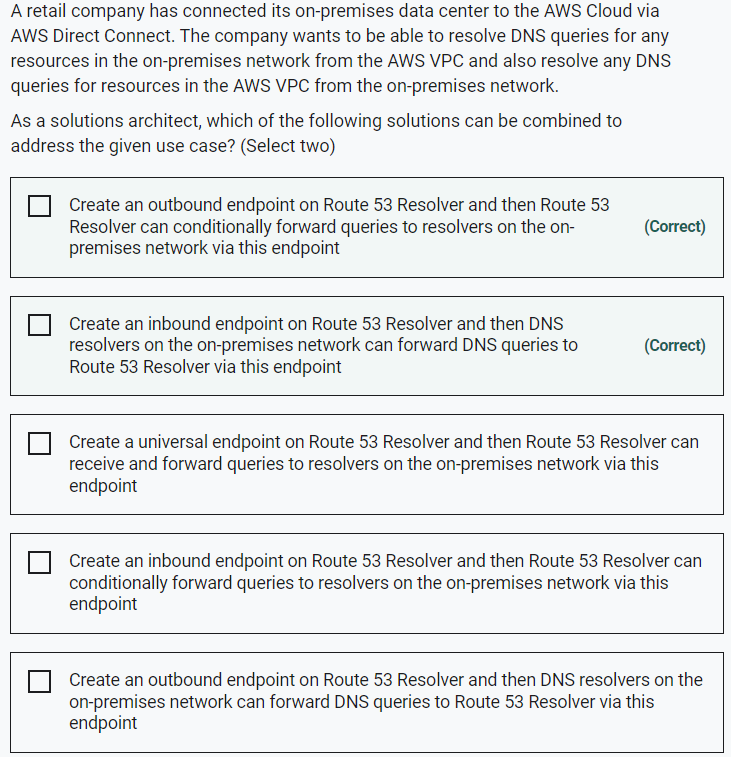
**Configure AWS Trusted Advisor to monitor the health check of EC2 instance and provide a remedial action in case an unhealthy flag is detected** - You can use Amazon CloudWatch Events to detect and react to changes in the status of Trusted Advisor checks. This support is only available with AWS Business Support and AWS Enterprise Support. Trusted Advisor by itself does not support health checks of EC2 instances or their recovery.



**Use VPC sharing to share one or more subnets with other AWS accounts belonging to the same parent organization from AWS Organizations**

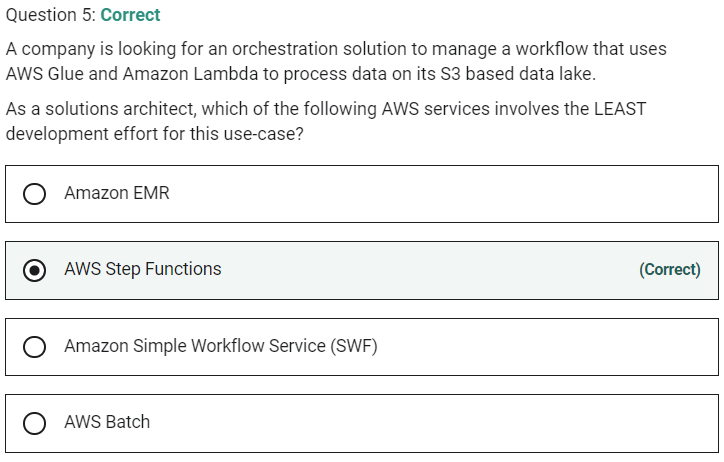
VPC sharing (part of Resource Access Manager) allows multiple AWS accounts to create their application resources such as EC2 instances, RDS databases, Redshift clusters, and Lambda functions, into shared and centrally-managed Amazon Virtual Private Clouds (VPCs). To set this up, the account that owns the VPC (owner) shares one or more subnets with other accounts (participants) that belong to the same organization from AWS Organizations. After a subnet is shared, the participants can view, create, modify, and delete their application resources in the subnets shared with them. Participants cannot view, modify, or delete resources that belong to other participants or the VPC owner.

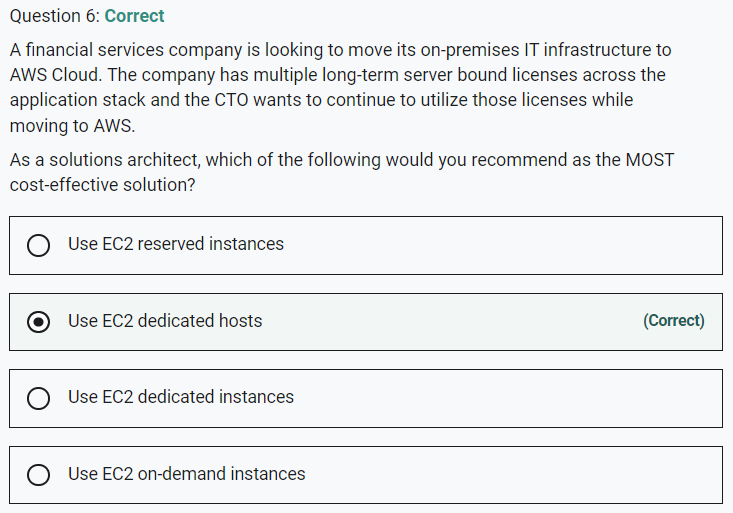
**Use VPC sharing to share a VPC with other AWS accounts belonging to the same parent organization from AWS Organizations** - Using VPC sharing, an account that owns the VPC (owner) shares one or more subnets with other accounts (participants) that belong to the same organization from AWS Organizations. The owner account cannot share the VPC itself. Therefore this option is incorrect.



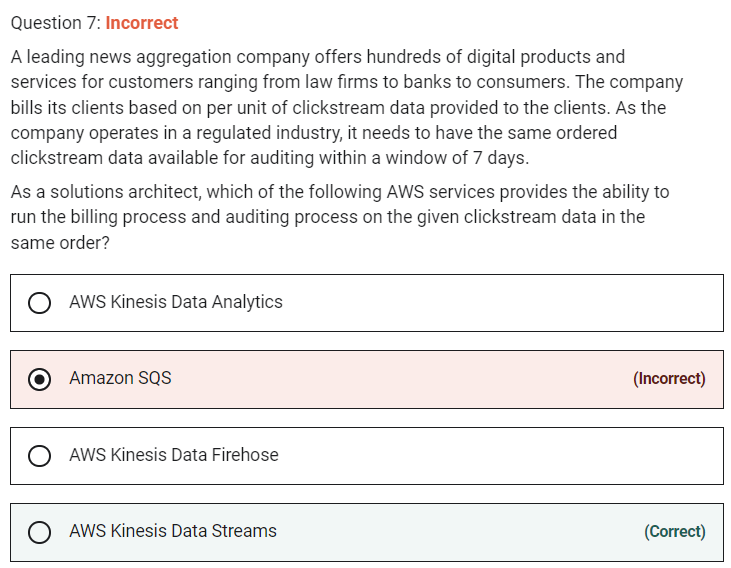
To resolve any DNS queries for resources in the AWS VPC from the on-premises network, you can create an inbound endpoint on Route 53 Resolver and then DNS resolvers on the on-premises network can forward DNS queries to Route 53 Resolver via this endpoint.

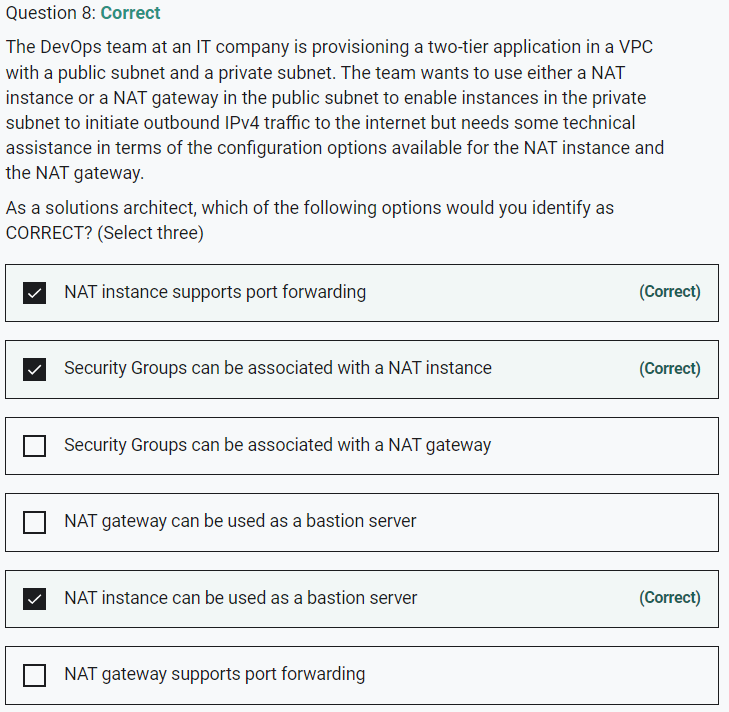
To resolve DNS queries for any resources in the on-premises network from the AWS VPC, you can create an outbound endpoint on Route 53 Resolver and then Route 53 Resolver can conditionally forward queries to resolvers on the on-premises network via this endpoint.

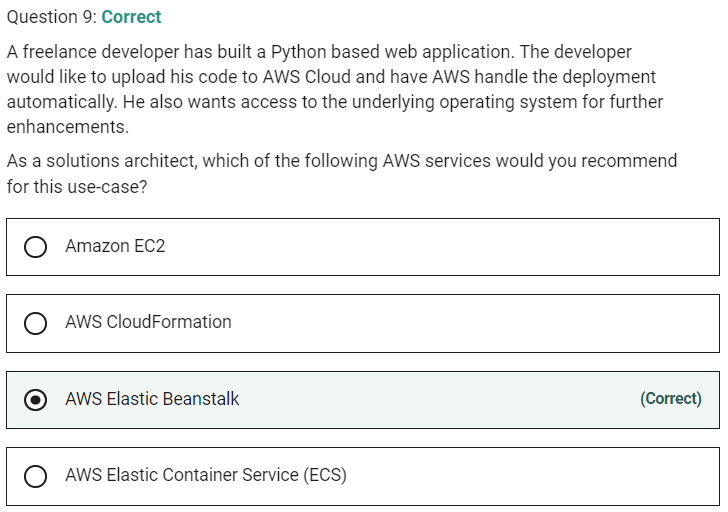


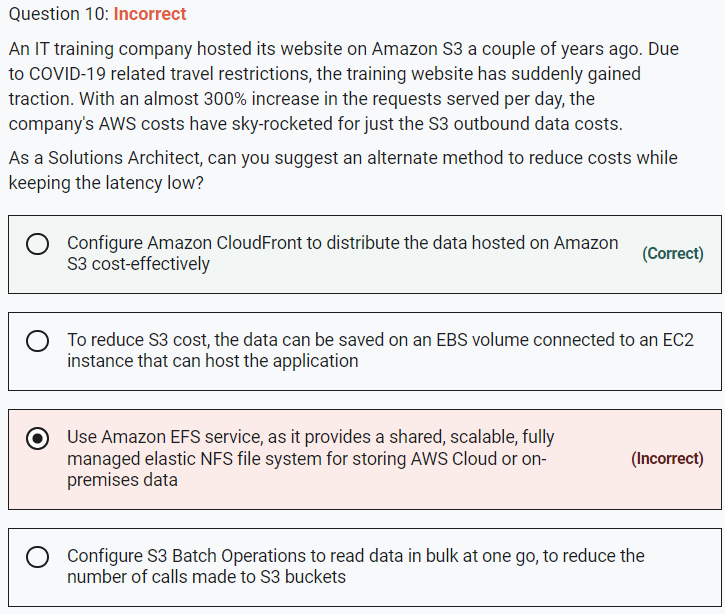


**Use EC2 dedicated instances** - Dedicated instances are Amazon EC2 instances that run in a VPC on hardware that's dedicated to a single customer. Your dedicated instances are physically isolated at the host hardware level from instances that belong to other AWS accounts. Dedicated instances may share hardware with other instances from the same AWS account that are not dedicated instances. Dedicated instances cannot be used for existing server-bound software licenses.









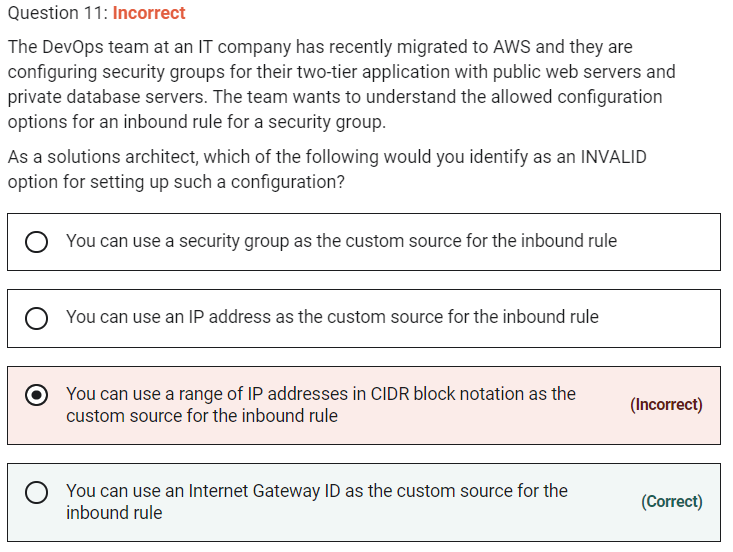
**Configure Amazon CloudFront to distribute the data hosted on Amazon S3, cost-effectively** - Storing content with S3 provides a lot of advantages. But to help optimize your application’s performance and security while effectively managing cost, AWS recommends that you also set up Amazon CloudFront to work with your S3 bucket to serve and protect the content.

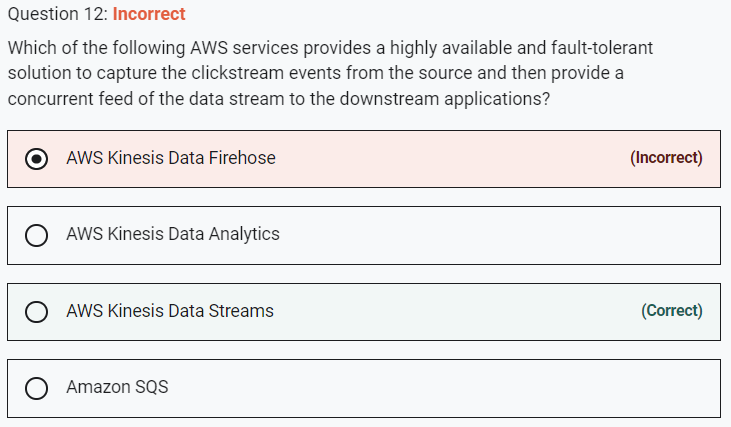
CloudFront is a content delivery network (CDN) service that delivers static and dynamic web content, video streams, and APIs around the world, securely and at scale. By design, delivering data out of CloudFront can be more cost-effective than delivering it from S3 directly to your users.

**To reduce S3 cost, the data can be saved on an EBS volume connected to an EC2 instance that can host the application** - EBS volumes are fast and are relatively cheap (though S3 is still a cheaper alternative). But, EBS volumes are accessible only through EC2 instances and are bound to a specific region.

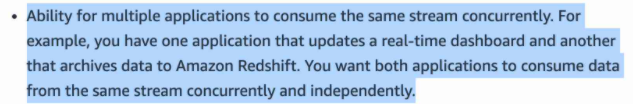
**Use Amazon Elastic File System (Amazon EFS), as it provides a shared, scalable, fully managed elastic NFS file system for storing AWS Cloud or on-premises data** - EFS is a shareable file system that can be mounted onto EC2 instances. EFS is costlier than EBS and not a solution if the company is looking at reducing costs.

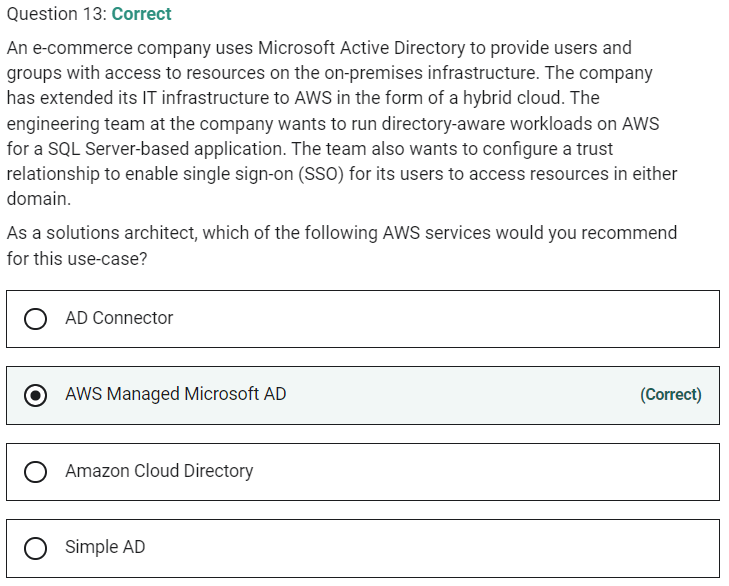
**Configure S3 Batch Operations to read data in bulk at one go, to reduce the number of calls made to S3 buckets** - This statement is incorrect and given only as a distractor. You can use S3 Batch Operations to perform large-scale batch operations on Amazon S3 objects, and it has nothing to do with content distribution.

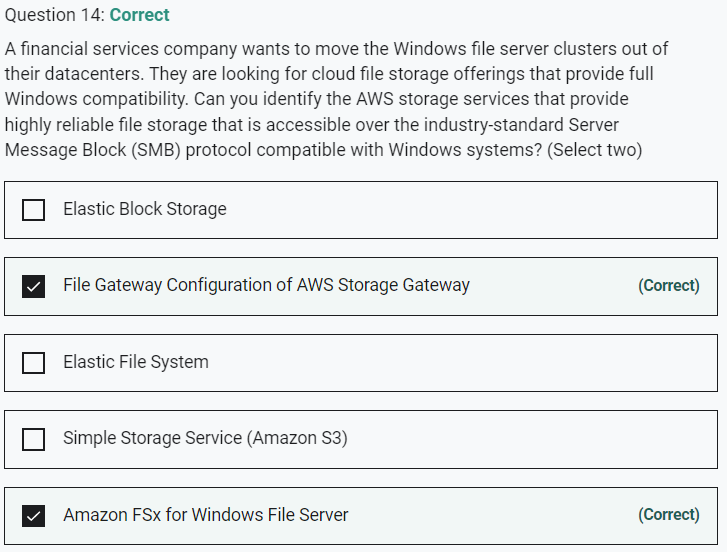


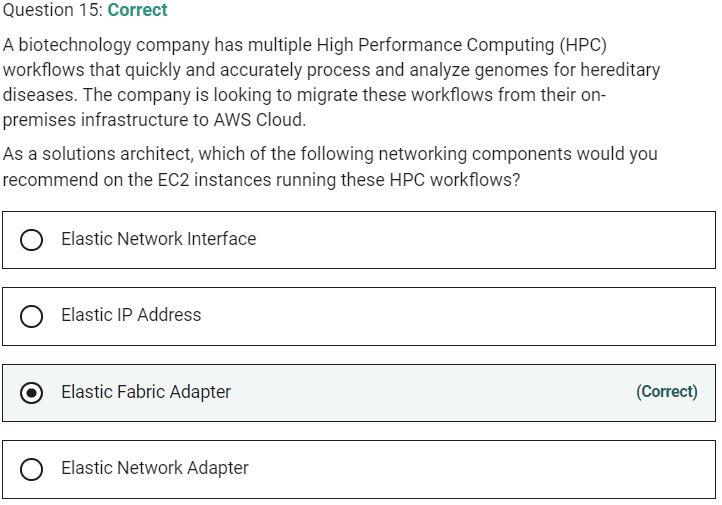


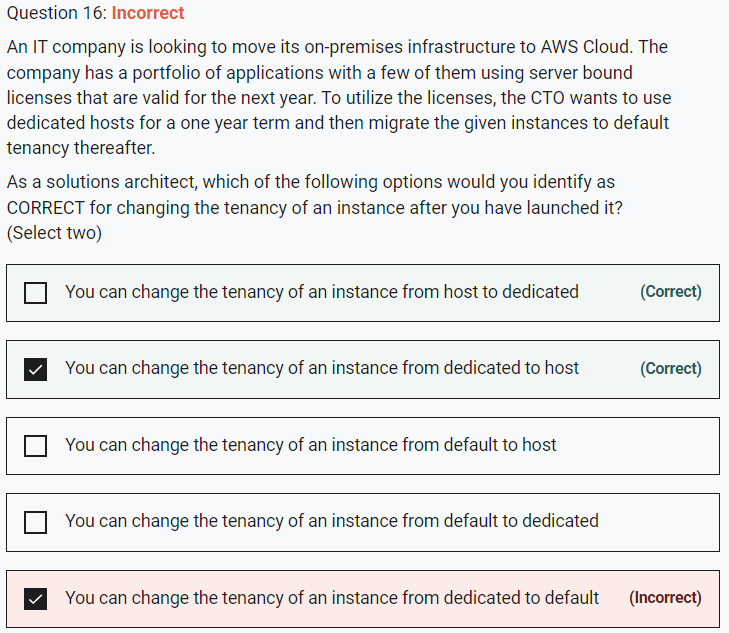
KDS provides the ability for multiple applications to consume the same stream concurrently

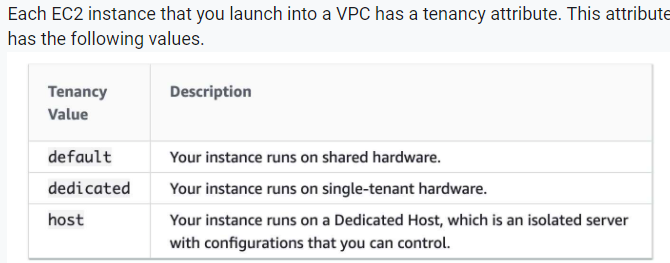


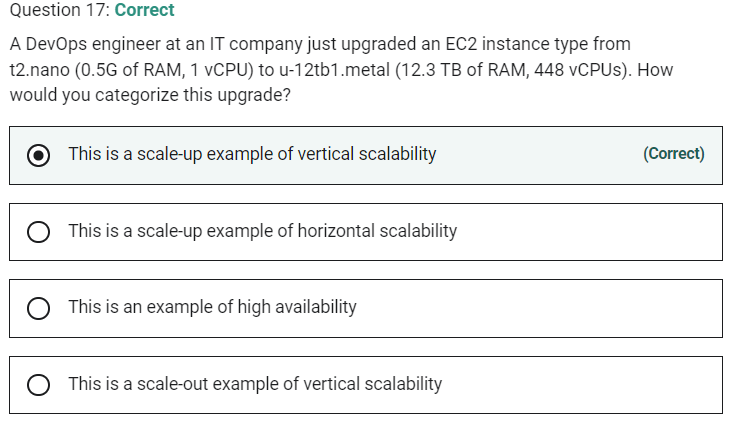


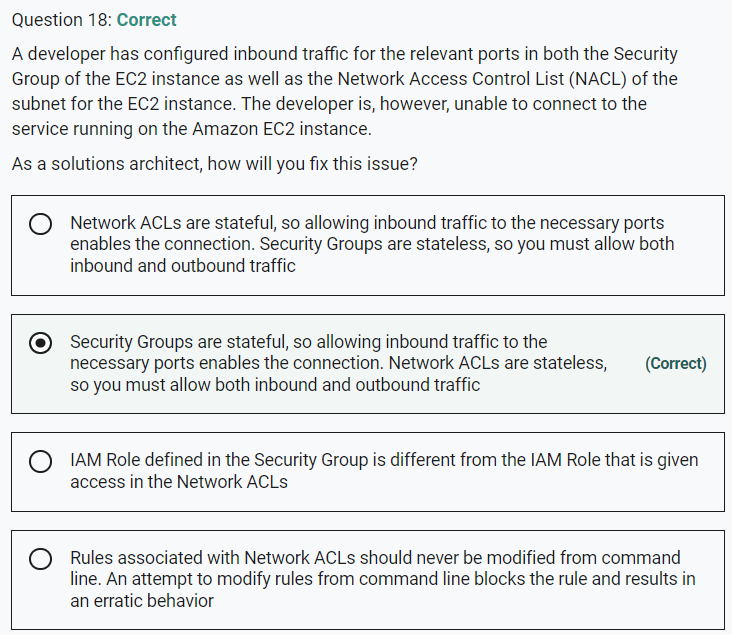


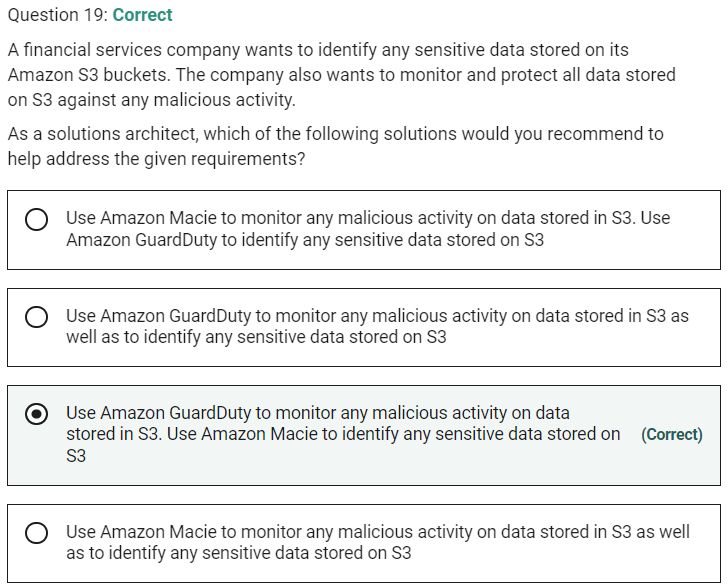


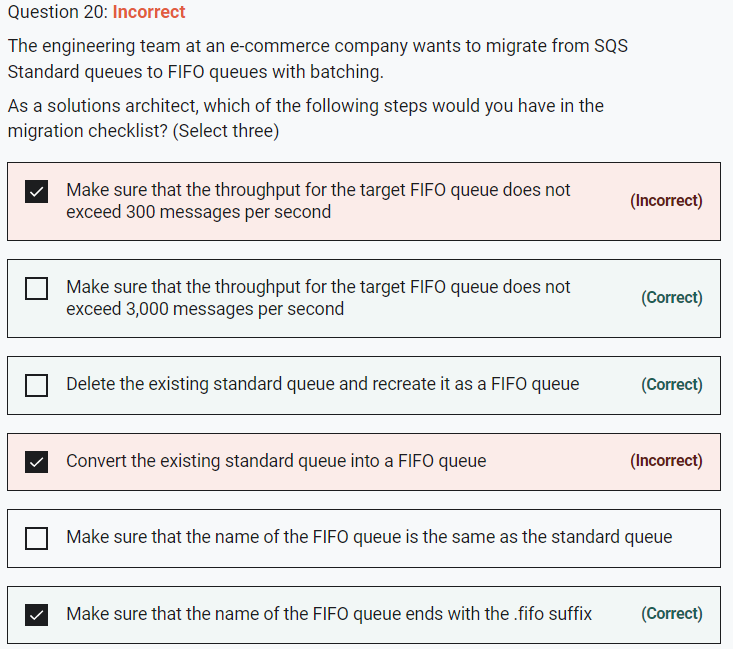








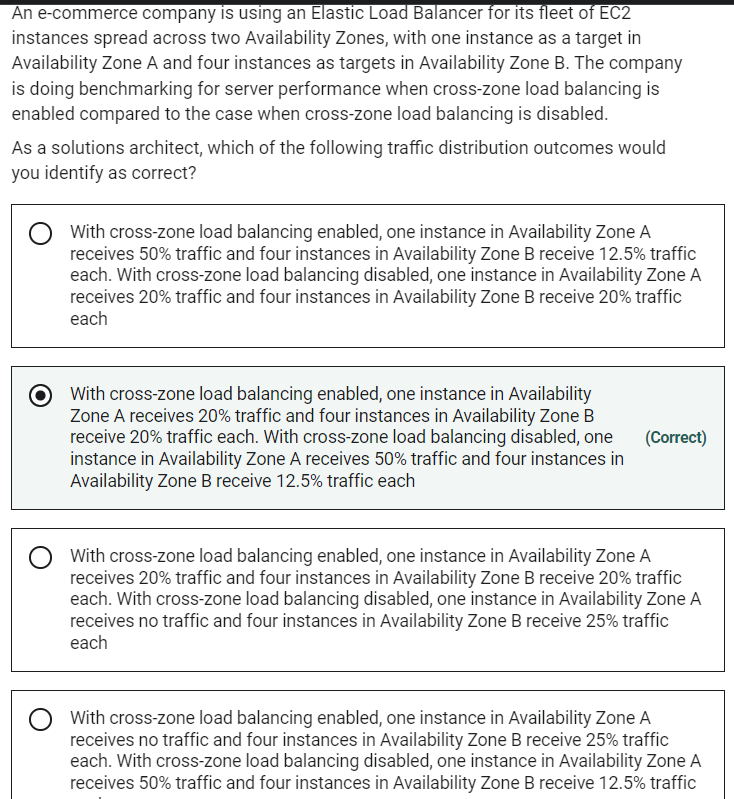


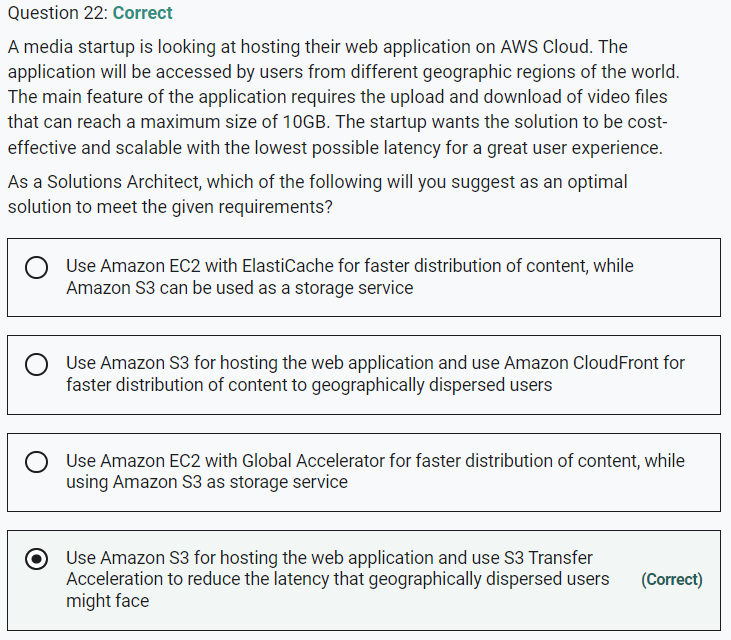


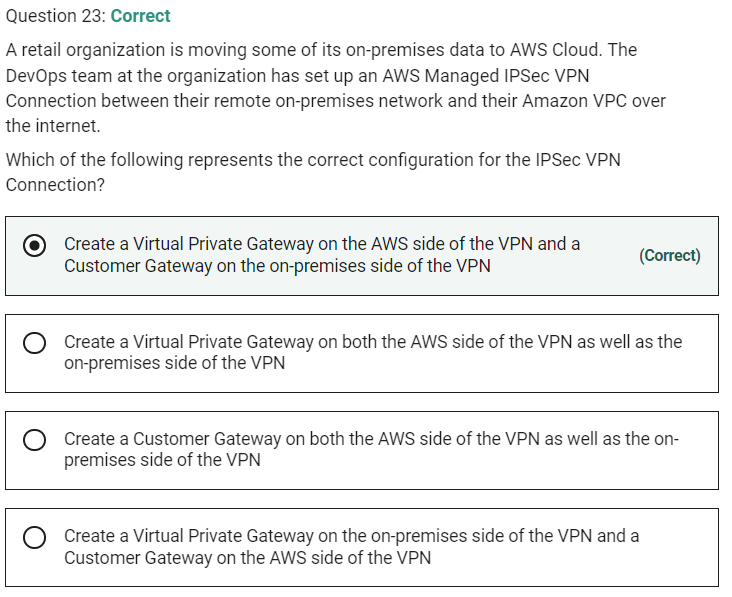
By default, FIFO queues support up to 3,000 messages per second with batching, or up to 300 messages per second (300 send, receive, or delete operations per second) without batching. Therefore, using batching you can meet a throughput requirement of upto 3,000 messages per second.

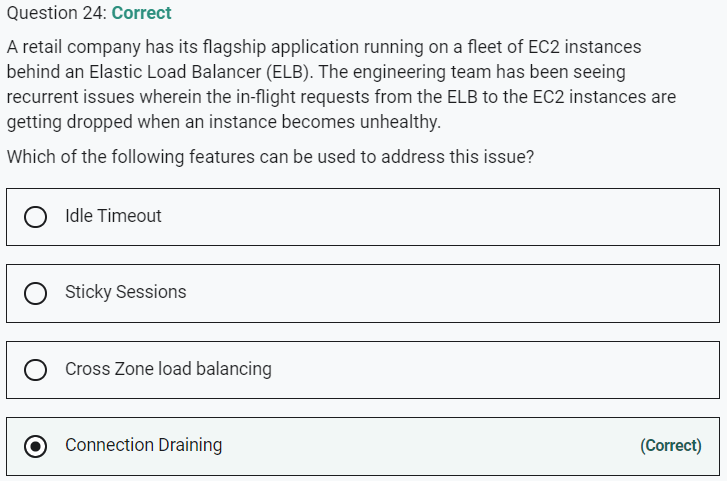
The name of a FIFO queue must end with the .fifo suffix. The suffix counts towards the 80-character queue name limit. To determine whether a queue is FIFO, you can check whether the queue name ends with the suffix.

If you have an existing application that uses standard queues and you want to take advantage of the ordering or exactly-once processing features of FIFO queues, you need to configure the queue and your application correctly. You can't convert an existing standard queue into a FIFO queue. To make the move, you must either create a new FIFO queue for your application or delete your existing standard queue and recreate it as a FIFO queue.

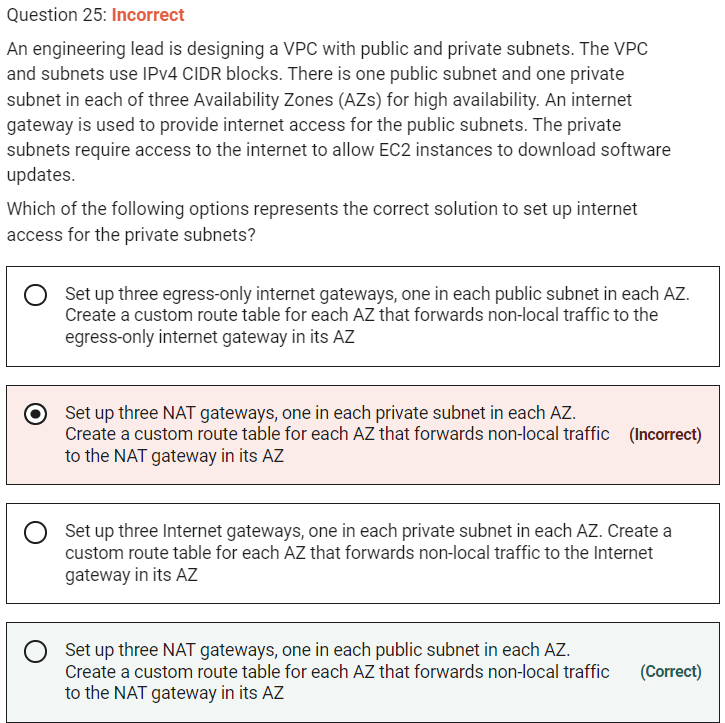


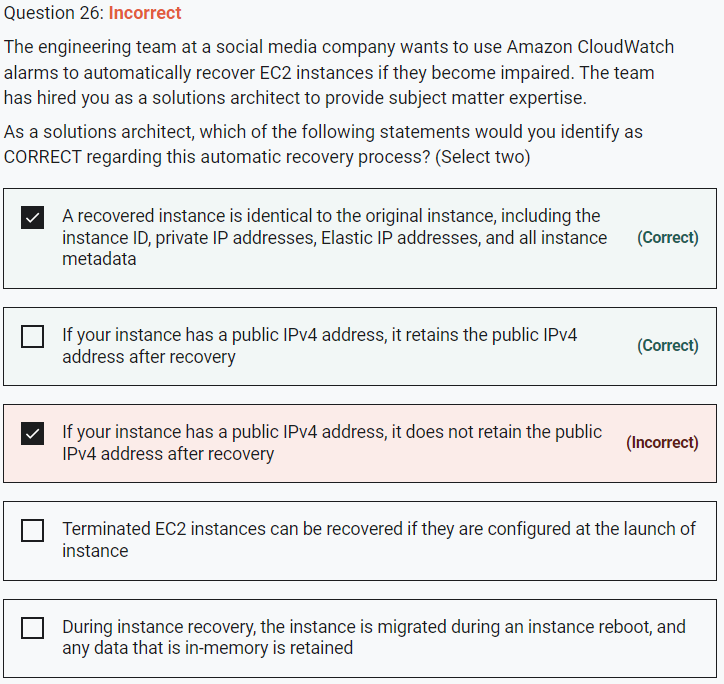






**Idle Timeout** - For each request that a client makes through an Elastic Load Balancer, the load balancer maintains two connections. The front-end connection is between the client and the load balancer. The back-end connection is between the load balancer and a registered EC2 instance. The load balancer has a configured "idle timeout" period that applies to its connections. If no data has been sent or received by the time that the "idle timeout" period elapses, the load balancer closes the connection. "Idle timeout" cannot be used to complete in-flight requests made to instances that are de-registering or unhealthy.





You can create an Amazon CloudWatch alarm to automatically recover the Amazon EC2 instance if it becomes impaired due to an underlying hardware failure or a problem that requires AWS involvement to repair.

Terminated instances cannot be recovered. A recovered instance is identical to the original instance, including the instance ID, private IP addresses, Elastic IP addresses, and all instance metadata.

If the impaired instance is in a placement group, the recovered instance runs in the placement group. If your instance has a public IPv4 address, it retains the public IPv4 address after recovery.

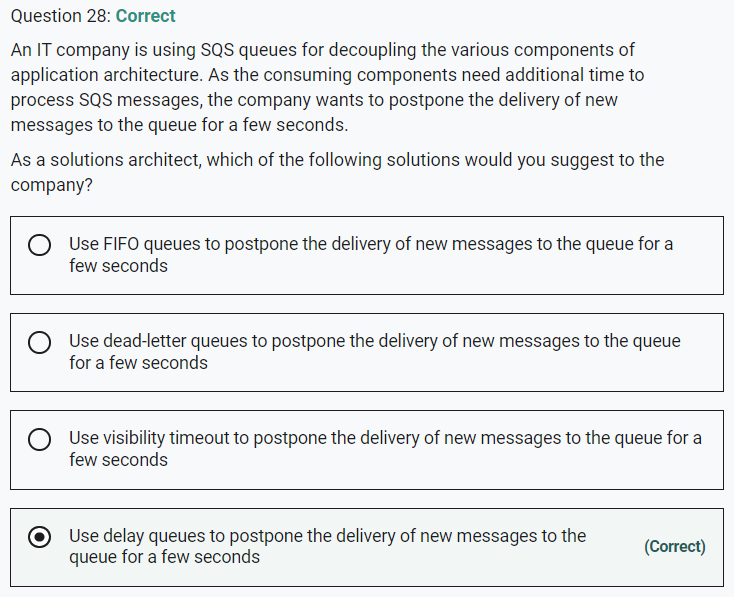
During instance recovery, the instance is migrated during an instance reboot, and any data that is in-memory is lost.

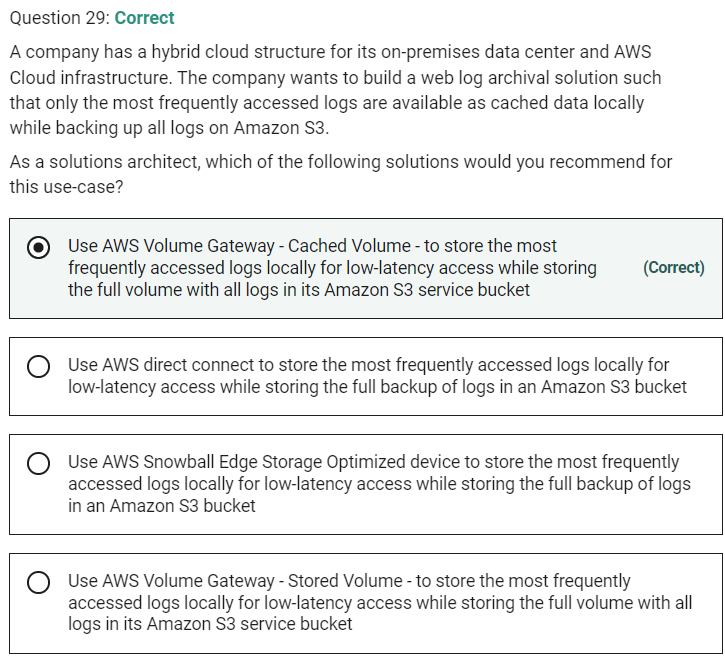
**Terminated EC2 instances can be recovered if they are configured at the launch of instance** - This is incorrect as terminated instances cannot be recovered.

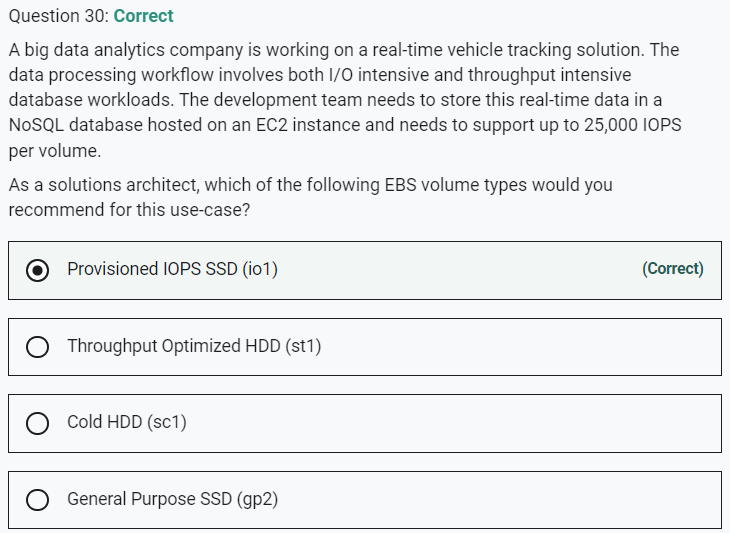
**During instance recovery, the instance is migrated during an instance reboot, and any data that is in-memory is retained** - As mentioned above, during instance recovery, the instance is migrated during an instance reboot, and any data that is in-memory is lost.

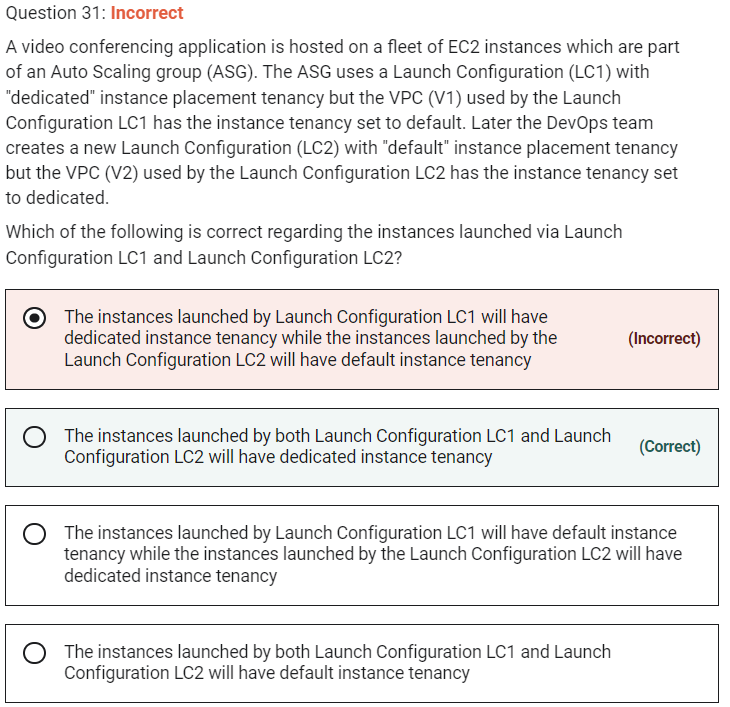
**If your instance has a public IPv4 address, it does not retain the public IPv4 address after recovery** - As mentioned above, if your instance has a public IPv4 address, it retains the public IPv4 address after recovery.





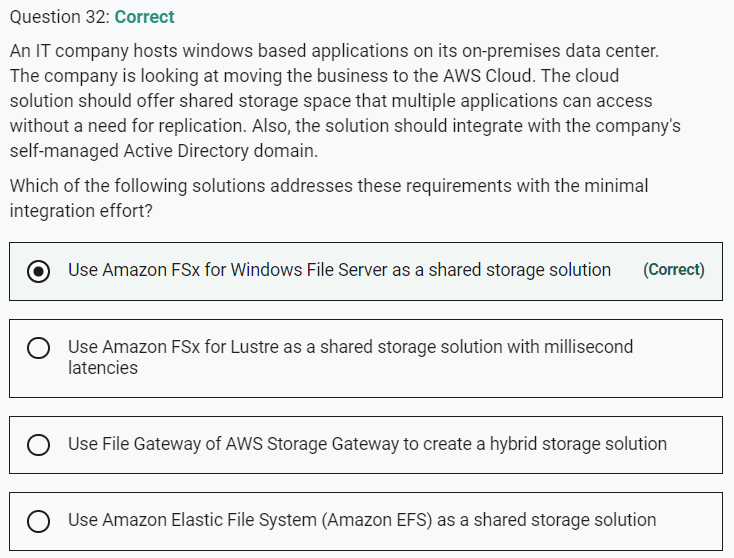


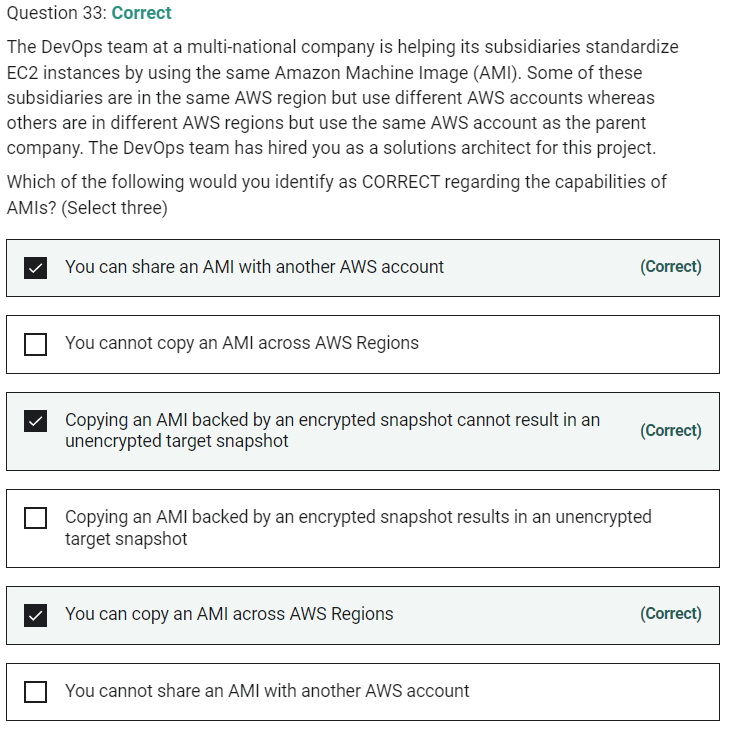


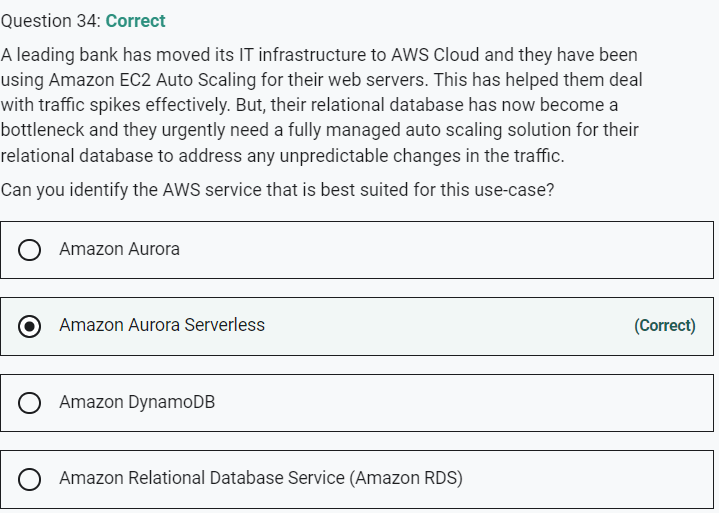


A launch configuration is an instance configuration template that an Auto Scaling group uses to launch EC2 instances. When you create a launch configuration, you specify information for the instances. Include the ID of the Amazon Machine Image (AMI), the instance type, a key pair, one or more security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

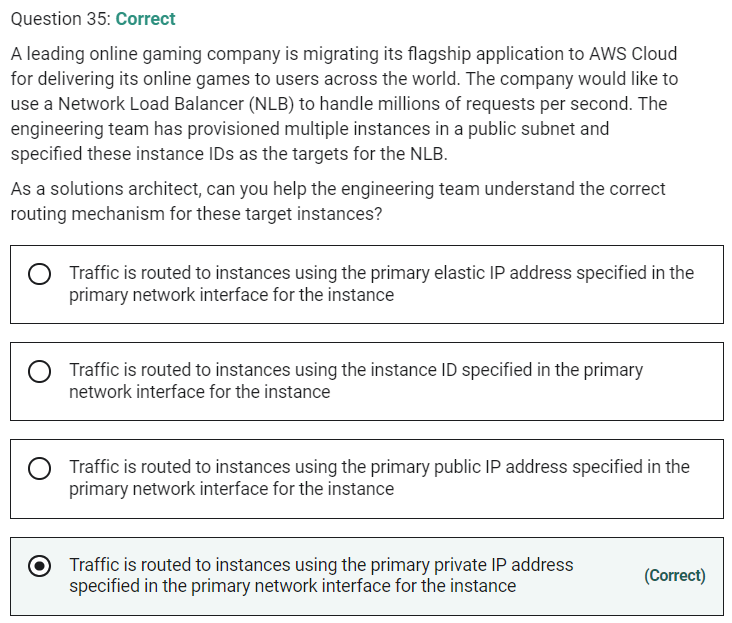
When you create a launch configuration, the default value for the instance placement tenancy is null and the instance tenancy is controlled by the tenancy attribute of the VPC. If you set the Launch Configuration Tenancy to default and the VPC Tenancy is set to dedicated, then the instances have dedicated tenancy. If you set the Launch Configuration Tenancy to dedicated and the VPC Tenancy is set to default, then again the instances have dedicated tenancy

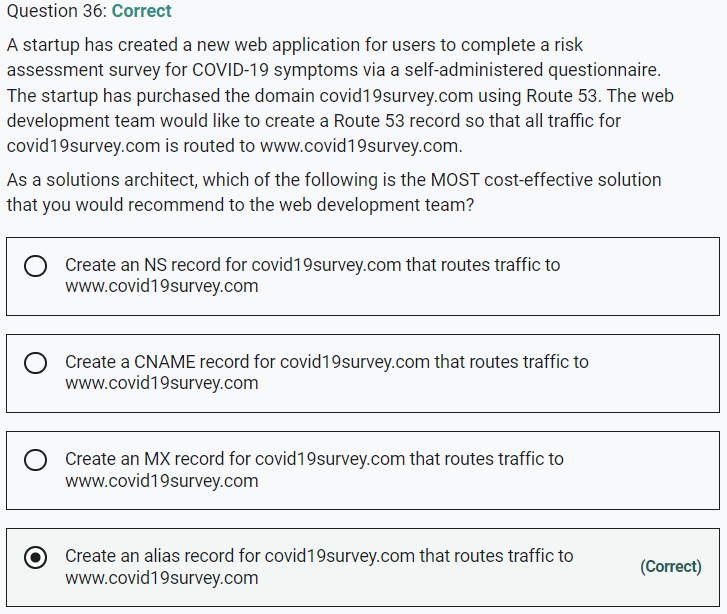


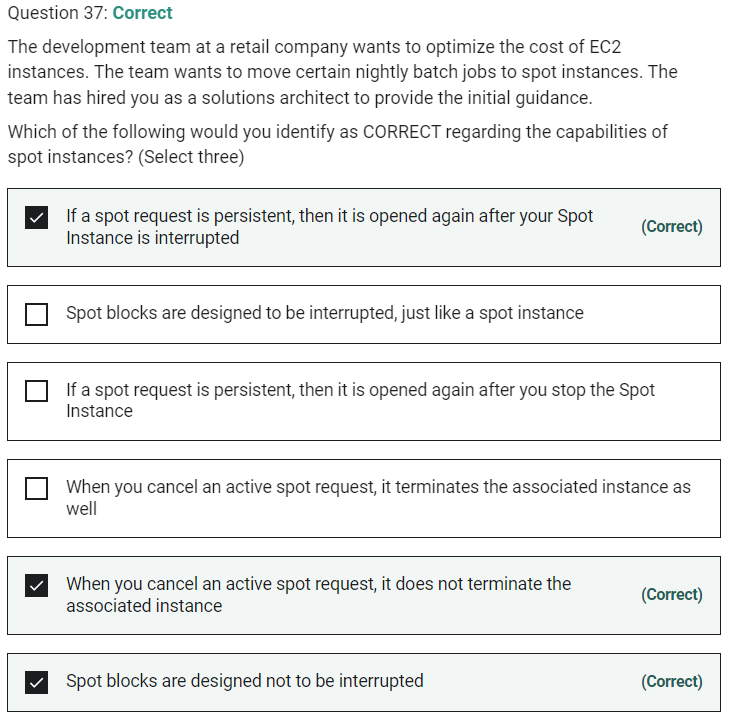




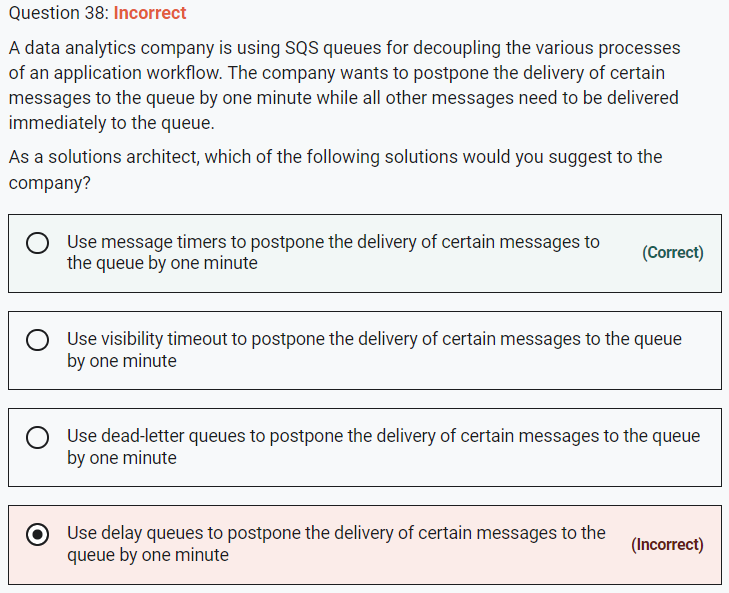
Aurora does not provide auto scaling solution





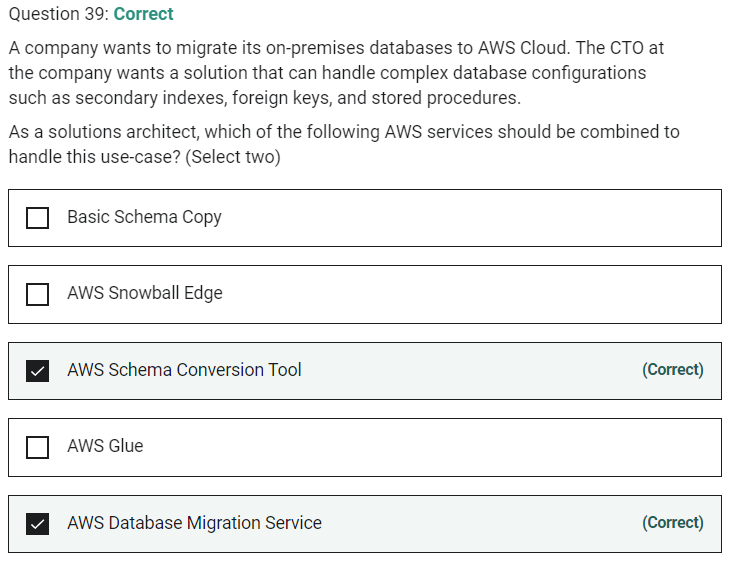


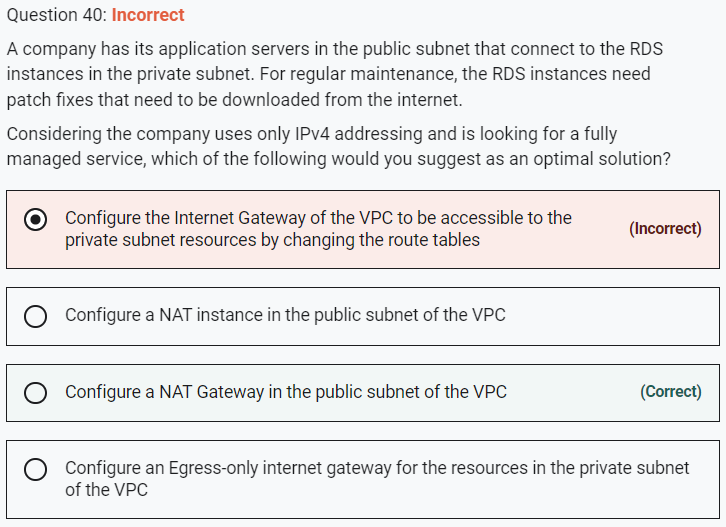
A Spot Instance request is either one-time or persistent. If the spot request is persistent, the request is opened again after your Spot Instance is interrupted. If the request is persistent and you stop your Spot Instance, the request only opens after you start your Spot Instance.

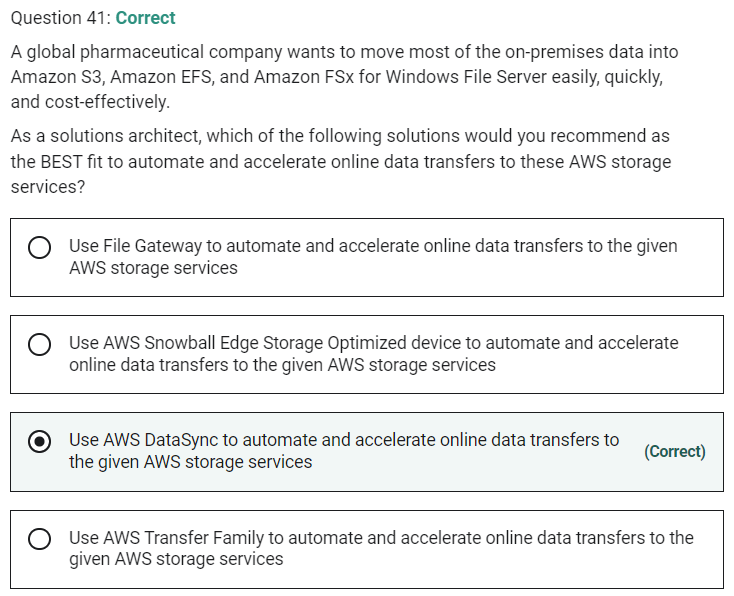


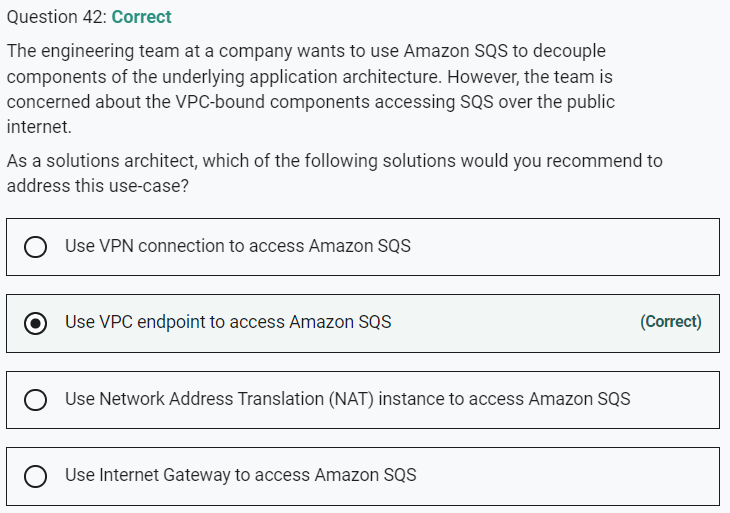
**Use message timers to postpone the delivery of certain messages to the queue by one minute**

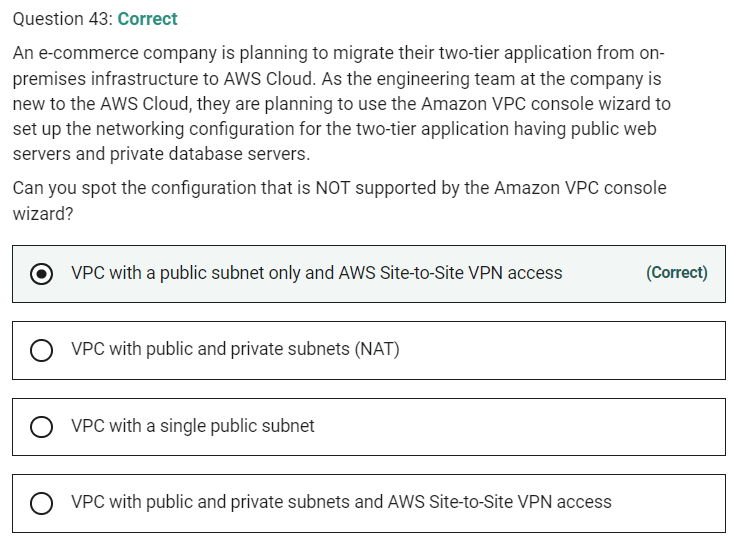
You can use message timers to set an initial invisibility period for a message added to a queue. So, if you send a message with a 60-second timer, the message isn't visible to consumers for its first 60 seconds in the queue. The default (minimum) delay for a message is 0 seconds. The maximum is 15 minutes. Therefore, you should use message timers to postpone the delivery of certain messages to the queue by one minute.











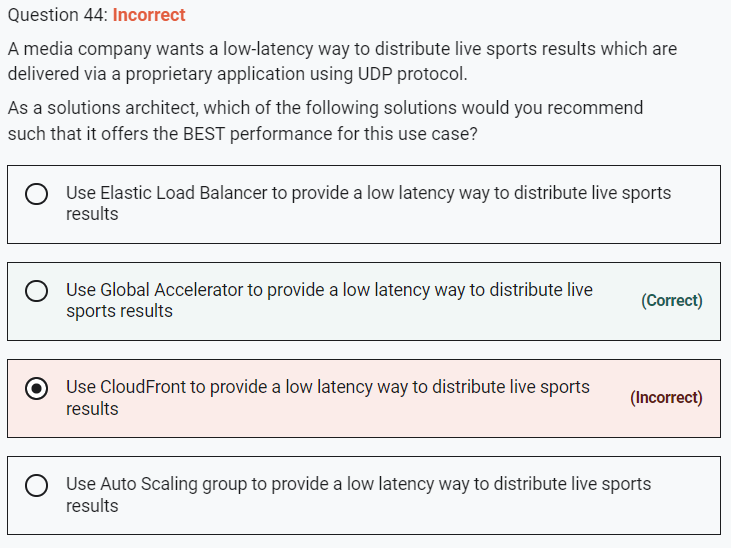
The Amazon VPC console wizard provides the following four configurations:

VPC with a single public subnet –

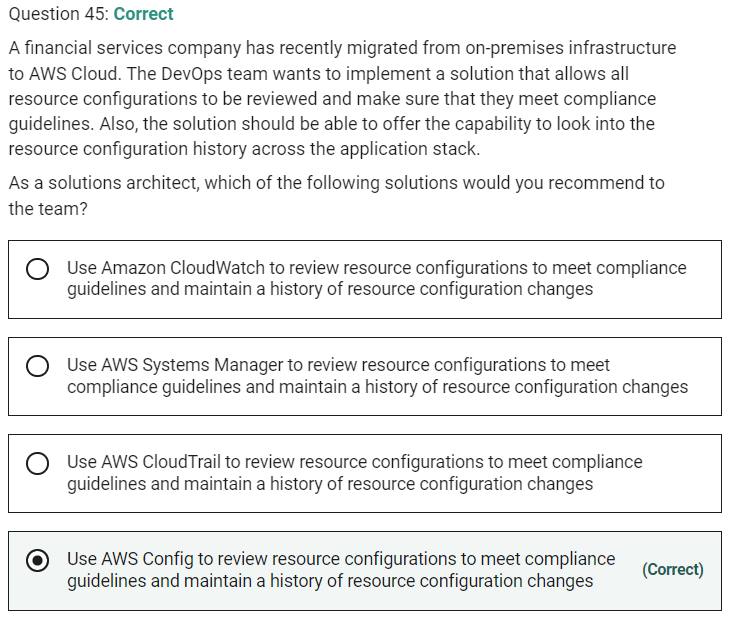
VPC with public and private subnets (NAT) –

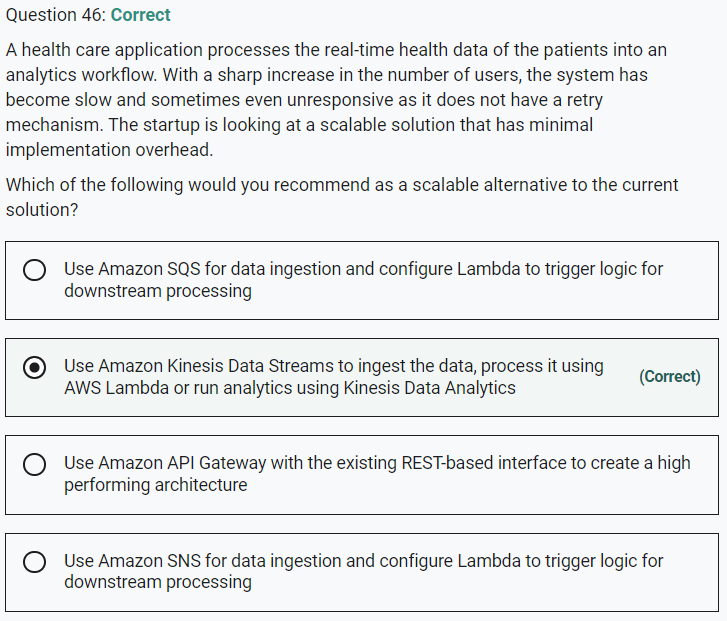
VPC with public and private subnets and AWS Site-to-Site VPN access –

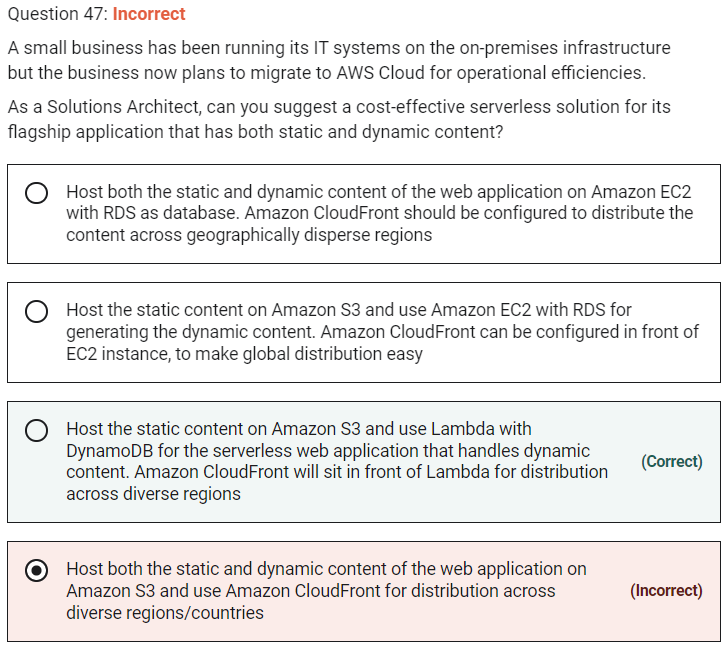
VPC with a private subnet only and AWS Site-to-Site VPN access -



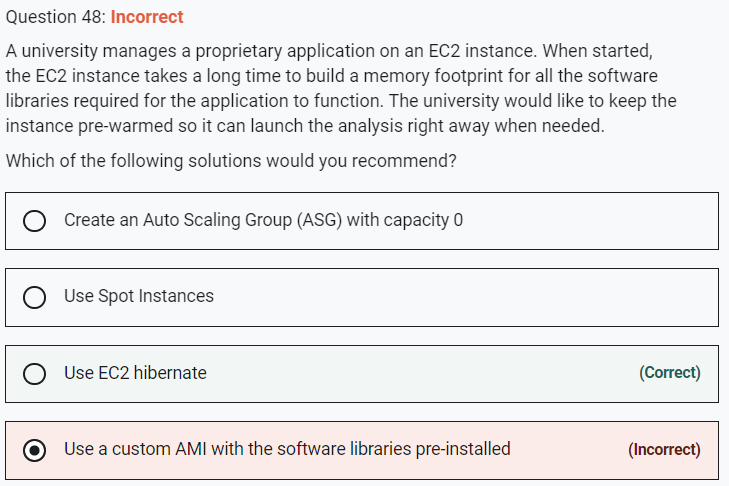
Global Accelerator is a good fit for non-HTTP use cases, such as gaming (UDP), IoT (MQTT), or Voice over IP. Therefore, this option is correct.

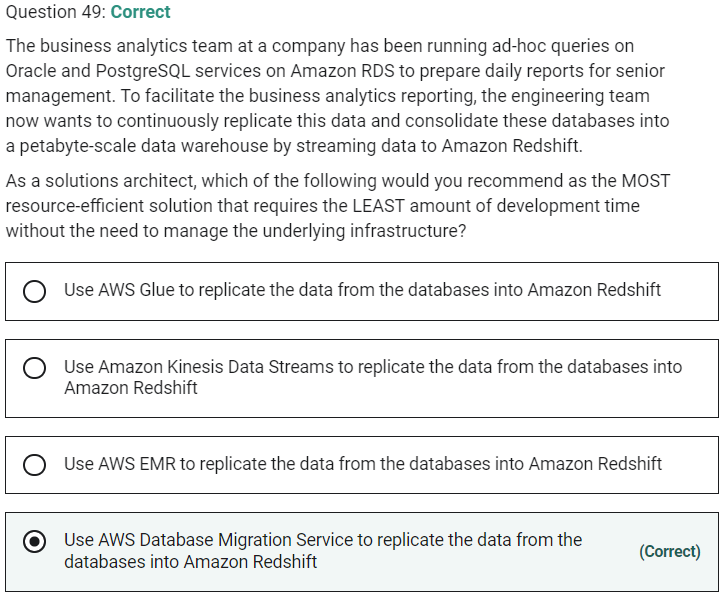


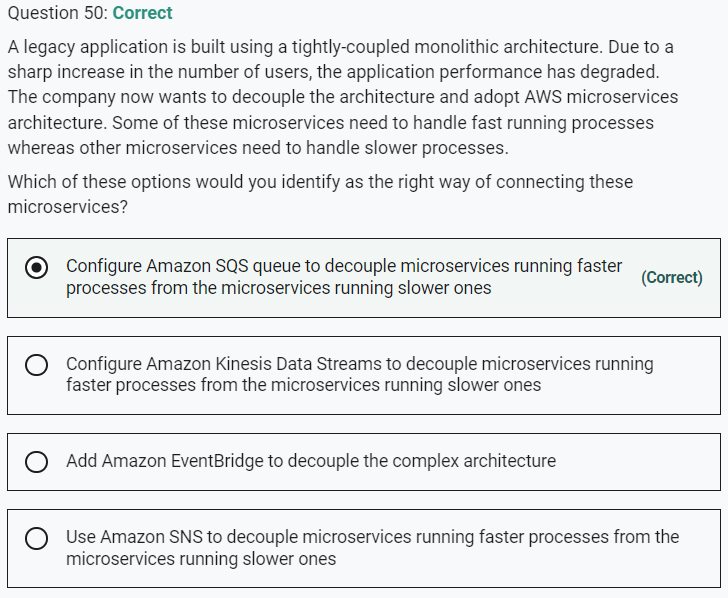


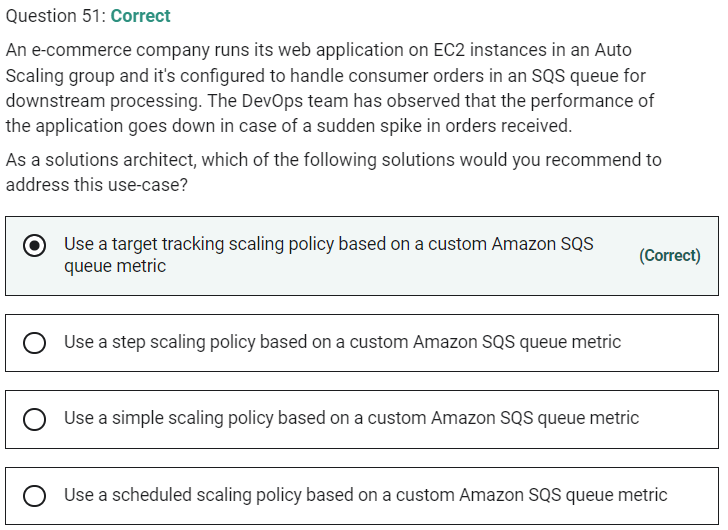


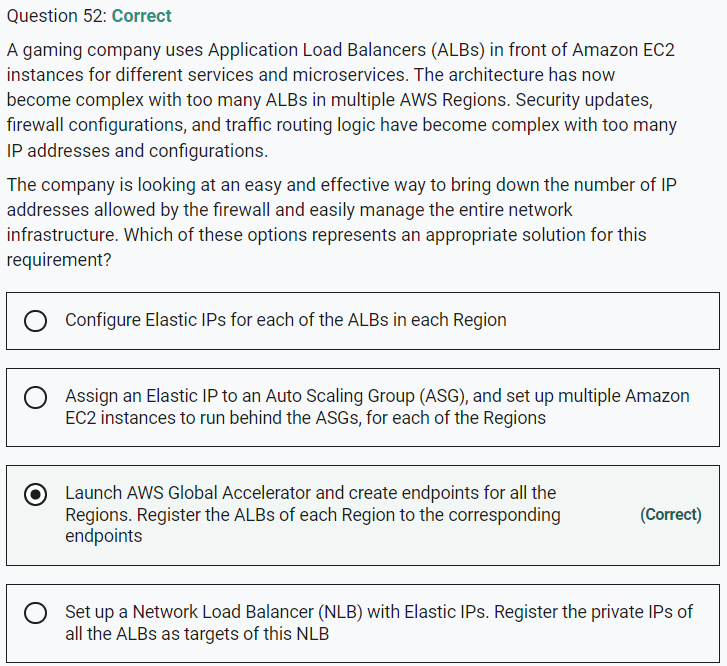
S3 is not the right fit for hosting Dynamic content





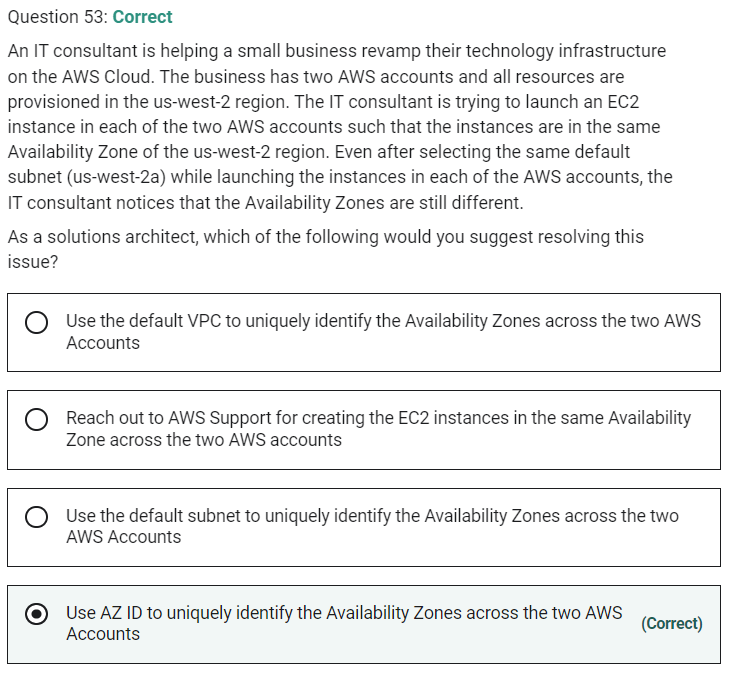






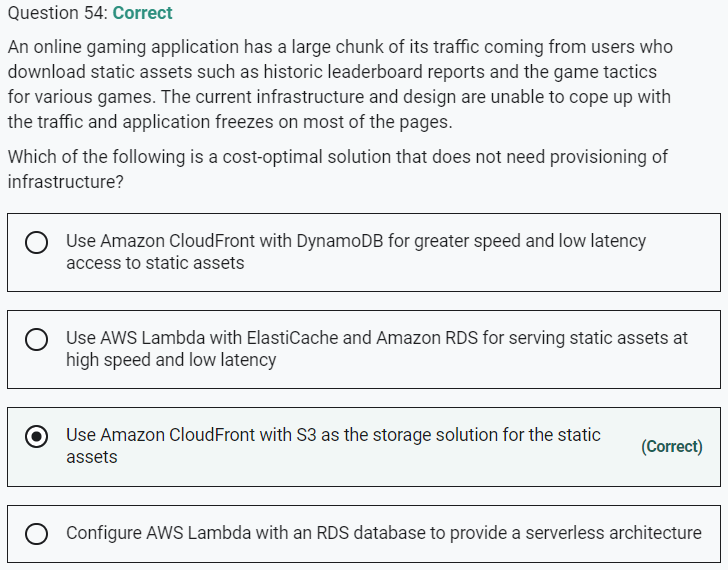
An Application Load Balancer cannot be assigned an Elastic IP address

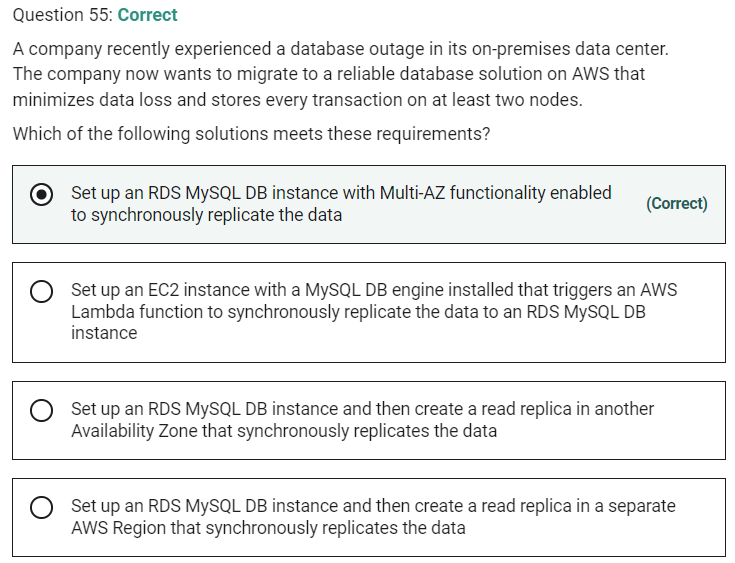
You cannot assign an Elastic IP to an Auto Scaling Group

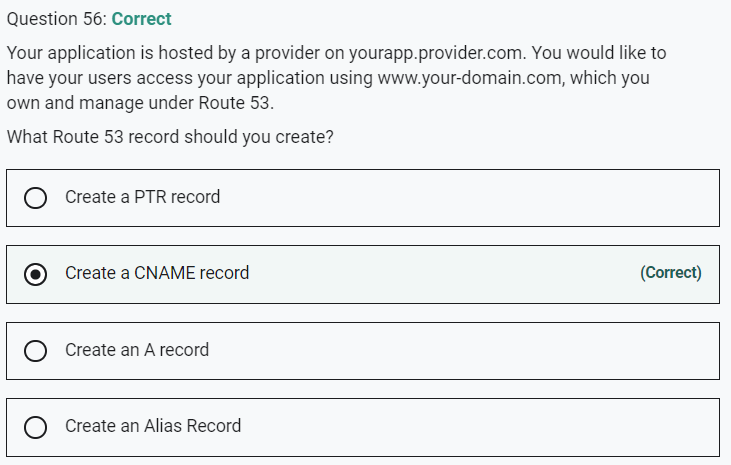


An Availability Zone is represented by a region code followed by a letter identifier; for example, us-east-1a. To ensure that resources are distributed across the Availability Zones for a region, AWS maps Availability Zones to names for each AWS account. For example, the Availability Zone us-west-2a for one AWS account might not be the same location as us-west-2a for another AWS account.

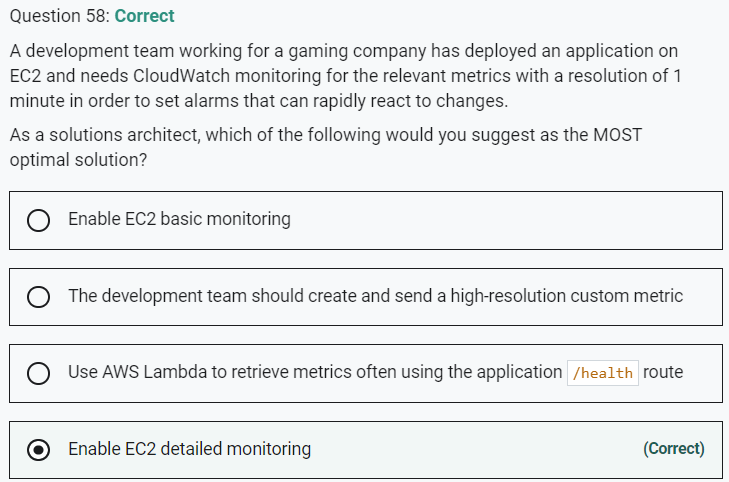
To coordinate Availability Zones across accounts, you must use the AZ ID, which is a unique and consistent identifier for an Availability Zone. For example, usw2-az2 is an AZ ID for the us-west-2 region and it has the same location in every AWS account.

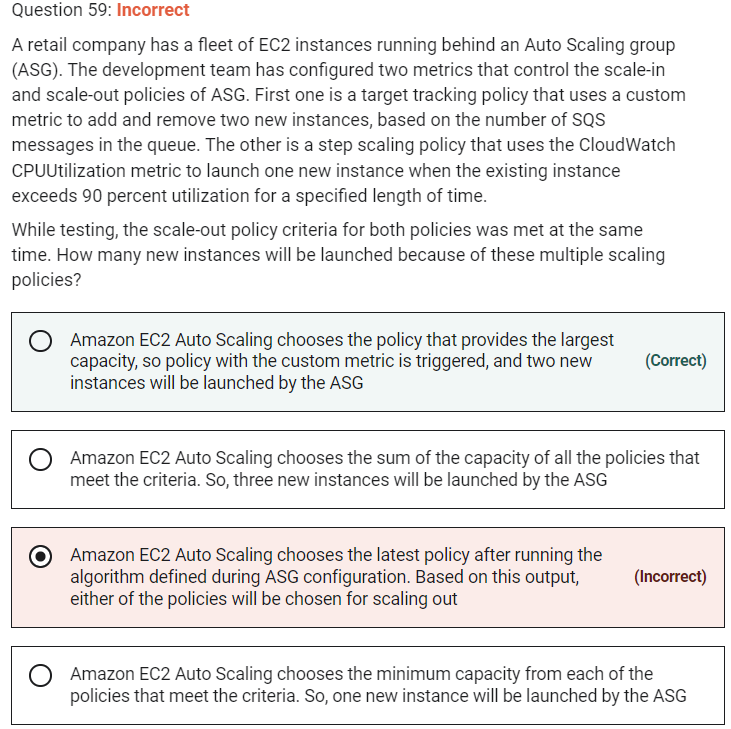








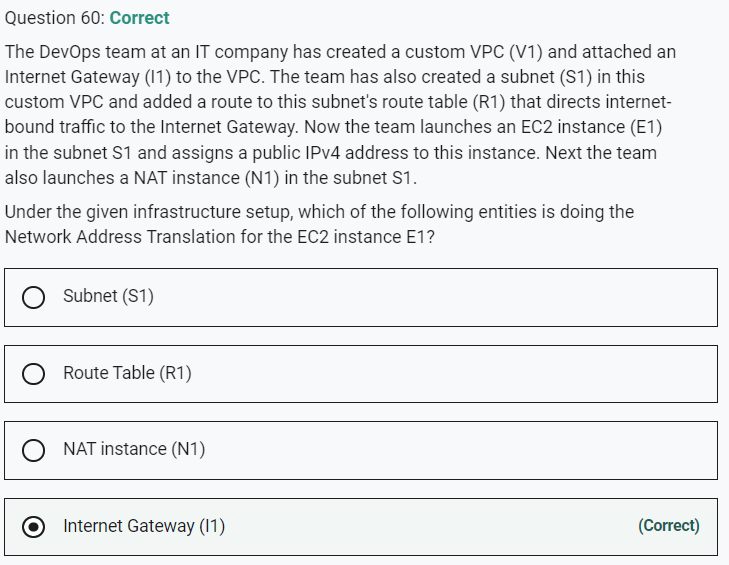


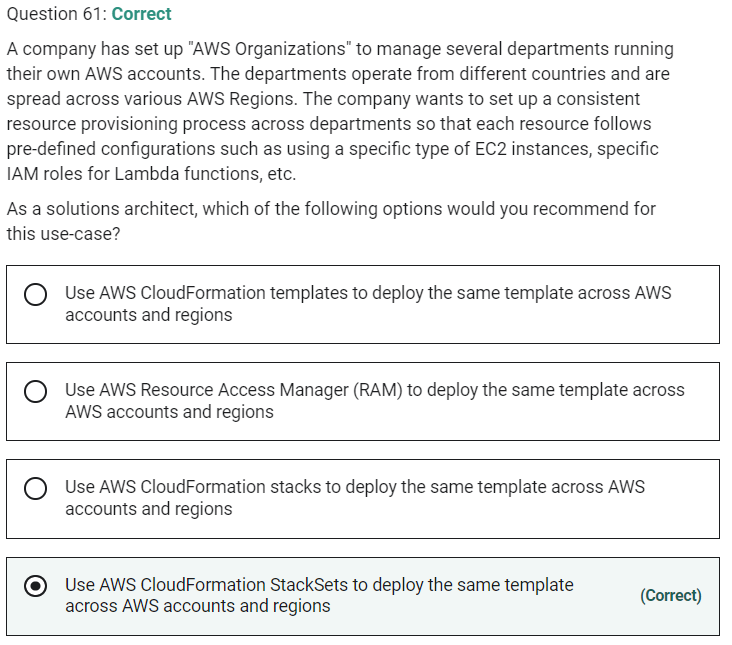


When there are multiple policies in force at the same time, there's a chance that each policy could instruct the Auto Scaling group to scale out (or in) at the same time. For example, it's possible that the CPUUtilization metric spikes and triggers the CloudWatch alarm at the same time that the SQS custom metric spikes and triggers the custom metric alarm.

When these situations occur, Amazon EC2 Auto Scaling chooses the policy that provides the largest capacity for both scale-out and scale-in. Suppose, for example, that the policy for CPUUtilization launches one instance, while the policy for the SQS queue launches two instances. If the scale-out criteria for both policies are met at the same time, Amazon EC2 Auto Scaling gives precedence to the SQS queue policy. This results in the Auto Scaling group launching two instances. The approach of giving precedence to the policy that provides the largest capacity applies even when the policies use different criteria for scaling in.

AWS recommends caution when using target tracking scaling policies with step scaling policies because conflicts between these policies can cause undesirable behavior. For example, if the step scaling policy initiates a scale-in activity before the target tracking policy is ready to scale in, the scale-in activity will not be blocked. After the scale-in activity completes, the target tracking policy could instruct the group to scale out again.





AWS CloudFormation StackSet extends the functionality of stacks by enabling you to create, update, or delete stacks across multiple accounts and regions with a single operation.

