Top of Form

Question 1: **Incorrect**

**A multi-national enterprise uses AWS Organizations to manage its users across different divisions. Even though CloudTrail is enabled on the member accounts, managers have noticed that access issues to CloudTrail logs across different divisions and AWS Regions is becoming a bottleneck in troubleshooting issues. They have decided to use the organization trail to keep things simple.**

**What are the important points to remember when configuring an organization trail? (Select two)**

* ​

By default, CloudTrail tracks only bucket-level actions. To track object-level actions, you need to enable Amazon S3 data events

**(Correct)**

* ​

Member accounts will be able to see the Organization trail, but cannot modify or delete it

**(Correct)**

* ​

Member accounts do not have access to organization trail, neither do they have access to the Amazon S3 bucket that logs the files

* ​

There is nothing called Organization Trail. The master account can, however, enable CloudTrail logging, to keep track of all activities across AWS accounts

**(Incorrect)**

* ​

By default, CloudTrail event log files are not encrypted

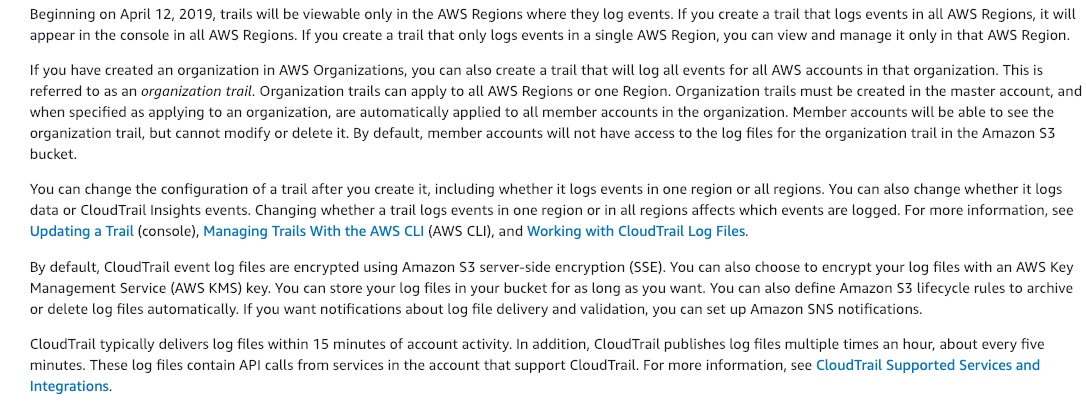
**Explanation**

Correct option:

If you have created an organization in AWS Organizations, you can also create a trail that will log all events for all AWS accounts in that organization. This is referred to as an organization trail.

**By default, CloudTrail tracks only bucket-level actions. To track object-level actions, you need to enable Amazon S3 data events** - This is a correct statement. AWS CloudTrail supports Amazon S3 Data Events, apart from bucket Events. You can record all API actions on S3 Objects and receive detailed information such as the AWS account of the caller, IAM user role of the caller, time of the API call, IP address of the API, and other details. All events are delivered to an S3 bucket and CloudWatch Events, allowing you to take programmatic actions on the events.

**Member accounts will be able to see the organization trail, but cannot modify or delete it** - Organization trails must be created in the master account, and when specified as applying to an organization, are automatically applied to all member accounts in the organization. Member accounts will be able to see the organization trail, but cannot modify or delete it. By default, member accounts will not have access to the log files for the organization trail in the Amazon S3 bucket.

Organization trail:  via - <https://docs.aws.amazon.com/awscloudtrail/latest/userguide/how-cloudtrail-works.html>

Incorrect options:

**There is nothing called Organization Trail. The master account can, however, enable CloudTrail logging, to keep track of all activities across AWS accounts** - This statement is incorrect. AWS offers Organization Trail for easy management and monitoring.

**Member accounts do not have access to the organization trail, neither do they have access to the Amazon S3 bucket that logs the files** - This statement is only partially correct. Member accounts will be able to see the organization trail, but cannot modify or delete it. By default, member accounts will not have access to the log files for the organization trail in the Amazon S3 bucket.

**By default, CloudTrail event log files are not encrypted** - This is an incorrect statement. By default, CloudTrail event log files are encrypted using Amazon S3 server-side encryption (SSE).

References:

<https://docs.aws.amazon.com/awscloudtrail/latest/userguide/how-cloudtrail-works.html>

<https://aws.amazon.com/about-aws/whats-new/2016/11/aws-cloudtrail-supports-s3-data-events/>

<https://aws.amazon.com/premiumsupport/knowledge-center/secure-s3-resources/>

Bottom of Form

Top of Form

Question 2: Skipped

**Your web application front end consists of 5 EC2 instances behind an Application Load Balancer. You have configured your web application to capture the IP address of the client making requests. When viewing the data captured you notice that every IP address being captured is the same, which also happens to be the IP address of the Application Load Balancer.**

**What should you do to identify the true IP address of the client?**

* ​

Look into the X-Forwarded-Proto header in the backend

* ​

Look into the client's cookie

* ​

Look into the X-Forwarded-For header in the backend

**(Correct)**

* ​

Modify the front-end of the website so that the users send their IP in the requests

**Explanation**

Correct option:

**Look into the X-Forwarded-For header in the backend**

The X-Forwarded-For request header helps you identify the IP address of a client when you use an HTTP or HTTPS load balancer. Because load balancers intercept traffic between clients and servers, your server access logs contain only the IP address of the load balancer. To see the IP address of the client, use the X-Forwarded-For request header. Elastic Load Balancing stores the IP address of the client in the X-Forwarded-For request header and passes the header to your server.

Incorrect options:

**Modify the front-end of the website so that the users send their IP in the requests** - When a user makes a request the IP address is sent with the request to the server and the load balancer intercepts it. There is no need to modify the application.

**Look into the X-Forwarded-Proto header in the backend** - The X-Forwarded-Proto request header helps you identify the protocol (HTTP or HTTPS) that a client used to connect to your load balancer.

**Look into the client's cookie** - For this, we would need to modify the client-side logic and server-side logic, which would not be efficient.

Reference:

<https://docs.aws.amazon.com/elasticloadbalancing/latest/classic/x-forwarded-headers.html>

Bottom of Form

Top of Form

Question 3: Skipped

**A client has hired you as an AWS Certified Developer Associate for a consulting project. The client wants to weigh their options of choosing between an Amazon SQS standard queue and Amazon Simple Workflow Service (SWF).**

**Which of the following statements are correct regarding the two services? (Select two)**

* ​

SWF has task-oriented APIs and SQS has message-oriented APIs

**(Correct)**

* ​

SWF offers synchronous programming option whereas SQS offers an asynchronous facility

* ​

SQS has task-oriented APIs and SWF has message-oriented APIs

* ​

SWF ensures the task is assigned only once while SQS may deliver the message multiple times

**(Correct)**

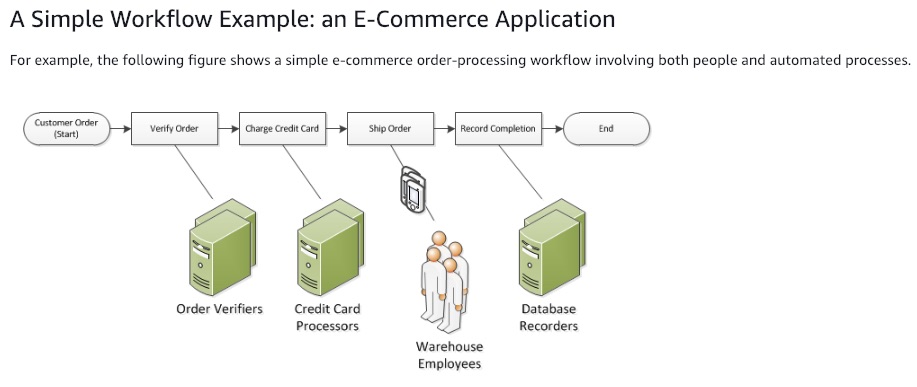
* ​

SQS ensures the task is assigned only once while SWF may deliver the message multiple times

**Explanation**

Correct option:

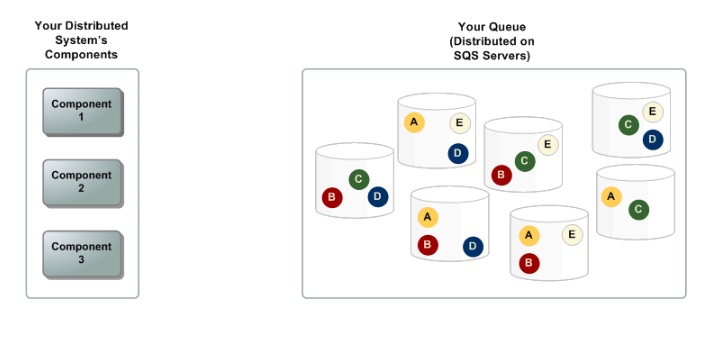
The Amazon Simple Workflow Service (Amazon SWF) makes it easy to build applications that coordinate work across distributed components. In Amazon SWF, a task represents a logical unit of work that is performed by a component of your application. Coordinating tasks across the application involves managing intertask dependencies, scheduling, and concurrency per the logical flow of the application. Amazon SWF gives you full control over implementing tasks and coordinating them without worrying about underlying complexities such as tracking their progress and maintaining their state.

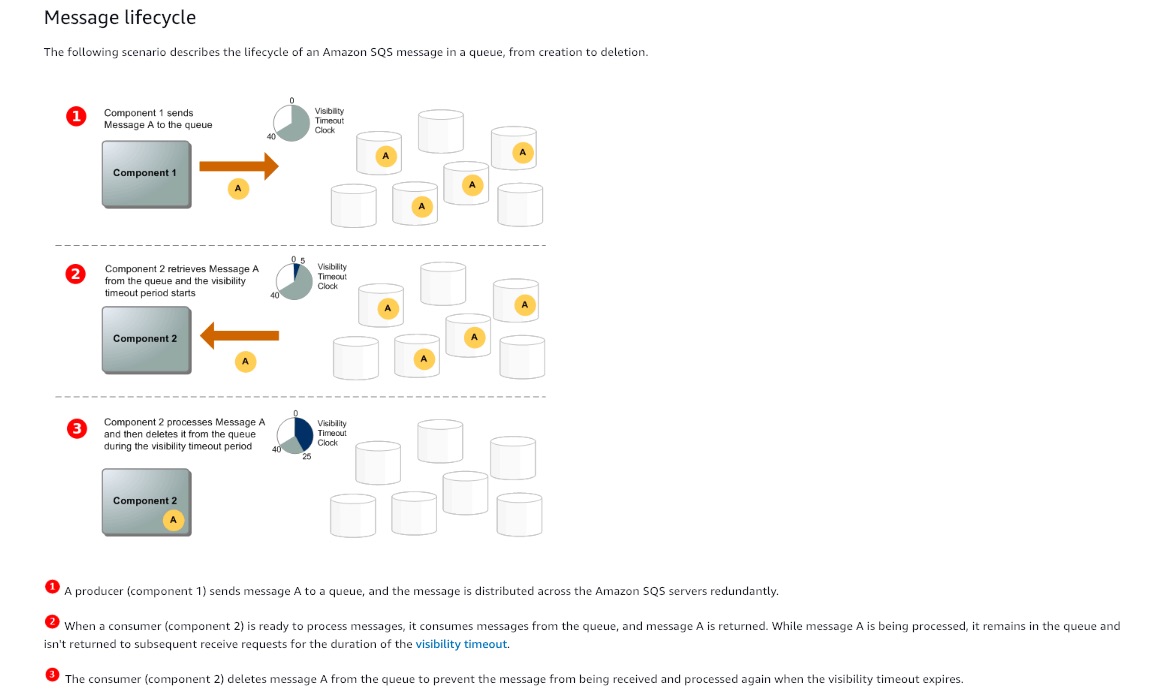
Reference workflow:  via - <https://docs.aws.amazon.com/amazonswf/latest/developerguide/swf-dev-about-workflows.html>

**SWF ensures the task is assigned only once while SQS may deliver the message multiple times** - SWF ensures that a task is assigned only once and is never duplicated. Since Amazon SWF maintains the application’s state durably, workers and deciders don’t have to keep track of execution state. SWF is a workflow solution whereas SQS is a message queue mechanism for decoupling system architecture. SQS does not require the queue consumer to be present at the time the producer adds the messages to the queue. This also implies, sometimes queues duplicate messages and it is up to the consumer to write logic to handle the duplicates.

**SWF has task-oriented APIs and SQS has message-oriented APIs** - The fundamental concept in Amazon SWF is the workflow. A workflow is a set of activities that carry out some objective, together with logic that coordinates the activities. Amazon SWF has the concept of an activity task that represents one invocation of an activity. The basic building block of SWF, work is tracked through tasks and hence SWF has task-oriented APIs.

Amazon Simple Queue Service (SQS) is a fully managed message queuing service that enables you to decouple and scale microservices, distributed systems, and serverless applications. Amazon SQS requires no administrative overhead and little configuration. Amazon SQS works on a massive scale, processing billions of messages per day. SQS is a queuing service and messages in these queues are the basic building blocks of work that are saved, maintained, and tracked to completion. So SQS has message-oriented APIs.

SQS architecture:  via - <https://docs.aws.amazon.com/AWSSimpleQueueService/latest/SQSDeveloperGuide/sqs-basic-architecture.html>

Message lifecycle:  via - <https://docs.aws.amazon.com/AWSSimpleQueueService/latest/SQSDeveloperGuide/sqs-basic-architecture.html>

Incorrect options:

**SQS has task-oriented APIs and SWF has message-oriented APIs** - This is an incorrect statement. As discussed above, SQS has message-oriented APIs and SWF has task-oriented APIs.

**SQS ensures the task is assigned only once while SWF may deliver the message multiple times** - This is an incorrect statement. SWF ensures the task is assigned only once while SQS may deliver the message multiple times.

**SWF offers a synchronous programming option whereas SQS offers an asynchronous facility** - This is an incorrect statement. Both SWF and SQS offer asynchronous programming options in the development of applications.

References:

<https://docs.aws.amazon.com/AWSSimpleQueueService/latest/SQSDeveloperGuide/sqs-basic-architecture.html>

<https://aws.amazon.com/swf/>

Bottom of Form

Top of Form

Question 4: Skipped

**A junior developer working on ECS instances terminated a container instance in Amazon Elastic Container Service (Amazon ECS) as per instructions from the team lead. But the container instance continues to appear as a resource in the ECS cluster.**

**As a Developer Associate, which of the following solutions would you recommend to fix this behavior?**

* ​

You terminated the container instance while it was in STOPPED state, that lead to this synchronization issues

**(Correct)**

* ​

The container instance has been terminated with AWS CLI, whereas, for ECS instances, Amazon ECS CLI should be used to avoid any synchronization issues

* ​

You terminated the container instance while it was in RUNNING state, that lead to this synchronization issues

* ​

A custom software on the container instance could have failed and resulted in the container hanging in an unhealthy state till restarted again

**Explanation**

Correct option:

**You terminated the container instance while it was in STOPPED state, that lead to this synchronization issues** - If you terminate a container instance while it is in the STOPPED state, that container instance isn't automatically removed from the cluster. You will need to deregister your container instance in the STOPPED state by using the Amazon ECS console or AWS Command Line Interface. Once deregistered, the container instance will no longer appear as a resource in your Amazon ECS cluster.

Incorrect options:

**You terminated the container instance while it was in RUNNING state, that lead to this synchronization issues** - This is an incorrect statement. If you terminate a container instance in the RUNNING state, that container instance is automatically removed, or deregistered, from the cluster.

**The container instance has been terminated with AWS CLI, whereas, for ECS instances, Amazon ECS CLI should be used to avoid any synchronization issues** - This is incorrect and has been added as a distractor.

**A custom software on the container instance could have failed and resulted in the container hanging in an unhealthy state till restarted again** - This is an incorrect statement. It is already mentioned in the question that the developer has terminated the instance.

References:

<https://aws.amazon.com/premiumsupport/knowledge-center/deregister-ecs-instance/>

<https://docs.aws.amazon.com/AmazonECS/latest/developerguide/ECS_instances.html>

Bottom of Form

Top of Form

Question 5: Skipped

**A development team has created AWS CloudFormation templates that are reusable by taking advantage of input parameters to name resources based on client names.**

**You would like to save your templates on the cloud, which storage option should you choose?**

* ​

S3

**(Correct)**

* ​

ECR

* ​

EFS

* ​

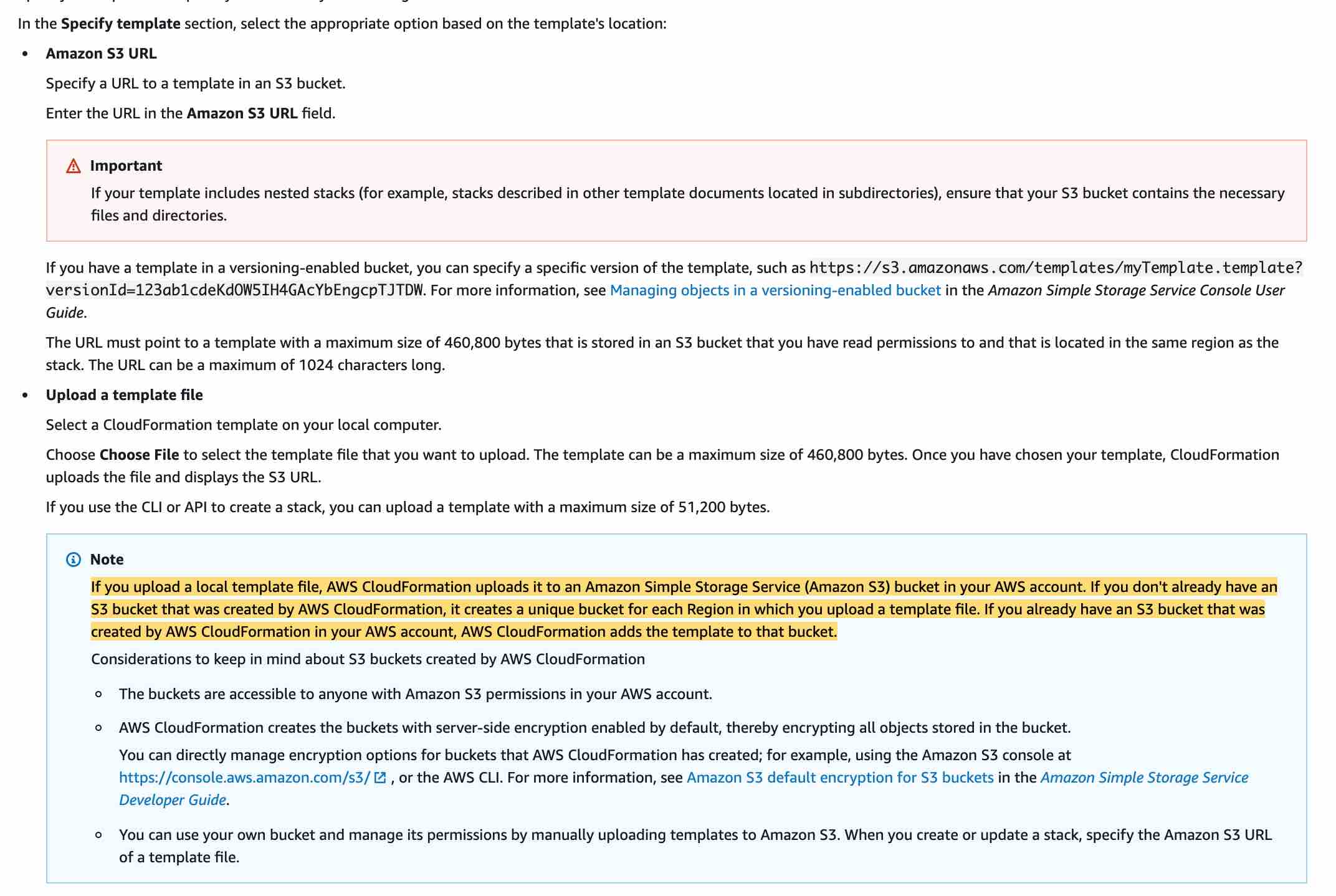
EBS

**Explanation**

Correct option:

**S3**

If you upload a local template file, AWS CloudFormation uploads it to an Amazon Simple Storage Service (Amazon S3) bucket in your AWS account. If you don't already have an S3 bucket that was created by AWS CloudFormation, it creates a unique bucket for each region in which you upload a template file. If you already have an S3 bucket that was created by AWS CloudFormation in your AWS account, AWS CloudFormation adds the template to that bucket.

Selecting a stack template for CloudFormation:  via - <https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/cfn-using-console-create-stack-template.html>

Incorrect options:

**EBS** - An Amazon EBS volume is a durable, block-level storage device that you can attach to your instances. After you attach a volume to an instance, you can use it as you would use a physical hard drive. EBS volumes are flexible. Amazon EBS is a recommended storage option when data must be quickly accessible and requires long-term persistence. EBS cannot be used for selecting a stack template for CloudFormation.

**EFS** - EFS is a file storage service where you mount the file system on an Amazon EC2 Linux-based instance which is not an option for CloudFormation.

**ECR** - Amazon ECR eliminates the need to operate your container repositories or worry about scaling the underlying infrastructure which does not apply to CloudFormation.

Reference:

<https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/cfn-using-console-create-stack-template.html>

Bottom of Form

Top of Form

Question 6: Skipped

**You are planning to build a fleet of EBS-optimized EC2 instances to handle the load of your new application. Due to security compliance, your organization wants any secret strings used in the application to be encrypted to prevent exposing values as clear text.**

**The solution requires that decryption events be audited and API calls to be simple. How can this be achieved? (select two)**

* ​

Encrypt first with KMS then store in SSM Parameter store

* ​

Store the secret as PlainText in SSM Parameter Store

* ​

Audit using SSM Audit Trail

* ​

Store the secret as SecureString in SSM Parameter Store

**(Correct)**

* ​

Audit using CloudTrail

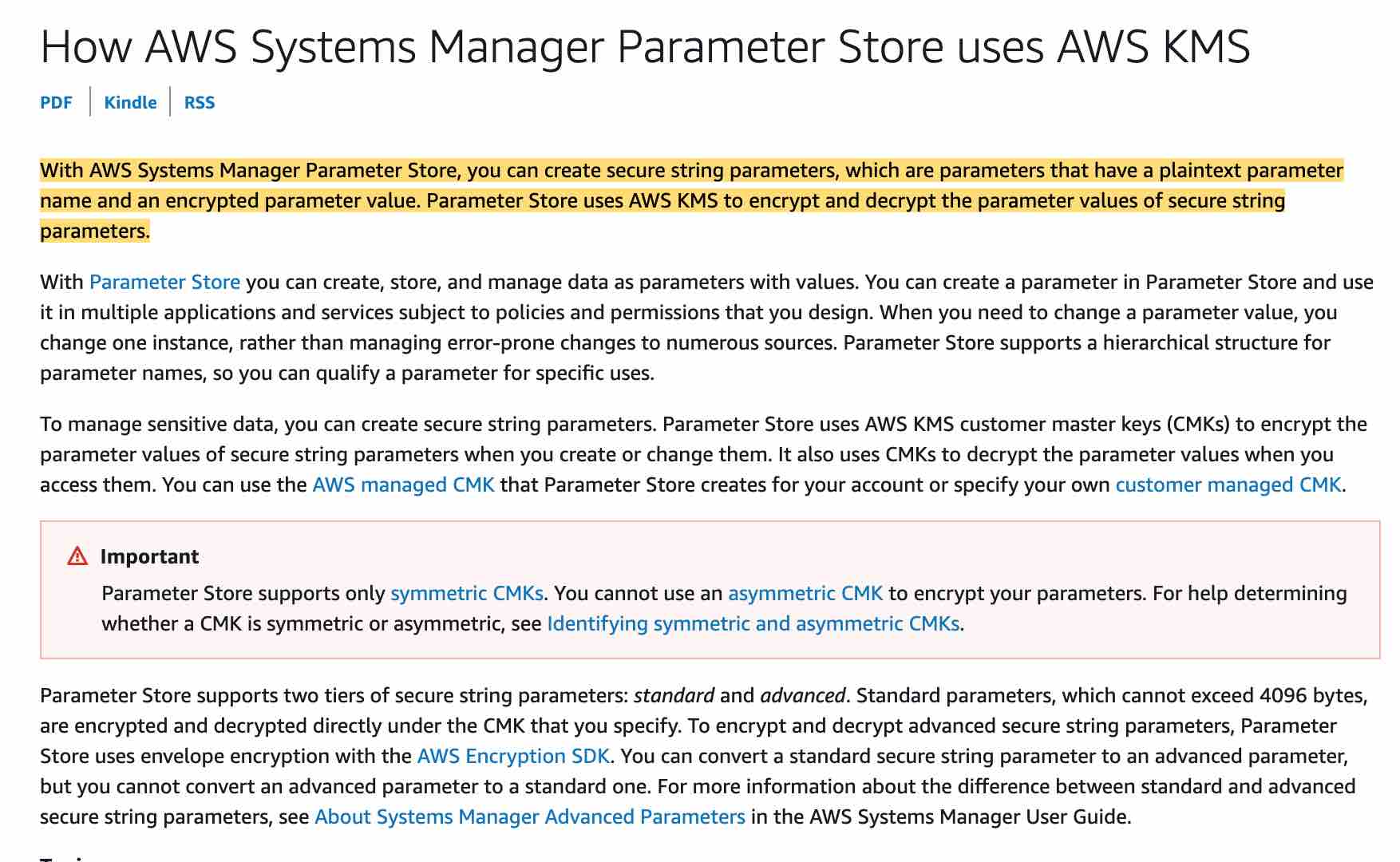
**(Correct)**

**Explanation**

Correct options:

**Store the secret as SecureString in SSM Parameter Store**

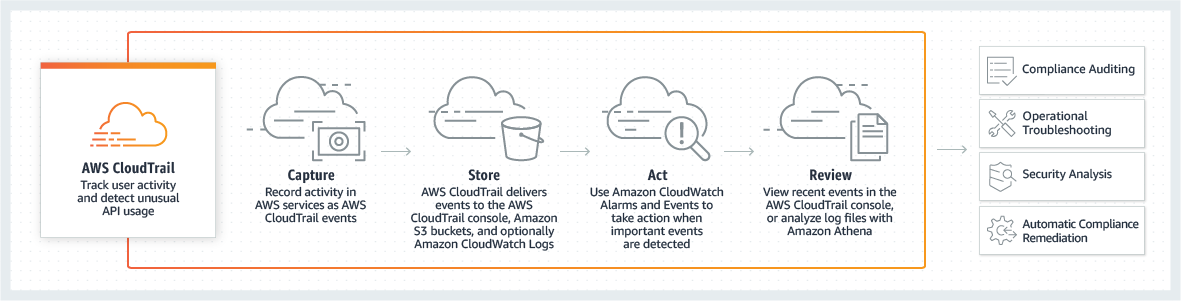
With AWS Systems Manager Parameter Store, you can create SecureString parameters, which are parameters that have a plaintext parameter name and an encrypted parameter value. Parameter Store uses AWS KMS to encrypt and decrypt the parameter values of Secure String parameters. Also, if you are using customer-managed CMKs, you can use IAM policies and key policies to manage to encrypt and decrypt permissions. To retrieve the decrypted value you only need to do one API call.

 via - <https://docs.aws.amazon.com/kms/latest/developerguide/services-parameter-store.html>

**Audit using CloudTrail**

AWS CloudTrail is a service that enables governance, compliance, operational auditing, and risk auditing of your AWS account. With CloudTrail, you can log, continuously monitor, and retain account activity related to actions across your AWS infrastructure. CloudTrail provides an event history of your AWS account activity, including actions taken through the AWS Management Console, AWS SDKs, command-line tools, and other AWS services.

CloudTrail will allow you to see all API calls made to SSM and KMS.



Incorrect options:

**Encrypt first with KMS then store in SSM Parameter store** - This could work but will require two API calls to get the decrypted value instead of one. So this is not the right option.

**Store the secret as PlainText in SSM Parameter Store** - Plaintext parameters are not secure and shouldn't be used to store secrets.

**Audit using SSM Audit Trail** - This is a made-up option and has been added as a distractor.

Reference:

<https://docs.aws.amazon.com/kms/latest/developerguide/services-parameter-store.html>

Bottom of Form

Top of Form

Question 7: Skipped

**An Auto Scaling group has a maximum capacity of 3, a current capacity of 2, and a scaling policy that adds 3 instances.**

**When executing this scaling policy, what is the expected outcome?**

* ​

Amazon EC2 Auto Scaling adds 3 instances to the group

* ​

Amazon EC2 Auto Scaling adds 3 instances to the group and scales down 2 of those instances eventually

* ​

Amazon EC2 Auto Scaling does not add any instances to the group, but suggests changing the scaling policy to add one instance

* ​

Amazon EC2 Auto Scaling adds only 1 instance to the group

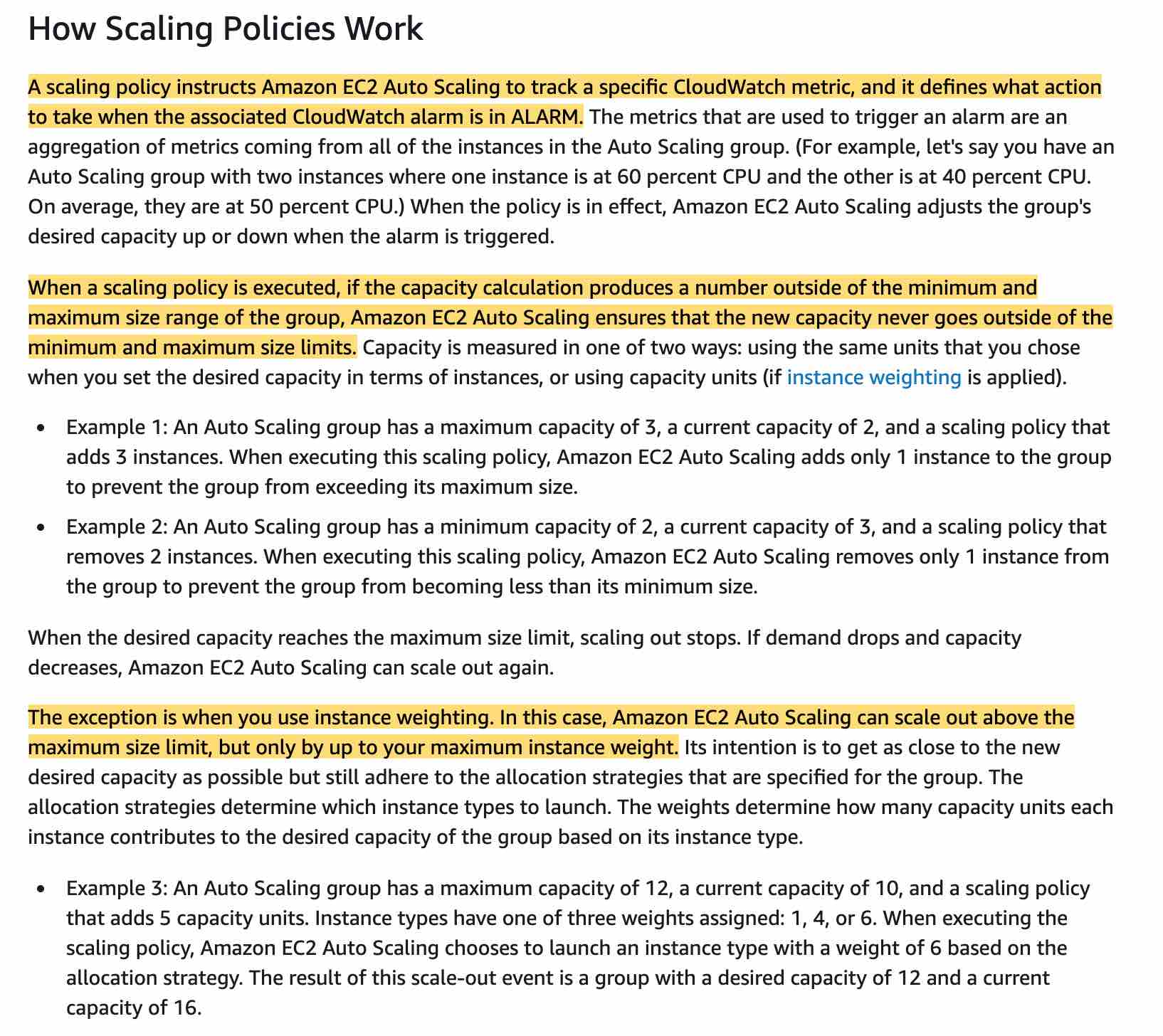
**(Correct)**

**Explanation**

Correct option:

A scaling policy instructs Amazon EC2 Auto Scaling to track a specific CloudWatch metric, and it defines what action to take when the associated CloudWatch alarm is in ALARM.

When a scaling policy is executed, if the capacity calculation produces a number outside of the minimum and maximum size range of the group, Amazon EC2 Auto Scaling ensures that the new capacity never goes outside of the minimum and maximum size limits.

 via - <https://docs.aws.amazon.com/autoscaling/ec2/userguide/as-scale-based-on-demand.html>

**Amazon EC2 Auto Scaling adds only 1 instance to the group**

For the given use-case, Amazon EC2 Auto Scaling adds only 1 instance to the group to prevent the group from exceeding its maximum size.

Incorrect options:

**Amazon EC2 Auto Scaling adds 3 instances to the group** - This is an incorrect statement. Auto Scaling ensures that the new capacity never goes outside of the minimum and maximum size limits.

**Amazon EC2 Auto Scaling adds 3 instances to the group and scales down 2 of those instances eventually** - This is an incorrect statement. Adding the instances initially and immediately downsizing them is impractical.

**Amazon EC2 Auto Scaling does not add any instances to the group, but suggests changing the scaling policy to add one instance** - This option has been added as a distractor.

Reference:

<https://docs.aws.amazon.com/autoscaling/ec2/userguide/as-scale-based-on-demand.html>

Bottom of Form

Top of Form

Question 8: Skipped

**You have a Java-based application running on EC2 instances loaded with AWS CodeDeploy agents. You are considering different options for deployment, one is the flexibility that allows for incremental deployment of your new application versions and replaces existing versions in the EC2 instances. The other option is a strategy in which an Auto Scaling group is used to perform a deployment.**

**Which of the following options will allow you to deploy in this manner? (Select two)**

* ​

Warm Standby Deployment

* ​

Cattle Deployment

* ​

Pilot Light Deployment

* ​

In-place Deployment

**(Correct)**

* ​

Blue/green Deployment

**(Correct)**

**Explanation**

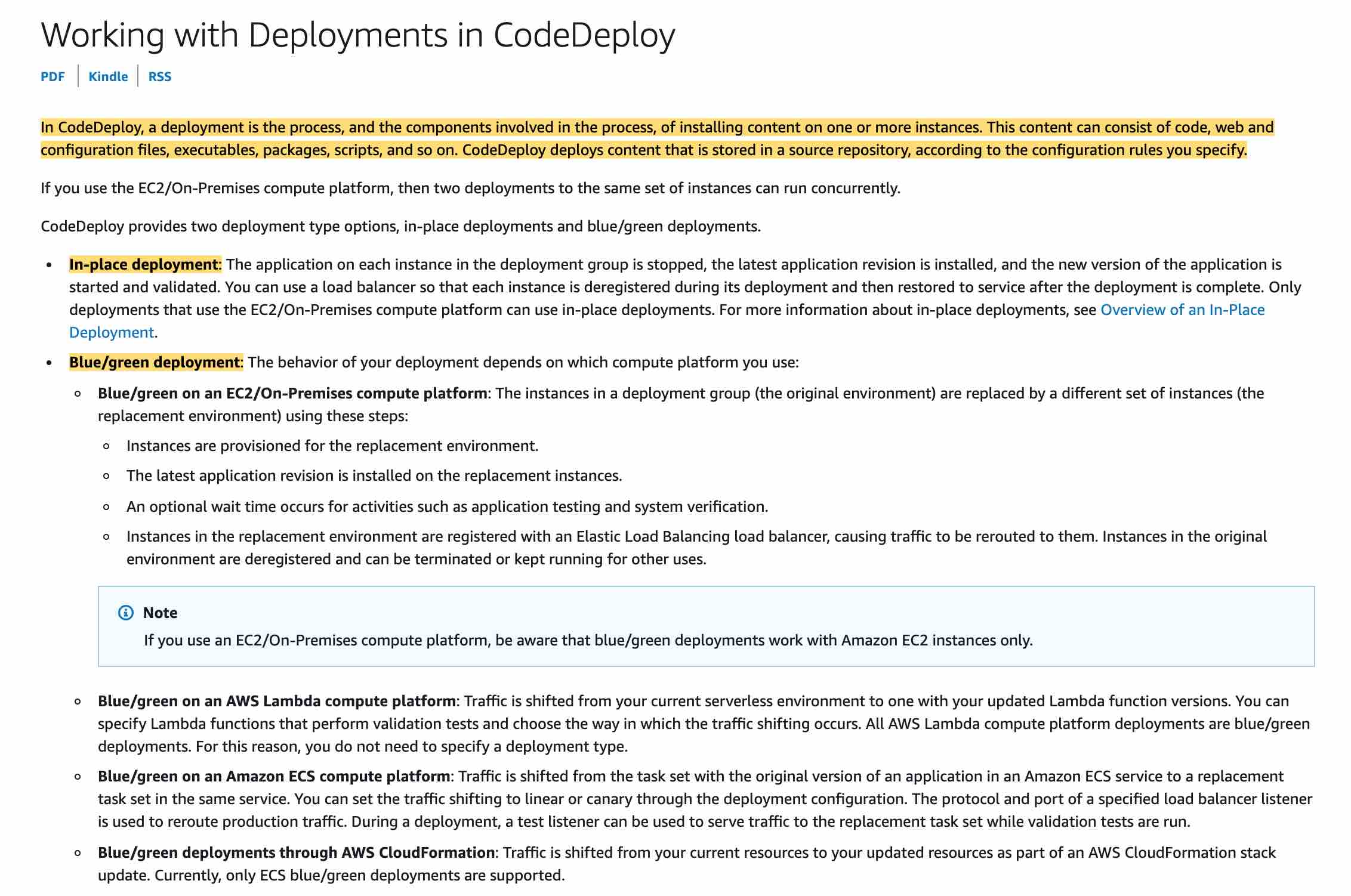
Correct options:

**In-place Deployment**

The application on each instance in the deployment group is stopped, the latest application revision is installed, and the new version of the application is started and validated. You can use a load balancer so that each instance is deregistered during its deployment and then restored to service after the deployment is complete.

**Blue/green Deployment**

With a blue/green deployment, you provision a new set of instances on which CodeDeploy installs the latest version of your application. CodeDeploy then re-routes load balancer traffic from an existing set of instances running the previous version of your application to the new set of instances running the latest version. After traffic is re-routed to the new instances, the existing instances can be terminated.

CodeDeploy Deployment Types:  via - <https://docs.aws.amazon.com/codedeploy/latest/userguide/deployments.html>

Incorrect options:

**Cattle Deployment** - This is a good option if you have cattle in a farm

**Warm Standby Deployment** - This is not a valid CodeDeploy deployment option. The term "Warm Standby" is used to describe a Disaster Recovery scenario in which a scaled-down version of a fully functional environment is always running in the cloud.

**Pilot Light Deployment** - This is not a valid CodeDeploy deployment option. "Pilot Light" is a Disaster Recovery approach where you simply replicate part of your IT structure for a limited set of core services so that the AWS cloud environment seamlessly takes over in the event of a disaster.

References:

<https://docs.aws.amazon.com/codedeploy/latest/userguide/deployments.html>

<https://aws.amazon.com/blogs/publicsector/rapidly-recover-mission-critical-systems-in-a-disaster/>

Bottom of Form

Top of Form

Question 9: Skipped

**A developer at a university is encrypting a large XML payload transferred over the network using AWS KMS and wants to test the application before going to production.**

**What is the maximum data size supported by AWS KMS?**

* ​

1MB

* ​

16KB

* ​

10MB

* ​

4KB

**(Correct)**

**Explanation**

Correct option:

**4 KB**

You can encrypt up to 4 kilobytes (4096 bytes) of arbitrary data such as an RSA key, a database password, or other sensitive information.

While AWS KMS does support sending data up to 4 KB to be encrypted directly, envelope encryption can offer significant performance benefits. When you encrypt data directly with AWS KMS it must be transferred over the network. Envelope encryption reduces the network load since only the request and delivery of the much smaller data key go over the network. The data key is used locally in your application or encrypting AWS service, avoiding the need to send the entire block of data to AWS KMS and suffer network latency.

Incorrect options:

**1MB** - For anything over 4 KB, you may want to look at envelope encryption

**10MB** - For anything over 4 KB, you may want to look at envelope encryption

**16KB** - For anything over 4 KB, you may want to look at envelope encryption

Reference:

<https://aws.amazon.com/kms/faqs/>

Bottom of Form

Top of Form

Question 10: Skipped

**An IT company has a HealthCare application with data security requirements such that the encryption key must be stored in a custom application running on-premises. The company wants to offload the data storage as well as the encryption process to Amazon S3 but continue to use the existing encryption keys.**

**Which of the following S3 encryption options allows the company to leverage Amazon S3 for storing data with given constraints?**

* ​

Server-Side Encryption with Customer-Provided Keys (SSE-C)

**(Correct)**

* ​

Client-Side Encryption with data encryption is done on the client-side before sending it to Amazon S3

* ​

Server-Side Encryption with Amazon S3-Managed Keys (SSE-S3)

* ​

Server-Side Encryption with Customer Master Keys (CMKs) Stored in AWS Key Management Service (SSE-KMS)

**Explanation**

Correct option:

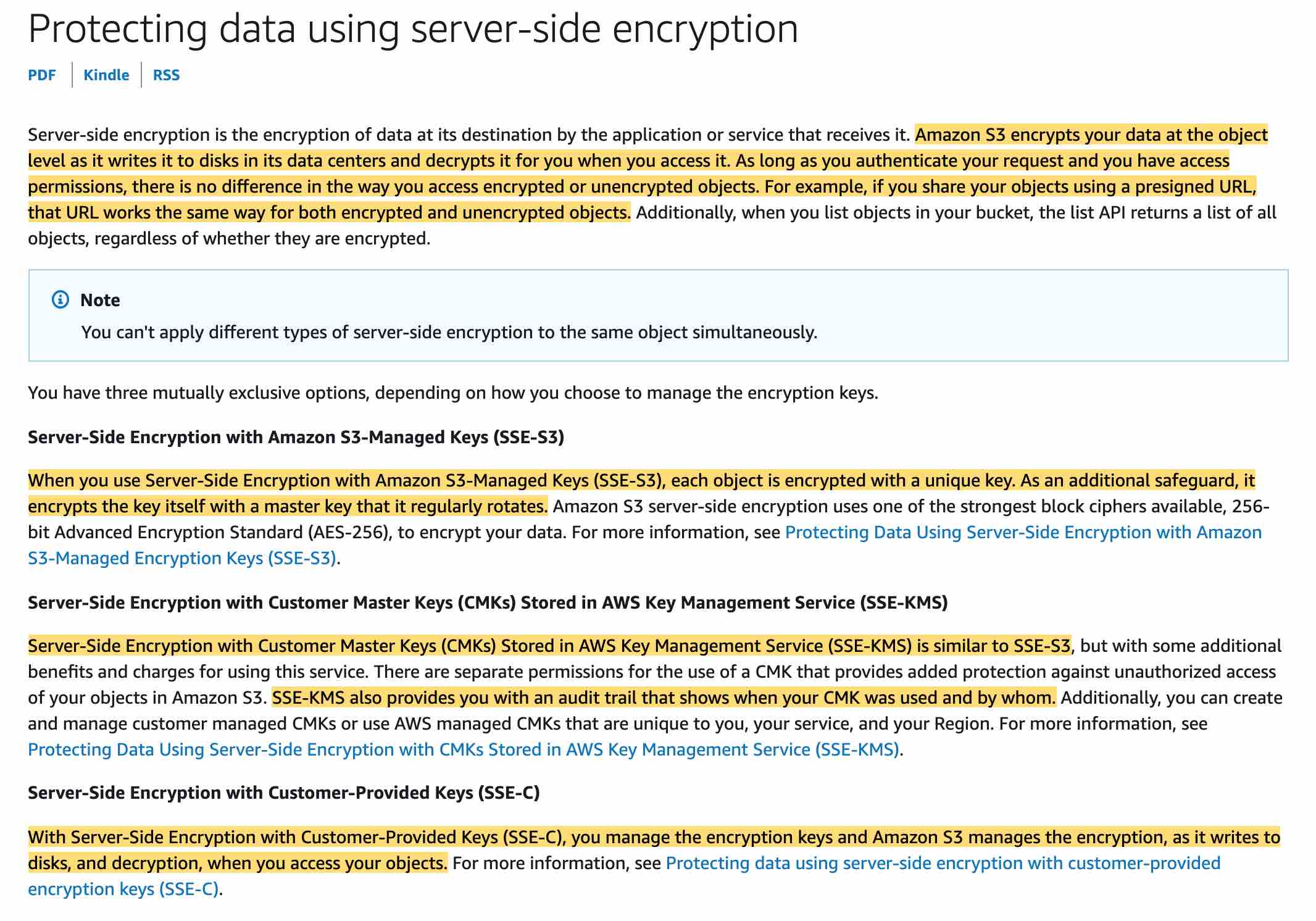
**Server-Side Encryption with Customer-Provided Keys (SSE-C)**

You have the following options for protecting data at rest in Amazon S3:

Server-Side Encryption – Request Amazon S3 to encrypt your object before saving it on disks in its data centers and then decrypt it when you download the objects.

Client-Side Encryption – Encrypt data client-side and upload the encrypted data to Amazon S3. In this case, you manage the encryption process, the encryption keys, and related tools.

For the given use-case, the company wants to manage the encryption keys via its custom application and let S3 manage the encryption, therefore you must use Server-Side Encryption with Customer-Provided Keys (SSE-C).

Please review these three options for Server Side Encryption on S3:  via - <https://docs.aws.amazon.com/AmazonS3/latest/dev/serv-side-encryption.html>

Incorrect options:

**Server-Side Encryption with Amazon S3-Managed Keys (SSE-S3)** - When you use Server-Side Encryption with Amazon S3-Managed Keys (SSE-S3), each object is encrypted with a unique key. As an additional safeguard, it encrypts the key itself with a master key that it regularly rotates. So this option is incorrect.

**Server-Side Encryption with Customer Master Keys (CMKs) Stored in AWS Key Management Service (SSE-KMS)** - Server-Side Encryption with Customer Master Keys (CMKs) stored in AWS Key Management Service (SSE-KMS) is similar to SSE-S3. SSE-KMS provides you with an audit trail that shows when your CMK was used and by whom. Additionally, you can create and manage customer-managed CMKs or use AWS managed CMKs that are unique to you, your service, and your Region.

**Client-Side Encryption with data encryption is done on the client-side before sending it to Amazon S3** - You can encrypt the data client-side and upload the encrypted data to Amazon S3. In this case, you manage the encryption process, the encryption keys, and related tools.

Reference:

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

Bottom of Form

Top of Form

Question 11: Skipped

**A developer from your team has configured the load balancer to route traffic equally between instances or across Availability Zones. However, Elastic Load Balancing (ELB) routes more traffic to one instance or Availability Zone than the others.**

**Why is this happening and how can it be fixed? (Select two)**

* ​

For Application Load Balancers, cross-zone load balancing is disabled by default

* ​

Instances of a specific capacity type aren’t equally distributed across Availability Zones

**(Correct)**

* ​

After you disable an Availability Zone, the targets in that Availability Zone remain registered with the load balancer, thereby receiving random bursts of traffic

* ​

There could be short-lived TCP connections between clients and instances

* ​

Sticky sessions are enabled for the load balancer

**(Correct)**

**Explanation**

Correct option:

**Sticky sessions are enabled for the load balancer** - This can be the reason for potential unequal traffic routing by the load balancer. Sticky sessions are a mechanism to route requests to the same target in a target group. This is useful for servers that maintain state information in order to provide a continuous experience to clients. To use sticky sessions, the clients must support cookies.

When a load balancer first receives a request from a client, it routes the request to a target, generates a cookie named AWSALB that encodes information about the selected target, encrypts the cookie, and includes the cookie in the response to the client. The client should include the cookie that it receives in subsequent requests to the load balancer. When the load balancer receives a request from a client that contains the cookie, if sticky sessions are enabled for the target group and the request goes to the same target group, the load balancer detects the cookie and routes the request to the same target.

If you use duration-based session stickiness, configure an appropriate cookie expiration time for your specific use case. If you set session stickiness from individual applications, use session cookies instead of persistent cookies where possible.

**Instances of a specific capacity type aren’t equally distributed across Availability Zones** - A Classic Load Balancer with HTTP or HTTPS listeners might route more traffic to higher-capacity instance types. This distribution aims to prevent lower-capacity instance types from having too many outstanding requests. It’s a best practice to use similar instance types and configurations to reduce the likelihood of capacity gaps and traffic imbalances.

A traffic imbalance might also occur if you have instances of similar capacities running on different Amazon Machine Images (AMIs). In this scenario, the imbalance of the traffic in favor of higher-capacity instance types is desirable.

Incorrect options:

**There could be short-lived TCP connections between clients and instances** - This is an incorrect statement. Long-lived TCP connections between clients and instances can potentially lead to unequal distribution of traffic by the load balancer. Long-lived TCP connections between clients and instances cause uneven traffic load distribution by design. As a result, new instances take longer to reach connection equilibrium. Be sure to check your metrics for long-lived TCP connections that might be causing routing issues in the load balancer.

**For Application Load Balancers, cross-zone load balancing is disabled by default** - This is an incorrect statement. With Application Load Balancers, cross-zone load balancing is always enabled.

**After you disable an Availability Zone, the targets in that Availability Zone remain registered with the load balancer, thereby receiving random bursts of traffic** - This is an incorrect statement. After you disable an Availability Zone, the targets in that Availability Zone remain registered with the load balancer. However, even though they remain registered, the load balancer does not route traffic to them.

References:

<https://docs.aws.amazon.com/elasticloadbalancing/latest/application/load-balancer-target-groups.html#sticky-sessions>

<https://aws.amazon.com/premiumsupport/knowledge-center/elb-fix-unequal-traffic-routing/>

<https://docs.aws.amazon.com/elasticloadbalancing/latest/userguide/how-elastic-load-balancing-works.html#availability-zones>

Bottom of Form

Top of Form

Question 12: Skipped

**You are storing your video files in a separate S3 bucket than your main static website in an S3 bucket. When accessing the video URLs directly the users can view the videos on the browser, but they can't play the videos while visiting the main website.**

**What is the root cause of this problem?**

* ​

Change the bucket policy

* ​

Enable CORS

**(Correct)**

* ​

Disable Server-Side Encryption

* ​

Amend the IAM policy

**Explanation**

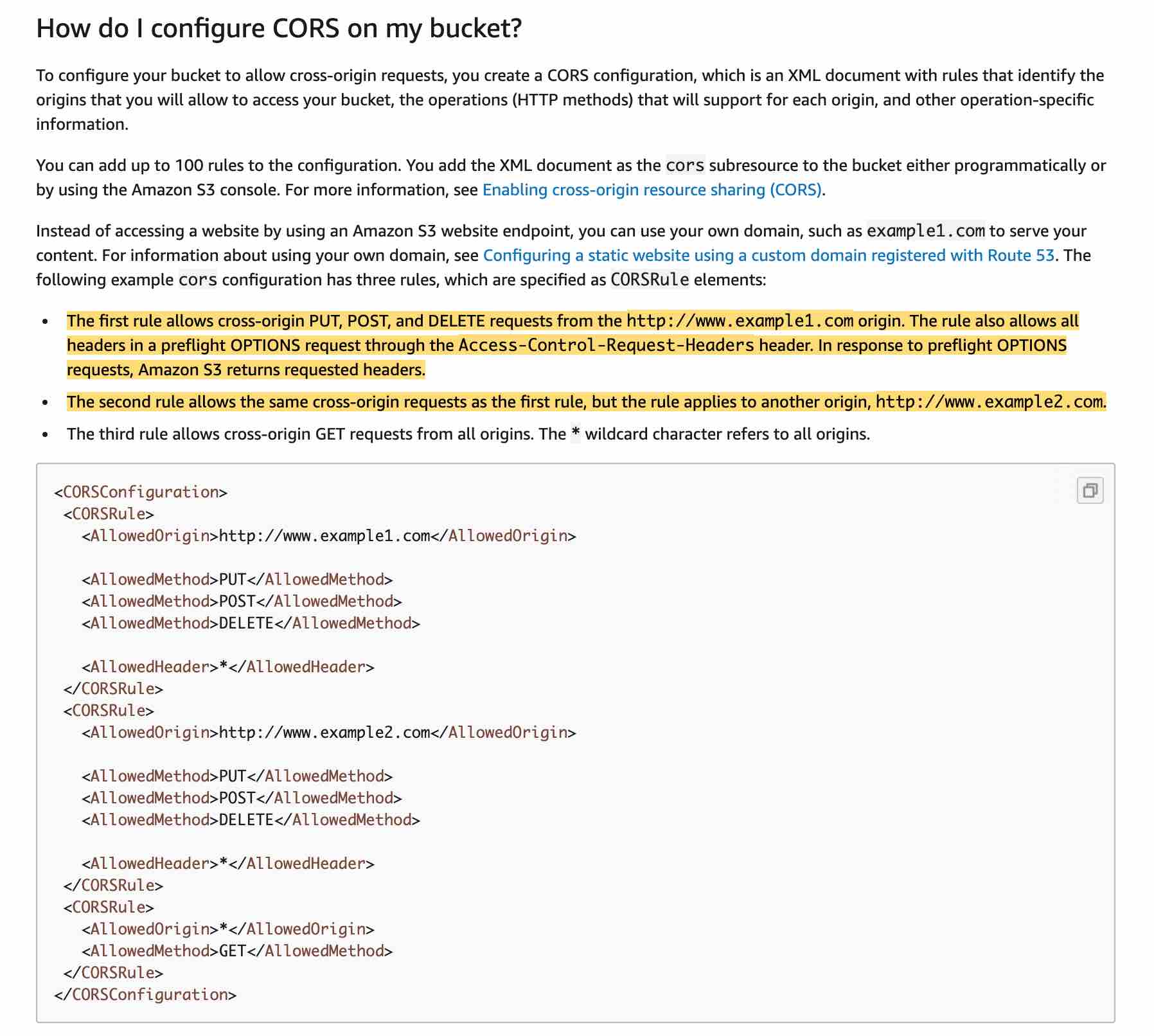
Correct option:

**Enable CORS**

Cross-origin resource sharing (CORS) defines a way for client web applications that are loaded in one domain to interact with resources in a different domain. With CORS support, you can build rich client-side web applications with Amazon S3 and selectively allow cross-origin access to your Amazon S3 resources.

To configure your bucket to allow cross-origin requests, you create a CORS configuration, which is an XML document with rules that identify the origins that you will allow to access your bucket, the operations (HTTP methods) that will support for each origin, and other operation-specific information.

For the given use-case, you would create a <CORSRule> in <CORSConfiguration> for bucket B to allow access from the S3 website origin hosted on bucket A.

Please see this note for more details:  via - <https://docs.aws.amazon.com/AmazonS3/latest/dev/cors.html>

Incorrect options:

**Change the bucket policy** - A bucket policy is a resource-based AWS Identity and Access Management (IAM) policy that grants permissions. With this policy, you can do things such as allow one IP address to access the video file in the S3 bucket. In this scenario, we know that's not the case because it works using the direct URL but it doesn't work when you click on a link to access the video.

**Amend the IAM policy** - You attach IAM policies to IAM users, groups, or roles, which are then subject to the permissions you've defined. This scenario refers to public users of a website and they need not have an IAM user account.

**Disable Server-Side Encryption** - Amazon S3 encrypts your data at the object level as it writes it to disks in its data centers and decrypts it for you when you access it, if the video file is encrypted at rest then there is nothing you need to do because AWS handles encrypt and decrypt. Disabling encryption is not an issue because you can access the video directly using an URL but not from the main website.

Reference:

<https://docs.aws.amazon.com/AmazonS3/latest/dev/cors.html>

Bottom of Form

Top of Form

Question 13: Skipped

**You were assigned to a project that requires the use of the AWS CLI to build a project with AWS CodeBuild. Your project's root directory includes the buildspec.yml file to run build commands and would like your build artifacts to be automatically encrypted at the end.**

**How should you configure CodeBuild to accomplish this?**

* ​

Use In Flight encryption (SSL)

* ​

Use an AWS Lambda Hook

* ​

Specify a KMS key to use

**(Correct)**

* ​

Use the AWS Encryption SDK

**Explanation**

Correct option:

**Specify a KMS key to use**

AWS Key Management Service (KMS) makes it easy for you to create and manage cryptographic keys and control their use across a wide range of AWS services and in your applications.

For AWS CodeBuild to encrypt its build output artifacts, it needs access to an AWS KMS customer master key (CMK). By default, AWS CodeBuild uses the AWS-managed CMK for Amazon S3 in your AWS account. The following environment variable provides these details:

CODEBUILD\_KMS\_KEY\_ID: The identifier of the AWS KMS key that CodeBuild is using to encrypt the build output artifact (for example, arn:aws:kms:region-ID:account-ID:key/key-ID or alias/key-alias).

Incorrect options:

**Use an AWS Lambda Hook** - Code hook is used for integration with Lambda and is not relevant for the given use-case.

**Use the AWS Encryption SDK** - The SDK just makes it easier for you to implement encryption best practices in your application and is not relevant for the given use-case.

**Use In-Flight encryption (SSL)** - SSL is usually for internet traffic which in this case will be using internal traffic through AWS and is not relevant for the given use-case.

References:

<https://docs.aws.amazon.com/codebuild/latest/userguide/build-env-ref-env-vars.html>

<https://docs.aws.amazon.com/codebuild/latest/userguide/setting-up.html>

Bottom of Form

Top of Form

Question 14: Skipped

**Your company has been hired to build a resilient mobile voting app for an upcoming music award show that expects to have 5 to 20 million viewers. The mobile voting app will be marketed heavily months in advance so you are expected to handle millions of messages in the system. You are configuring Amazon Simple Queue Service (SQS) queues for your architecture that should receive messages from 20 KB to 200 KB.**

**Is it possible to send these messages to SQS?**

* ​

No, the max message size is 64KB

* ​

Yes, the max message size is 512KB

* ​

Yes, the max message size is 256KB

**(Correct)**

* ​

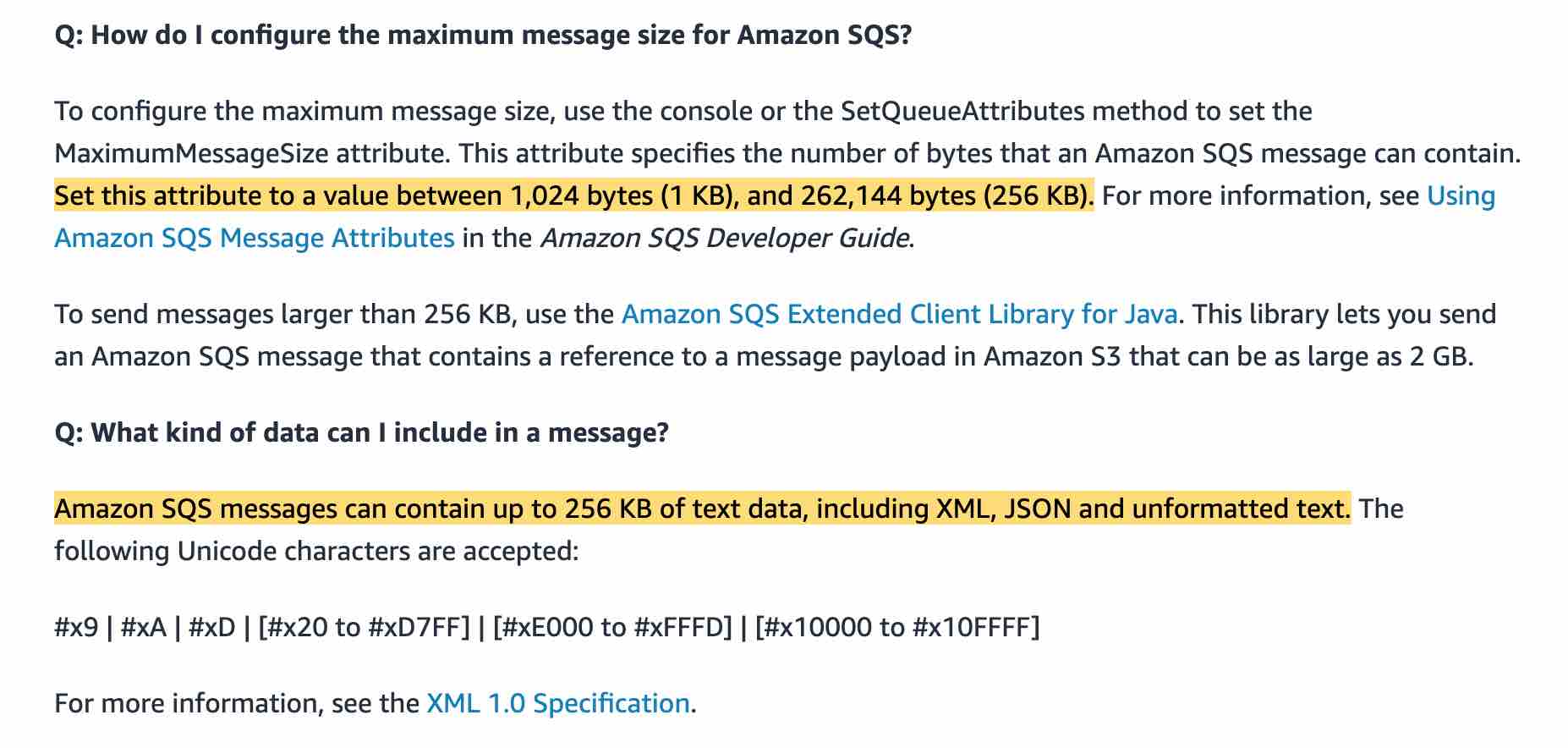
No, the max message size is 128KB

**Explanation**

Correct option:

**Yes, the max message size is 256KB**

The minimum message size is 1 byte (1 character). The maximum is 262,144 bytes (256 KB).

 via - <https://aws.amazon.com/sqs/faqs/>

Incorrect options:

**Yes, the max message size is 512KB** - The max size is 256KB

**No, the max message size is 128KB** - The max size is 256KB

**No, the max message size is 64KB** - The max size is 256KB

Reference:

<https://aws.amazon.com/sqs/faqs/>

Bottom of Form

Top of Form

Question 15: Skipped

**Your team has just signed up an year-long contract with a client maintaining a three-tier web application, that needs to be moved to AWS Cloud. The application has steady traffic throughout the day and needs to be on a reliable system with no down-time or access issues. The solution needs to be cost-optimal for this startup.**

**Which of the following options should you choose?**

* ​

Amazon EC2 On Demand Instances

* ​

On-premise EC2 instance

* ​

Amazon EC2 Reserved Instances

**(Correct)**

* ​

Amazon EC2 Spot Instances

**Explanation**

Correct option:

**Amazon EC2 Reserved Instances** - Reserved instances can provide a capacity reservation, offering additional confidence in your ability to launch the number of instances you have reserved when you need them. You save money going with Reserved instances vs on-demand especially in a year's worth of time.

Reserved Instances are not physical instances, but rather a billing discount applied to the use of On-Demand Instances in your account. These On-Demand Instances must match certain attributes, such as instance type and Region, to benefit from the billing discount. So, there is no performance difference between an On-Demand instance or a Reserved instance.

How RIs work:  via - <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-reserved-instances.html>

Incorrect options:

**Amazon EC2 Spot Instances** - A Spot Instance is an unused EC2 instance that is available for less than the On-Demand price. Because Spot Instances enable you to request unused EC2 instances at steep discounts, you can lower your Amazon EC2 costs significantly. Spot instances are useful if your applications can be interrupted, like data analysis, batch jobs, background processing, and optional tasks. Spot instances can be pulled down anytime without prior notice. Hence, not the right choice for the current scenario.

**Amazon EC2 On-Demand Instances** - With On-Demand Instances, you pay for 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. But, On-Demand instances cost a lot more than Reserved instances. Here, in our use case, we already know that the systems are required for a complete year, so making use of Reserved Instances discount makes a lot more sense.

**On-premise EC2 instance** - On-premise implies the client has to maintain the physical machines, their capacity provisioning and maintenance. Not an option when the client is planning to move to AWS Cloud.

References:

<https://aws.amazon.com/ec2/pricing/reserved-instances/>

<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-reserved-instances.html>

<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-spot-instances.html>

<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-on-demand-instances.html>

Bottom of Form

Top of Form

Question 16: Skipped

**A cyber forensics application, running behind an ALB, wants to analyze patterns for the client IPs.**

**Which of the following headers can be used for this requirement?**

* ​

X-Forwarded-IP

* ​

X-Forwarded-For

**(Correct)**

* ​

X-Forwarded-Port

* ​

X-Forwarded-Proto

**Explanation**

Correct option:

HTTP requests and HTTP responses use header fields to send information about the HTTP messages. Header fields are colon-separated name-value pairs that are separated by a carriage return (CR) and a line feed (LF).

**X-Forwarded-For** - The X-Forwarded-For request header helps you identify the IP address of a client when you use an HTTP or HTTPS load balancer. Because load balancers intercept traffic between clients and servers, your server access logs contain only the IP address of the load balancer. To see the IP address of the client, use the X-Forwarded-For request header.

Incorrect options:

**X-Forwarded-Proto** - The X-Forwarded-Proto request header helps you identify the protocol (HTTP or HTTPS) that a client used to connect to your load balancer. Your server access logs contain only the protocol used between the server and the load balancer; they contain no information about the protocol used between the client and the load balancer. To determine the protocol used between the client and the load balancer, use the X-Forwarded-Proto request header.

**X-Forwarded-Port** - The X-Forwarded-Port request header helps you identify the destination port that the client used to connect to the load balancer.

**X-Forwarded-IP** - This is a made-up option and has been added as a distractor.

Reference:

<https://docs.aws.amazon.com/elasticloadbalancing/latest/classic/x-forwarded-headers.html#x-forwarded-for>

Bottom of Form

Top of Form

Question 17: Skipped

**An e-commerce company has a fleet of EC2 based web servers running into very high CPU utilization issues. The development team has determined that serving secure traffic via HTTPS is a major contributor to the high CPU load.**

**Which of the following steps can take the high CPU load off the web servers? (Select two)**

* ​

Configure an SSL/TLS certificate on an Application Load Balancer via AWS Certificate Manager (ACM)

**(Correct)**

* ​

Create an HTTPS listener on the Application Load Balancer with SSL termination

**(Correct)**

* ​

Create an HTTPS listener on the Application Load Balancer with SSL pass-through

* ​

Create an HTTP listener on the Application Load Balancer with SSL termination

* ​

Create an HTTP listener on the Application Load Balancer with SSL pass-through

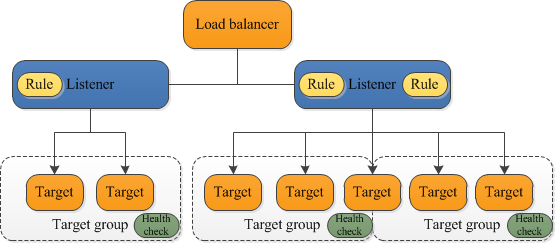
**Explanation**

Correct option:

"Configure an SSL/TLS certificate on an Application Load Balancer via AWS Certificate Manager (ACM)"

"Create an HTTPS listener on the Application Load Balancer with SSL termination"

An Application load balancer distributes incoming application traffic across multiple targets, such as EC2 instances, in multiple Availability Zones. A listener checks for connection requests from clients, using the protocol and port that you configure. The rules that you define for a listener determine how the load balancer routes requests to its registered targets. Each rule consists of a priority, one or more actions, and one or more conditions.

 via - <https://docs.aws.amazon.com/elasticloadbalancing/latest/application/introduction.html>

To use an HTTPS listener, you must deploy at least one SSL/TLS server certificate on your load balancer. You can create an HTTPS listener, which uses encrypted connections (also known as SSL offload). This feature enables traffic encryption between your load balancer and the clients that initiate SSL or TLS sessions. As the EC2 instances are under heavy CPU load, the load balancer will use the server certificate to terminate the front-end connection and then decrypt requests from clients before sending them to the EC2 instances.

Please review this resource to understand how to associate an ACM SSL/TLS certificate with an Application Load Balancer: <https://aws.amazon.com/premiumsupport/knowledge-center/associate-acm-certificate-alb-nlb/>

Incorrect options:

"Create an HTTPS listener on the Application Load Balancer with SSL pass-through" - If you use an HTTPS listener with SSL pass-through, then the EC2 instances would continue to be under heavy CPU load as they would still need to decrypt the secure traffic at the instance level. Hence this option is incorrect.

"Create an HTTP listener on the Application Load Balancer with SSL termination"

"Create an HTTP listener on the Application Load Balancer with SSL pass-through"

You cannot have an HTTP listener for an Application Load Balancer to support SSL termination or SSL pass-through, so both these options are incorrect.

References:

<https://docs.aws.amazon.com/elasticloadbalancing/latest/application/introduction.html>

<https://aws.amazon.com/premiumsupport/knowledge-center/associate-acm-certificate-alb-nlb/>

Bottom of Form

Top of Form

Question 18: Skipped

**A popular mobile app retrieves data from an AWS DynamoDB table that was provisioned with read-capacity units (RCU’s) that are evenly shared across four partitions. One of those partitions is receiving more traffic than the other partitions, causing hot partition issues.**

**What technology will allow you to reduce the read traffic on your AWS DynamoDB table with minimal effort?**

* ​

DynamoDB Streams

* ​

More partitions

* ​

ElastiCache

* ​

DynamoDB DAX

**(Correct)**

**Explanation**

Correct option:

**DynamoDB DAX**

Amazon DynamoDB Accelerator (DAX) is a fully managed, highly available, in-memory cache for DynamoDB that delivers up to a 10x performance improvement: from milliseconds to microseconds: even at millions of requests per second.

Incorrect options:

**DynamoDB Streams** - A stream record contains information about a data modification to a single item in a DynamoDB table. This is not the correct option for the given use-case.

**ElastiCache** - ElastiCache can cache the results from anything but you will need to modify your code to check the cache before querying the main query store. As the given use-case mandates minimal effort, so this option is not correct.

**More partitions** - This option has been added as a distractor as DynamoDB handles that for you automatically.

Reference:

<https://aws.amazon.com/dynamodb/dax/>

Bottom of Form

Top of Form

Question 19: Skipped

**You are working with a t2.small instance bastion host that has the AWS CLI installed to help manage all the AWS services installed on it. You would like to know the security group and the instance id of the current instance.**

**Which of the following will help you fetch the needed information?**

* ​

Query the metadata at http://169.254.169.254/latest/meta-data

**(Correct)**

* ​

Query the user data at http://254.169.254.169/latest/meta-data

* ​

Query the user data at http://169.254.169.254/latest/user-data

* ​

Create an IAM role and attach it to your EC2 instance that helps you perform a 'describe' API call

**Explanation**

Correct option:

**Query the metadata at http://169.254.169.254/latest/meta-data** - Because your instance metadata is available from your running instance, you do not need to use the Amazon EC2 console or the AWS CLI. This can be helpful when you're writing scripts to run from your instance. For example, you can access the local IP address of your instance from instance metadata to manage a connection to an external application. To view all categories of instance metadata from within a running instance, use the following URI - http://169.254.169.254/latest/meta-data/. The IP address 169.254.169.254 is a link-local address and is valid only from the instance. All instance metadata is returned as text (HTTP content type text/plain).

Incorrect options:

**Create an IAM role and attach it to your EC2 instance that helps you perform a 'describe' API call** - The AWS CLI has a describe-instances API call needs instance ID as an input. So, this will not work for the current use case wherein we do not know the instance ID.

**Query the user data at http://169.254.169.254/latest/user-data** - This address retrieves the user data that you specified when launching your instance.

**Query the user data at http://254.169.254.169/latest/meta-data** - The IP address specified is wrong.

References:

<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/instancedata-data-retrieval.html>

<https://awscli.amazonaws.com/v2/documentation/api/latest/reference/ec2/describe-instances.html>

Bottom of Form

Top of Form

Question 20: Skipped

**A .NET developer team works with many ASP.NET web applications that use EC2 instances to host them on IIS. The deployment process needs to be configured so that multiple versions of the application can run in AWS Elastic Beanstalk. One version would be used for development, testing, and another version for load testing.**

**Which of the following methods do you recommend?**

* ​

Create an Application Load Balancer to route based on hostname so you can pass on parameters to the development Elastic Beanstalk environment. Create a file in .ebextensions/ to know how to handle the traffic coming from the ALB

* ​

Use only one Beanstalk environment and perform configuration changes using an Ansible script

* ​

You cannot have multiple development environments in Elastic Beanstalk, just one development and one production environment

* ​

Define a dev environment with a single instance and a 'load test' environment that has settings close to production environment

**(Correct)**

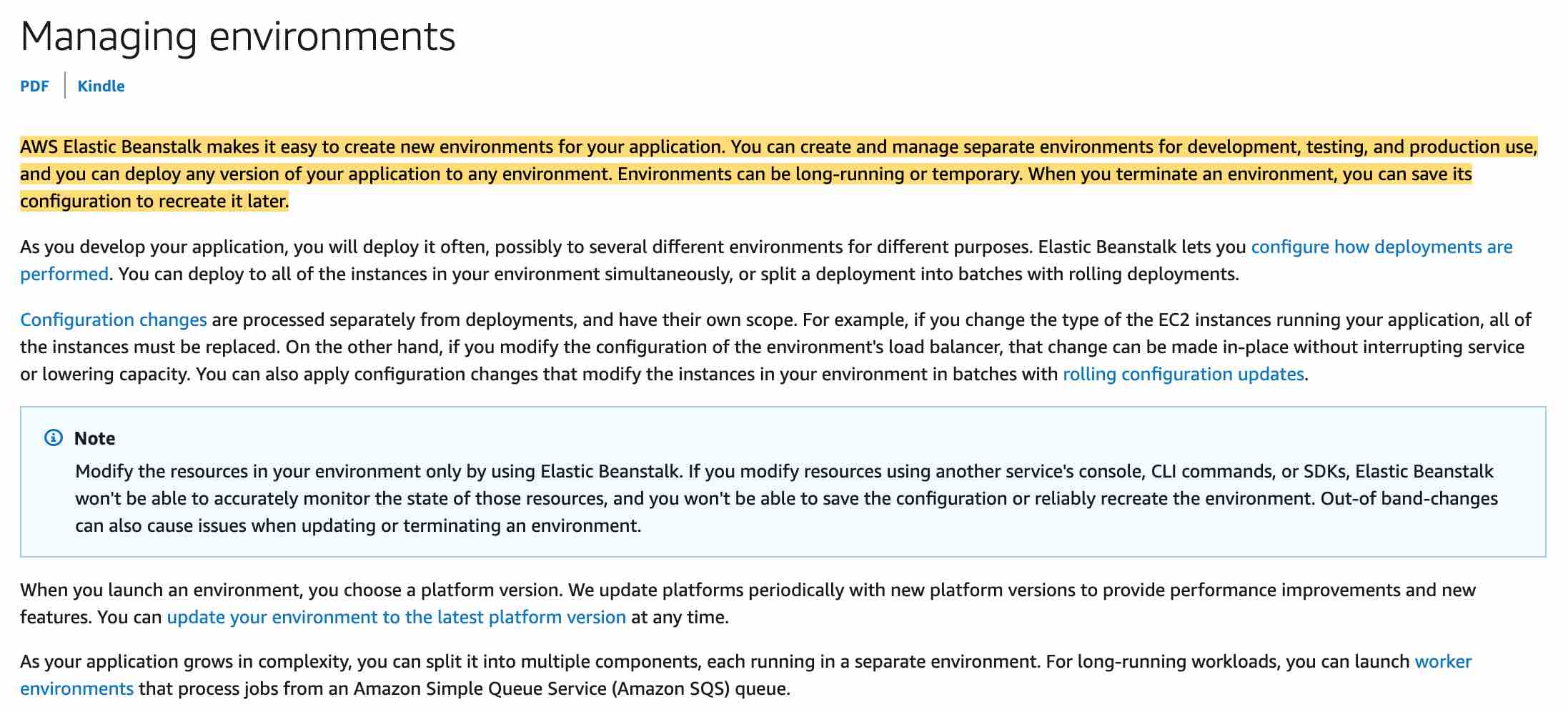
**Explanation**

Correct option:

**Define a dev environment with a single instance and a 'load test' environment that has settings close to production environment**

AWS Elastic Beanstalk makes it easy to create new environments for your application. You can create and manage separate environments for development, testing, and production use, and you can deploy any version of your application to any environment. Environments can be long-running or temporary. When you terminate an environment, you can save its configuration to recreate it later.

It is common practice to have many environments for the same application. You can deploy multiple environments when you need to run multiple versions of an application. So for the given use-case, you can set up 'dev' and 'load test' environment.

 via - <https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/using-features.managing.html>

**You cannot have multiple development environments in Elastic Beanstalk, just one development, and one production environment** - Incorrect, use the Create New Environment wizard in the AWS Management Console for BeanStalk to guide you on this.

**Use only one Beanstalk environment and perform configuration changes using an Ansible script** - Ansible is an open-source deployment tool that integrates with AWS. It allows us to deploy the infrastructure. Elastic Beanstalk provisions the servers that you need for hosting the application and it also handles multiple environments, so Beanstalk is a better option.

**Create an Application Load Balancer to route based on hostname so you can pass on parameters to the development Elastic Beanstalk environment. Create a file in .ebextensions/ to know how to handle the traffic coming from the ALB** - This is not a good design if you need to load test because you will have two versions on the same instances and may not be able to access resources in the system due to the load testing.

Reference:

<https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/using-features.environments.html>

Bottom of Form

Top of Form

Question 21: Skipped

**The development team at an IT company has configured an Application Load Balancer (ALB) with a Lambda function A as the target but the Lambda function A is not able to process any request from the ALB. Upon investigation, the team finds that there is another Lambda function B in the AWS account that is exceeding the concurrency limits.**

**How can the development team address this issue?**

* ​

Use an API Gateway instead of an Application Load Balancer (ALB) for Lambda function A

* ​

Set up provisioned concurrency for the Lambda function B so that it throttles if it goes above a certain concurrency limit

* ​

Set up reserved concurrency for the Lambda function B so that it throttles if it goes above a certain concurrency limit

**(Correct)**

* ​

Use a Cloudfront Distribution instead of an Application Load Balancer (ALB) for Lambda function A

**Explanation**

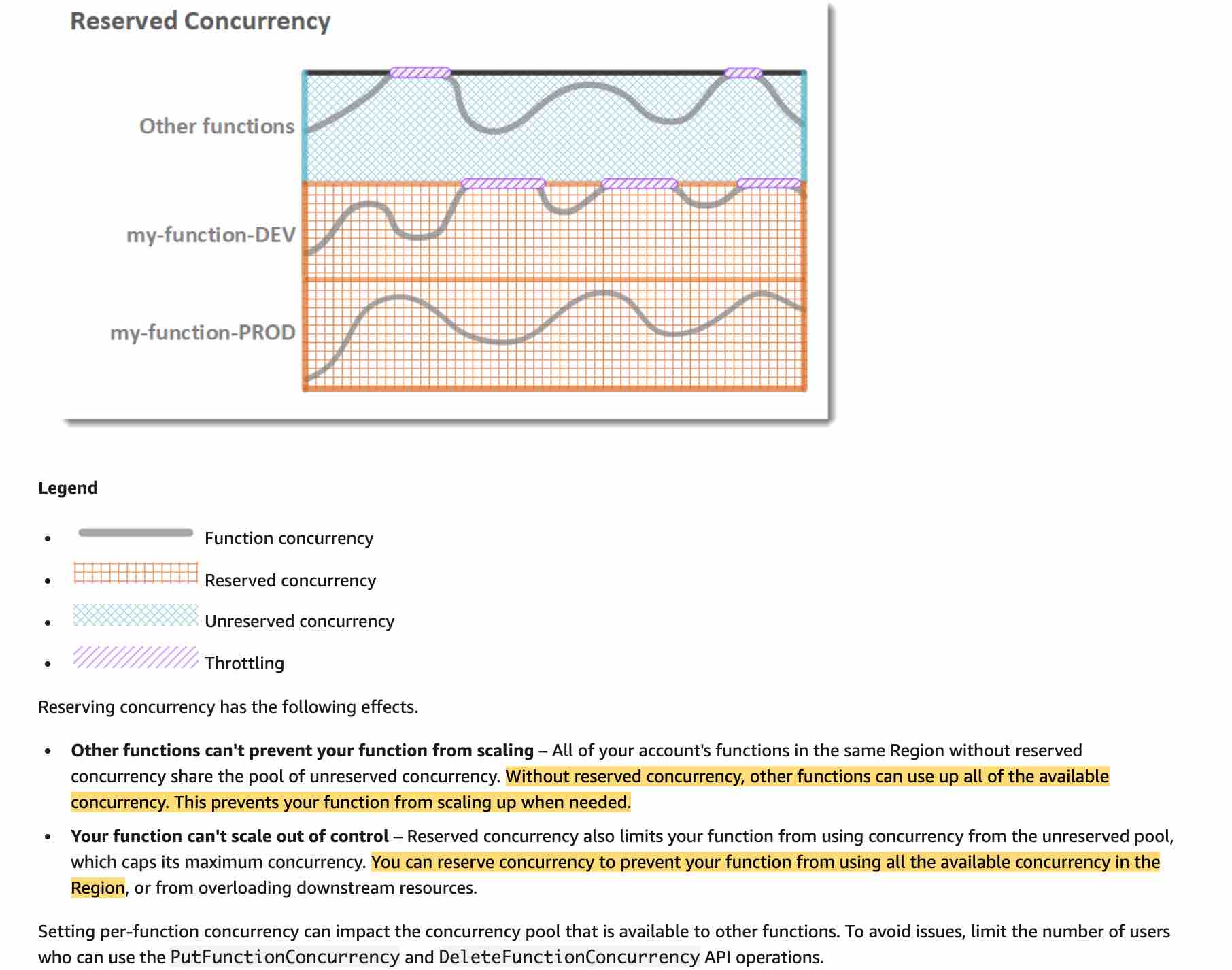
Correct option:

**Set up reserved concurrency for the Lambda function B so that it throttles if it goes above a certain concurrency limit**

Concurrency is the number of requests that a Lambda function is serving at any given time. If a Lambda function is invoked again while a request is still being processed, another instance is allocated, which increases the function's concurrency.

To ensure that a function can always reach a certain level of concurrency, you can configure the function with reserved concurrency. When a function has reserved concurrency, no other function can use that concurrency. More importantly, reserved concurrency also limits the maximum concurrency for the function, and applies to the function as a whole, including versions and aliases.

Please review this note to understand how reserved concurrency works:

 via - <https://docs.aws.amazon.com/lambda/latest/dg/configuration-concurrency.html#configuration-concurrency-reserved>

Therefore using reserved concurrency for Lambda function B would limit its maximum concurrency and allow Lambda function A to execute without getting throttled.

Incorrect options:

**Set up provisioned concurrency for the Lambda function B so that it throttles if it goes above a certain concurrency limit** - You should use provisioned concurrency to enable your function to scale without fluctuations in latency. By allocating provisioned concurrency before an increase in invocations, you can ensure that all requests are served by initialized instances with very low latency. Provisioned concurrency is not used to limit the maximum concurrency for a given Lambda function, so this option is incorrect.

**Use an API Gateway instead of an Application Load Balancer (ALB) for Lambda function A** - This has been added as a distractor as using an API Gateway for Lambda function A has no bearing on limiting the concurrency of Lambda function B, so this option is incorrect.

**Use a Cloudfront Distribution instead of an Application Load Balancer (ALB) for Lambda function A** - When you associate a CloudFront distribution with a Lambda function (known as Lambda@Edge), CloudFront intercepts requests and responses at CloudFront edge locations and runs the function. Again, this has no bearing on limiting the concurrency of Lambda function B, so this option is incorrect.

References:

<https://docs.aws.amazon.com/lambda/latest/dg/configuration-concurrency.html#configuration-concurrency-reserved>

<https://aws.amazon.com/blogs/networking-and-content-delivery/lambda-functions-as-targets-for-application-load-balancers/>

Bottom of Form

Top of Form

Question 22: Skipped

**You are designing a high-performance application that requires millions of connections. You have several EC2 instances running Apache2 web servers and the application will require capturing the user’s source IP address and source port without the use of X-Forwarded-For.**

**Which of the following options will meet your needs?**

* ​

Elastic Load Balancer

* ​

Classic Load Balancer

* ​

Network Load Balancer

**(Correct)**

* ​

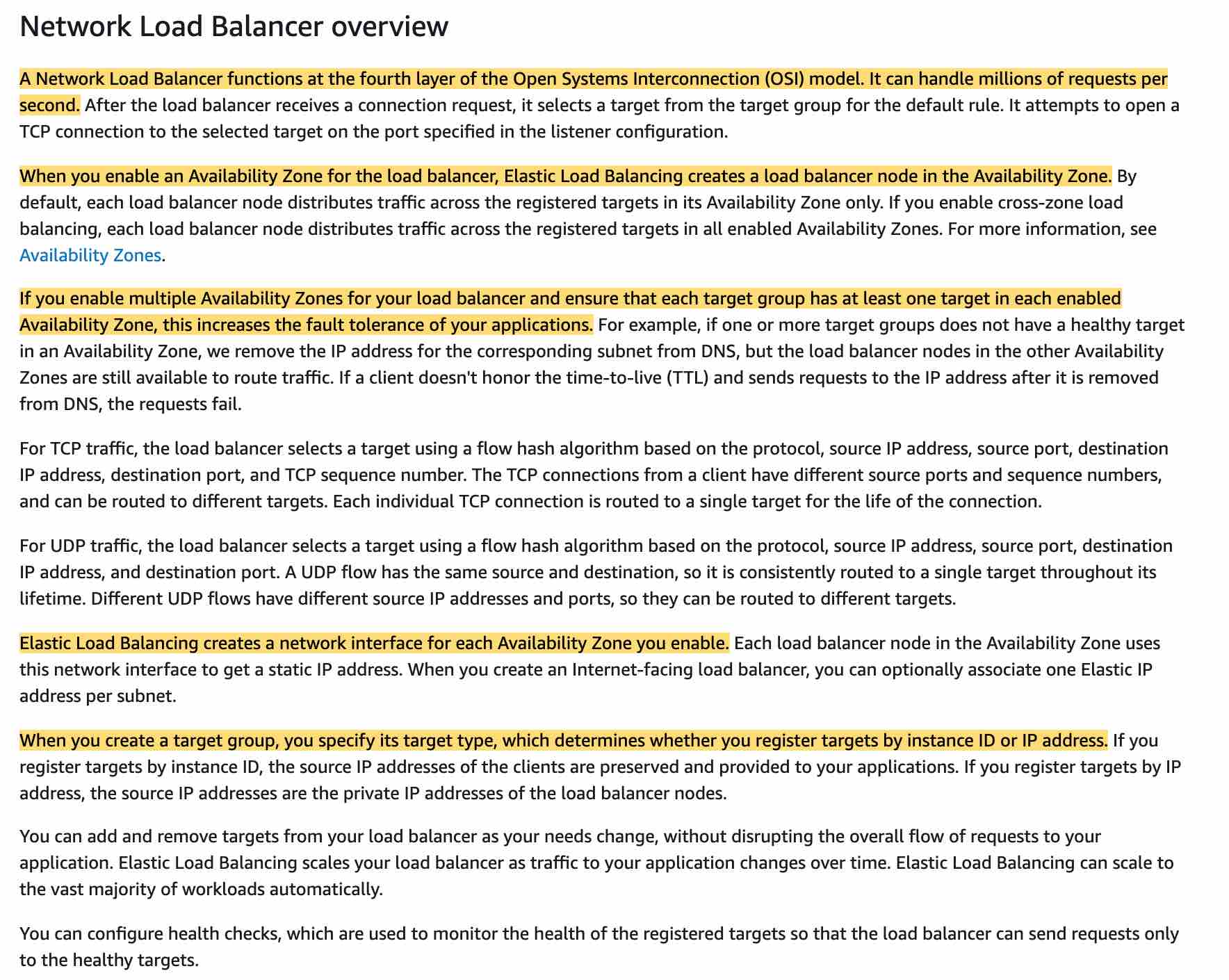
Application Load Balancer

**Explanation**

Correct option:

**Network Load Balancer**

A Network Load Balancer functions at the fourth layer of the Open Systems Interconnection (OSI) model. It can handle millions of requests per second. After the load balancer receives a connection request, it selects a target from the target group for the default rule. It attempts to open a TCP connection to the selected target on the port specified in the listener configuration. Incoming connections remain unmodified, so application software need not support X-Forwarded-For.

 via - <https://docs.aws.amazon.com/elasticloadbalancing/latest/network/introduction.html>

Incorrect options:

**Application Load Balancer** - An Application Load Balancer functions at the application layer, the seventh layer of the Open Systems Interconnection (OSI) model. After the load balancer receives a request, it evaluates the listener rules in priority order to determine which rule to apply and then selects a target from the target group for the rule action.

One of many benefits of the Application Load Balancer is its support for path-based routing. You can configure rules for your listener that forward requests based on the URL in the request. This enables you to structure your application as smaller services, and route requests to the correct service based on the content of the URL. For needs relating to network traffic go with Network Load Balancer.

**Elastic Load Balancer** - Elastic Load Balancing is the service itself that offers different types of load balancers.

**Classic Load Balancer** - It is a basic load balancer that distributes traffic. If your account was created before 2013-12-04, your account supports EC2-Classic instances and you will benefit in using this type of load balancer. The classic load balancer can be used regardless of when your account was created and whether you use EC2-Classic or whether your instances are in a VPC but just remember its the basic load balancer AWS offers and not advanced as the others.

Reference:

<https://docs.aws.amazon.com/elasticloadbalancing/latest/network/introduction.html>

Bottom of Form

Top of Form

Question 23: Skipped

**A company has a workload that requires 14,000 consistent IOPS for data that must be durable and secure. The compliance standards of the company state that the data should be secure at every stage of its lifecycle on all of the EBS volumes they use.**

**Which of the following statements are true regarding data security on EBS?**

* ​

EBS volumes don't support any encryption

* ​

EBS volumes support both in-flight encryption and encryption at rest using KMS

**(Correct)**

* ​

EBS volumes do not support in-flight encryption but do support encryption at rest using KMS

* ​

EBS volumes support in-flight encryption but does not support encryption at rest

**Explanation**

Correct option:

Amazon EBS works with AWS KMS to encrypt and decrypt your EBS volume. You can encrypt both the boot and data volumes of an EC2 instance. When you create an encrypted EBS volume and attach it to a supported instance type, the following types of data are encrypted:

1. Data at rest inside the volume
2. All data moving between the volume and the instance
3. All snapshots created from the volume
4. All volumes created from those snapshots

**EBS volumes support both in-flight encryption and encryption at rest using KMS** - This is a correct statement. Encryption operations occur on the servers that host EC2 instances, ensuring the security of both data-at-rest and data-in-transit between an instance and its attached EBS storage.

Incorrect options:

**EBS volumes support in-flight encryption but do not support encryption at rest** - This is an incorrect statement. As discussed above, all data moving between the volume and the instance is encrypted.

**EBS volumes do not support in-flight encryption but do support encryption at rest using KMS** - This is an incorrect statement. As discussed above, data at rest is also encrypted.

**EBS volumes don't support any encryption** - This is an incorrect statement. Amazon EBS encryption offers a straight-forward encryption solution for your EBS resources associated with your EC2 instances. With Amazon EBS encryption, you aren't required to build, maintain, and secure your own key management infrastructure.

Reference:

<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EBSEncryption.html>

Bottom of Form

Top of Form

Question 24: Skipped

**You are working for a technology startup building web and mobile applications. You would like to pull Docker images from the ECR repository called demo so you can start running local tests against the latest application version.**

**Which of the following commands must you run to pull existing Docker images from ECR? (Select two)**

* ​

aws docker push 1234567890.dkr.ecr.eu-west-1.amazonaws.com/demo:latest

* ​

docker build -t 1234567890.dkr.ecr.eu-west-1.amazonaws.com/demo:latest

* ​

docker login -u $AWS\_ACCESS\_KEY\_ID -p $AWS\_SECRET\_ACCESS\_KEY

* ​

docker pull 1234567890.dkr.ecr.eu-west-1.amazonaws.com/demo:latest

**(Correct)**

* ​

$(aws ecr get-login --no-include-email)

**(Correct)**

**Explanation**

Correct options:

**$(aws ecr get-login --no-include-email)**

**docker pull 1234567890.dkr.ecr.eu-west-1.amazonaws.com/demo:latest**

The get-login command retrieves a token that is valid for a specified registry for 12 hours, and then it prints a docker login command with that authorization token. You can execute the printed command to log in to your registry with Docker, or just run it automatically using the $() command wrapper. After you have logged in to an Amazon ECR registry with this command, you can use the Docker CLI to push and pull images from that registry until the token expires. The docker pull command is used to pull an image from the ECR registry.

Incorrect options:

**docker login -u $AWS\_ACCESS\_KEY\_ID -p $AWS\_SECRET\_ACCESS\_KEY** - You cannot login to AWS ECR this way. AWS\_ACCESS\_KEY\_ID and AWS\_SECRET\_ACCESS\_KEY are only used by the CLI and not by docker.

**aws docker push 1234567890.dkr.ecr.eu-west-1.amazonaws.com/demo:latest** - docker push here is the wrong answer, you need to use docker pull.

**docker build -t 1234567890.dkr.ecr.eu-west-1.amazonaws.com/demo:latest** - This is a docker command that is used to build Docker images from a Dockerfile.

Reference:

<https://docs.aws.amazon.com/cli/latest/reference/ecr/get-login.html>

Bottom of Form

Top of Form

Question 25: Skipped

**As an AWS certified developer associate, you are working on an AWS CloudFormation template that will create resources for a company's cloud infrastructure. Your template is composed of three stacks which are Stack-A, Stack-B, and Stack-C. Stack-A will provision a VPC, a security group, and subnets for public web applications that will be referenced in Stack-B and Stack-C.**

**After running the stacks you decide to delete them, in which order should you do it?**

* ​

Stack A, then Stack B, then Stack C

* ​

Stack B, then Stack C, then Stack A

**(Correct)**

* ​

Stack C then Stack A then Stack B

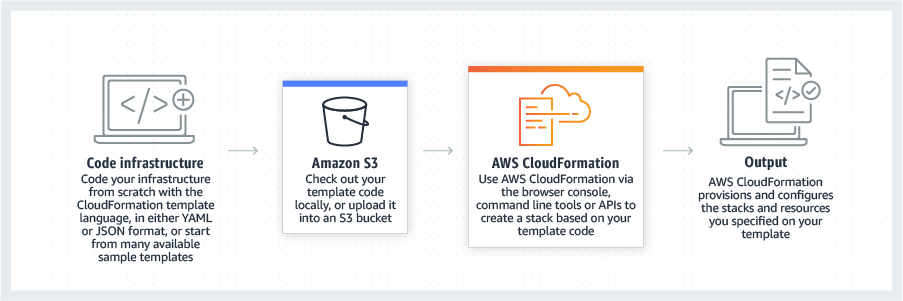
* ​

Stack A, Stack C then Stack B

**Explanation**

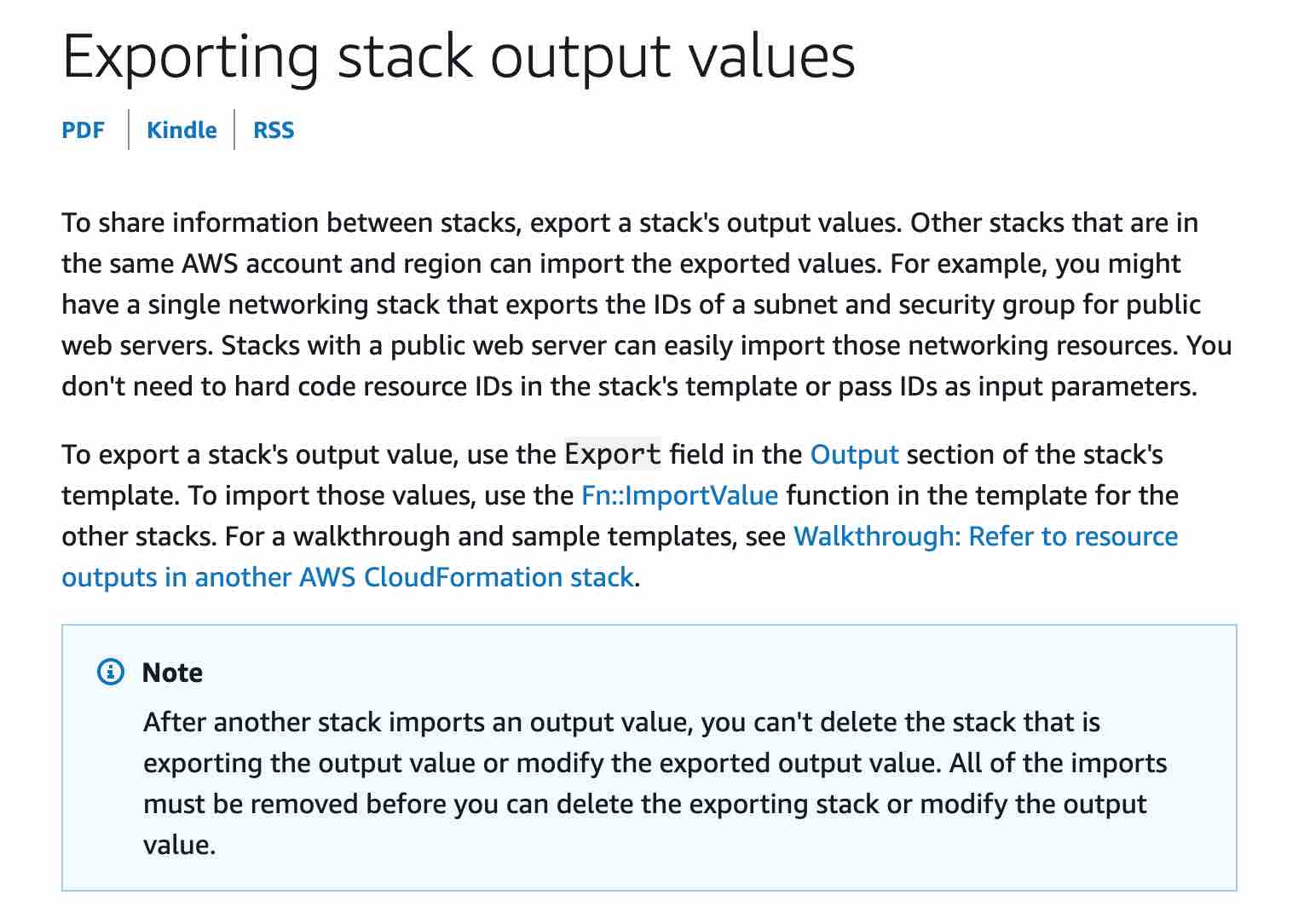
Correct option:

AWS CloudFormation gives developers and businesses an easy way to create a collection of related AWS and third-party resources and provision them in an orderly and predictable fashion.

How CloudFormation Works:  via - <https://aws.amazon.com/cloudformation/>

**Stack B, then Stack C, then Stack A**

All of the imports must be removed before you can delete the exporting stack or modify the output value. In this case, you must delete either Stack B or Stack C, then you delete Stack A.

 via - <https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/using-cfn-stack-exports.html>

Incorrect options:

**Stack A, then Stack B, then Stack C** - All of the imports must be removed before you can delete the exporting stack or modify the output value. In this case, you cannot delete Stack A first because that's being referenced in the other Stacks.

**Stack A, Stack C then Stack B** - All of the imports must be removed before you can delete the exporting stack or modify the output value. In this case, you cannot delete Stack A first because that's being referenced in the other Stacks.

**Stack C then Stack A then Stack B** - Stack C is fine but you should delete Stack B before Stack A because all of the imports must be removed before you can delete the exporting stack or modify the output value.

For more information visit

Bottom of Form

Top of Form

Question 26: Skipped

**A team lead has asked you to create an AWS CloudFormation template that creates EC2 instances and RDS databases. The template should be reusable by allowing the user to input a parameter value for an Amazon EC2 AMI ID.**

**Which of the following intrinsic function should you choose to reference the parameter?**

* ​

!Param

* ​

!Ref

**(Correct)**

* ​

!Join

* ​

!GetAtt

**Explanation**

Correct option:

**!Ref**

The intrinsic function Ref returns the value of the specified parameter or resource. When you specify a parameter's logical name, it returns the value of the parameter, when you specify a resource's logical name, it returns a value that you can typically use to refer to that resource such as a physical ID. Take a look at this YAML sample template:

MyEIP:

Type: "AWS::EC2::EIP"

Properties:

InstanceId: !Ref MyEC2Instance

Incorrect options:

**!GetAtt** - This function returns the value of an attribute from a resource in the template. The YAML syntax is like so:

!GetAtt logicalNameOfResource.attributeName

**!Param** - This is not a valid function name. This option has been added as a distractor.

**!Join** - This function appends a set of values into a single value, separated by the specified delimiter. The YAML syntax is like so:

!Join [ delimiter, [ comma-delimited list of values ] ]

Reference:

<https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/intrinsic-function-reference.html>

Bottom of Form

Top of Form

Question 27: Skipped

**As an AWS Certified Developer Associate, you have been hired to consult with a company that uses the NoSQL database for mobile applications. The developers are using DynamoDB to perform operations such as GetItem but are limited in knowledge. They would like to be more efficient with retrieving some attributes rather than all.**

**Which of the following recommendations would you provide?**

* ​

Use a Scan

* ​

Use a FilterExpression

* ​

Use the --query parameter

* ​

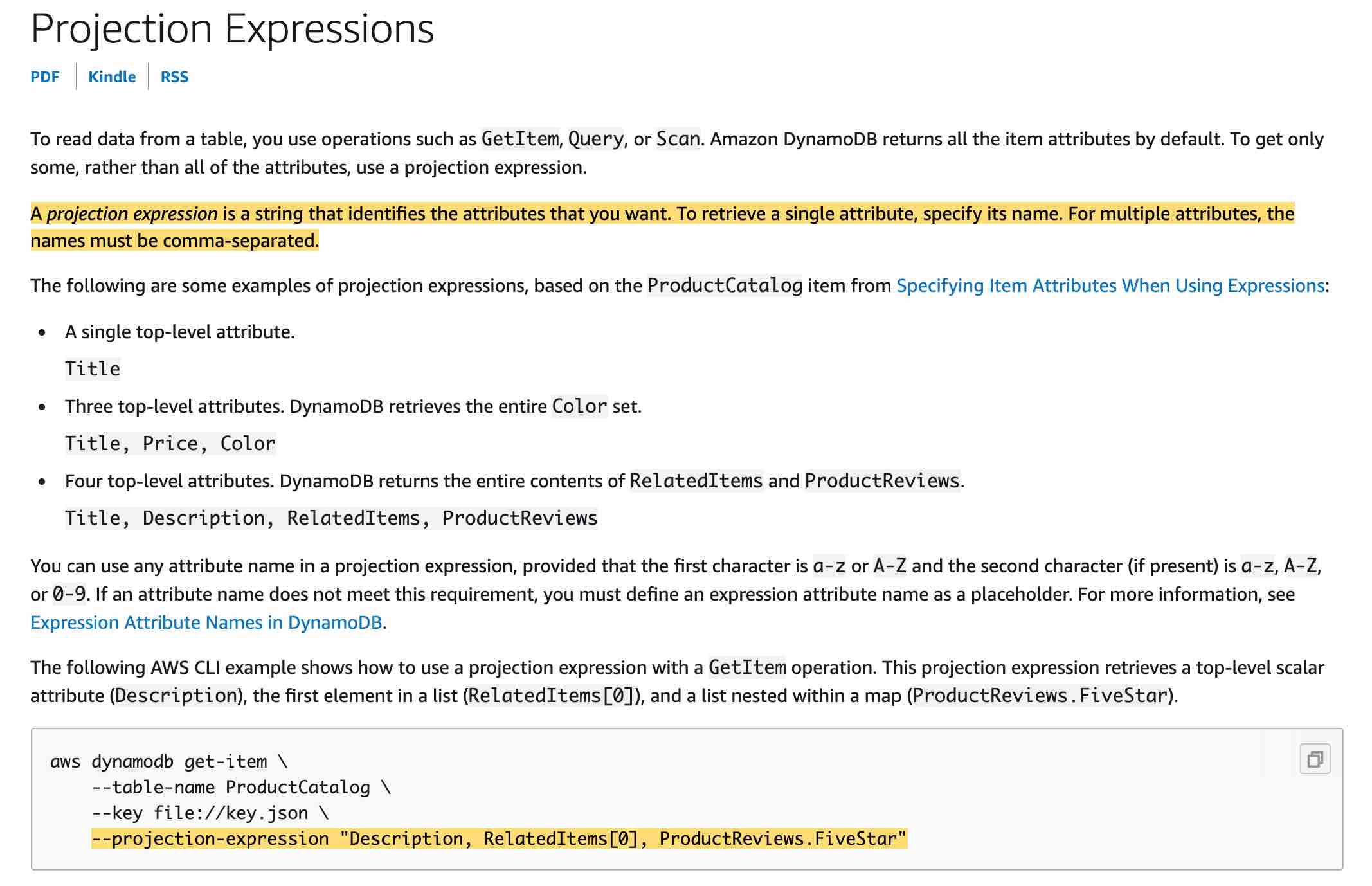
Specify a ProjectionExpression

**(Correct)**

**Explanation**

Correct option:

**Specify a ProjectionExpression**: A projection expression is a string that identifies the attributes you want. To retrieve a single attribute, specify its name. For multiple attributes, the names must be comma-separated.

 via - <https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/Expressions.ProjectionExpressions.html>

Incorrect options:

**Use a FilterExpression** - If you need to further refine the Query results, you can optionally provide a filter expression. A filter expression determines which items within the Query results should be returned to you. All of the other results are discarded. A filter expression is applied after Query finishes, but before the results are returned. Therefore, a Query consumes the same amount of read capacity, regardless of whether a filter expression is present. A Query operation can retrieve a maximum of 1 MB of data. This limit applies before the filter expression is evaluated.

**Use the --query parameter** - The Query operation in Amazon DynamoDB finds items based on primary key values. You must provide the name of the partition key attribute and a single value for that attribute. The Query returns all items with that partition key value. Optionally, you can provide a sort key attribute and use a comparison operator to refine the search results.

**Use a Scan** - A Scan operation in Amazon DynamoDB reads every item in a table or a secondary index. By default, a Scan operation returns all of the data attributes for every item in the table or index. You can also use the ProjectionExpression parameter so that Scan only returns some of the attributes, rather than all of them.

Reference:

<https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/Expressions.ProjectionExpressions.html>

Bottom of Form

Top of Form

Question 28: Skipped

**An IT company uses AWS CloudFormation templates to provision their AWS infrastructure for Amazon EC2, Amazon VPC, and Amazon S3 resources. Using cross-stack referencing, a developer creates a stack called NetworkStack which will export the subnetId that can be used when creating EC2 instances in another stack.**

**To use the exported value in another stack, which of the following functions must be used?**

* ​

!GetAtt

* ​

!Sub

* ​

!Ref

* ​

!ImportValue

**(Correct)**

**Explanation**

Correct option:

**!ImportValue**

The intrinsic function Fn::ImportValue returns the value of an output exported by another stack. You typically use this function to create cross-stack references.

Incorrect options:

**!Ref** - Returns the value of the specified parameter or resource.

**!GetAtt** - Returns the value of an attribute from a resource in the template.

**!Sub** - Substitutes variables in an input string with values that you specify.

Reference:

<https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/intrinsic-function-reference-importvalue.html>

Bottom of Form

Top of Form

Question 29: Skipped

**A firm runs its technology operations on a fleet of Amazon EC2 instances. The firm needs a certain software to be available on the instances to support their daily workflows. The developer team has been told to use the user data feature of EC2 instances.**

**Which of the following are true about the user data EC2 configuration? ( Select two)**

* ​

By default, scripts entered as user data do not have root user privileges for executing

* ​

By default, scripts entered as user data are executed with root user privileges

**(Correct)**

* ​

When an instance is running, you can update user data by using root user credentials

* ​

By default, user data is executed every time an EC2 instance is re-started

* ​

By default, user data runs only during the boot cycle when you first launch an instance

**(Correct)**

**Explanation**

Correct options:

User Data is generally used to perform common automated configuration tasks and even run scripts after the instance starts. When you launch an instance in Amazon EC2, you can pass two types of user data - shell scripts and cloud-init directives. You can also pass this data into the launch wizard as plain text or as a file.

**By default, scripts entered as user data are executed with root user privileges** - Scripts entered as user data are executed as the root user, hence do not need the sudo command in the script. Any files you create will be owned by root; if you need non-root users to have file access, you should modify the permissions accordingly in the script.

**By default, user data runs only during the boot cycle when you first launch an instance** - By default, user data scripts and cloud-init directives run only during the boot cycle when you first launch an instance. You can update your configuration to ensure that your user data scripts and cloud-init directives run every time you restart your instance.

Incorrect options:

**By default, user data is executed every time an EC2 instance is re-started** - As discussed above, this is not a default configuration of the system. But, can be achieved by explicitly configuring the instance.

**When an instance is running, you can update user data by using root user credentials** - You can't change the user data if the instance is running (even by using root user credentials), but you can view it.

**By default, scripts entered as user data do not have root user privileges for executing** - Scripts entered as user data are executed as the root user, hence do not need the sudo command in the script.

Reference:

<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/user-data.html>

Bottom of Form

Top of Form

Question 30: Skipped

**DevOps engineers are developing an order processing system where notifications are sent to a department whenever an order is placed for a product. The system also pushes identical notifications of the new order to a processing module that would allow EC2 instances to handle the fulfillment of the order. In the case of processing errors, the messages should be allowed to be re-processed at a later stage and never lost.**

**Which of the following solutions can be used to address this use-case?**

* ​

SNS + SQS

**(Correct)**

* ​

SNS + Kinesis

* ​

SNS + Lambda

* ​

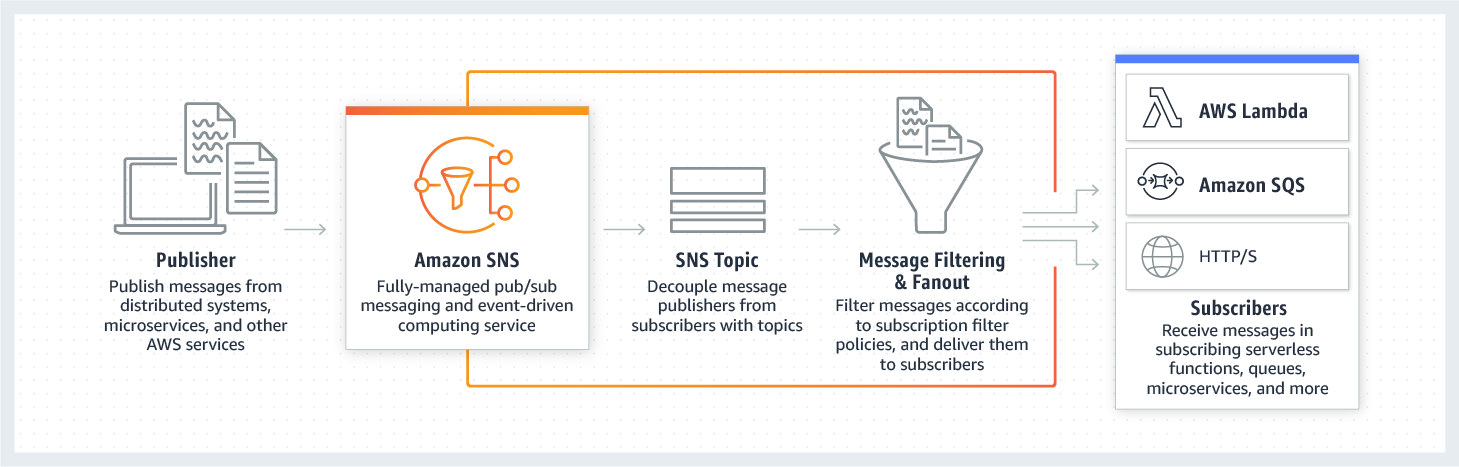
SQS + SES

**Explanation**

Correct option:

**SNS + SQS**

Amazon SNS enables message filtering and fanout to a large number of subscribers, including serverless functions, queues, and distributed systems. Additionally, Amazon SNS fans out notifications to end users via mobile push messages, SMS, and email.

How SNS Works:  via - <https://aws.amazon.com/sns/>

Amazon Simple Queue Service (SQS) is a fully managed message queuing service that enables you to decouple and scale microservices, distributed systems, and serverless applications. SQS offers two types of message queues. Standard queues offer maximum throughput, best-effort ordering, and at-least-once delivery. SQS FIFO queues are designed to guarantee that messages are processed exactly once, in the exact order that they are sent.

SNS and SQS can be used to create a fanout messaging scenario in which messages are "pushed" to multiple subscribers, which eliminates the need to periodically check or poll for updates and enables parallel asynchronous processing of the message by the subscribers. SQS can allow for later re-processing and dead letter queues. This is called the fan-out pattern.

Incorrect options:

**SNS + Kinesis** - You can use Amazon Kinesis Data Streams to collect and process large streams of data records in real-time. The only issue is that the retention day period is 7 days, and processing issues would completely block all future messages.

**SNS + Lambda** - Amazon SNS and AWS Lambda are integrated so you can invoke Lambda functions with Amazon SNS notifications. The Lambda function receives the message payload as an input parameter and can manipulate the information in the message, publish the message to other SNS topics, or send the message to other AWS services. However, your EC2 instances cannot "poll" from Lambda functions and as such, this would not work.

**SQS + SES** - This will not work as the messages need to be processed twice (once for sending the notification and later for order fulfillment) and SQS only allows for one consuming application.

References:

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

<https://aws.amazon.com/getting-started/tutorials/send-fanout-event-notifications/>

Bottom of Form

Top of Form

Question 31: Skipped

**You are responsible for an application that runs on multiple Amazon EC2 instances. In front of the instances is an Internet-facing load balancer that takes requests from clients over the internet and distributes them to the EC2 instances. A health check is configured to ping the index.html page found in the root directory for the health status. When accessing the website via the internet visitors of the website receive timeout errors.**

**What should be checked first to resolve the issue?**

* ​

The application is down

* ​

Security Groups

**(Correct)**

* ​

IAM Roles

* ​

The ALB is warming up

**Explanation**

Correct option:

**Security Groups**

A security group acts as a virtual firewall for your EC2 instances to control incoming and outgoing traffic. Inbound rules control the incoming traffic to your instance, and outbound rules control the outgoing traffic from your instance.

Check the security group rules of your EC2 instance. You need a security group rule that allows inbound traffic from your public IPv4 address on the proper port.

Incorrect options:

**IAM Roles** - Usually you run into issues with authorization of APIs with roles but not for timeout, so this option does not fit the given use-case.

**The application is down** - Although you can set a health check for application ping or HTTP, timeouts are usually caused by blocked firewall access.

**The ALB is warming up** - ALB has a slow start mode which allows a warm-up period before being able to respond to requests with optimal performance. So this is not the issue.

Reference:

<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/TroubleshootingInstancesConnecting.html#TroubleshootingInstancesConnectionTimeout>

Bottom of Form

Top of Form

Question 32: Skipped

**A new recruit is trying to understand the nuances of EC2 Auto Scaling. As an AWS Certified Developer Associate, you have been asked to mentor the new recruit.**

**Can you identify and explain the correct statements about Auto Scaling to the new recruit? (Select two).**

* ​

Amazon EC2 Auto Scaling cannot add a volume to an existing instance if the existing volume is approaching capacity

**(Correct)**

* ​

Amazon EC2 Auto Scaling works with both Application Load Balancers and Network Load Balancers

**(Correct)**

* ​

Every time you create an Auto Scaling group from an existing instance, it creates a new AMI (Amazon Machine Image)

* ​

You cannot use Amazon EC2 Auto Scaling for health checks (to replace unhealthy instances) if you are not using Elastic Load Balancing (ELB)

* ​

EC2 Auto Scaling groups are regional constructs. They span across Availability Zones and AWS regions

**Explanation**

Correct options:

Amazon EC2 Auto Scaling is a fully managed service designed to launch or terminate Amazon EC2 instances automatically to help ensure you have the correct number of Amazon EC2 instances available to handle the load for your application.

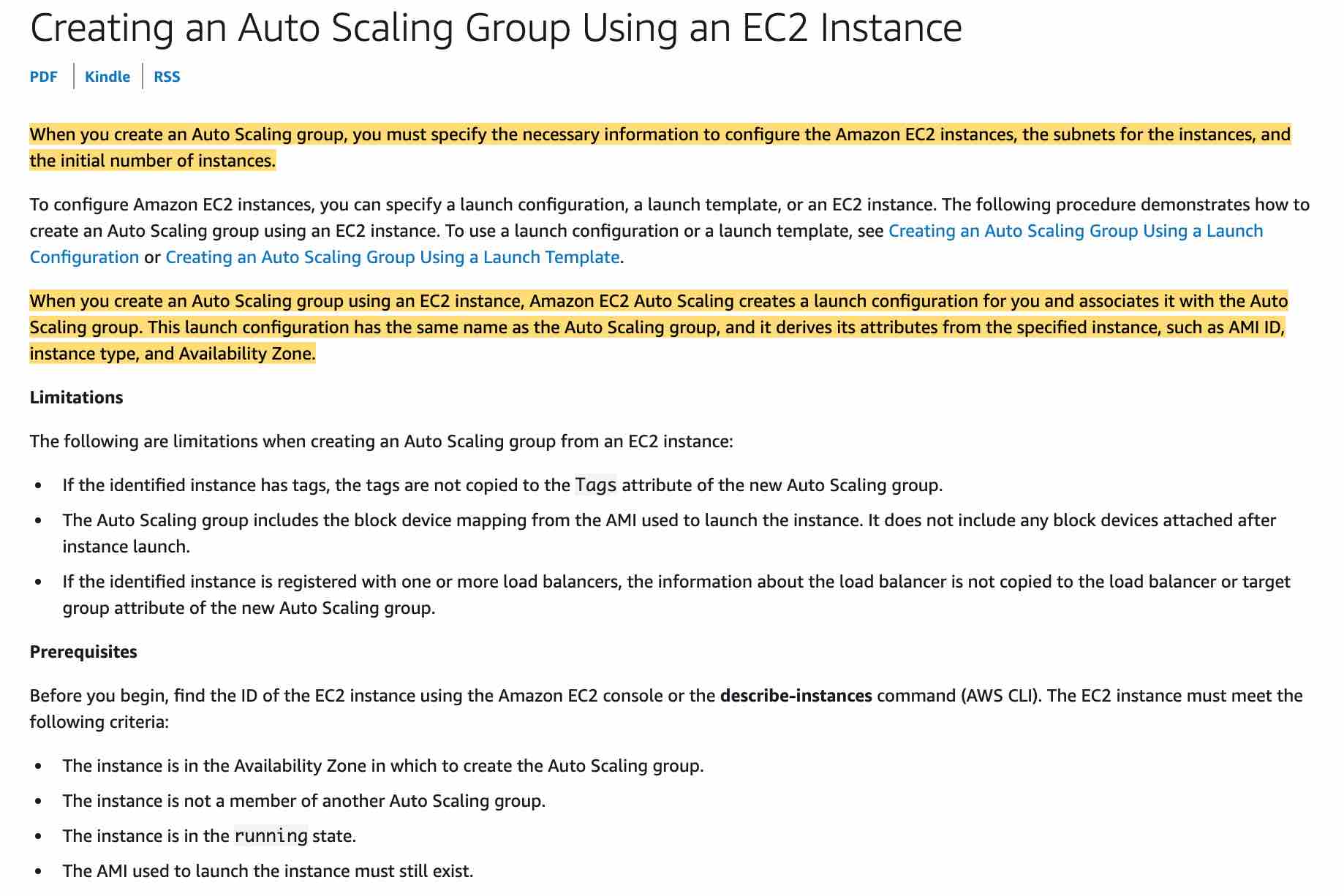
**Amazon EC2 Auto Scaling cannot add a volume to an existing instance if the existing volume is approaching capacity** - A volume is attached to a new instance when it is added. Amazon EC2 Auto Scaling doesn't automatically add a volume when the existing one is approaching capacity. You can use the EC2 API to add a volume to an existing instance.

**Amazon EC2 Auto Scaling works with both Application Load Balancers and Network Load Balancers** - Amazon EC2 Auto Scaling works with Application Load Balancers and Network Load Balancers including their health check feature.

Incorrect options:

**EC2 Auto Scaling groups are regional constructs. They span across Availability Zones and AWS regions** - This is an incorrect statement. EC2 Auto Scaling groups are regional constructs. They can span Availability Zones, but not AWS regions.

**Every time you create an Auto Scaling group from an existing instance, it creates a new AMI (Amazon Machine Image)** - This is an incorrect statement. When you create an Auto Scaling group from an existing instance, it does not create a new AMI.

 via - <https://docs.aws.amazon.com/autoscaling/ec2/userguide/create-asg-from-instance.html>

**You cannot use Amazon EC2 Auto Scaling for health checks (to replace unhealthy instances) if you are not using Elastic Load Balancing (ELB)** - This is an incorrect statement. You don't have to use ELB to use Auto Scaling. You can use the EC2 health check to identify and replace unhealthy instances.

References:

<https://aws.amazon.com/ec2/autoscaling/faqs/>

<https://docs.aws.amazon.com/autoscaling/ec2/userguide/create-asg-from-instance.html>

Bottom of Form

Top of Form

Question 33: Skipped

**A data analytics company wants to use clickstream data for Machine Learning tasks, develop algorithms, and create visualizations and dashboards to support the business stakeholders. Each of these business units works independently and would need real-time access to this clickstream data for their applications.**

**As a Developer Associate, which of the following AWS services would you recommend such that it provides a highly available and fault-tolerant solution to capture the clickstream events from the source and then provide a simultaneous feed of the data stream to the consumer applications?**

* ​

Amazon SQS

* ​

AWS Kinesis Data Streams

**(Correct)**

* ​

AWS Kinesis Data Firehose

* ​

AWS Kinesis Data Analytics

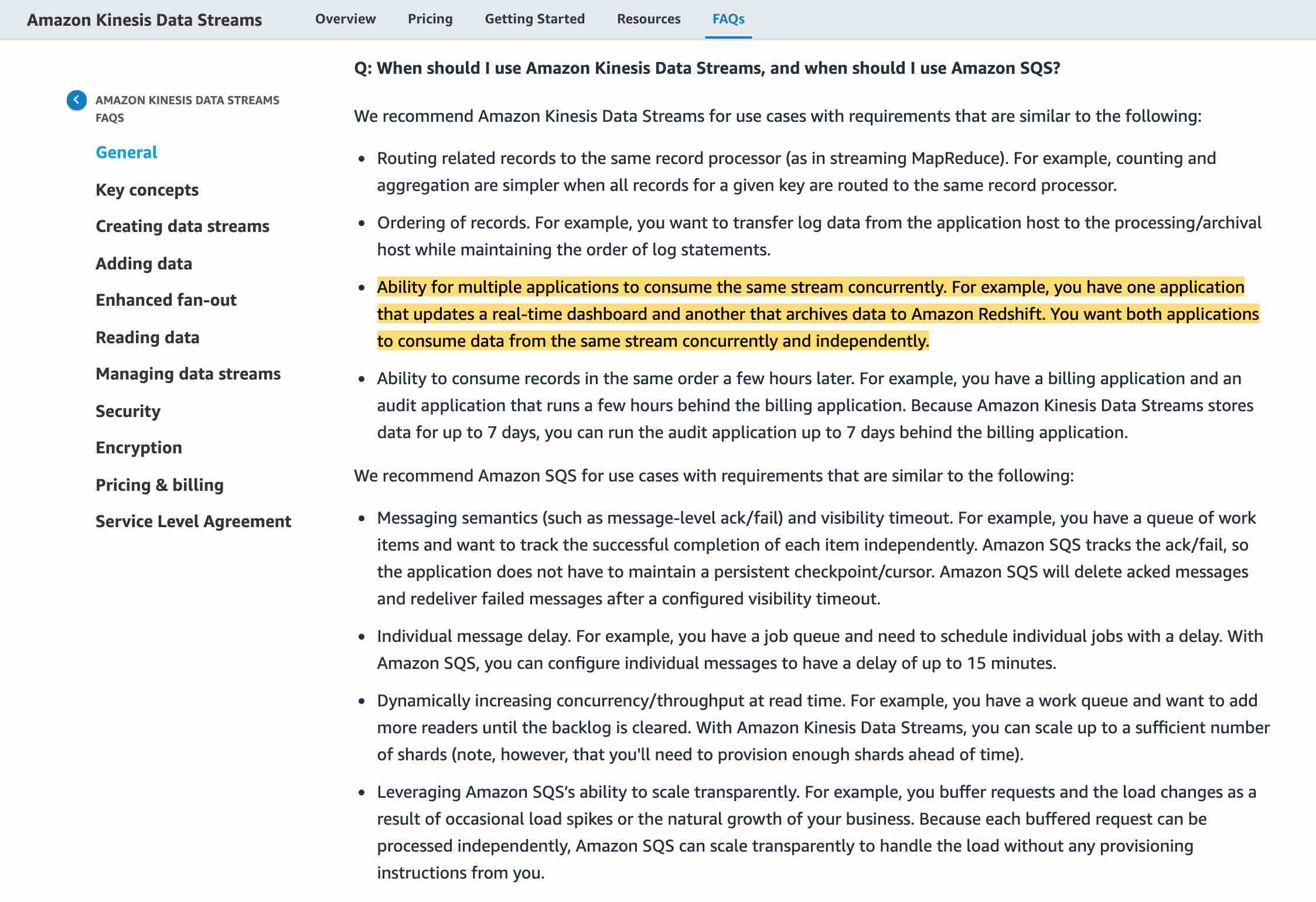
**Explanation**

Correct option:

**AWS 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. The data collected is available in milliseconds to enable real-time analytics use cases such as real-time dashboards, real-time anomaly detection, dynamic pricing, and more.

Amazon Kinesis Data Streams enables real-time processing of streaming big data. It provides ordering of records, as well as the ability to read and/or replay records in the same order to multiple Amazon Kinesis Applications. The Amazon Kinesis Client Library (KCL) delivers all records for a given partition key to the same record processor, making it easier to build multiple applications reading from the same Amazon Kinesis data stream (for example, to perform counting, aggregation, and filtering). Amazon Kinesis Data Streams is recommended when you need the ability for multiple applications to consume the same stream concurrently. For example, you have one application that updates a real-time dashboard and another application that archives data to Amazon Redshift. You want both applications to consume data from the same stream concurrently and independently.

KDS provides the ability for multiple applications to consume the same stream concurrently  via - <https://aws.amazon.com/kinesis/data-streams/faqs/>

Incorrect options:

**AWS Kinesis Data Firehose** - Amazon Kinesis Data Firehose is the easiest way to load streaming data into data stores and analytics tools. It can capture, transform, and load streaming data into Amazon S3, Amazon Redshift, Amazon Elasticsearch Service, and Splunk, enabling near real-time analytics with existing business intelligence tools and dashboards you’re already using today. It is a fully managed service that automatically scales to match the throughput of your data and requires no ongoing administration. It can also batch, compress, and encrypt the data before loading it, minimizing the amount of storage used at the destination and increasing security. As Kinesis Data Firehose is used to load streaming data into data stores, therefore this option is incorrect.

**AWS Kinesis Data Analytics** - Amazon Kinesis Data Analytics is the easiest way to analyze streaming data in real-time. You can quickly build SQL queries and sophisticated Java applications using built-in templates and operators for common processing functions to organize, transform, aggregate, and analyze data at any scale. Kinesis Data Analytics enables you to easily and quickly build queries and sophisticated streaming applications in three simple steps: setup your streaming data sources, write your queries or streaming applications and set up your destination for processed data. As Kinesis Data Analytics is used to build SQL queries and sophisticated Java applications, therefore this option is incorrect.

**Amazon SQS** - Amazon Simple Queue Service (SQS) is a fully managed message queuing service that enables you to decouple and scale microservices, distributed systems, and serverless applications. SQS offers two types of message queues. Standard queues offer maximum throughput, best-effort ordering, and at-least-once delivery. SQS FIFO queues are designed to guarantee that messages are processed exactly once, in the exact order that they are sent. For SQS, you cannot have the same message being consumed by multiple consumers at the same time, therefore this option is incorrect.

Exam alert:

Please remember that Kinesis Data Firehose is used to load streaming data into data stores (Amazon S3, Amazon Redshift, Amazon Elasticsearch Service, and Splunk) whereas Kinesis Data Streams provides support for real-time processing of streaming data. It provides ordering of records, as well as the ability to read and/or replay records in the same order to multiple downstream Amazon Kinesis Applications.

References:

<https://aws.amazon.com/kinesis/data-streams/faqs/>

<https://aws.amazon.com/kinesis/data-firehose/faqs/>

<https://aws.amazon.com/kinesis/data-analytics/faqs/>

Bottom of Form

Top of Form

Question 34: Skipped

**A startup manages its Cloud resources with Elastic Beanstalk. The environment consists of few Amazon EC2 instances, an Auto Scaling Group (ASG), and an Elastic Load Balancer. Even after the Load Balancer marked an EC2 instance as unhealthy, the ASG has not replaced it with a healthy instance.**

**As a Developer, suggest the necessary configurations to automate the replacement of unhealthy instance.**

* ​

The health check type of your instance's Auto Scaling group, must be changed from EC2 to ELB by using a configuration file

**(Correct)**

* ​

Auto Scaling group doesn't automatically replace the unhealthy instances marked by the load balancer. They have to be manually replaced from AWS Console

* ​

The ping path field of the Load Balancer is configured incorrectly

* ​

Health check parameters were configured for checking the instance health alone. The instance failed because of application failure which was not configured as a parameter for health check status

**Explanation**

Correct option:

**The health check type of your instance's Auto Scaling group, must be changed from EC2 to ELB by using a configuration file** - By default, the health check configuration of your Auto Scaling group is set as an EC2 type that performs a status check of EC2 instances. To automate the replacement of unhealthy EC2 instances, you must change the health check type of your instance's Auto Scaling group from EC2 to ELB by using a configuration file.

Incorrect options:

**Health check parameters were configured for checking the instance health alone. The instance failed because of application failure which was not configured as a parameter for health check status** - This is an incorrect statement. Status checks, by definition, cover only an EC2 instance's health, and not the health of your application, server, or any Docker containers running on the instance.

**Auto Scaling group doesn't automatically replace the unhealthy instances marked by the load balancer. They have to be manually replaced from AWS Console** - Incorrect statement. As discussed above, if the health check type of ASG is changed from EC2 to ELB, Auto Scaling will be able to replace the unhealthy instance.

**The ping path field of the Load Balancer is configured incorrectly** - Ping path is a health check configuration field of Elastic Load Balancer. If the ping path is configured wrong, ELB will not be able to reach the instance and hence will consider the instance unhealthy. However, this would then apply to all instances, not just once instance. So it does not address the issue given in the use-case.

References:

<https://aws.amazon.com/premiumsupport/knowledge-center/elastic-beanstalk-instance-automation/>

<https://docs.aws.amazon.com/elasticloadbalancing/latest/classic/elb-healthchecks.html>

Bottom of Form

Top of Form

Question 35: Skipped

**AWS CloudFormation helps model and provision all the cloud infrastructure resources needed for your business.**

**Which of the following services rely on CloudFormation to provision resources (Select two)?**

* ​

AWS Lambda

* ​

AWS Serverless Application Model (AWS SAM)

**(Correct)**

* ​

AWS CodeBuild

* ​

AWS Autoscaling

* ​

AWS Elastic Beanstalk

**(Correct)**

**Explanation**

Correct option:

**AWS Elastic Beanstalk** - AWS Elastic Beanstalk is an easy-to-use service for deploying and scaling web applications and services developed with Java, .NET, PHP, Node.js, Python, Ruby, Go, and Docker on familiar servers such as Apache, Nginx, Passenger, and IIS. Elastic Beanstalk uses AWS CloudFormation to launch the resources in your environment and propagate configuration changes.

**AWS Serverless Application Model (AWS SAM)** - You use the AWS SAM specification to define your serverless application. AWS SAM templates are an extension of AWS CloudFormation templates, with some additional components that make them easier to work with. AWS SAM needs CloudFormation templates as a basis for its configuration.

Incorrect options:

**AWS Lambda** - AWS Lambda lets you run code without provisioning or managing servers. You pay only for the compute time you consume. Hence, Lamda does not need CloudFormation to run its services.

**AWS Autoscaling** - AWS Auto Scaling monitors your applications and automatically adjusts the capacity to maintain steady, predictable performance at the lowest possible cost. Using AWS Auto Scaling, it’s easy to setup application scaling for multiple resources across multiple services in minutes. Auto Scaling used CloudFormation but is not a mandatory requirement.

**AWS CodeBuild** - AWS CodeBuild is a fully managed continuous integration service that compiles source code, runs tests, and produces software packages that are ready to deploy. With CodeBuild, you don’t need to provision, manage, and scale your own build servers. AWS CodePipeline uses AWS CloudFormation as a deployment action but is not a mandatory service.

References:

<https://docs.aws.amazon.com/serverless-application-model/latest/developerguide/sam-specification.html>

<https://aws.amazon.com/elasticbeanstalk/>

Bottom of Form

Top of Form

Question 36: Skipped

**The development team at an e-commerce company is preparing for the upcoming Thanksgiving sale. The product manager wants the development team to implement appropriate caching strategy on Amazon ElastiCache to withstand traffic spikes on the website during the sale. A key requirement is to facilitate consistent updates to the product prices and product description, so that the cache never goes out of sync with the backend.**

**As a Developer Associate, which of the following solutions would you recommend for the given use-case?**

* ​

Use a caching strategy to write to the backend first and then invalidate the cache

**(Correct)**

* ​

Use a caching strategy to write to the cache directly and sync the backend at a later time

* ​

Use a caching strategy to write to the backend first and wait for the cache to expire via TTL

* ​

Use a caching strategy to update the cache and the backend at the same time

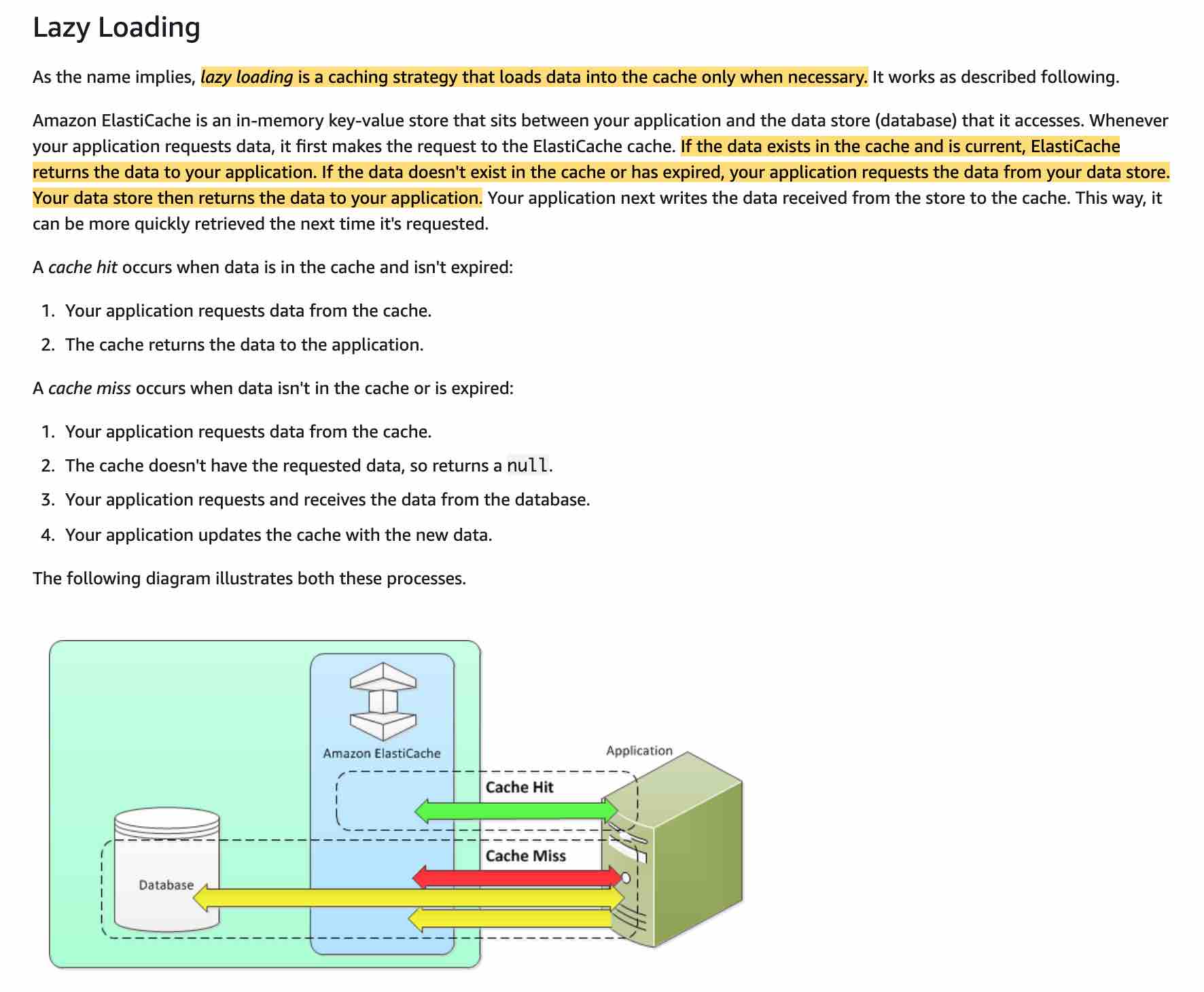
**Explanation**

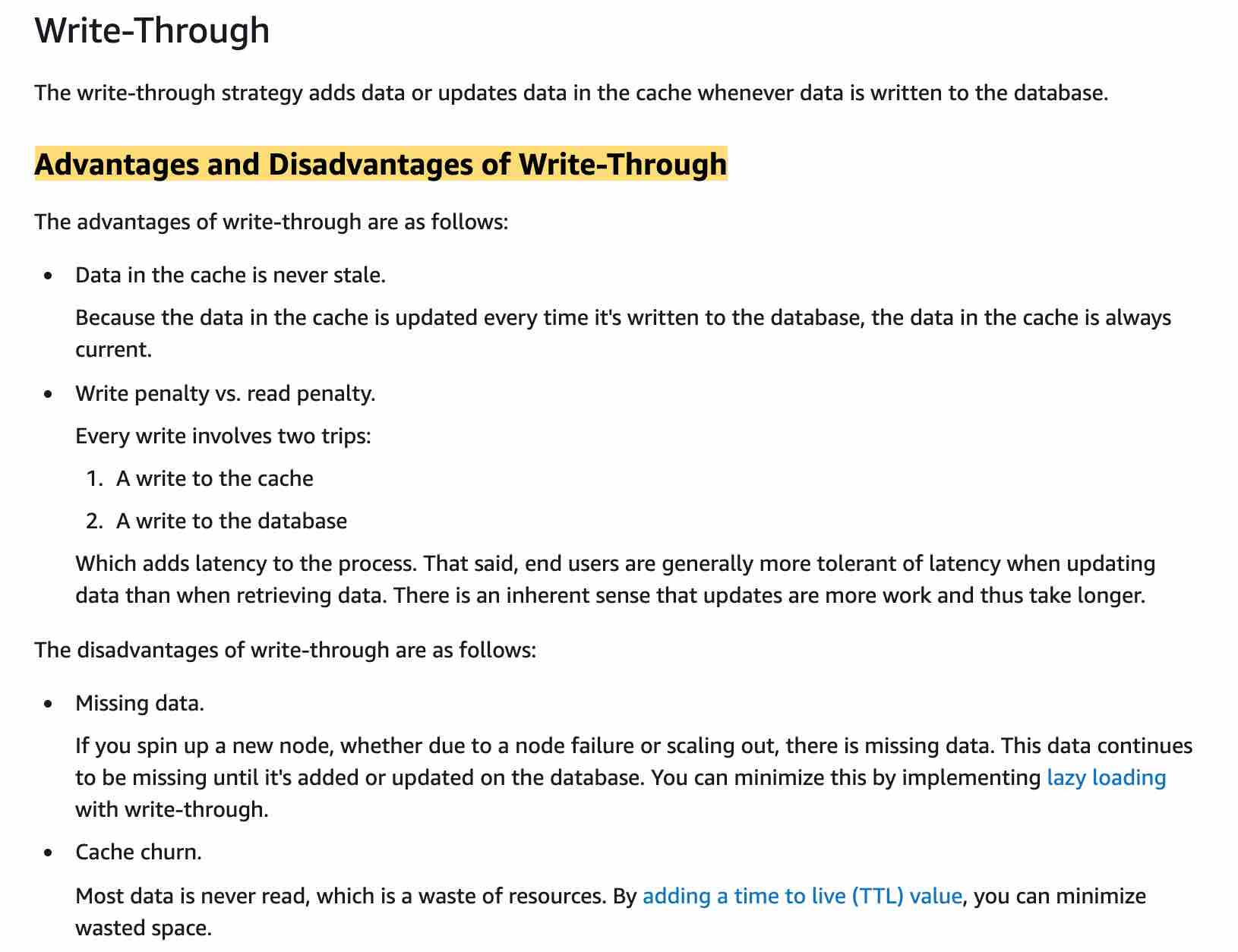
Correct option:

Amazon ElastiCache allows you to seamlessly set up, run, and scale popular open-Source compatible in-memory data stores in the cloud. Build data-intensive apps or boost the performance of your existing databases by retrieving data from high throughput and low latency in-memory data stores. Amazon ElastiCache is a popular choice for real-time use cases like Caching, Session Stores, Gaming, Geospatial Services, Real-Time Analytics, and Queuing.

Broadly, you can set up two types of caching strategies:

1. Lazy Loading
2. Write-Through

 via - <https://docs.aws.amazon.com/AmazonElastiCache/latest/mem-ug/Strategies.html>

 via - <https://docs.aws.amazon.com/AmazonElastiCache/latest/mem-ug/Strategies.html>

**Use a caching strategy to write to the backend first and then invalidate the cache**

This option is similar to the write-through strategy wherein the application writes to the backend first and then invalidate the cache. As the cache gets invalidated, the caching engine would then fetch the latest value from the backend, thereby making sure that the product prices and product description stay consistent with the backend.

Incorrect options:

**Use a caching strategy to update the cache and the backend at the same time** - The cache and the backend cannot be updated at the same time via a single atomic operation as these are two separate systems. Therefore this option is incorrect.

**Use a caching strategy to write to the backend first and wait for the cache to expire via TTL** - This strategy could work if the TTL is really short. However, for the duration of this TTL, the cache would be out of sync with the backend, hence this option is not correct for the given use-case.

**Use a caching strategy to write to the cache directly and sync the backend at a later time** - This option is given as a distractor as this strategy is not viable to address the given use-case. The product prices and description on the cache must always stay consistent with the backend. You cannot sync the backend at a later time.

Reference:

<https://docs.aws.amazon.com/AmazonElastiCache/latest/mem-ug/Strategies.html>

Bottom of Form

Top of Form

Question 37: Skipped

**A video encoding application running on an EC2 instance takes about 20 seconds on average to process each raw footage file. The application picks the new job messages from an SQS queue. The development team needs to account for the use-case when the video encoding process takes longer than usual so that the same raw footage is not processed by multiple consumers.**

**As a Developer Associate, which of the following solutions would you recommend to address this use-case?**

* ​

Use ChangeMessageVisibility action to extend a message's visibility timeout

**(Correct)**

* ​

Use WaitTimeSeconds action to short poll and extend a message's visibility timeout

* ​

Use DelaySeconds action to delay a message's visibility timeout

* ​

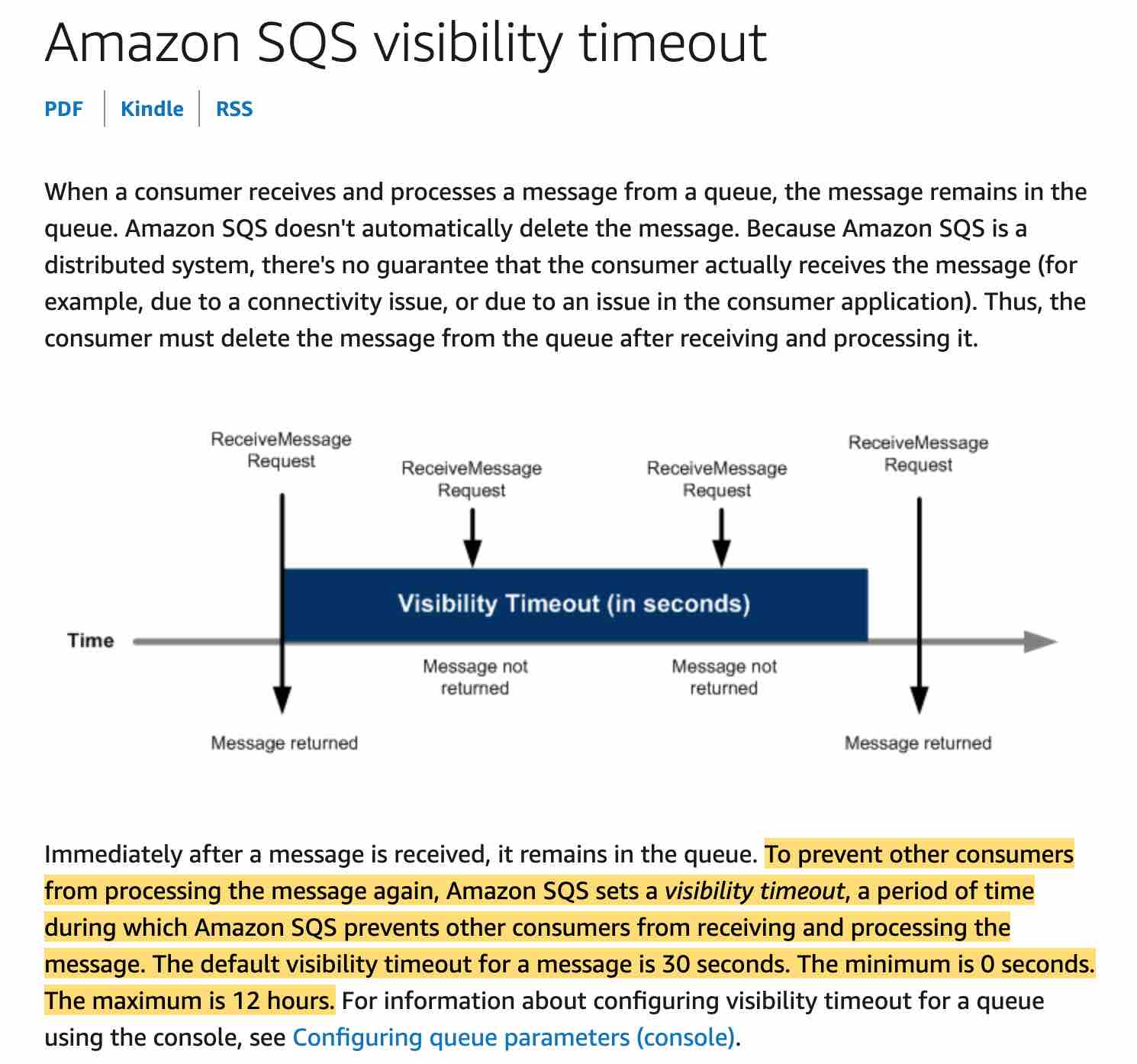
Use WaitTimeSeconds action to long poll and extend a message's visibility timeout

**Explanation**

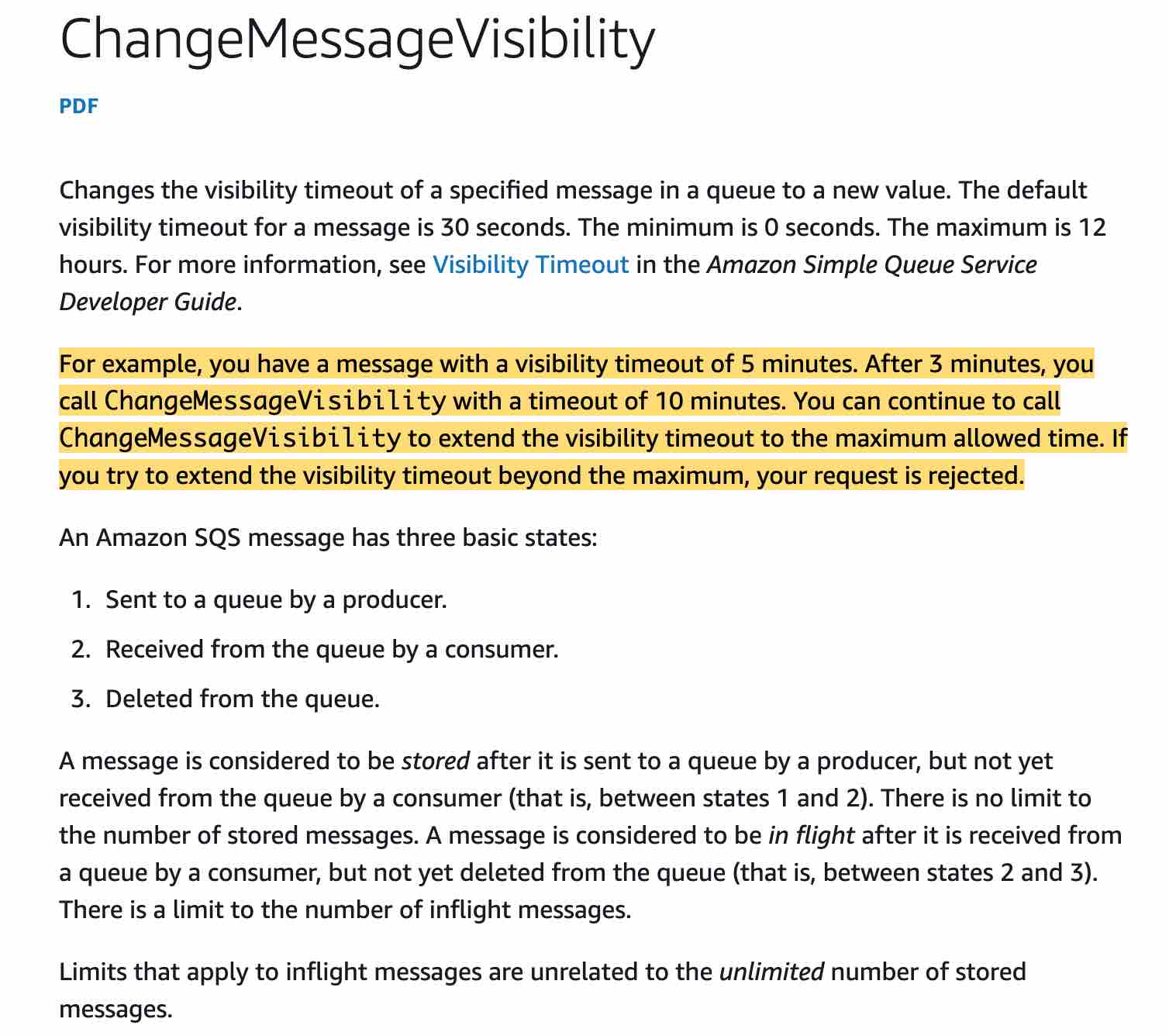
Correct option:

**Use ChangeMessageVisibility action to extend a message's visibility timeout**

Amazon SQS uses a visibility timeout to prevent other consumers from receiving and processing the same message. The default visibility timeout for a message is 30 seconds. The minimum is 0 seconds. The maximum is 12 hours.

 via - <https://docs.aws.amazon.com/AWSSimpleQueueService/latest/SQSDeveloperGuide/sqs-visibility-timeout.html>

For example, you have a message with a visibility timeout of 5 minutes. After 3 minutes, you call ChangeMessageVisibility with a timeout of 10 minutes. You can continue to call ChangeMessageVisibility to extend the visibility timeout to the maximum allowed time. If you try to extend the visibility timeout beyond the maximum, your request is rejected. So, for the given use-case, the application can set the initial visibility timeout to 1 minute and then continue to update the ChangeMessageVisibility value if required.

 via - <https://docs.aws.amazon.com/AWSSimpleQueueService/latest/SQSDeveloperGuide/sqs-visibility-timeout.html>

Incorrect options:

**Use DelaySeconds action to delay a message's visibility timeout** - Delay queues let you postpone the delivery of new messages to a queue for a number of seconds. To set delay seconds on individual messages, rather than on an entire queue, use message timers to allow Amazon SQS to use the message timer's DelaySeconds value instead of the delay queue's DelaySeconds value. You cannot use DelaySeconds to alter the visibility of a message which has been picked for processing.

**Use WaitTimeSeconds action to short poll and extend a message's visibility timeout**

**Use WaitTimeSeconds action to long poll and extend a message's visibility timeout**

Amazon SQS provides short polling and long polling to receive messages from a queue. Both these options have been added as distractors as WaitTimeSeconds (via short polling or long polling) cannot be used to influence the message's visibility.

References:

<https://docs.aws.amazon.com/AWSSimpleQueueService/latest/SQSDeveloperGuide/sqs-visibility-timeout.html>

<https://docs.aws.amazon.com/AWSSimpleQueueService/latest/APIReference/API_ChangeMessageVisibility.html>

Bottom of Form

Top of Form

Question 38: Skipped

**You team maintains a public API Gateway that is accessed by clients from another domain. Usage has been consistent for the last few months but recently it has more than doubled. As a result, your costs have gone up and would like to prevent other unauthorized domains from accessing your API.**

**Which of the following actions should you take?**

* ​

Use Mapping Templates

* ​

Assign a Security Group to your API Gateway

* ​

Use Account-level throttling

* ​

Restrict access by using CORS

**(Correct)**

**Explanation**

Correct option:

**Restrict access by using CORS** - Cross-origin resource sharing (CORS) defines a way for client web applications that are loaded in one domain to interact with resources in a different domain. When your API's resources receive requests from a domain other than the API's own domain and you want to restrict servicing these requests, you must disable cross-origin resource sharing (CORS) for selected methods on the resource.

Incorrect options:

**Use Account-level throttling** - To prevent your API from being overwhelmed by too many requests, Amazon API Gateway throttles requests to your API. By default, API Gateway limits the steady-state request rate to 10,000 requests per second (rps). It limits the burst (that is, the maximum bucket size) to 5,000 requests across all APIs within an AWS account. This is Account-level throttling. As you see, this is about limit on the number of requests and is not a suitable answer for the current scenario.

**Use Mapping Templates** - A mapping template is a script expressed in Velocity Template Language (VTL) and applied to the payload using JSONPath expressions. Mapping templates help format/structure the data in a way that it is easily readable, unlike a server response that might always be easy to ready. Mapping Templates have nothing to do with access and are not useful for the current scenario.

**Assign a Security Group to your API Gateway** - API Gateway does not use security groups but uses resource policies, which are JSON policy documents that you attach to an API to control whether a specified principal (typically an IAM user or role) can invoke the API. You can restrict IP address using this, the downside being, an IP address can be changed by the accessing user. So, this is not an optimal solution for the current use case.

References:

<https://docs.aws.amazon.com/AmazonS3/latest/dev/cors.html>

<https://docs.aws.amazon.com/apigateway/latest/developerguide/http-api-protect.html>

<https://docs.aws.amazon.com/apigateway/latest/developerguide/rest-api-data-transformations.html>

Bottom of Form

Top of Form

Question 39: Skipped

**You are a DynamoDB developer for an aerospace company that requires you to write 6 objects per second of 4.5KB in size each.**

**What write capacity unit is needed for your project?**

* ​

30

**(Correct)**

* ​

46

* ​

15

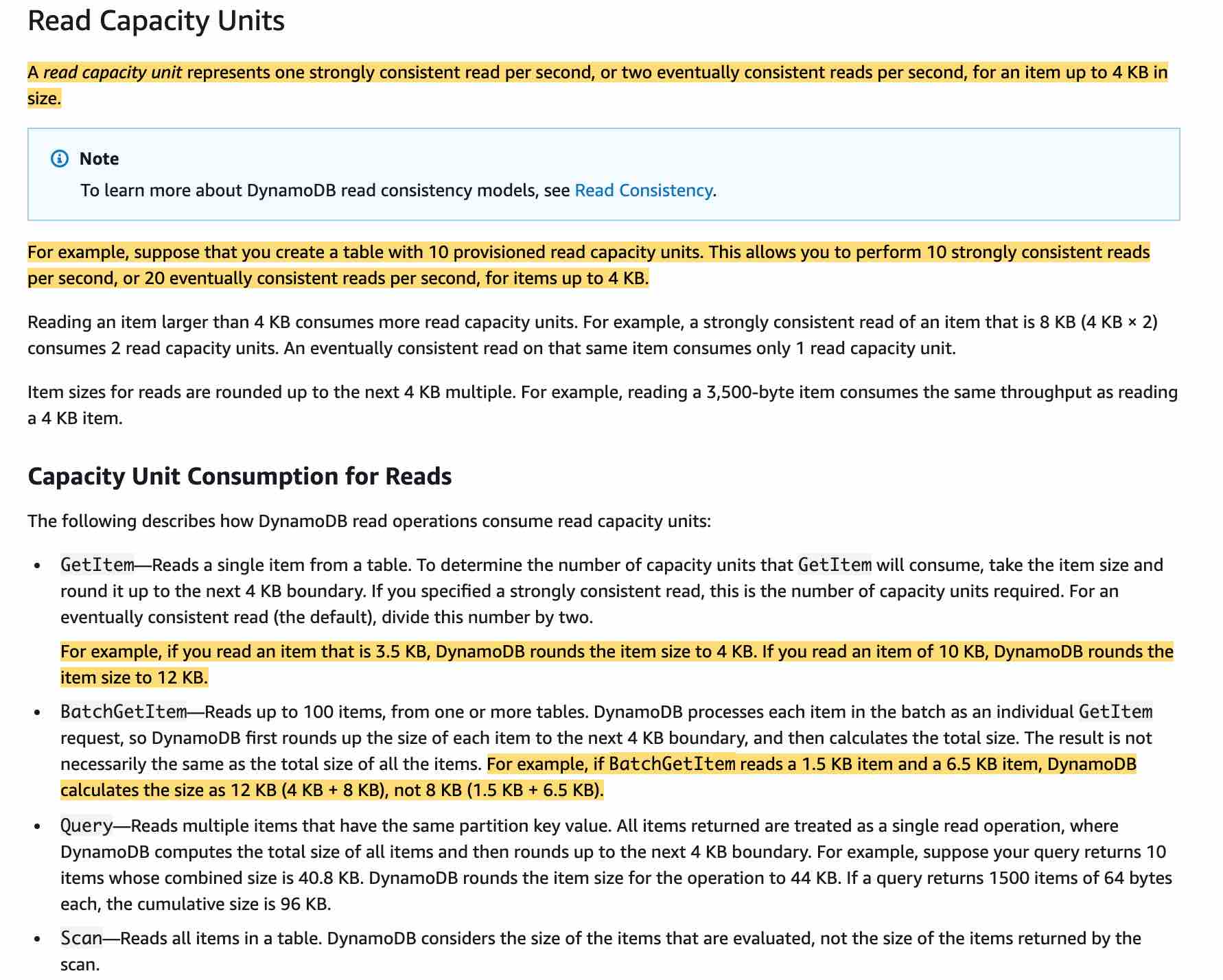
* ​

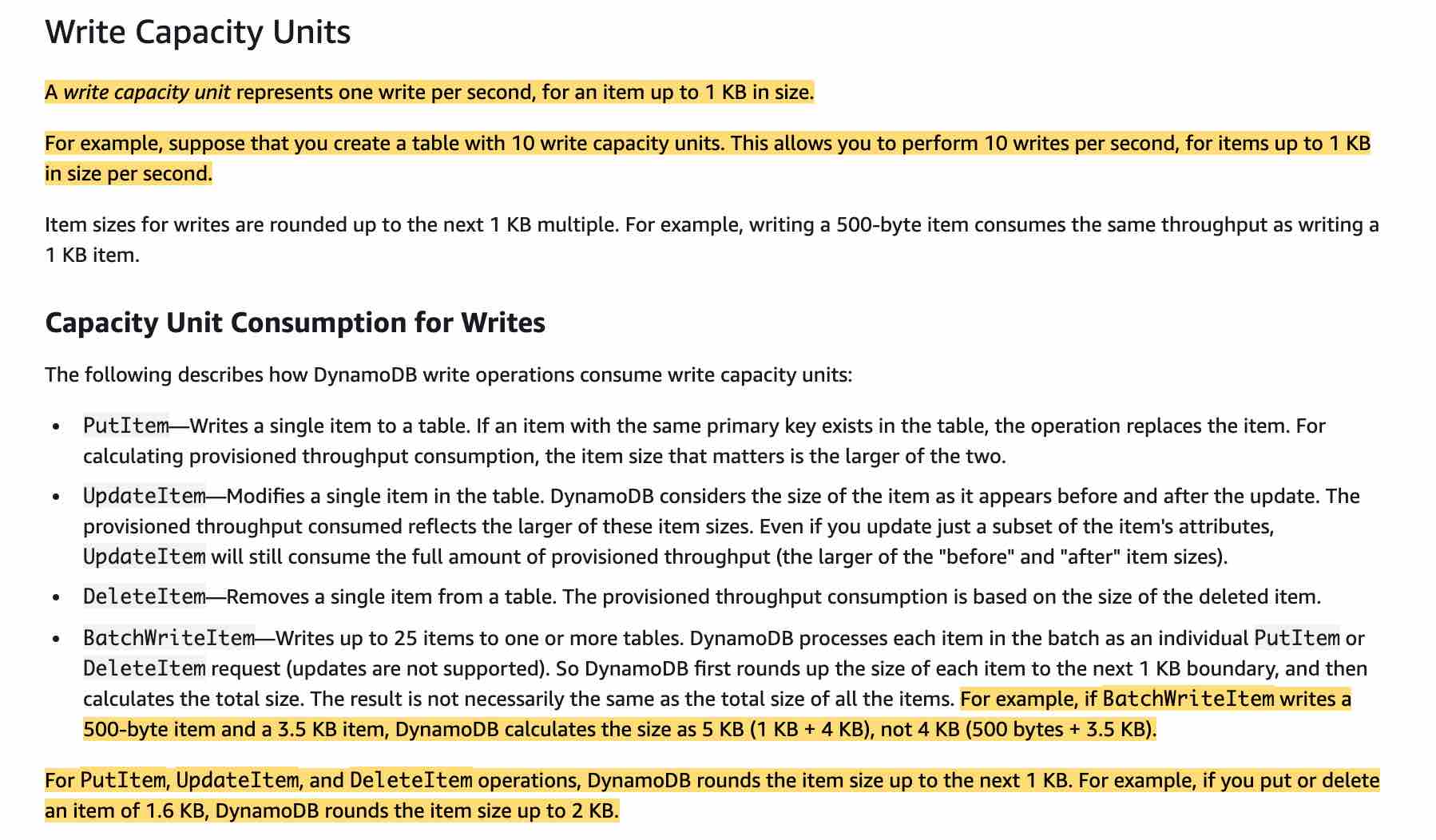
24

**Explanation**

Correct option:

Before proceeding with the calculations, please review the following:





via - <https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/ProvisionedThroughput.html>

**30**

A write capacity unit represents one write per second, for an item up to 1 KB in size.

Item sizes for writes are rounded up to the next 1 KB multiple. For example, writing a 500-byte item consumes the same throughput as writing a 1 KB item. So, for the given use-case, each object is of size 4.5 KB, which will be rounded up to 5KB.

Therefore, for 6 objects, you need 6x5 = 30 WCUs.

Incorrect options:

**24**

**15**

**46**

These three options contradict the details provided in the explanation above, so these are incorrect.

Reference:

<https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/ProvisionedThroughput.html>

Bottom of Form

Top of Form

Question 40: Skipped

**As a site reliability engineer, you are responsible for improving the company’s deployment by scaling and automating applications. As new application versions are ready for production you ensure that the application gets deployed to different sets of EC2 instances at different times allowing for a smooth transition.**

**Using AWS CodeDeploy, which of the following options will allow you to do this?**

* ​

CodeDeploy Hooks

* ​

CodeDeploy Agent

* ​

CodeDeploy Deployment Groups

**(Correct)**

* ​

Define multiple CodeDeploy Applications

**Explanation**

Correct option:

**CodeDeploy Deployment Groups**

You can specify one or more deployment groups for a CodeDeploy application. The deployment group contains settings and configurations used during the deployment. Most deployment group settings depend on the compute platform used by your application. Some settings, such as rollbacks, triggers, and alarms can be configured for deployment groups for any compute platform.

In an EC2/On-Premises deployment, a deployment group is a set of individual instances targeted for deployment. A deployment group contains individually tagged instances, Amazon EC2 instances in Amazon EC2 Auto Scaling groups, or both.

Incorrect options:

**CodeDeploy Agent** - The CodeDeploy agent is a software package that, when installed and configured on an instance, makes it possible for that instance to be used in CodeDeploy deployments. The agent connects the EC2 instances to the CodeDeploy service.

**CodeDeploy Hooks** - Hooks are found in the AppSec file used by AWS CodeDeploy to manage deployment. Hooks correspond to lifecycle events such as ApplicationStart, ApplicationStop, etc. to which you can assign a script.

**Define multiple CodeDeploy Applications** - This option has been added as a distractor. Instead, you want to use deployment groups to use the same deployment and maybe separate the times when a group of instances receives the software updates.

Reference:

<https://docs.aws.amazon.com/codedeploy/latest/userguide/deployment-groups.html>

Bottom of Form

Top of Form

Question 41: Skipped

**A company that specializes in cloud communications platform as a service allows software developers to programmatically use their services to send and receive text messages. The initial platform did not have a scalable architecture as all components were hosted on one server and should be redesigned for high availability and scalability.**

**Which of the following options can be used to implement the new architecture? (select two)**

* ​

API Gateway + Lambda

**(Correct)**

* ​

SES + S3

* ​

CloudWatch + CloudFront

* ​

ALB + ECS

**(Correct)**

* ​

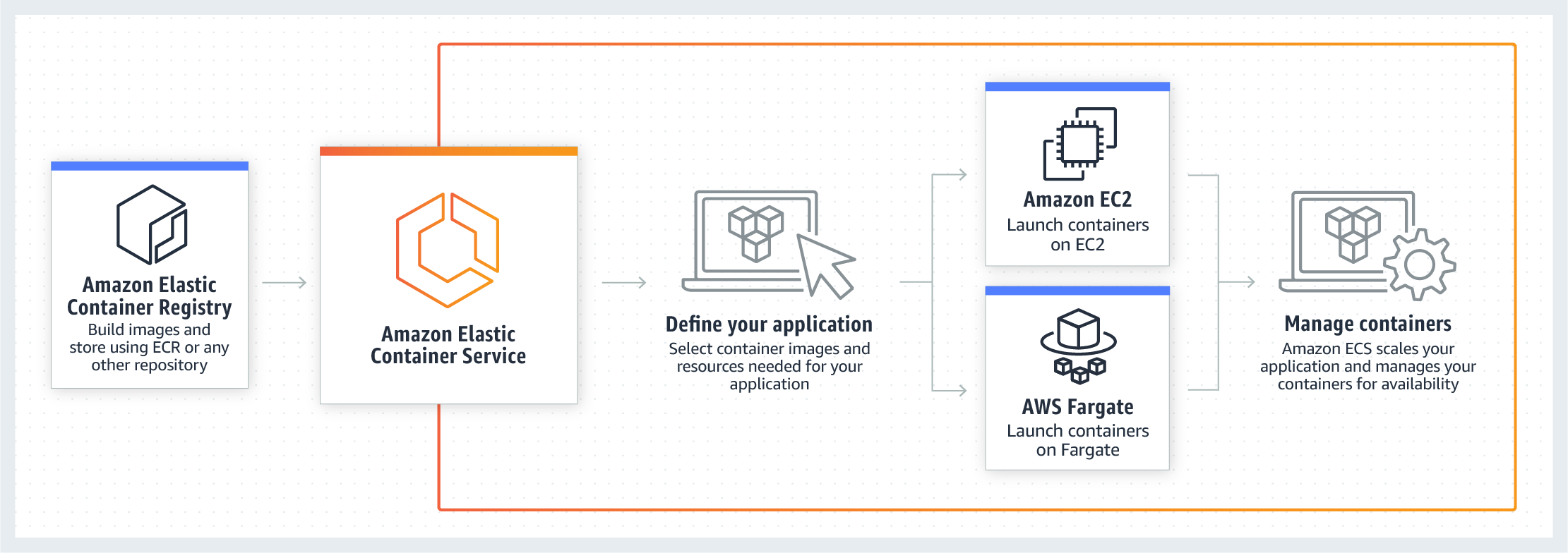
EBS + RDS

**Explanation**

Correct options:

**ALB + ECS**

Amazon Elastic Container Service (ECS) is a highly scalable, high-performance container management service that supports Docker containers and allows you to easily run applications on a managed cluster of Amazon EC2 instances.

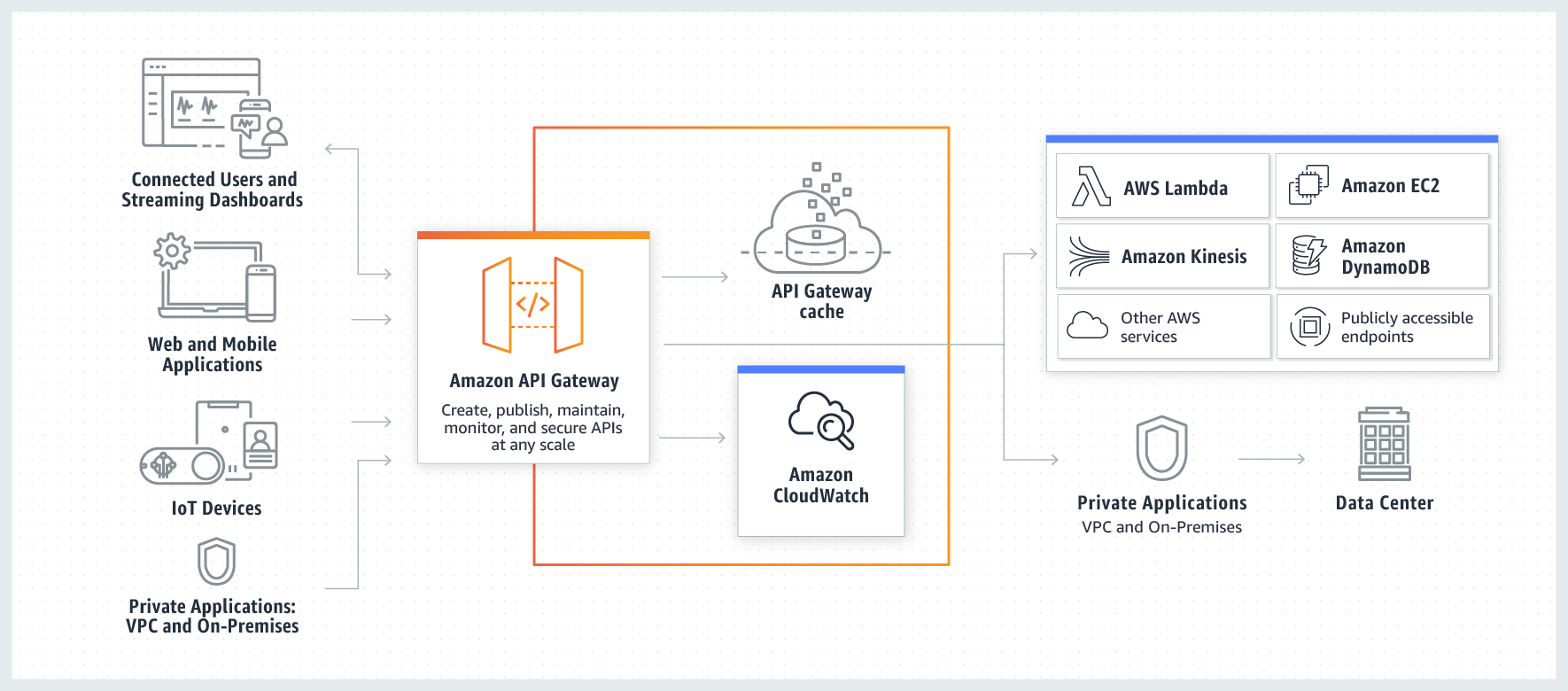
How ECS Works:  via - <https://aws.amazon.com/ecs/>

Elastic Load Balancing automatically distributes incoming application traffic across multiple targets, such as Amazon EC2 instances, containers, IP addresses, and Lambda functions. It can handle the varying load of your application traffic in a single Availability Zone or across multiple Availability Zones.

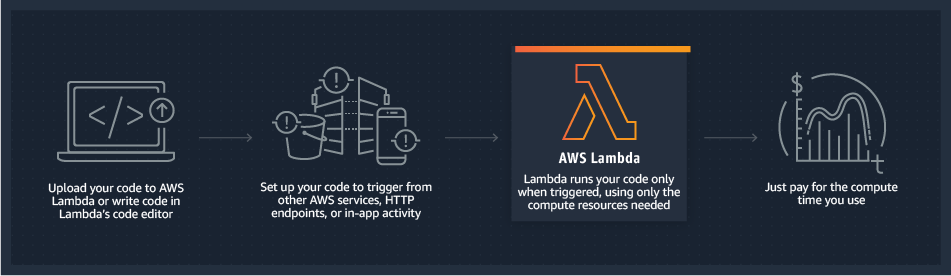
When you use ECS with a load balancer such as ALB deployed across multiple Availability Zones, it helps provide a scalable and highly available REST API.

**API Gateway + Lambda**

Amazon API Gateway is a fully managed service that makes it easy for developers to publish, maintain, monitor, and secure APIs at any scale. Using API Gateway, you can create an API that acts as a “front door” for applications to access data, business logic, or functionality from your back-end services, such as EC2 or Lambda functions.

How API Gateway Works:  via - <https://aws.amazon.com/api-gateway/>

AWS Lambda lets you run code without provisioning or managing servers. You pay only for the compute time you consume.

How Lambda function works:  via - <https://aws.amazon.com/lambda/>

API Gateway and Lambda help achieve the same purpose integrating some capabilities such as authentication in a serverless fashion, with fully scalable and highly available architectures.

Incorrect options:

**SES + S3** - The combination of these services only provide email and object storage services.

**CloudWatch + CloudFront** - The combination of these services only provide monitoring and fast content delivery network (CDN) services.

**EBS + RDS** - The combination of these services only provide elastic block storage and database services.

References:

<https://aws.amazon.com/getting-started/projects/build-serverless-web-app-lambda-apigateway-s3-dynamodb-cognito/module-4/>

<https://aws.amazon.com/blogs/compute/microservice-delivery-with-amazon-ecs-and-application-load-balancers/>

Bottom of Form

Top of Form

Question 42: Skipped

**You work as a developer doing contract work for the government on AWS gov cloud. Your applications use Amazon Simple Queue Service (SQS) for its message queue service. Due to recent hacking attempts, security measures have become stricter and require you to store data in encrypted queues.**

**Which of the following steps can you take to meet your requirements without making changes to the existing code?**

* ​

Enable SQS KMS encryption

**(Correct)**

* ​

Use Client side encryption

* ​

Use the SSL endpoint

* ​

Use Secrets Manager

**Explanation**

Correct option:

**Enable SQS KMS encryption**

Server-side encryption (SSE) lets you transmit sensitive data in encrypted queues. SSE protects the contents of messages in queues using keys managed in AWS Key Management Service (AWS KMS).

AWS KMS combines secure, highly available hardware and software to provide a key management system scaled for the cloud. When you use Amazon SQS with AWS KMS, the data keys that encrypt your message data are also encrypted and stored with the data they protect.

You can choose to have SQS encrypt messages stored in both Standard and FIFO queues using an encryption key provided by AWS Key Management Service (KMS).

Incorrect options:

**Use the SSL endpoint** - The given use-case needs encryption at rest. When using SSL, the data is encrypted during transit, but the data needs to be encrypted at rest as well, so this option is incorrect.

**Use Client-side encryption** - For additional security, you can build your application to encrypt messages before they are placed in a message queue but will require a code change, so this option is incorrect.

*\*Use Secrets Manager \** - AWS Secrets Manager enables you to easily rotate, manage, and retrieve database credentials, API keys, and other secrets throughout their lifecycle. Users and applications retrieve secrets with a call to Secrets Manager APIs, eliminating the need to hardcode sensitive information in plain text. Secrets Manager offers secret rotation with built-in integration for Amazon RDS, Amazon Redshift, and Amazon DocumentDB. Secrets Manager cannot be used for encrypting data at rest.

Reference:

<https://docs.aws.amazon.com/AWSSimpleQueueService/latest/SQSDeveloperGuide/sqs-server-side-encryption.html>

Bottom of Form

Top of Form

Question 43: Skipped

**A security company is requiring all developers to perform server-side encryption with customer-provided encryption keys when performing operations in AWS S3. Developers should write software with C# using the AWS SDK and implement the requirement in the PUT, GET, Head, and Copy operations.**

**Which of the following encryption methods meets this requirement?**

* ​

Client-Side Encryption

* ​

SSE-C

**(Correct)**

* ​

SSE-S3

* ​

SSE-KMS

**Explanation**

Correct option:

**SSE-C**

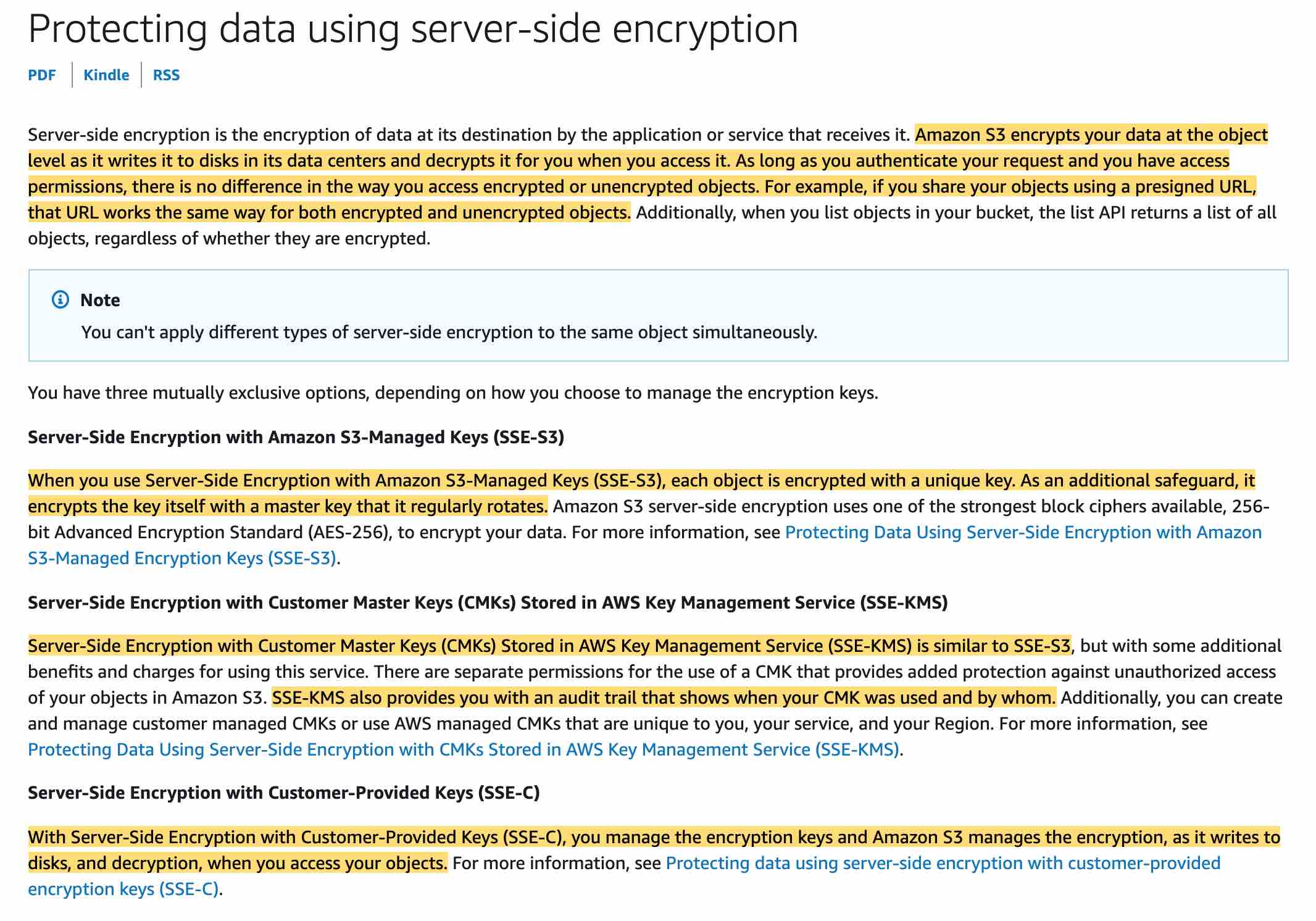
You have the following options for protecting data at rest in Amazon S3:

Server-Side Encryption – Request Amazon S3 to encrypt your object before saving it on disks in its data centers and then decrypt it when you download the objects.

Client-Side Encryption – Encrypt data client-side and upload the encrypted data to Amazon S3. In this case, you manage the encryption process, the encryption keys, and related tools.

For the given use-case, the company wants to manage the encryption keys via its custom application and let S3 manage the encryption, therefore you must use Server-Side Encryption with Customer-Provided Keys (SSE-C).

Using server-side encryption with customer-provided encryption keys (SSE-C) allows you to set your encryption keys. With the encryption key you provide as part of your request, Amazon S3 manages both the encryption, as it writes to disks, and decryption, when you access your objects.

Please review these three options for Server Side Encryption on S3:  via - <https://docs.aws.amazon.com/AmazonS3/latest/dev/serv-side-encryption.html>

Incorrect options:

**SSE-KMS** - Server-Side Encryption with Customer Master Keys (CMKs) stored in AWS Key Management Service (SSE-KMS) is similar to SSE-S3. SSE-KMS provides you with an audit trail that shows when your CMK was used and by whom. Additionally, you can create and manage customer-managed CMKs or use AWS managed CMKs that are unique to you, your service, and your Region.

**Client-Side Encryption** - You can encrypt the data client-side and upload the encrypted data to Amazon S3. In this case, you manage the encryption process, the encryption keys, and related tools.

**SSE-S3** - When you use Server-Side Encryption with Amazon S3-Managed Keys (SSE-S3), each object is encrypted with a unique key. As an additional safeguard, it encrypts the key itself with a master key that it regularly rotates. So this option is incorrect.

Reference:

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

Bottom of Form

Top of Form

Question 44: Skipped

**A company developed an app-based service for citizens to book transportation rides in the local community. The platform is running on AWS EC2 instances and uses Amazon Relational Database Service (RDS) for storing transportation data. A new feature has been requested where receipts would be emailed to customers with PDF attachments retrieved from Amazon Simple Storage Service (S3).**

**Which of the following options will provide EC2 instances with the right permissions to upload files to Amazon S3 and generate S3 Signed URL?**

* ​

Run aws configure on the EC2 instance

* ​

EC2 User Data

* ​

CloudFormation

* ​

Create an IAM Role for EC2

**(Correct)**

**Explanation**

Correct option:

**Create an IAM Role for EC2**

IAM roles have been incorporated so that your applications can securely make API requests from your instances, without requiring you to manage the security credentials that the applications use. Instead of creating and distributing your AWS credentials, you can delegate permission to make API requests using IAM roles.

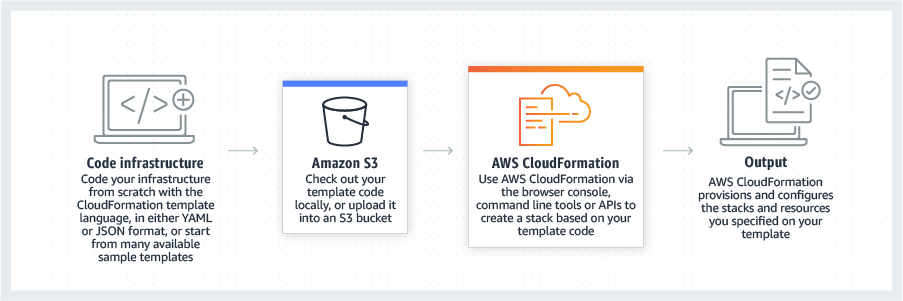
Amazon EC2 uses an instance profile as a container for an IAM role. When you create an IAM role using the IAM console, the console creates an instance profile automatically and gives it the same name as the role to which it corresponds.

Incorrect options:

**EC2 User Data** - You can specify user data when you launch an instance and you would not want to hard code the AWS credentials in the user data.

**Run aws configure on the EC2 instance** - When you first configure the CLI you have to run this command, afterward you should not need to if you want to obtain credentials to authenticate to other AWS services. An IAM role will receive temporary credentials for you so you can focus on using the CLI to get access to other AWS services if you have the permissions.

**CloudFormation** - AWS CloudFormation gives developers and businesses an easy way to create a collection of related AWS and third-party resources and provision them in an orderly and predictable fashion.

How CloudFormation Works:  via - <https://aws.amazon.com/cloudformation/>

Reference:

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

Bottom of Form

Top of Form

Question 45: Skipped

**Your company leverages Amazon CloudFront to provide content via the internet to customers with low latency. Aside from latency, security is another concern and you are looking for help in enforcing end-to-end connections using HTTPS so that content is protected.**

**Which of the following options is available for HTTPS in AWS CloudFront?**

* ​

Between clients and CloudFront only

* ​

Neither between clients and CloudFront nor between CloudFront and backend

* ​

Between clients and CloudFront as well as between CloudFront and backend

**(Correct)**

* ​

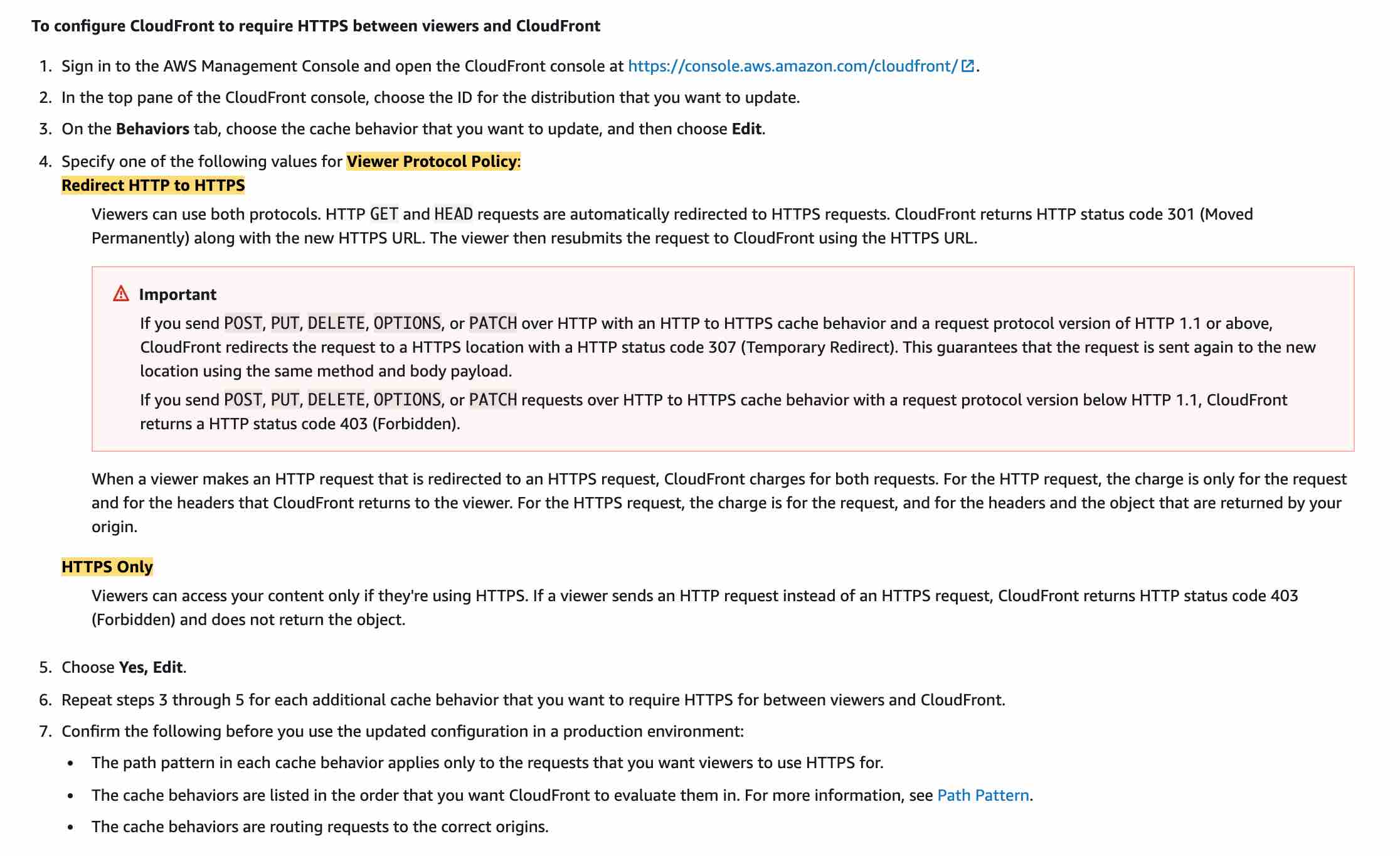
Between CloudFront and backend only

**Explanation**

Correct option:

**Between clients and CloudFront as well as between CloudFront and backend**

For web distributions, you can configure CloudFront to require that viewers use HTTPS to request your objects, so connections are encrypted when CloudFront communicates with viewers.

Requiring HTTPS for Communication Between Viewers and CloudFront:  via - <https://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/using-https-viewers-to-cloudfront.html>

You also can configure CloudFront to use HTTPS to get objects from your origin, so connections are encrypted when CloudFront communicates with your origin.

Requiring HTTPS for Communication Between CloudFront and Your Custom Origin:  via - <https://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/using-https-cloudfront-to-custom-origin.html>

Incorrect options:

**Between clients and CloudFront only** - This is incorrect as you can choose to require HTTPS between CloudFront and your origin.

**Between CloudFront and backend only** - This is incorrect as you can choose to require HTTPS between viewers and CloudFront.

**Neither between clients and CloudFront nor between CloudFront and backend** - This is incorrect as you can choose HTTPS settings both for communication between viewers and CloudFront as well as between CloudFront and your origin.

References:

<https://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/secure-connections-supported-viewer-protocols-ciphers.html#secure-connections-supported-ciphers-cloudfront-to-origin>

<https://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/using-https-viewers-to-cloudfront.html>

<https://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/using-https-cloudfront-to-custom-origin.html>

Bottom of Form

Top of Form

Question 46: Skipped

**An organization with online transaction processing (OLTP) workloads have successfully moved to DynamoDB after having many issues with traditional database systems. However, a few months into production, DynamoDB tables are consistently recording high latency.**

**As a Developer Associate, which of the following would you suggest to reduce the latency? (Select two)**

* ​

Reduce connection pooling, which keeps the connections alive even when user requests are not present, thereby, blocking the services

* ​

Use DynamoDB Accelerator (DAX) for businesses with heavy write-only workloads

* ​

Use eventually consistent reads in place of strongly consistent reads whenever possible

**(Correct)**

* ​

Increase the request timeout settings, so the client gets enough time to complete the requests, thereby reducing retries on the system

* ​

Consider using Global tables if your application is accessed by globally distributed users

**(Correct)**

**Explanation**

Correct option:

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 for internet-scale applications.

**Consider using Global tables if your application is accessed by globally distributed users** - If you have globally dispersed users, consider using global tables. With global tables, you can specify the AWS Regions where you want the table to be available. This can significantly reduce latency for your users. So, reducing the distance between the client and the DynamoDB endpoint is an important performance fix to be considered.

**Use eventually consistent reads in place of strongly consistent reads whenever possible** - If your application doesn't require strongly consistent reads, consider using eventually consistent reads. Eventually consistent reads are cheaper and are less likely to experience high latency. For more information

Incorrect options:

**Increase the request timeout settings, so the client gets enough time to complete the requests, thereby reducing retries on the system** - This statement is incorrect. The right way is to reduce the request timeout settings. This causes the client to abandon high latency requests after the specified time period and then send a second request that usually completes much faster than the first.

**Reduce connection pooling, which keeps the connections alive even when user requests are not present, thereby, blocking the services** - This is not correct. When you're not making requests, consider having the client send dummy traffic to a DynamoDB table. Alternatively, you can reuse client connections or use connection pooling. All of these techniques keep internal caches warm, which helps keep latency low.

**Use DynamoDB Accelerator (DAX) for businesses with heavy write-only workloads** - This is not correct. If your traffic is read-heavy, consider using a caching service such as DynamoDB Accelerator (DAX). DAX is a fully managed, highly available, in-memory cache for DynamoDB that delivers up to a 10x performance improvement—from milliseconds to microseconds—even at millions of requests per second.

References:

<https://aws.amazon.com/premiumsupport/knowledge-center/dynamodb-high-latency/>

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

Bottom of Form

Top of Form

Question 47: Skipped

**Your team-mate has configured an Amazon S3 event notification for an S3 bucket that holds sensitive audit data of a firm. As the Team Lead, you are receiving the SNS notifications for every event in this bucket. After validating the event data, you realized that few events are missing.**

**What could be the reason for this behavior and how to avoid this in the future?**

* ​

Your notification action is writing to the same bucket that triggers the notification

* ​

Someone could have created a new notification configuration and that has overridden your existing configuration

* ​

Versioning is enabled on the S3 bucket and event notifications are getting fired for only one version

* ​

If two writes are made to a single non-versioned object at the same time, it is possible that only a single event notification will be sent

**(Correct)**

**Explanation**

Correct option:

**If two writes are made to a single non-versioned object at the same time, it is possible that only a single event notification will be sent** - Amazon S3 event notifications are designed to be delivered at least once. Typically, event notifications are delivered in seconds but can sometimes take a minute or longer.

If two writes are made to a single non-versioned object at the same time, it is possible that only a single event notification will be sent. If you want to ensure that an event notification is sent for every successful write, you can enable versioning on your bucket. With versioning, every successful write will create a new version of your object and will also send event notification.

Incorrect options:

**Someone could have created a new notification configuration and that has overridden your existing configuration** - It is possible that the configuration can be overridden. But, in the current scenario, the team lead is receiving notifications for most of the events, which nullifies the claim that the configuration is overridden.

**Versioning is enabled on the S3 bucket and event notifications are getting fired for only one version** - This is an incorrect statement. If you want to ensure that an event notification is sent for every successful write, you should enable versioning on your bucket. With versioning, every successful write will create a new version of your object and will also send event notification.

**Your notification action is writing to the same bucket that triggers the notification** - If your notification ends up writing to the bucket that triggers the notification, this could cause an execution loop. But it will not result in missing events.

Reference:

<https://docs.aws.amazon.com/AmazonS3/latest/dev/NotificationHowTo.html>

Bottom of Form

Top of Form

Question 48: Skipped

**A photo-sharing application manages its EC2 server fleet running behind an Application Load Balancer and the traffic is fronted by a CloudFront distribution. The development team wants to decouple the user authentication process for the application so that the application servers can just focus on the business logic.**

**As a Developer Associate, which of the following solutions would you recommend to address this use-case?**

* ​

Use Cognito Authentication via Cognito User Pools for your Application Load Balancer

**(Correct)**

* ​

Use Cognito Authentication via Cognito Identity Pools for your CloudFront distribution

* ​

Use Cognito Authentication via Cognito User Pools for your CloudFront distribution

* ​

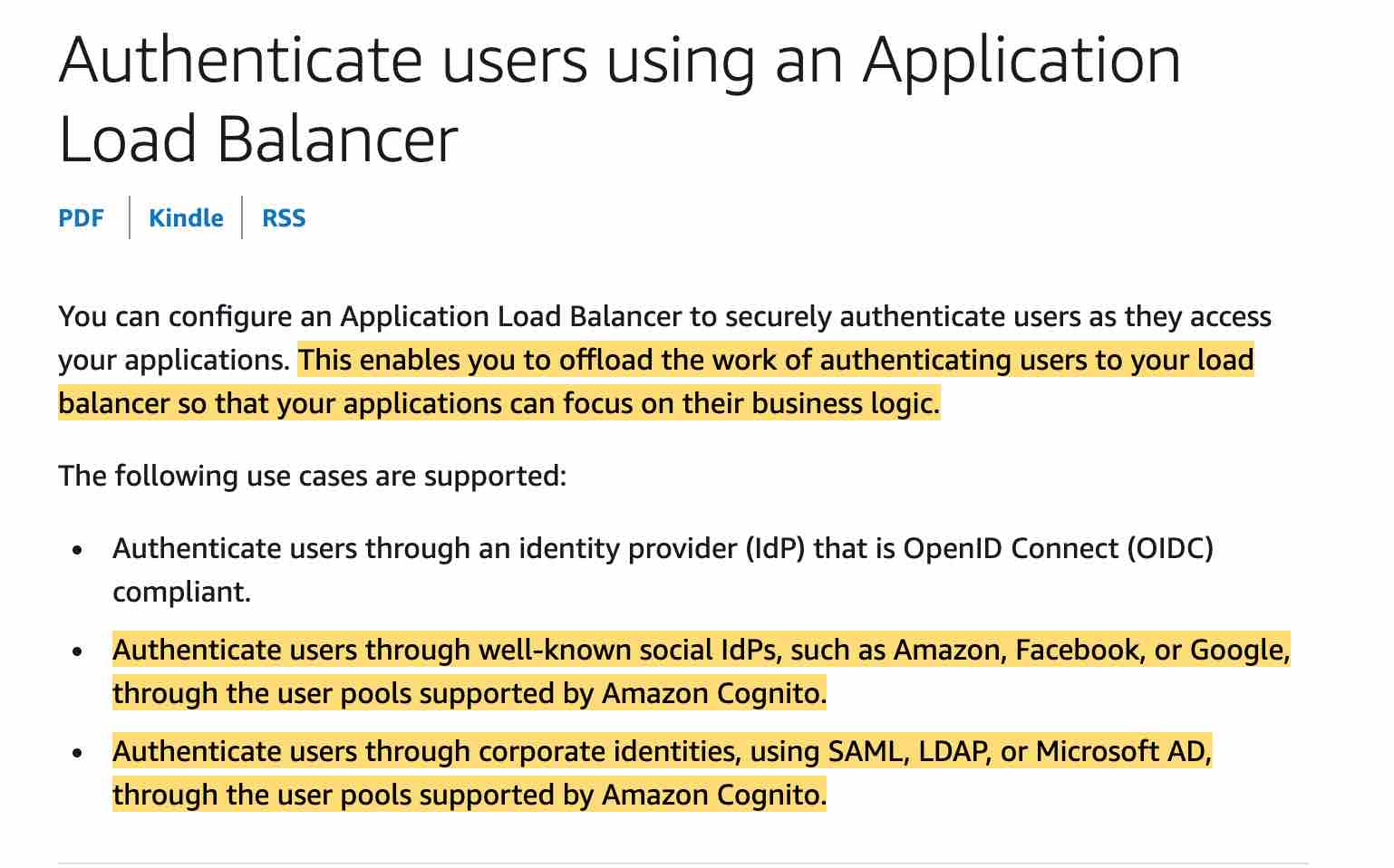
Use Cognito Authentication via Cognito Identity Pools for your Application Load Balancer

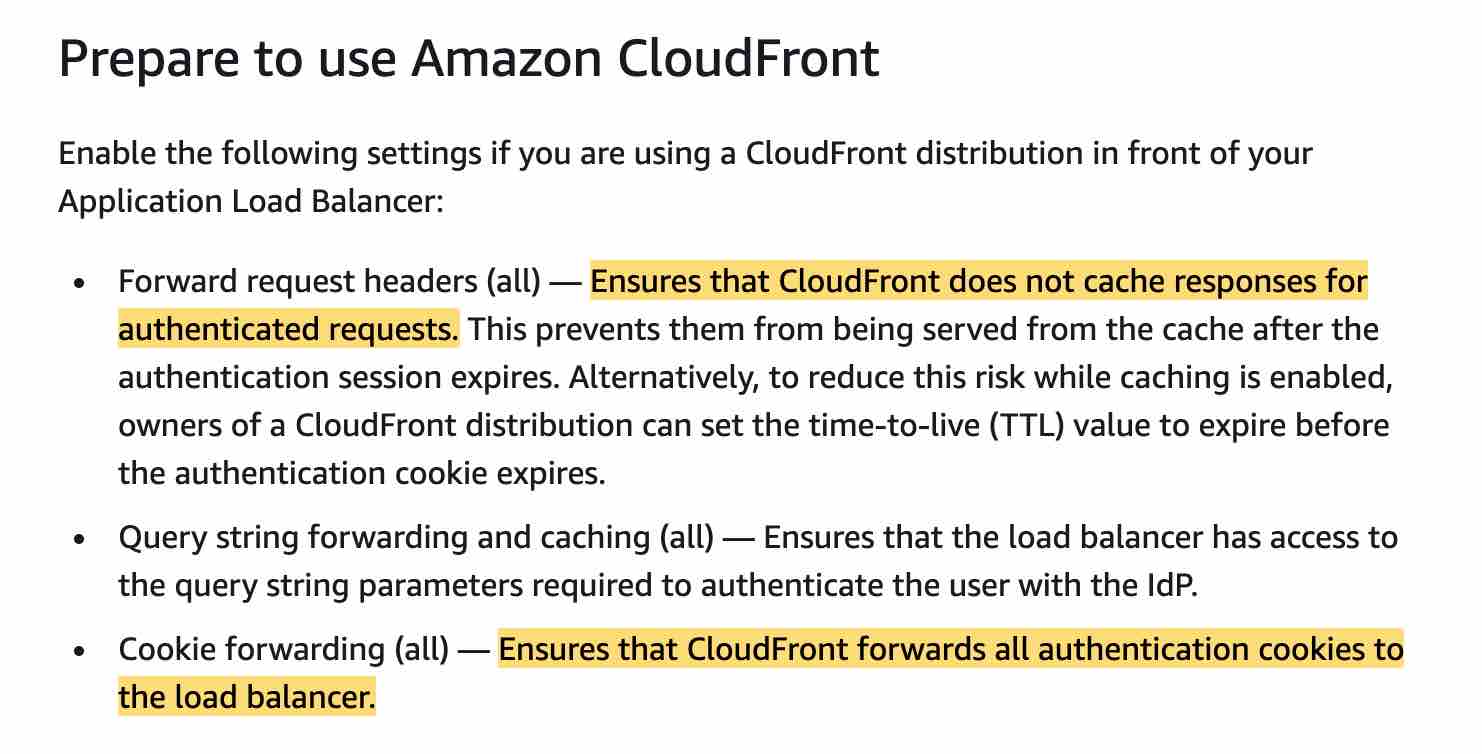
**Explanation**

Correct option:

**Use Cognito Authentication via Cognito User Pools for your Application Load Balancer**

Application Load Balancer can be used to securely authenticate users for accessing your applications. This enables you to offload the work of authenticating users to your load balancer so that your applications can focus on their business logic. You can use Cognito User Pools to authenticate users through well-known social IdPs, such as Amazon, Facebook, or Google, through the user pools supported by Amazon Cognito or through corporate identities, using SAML, LDAP, or Microsoft AD, through the user pools supported by Amazon Cognito.

 via - <https://docs.aws.amazon.com/elasticloadbalancing/latest/application/listener-authenticate-users.html>

Please make sure that you adhere to the following configurations while using CloudFront distribution in front of your Application Load Balancer:  via - <https://docs.aws.amazon.com/elasticloadbalancing/latest/application/listener-authenticate-users.html>

Exam Alert:

Please review the following note to understand the differences between Cognito User Pools and Cognito Identity Pools:  via - <https://docs.aws.amazon.com/cognito/latest/developerguide/what-is-amazon-cognito.html>

Incorrect options:

**Use Cognito Authentication via Cognito Identity Pools for your Application Load Balancer** - There is no such thing as using Cognito Authentication via Cognito Identity Pools for managing user authentication for the application. Application-specific user authentication can be provided via Cognito User Pools. Amazon Cognito identity pools provide temporary AWS credentials for users who are guests (unauthenticated) and for users who have been authenticated and received a token.

**Use Cognito Authentication via Cognito User Pools for your CloudFront distribution**

**Use Cognito Authentication via Cognito Identity Pools for your CloudFront distribution**

You cannot use Cognito Authentication with CloudFront distribution, so both these options are incorrect.

References:

<https://docs.aws.amazon.com/elasticloadbalancing/latest/application/listener-authenticate-users.html>

<https://docs.aws.amazon.com/cognito/latest/developerguide/cognito-user-identity-pools.html>

Bottom of Form

Top of Form

Question 49: Skipped

**A development team at a social media company uses AWS Lambda for its serverless stack on AWS Cloud. For a new deployment, the Team Lead wants to send only a certain portion of the traffic to the new Lambda version. In case the deployment goes wrong, the solution should also support the ability to roll back to a previous version of the Lambda function, with MIMINUM downtime for the application.**

**As a Developer Associate, which of the following options would you recommend to address this use-case?**

* ​

Set up the application to directly deploy the new Lambda version. If the deployment goes wrong, reset the application back to the current version using the version number in the ARN

* ​

Set up the application to have multiple alias of the Lambda function. Deploy the new version of the code. Configure a new alias that points to the current alias of the Lambda function for handling 10% of the traffic. If the deployment goes wrong, reset the new alias to point all traffic to the most recent working alias of the Lambda function

* ​

Set up the application to use an alias that points to the current version. Deploy the new version of the code and configure alias to send all users to this new version. If the deployment goes wrong, reset the alias to point to the current version

* ​

Set up the application to use an alias that points to the current version. Deploy the new version of the code and configure the alias to send 10% of the users to this new version. If the deployment goes wrong, reset the alias to point all traffic to the current version

**(Correct)**

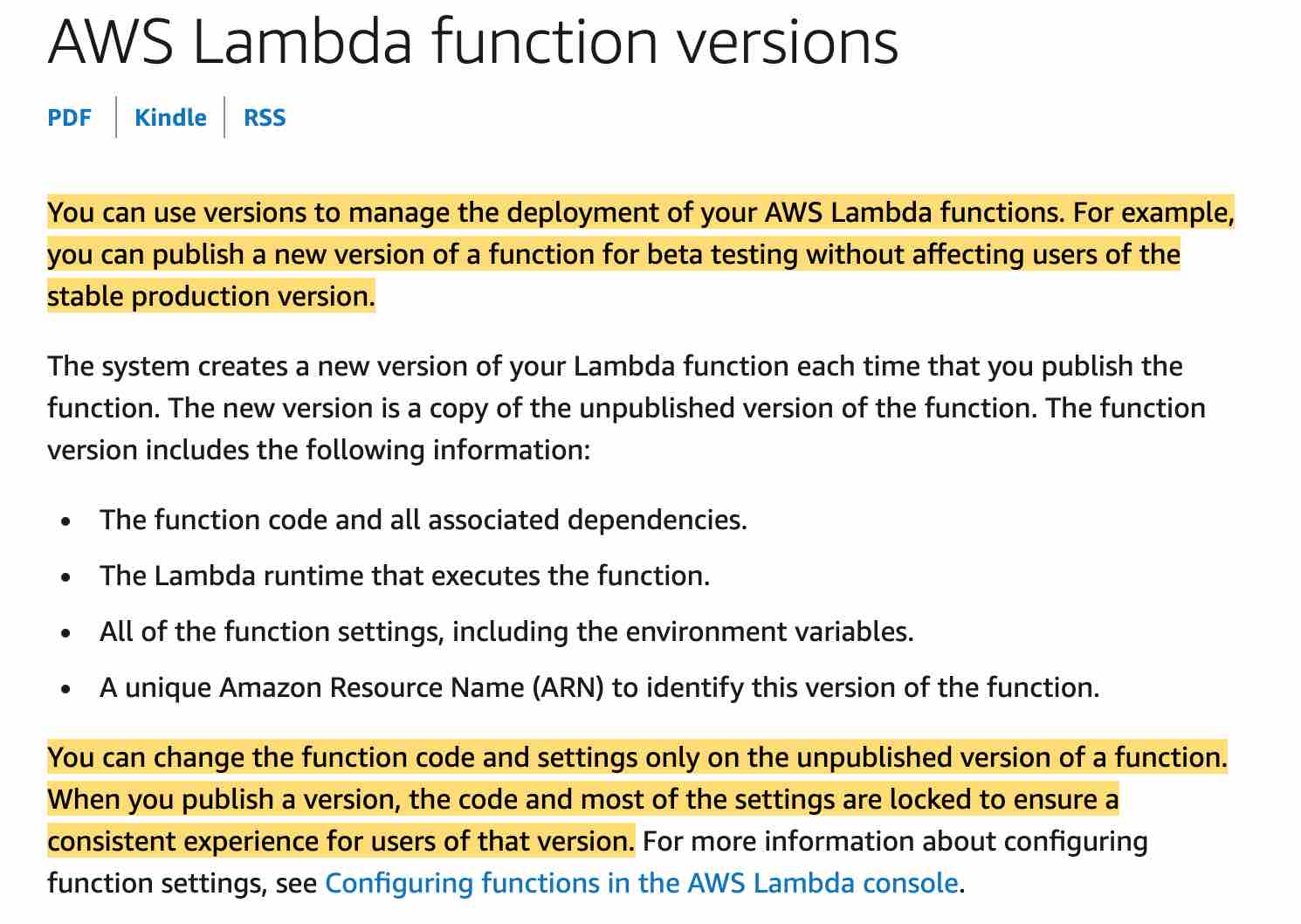
**Explanation**

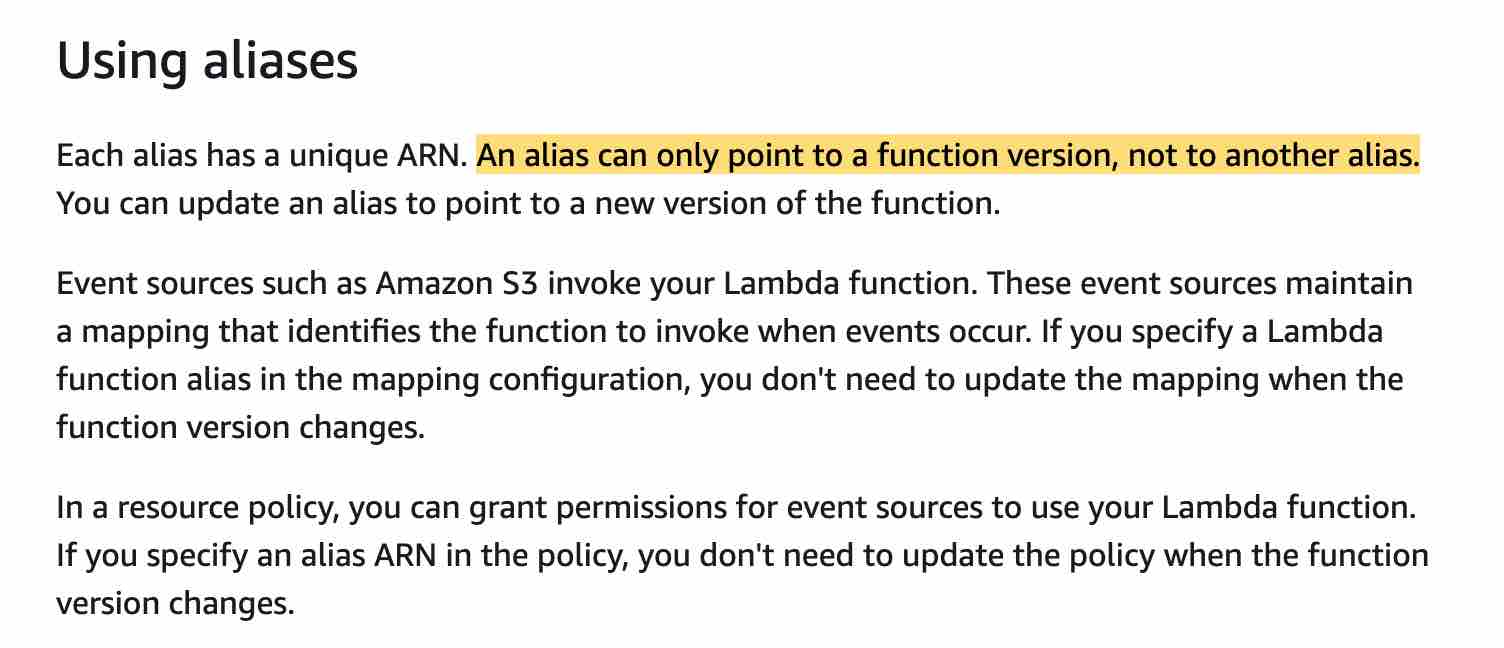
Correct option:

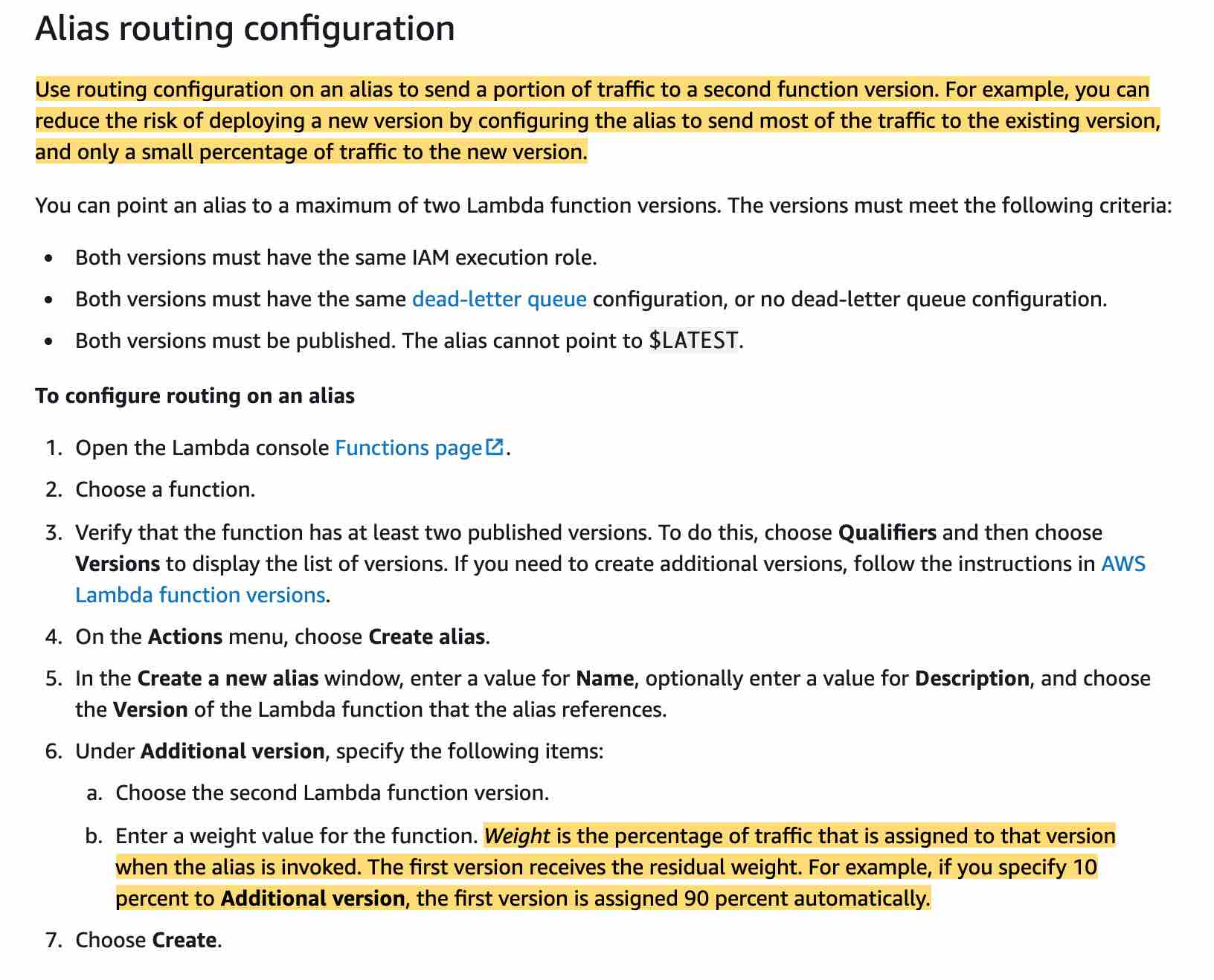
**Set up the application to use an alias that points to the current version. Deploy the new version of the code and configure the alias to send 10% of the users to this new version. If the deployment goes wrong, reset the alias to point all traffic to the current version**

You can use versions to manage the deployment of your AWS Lambda functions. For example, you can publish a new version of a function for beta testing without affecting users of the stable production version. You can change the function code and settings only on the unpublished version of a function. When you publish a version, the code and most of the settings are locked to ensure a consistent experience for users of that version.

You can create one or more aliases for your AWS Lambda function. A Lambda alias is like a pointer to a specific Lambda function version. You can use routing configuration on an alias to send a portion of traffic to a Lambda function version. For example, you can reduce the risk of deploying a new version by configuring the alias to send most of the traffic to the existing version, and only a small percentage of traffic to the new version.

 via - <https://docs.aws.amazon.com/lambda/latest/dg/configuration-versions.html>

 via - <https://docs.aws.amazon.com/lambda/latest/dg/configuration-aliases.html>

 via - <https://docs.aws.amazon.com/lambda/latest/dg/configuration-aliases.html>

Incorrect options:

**Set up the application to use an alias that points to the current version. Deploy the new version of the code and configure alias to send all users to this new version. If the deployment goes wrong, reset the alias to point to the current version** - In this case, the application uses an alias to send all traffic to the new version which does not meet the requirement of sending only a certain portion of the traffic to the new Lambda version. In addition, if the deployment goes wrong, the application would see a downtime. Hence this option is incorrect.

**Set up the application to directly deploy the new Lambda version. If the deployment goes wrong, reset the application back to the current version using the version number in the ARN** - In this case, the application sends all traffic to the new version which does not meet the requirement of sending only a certain portion of the traffic to the new Lambda version. In addition, if the deployment goes wrong, the application would see a downtime. Hence this option is incorrect.

**Set up the application to have multiple alias of the Lambda function. Deploy the new version of the code. Configure a new alias that points to the current alias of the Lambda function for handling 10% of the traffic. If the deployment goes wrong, reset the new alias to point all traffic to the most recent working alias of the Lambda function** - This option has been added as a distractor. The alias for a Lambda function can only point to a Lambda function version. It cannot point to another alias.

References:

<https://docs.aws.amazon.com/lambda/latest/dg/configuration-aliases.html>

<https://docs.aws.amazon.com/lambda/latest/dg/configuration-versions.html>

Bottom of Form

Top of Form

Question 50: Skipped

**The development team at a retail organization wants to allow a Lambda function in its AWS Account A to access a DynamoDB table in another AWS Account B.**

**As a Developer Associate, which of the following solutions would you recommend for the given use-case?**

* ​

Create an IAM role in Account B with access to DynamoDB. Modify the trust policy of the role in Account B to allow the execution role of Lambda to assume this role. Update the Lambda function code to add the AssumeRole API call

**(Correct)**

* ​

Create a clone of the Lambda function in AWS Account B so that it can access the DynamoDB table in the same account

* ​

Add a resource policy to the DynamoDB table in AWS Account B to give access to the Lambda function in Account A

* ​

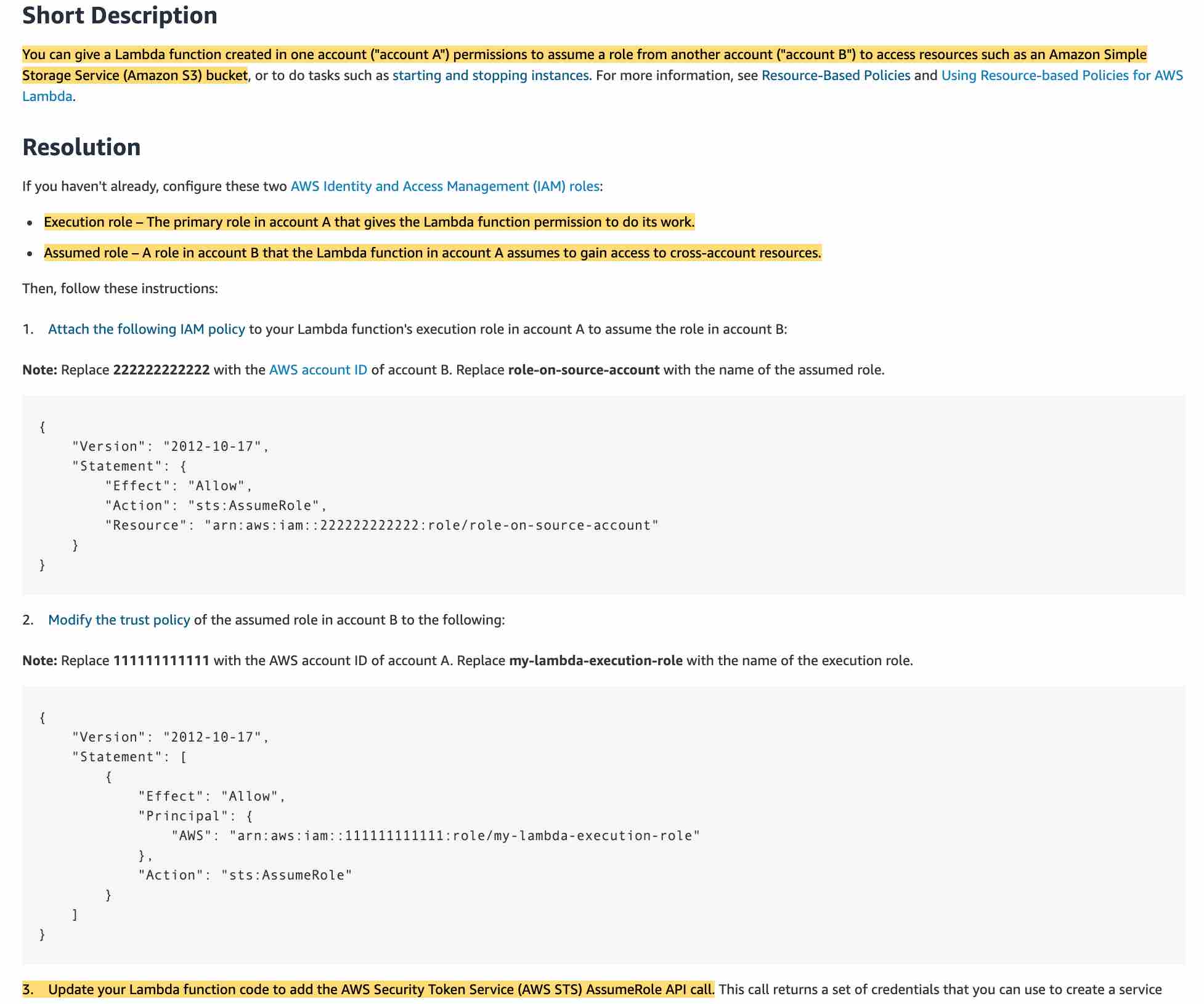
Create an IAM role in Account B with access to DynamoDB. Modify the trust policy of the execution role in Account A to allow the execution role of Lambda to assume the IAM role in Account B. Update the Lambda function code to add the AssumeRole API call

**Explanation**

Correct option:

**Create an IAM role in account B with access to DynamoDB. Modify the trust policy of the role in Account B to allow the execution role of Lambda to assume this role. Update the Lambda function code to add the AssumeRole API call**

You can give a Lambda function created in one account ("account A") permissions to assume a role from another account ("account B") to access resources such as DynamoDB or S3 bucket. You need to create an execution role in Account A that gives the Lambda function permission to do its work. Then you need to create a role in account B that the Lambda function in account A assumes to gain access to the cross-account DynamoDB table. Make sure that you modify the trust policy of the role in Account B to allow the execution role of Lambda to assume this role. Finally, update the Lambda function code to add the AssumeRole API call.

Sample use-case to configure a Lambda function to assume a role from another AWS account:  via - <https://aws.amazon.com/premiumsupport/knowledge-center/lambda-function-assume-iam-role/>

Incorrect options:

**Create a clone of the Lambda function in AWS Account B so that it can access the DynamoDB table in the same account** - Creating a clone of the Lambda function is a distractor as this does not solve the use-case outlined in the problem statement.

**Add a resource policy to the DynamoDB table in AWS Account B to give access to the Lambda function in Account A** - You cannot attach a resource policy to a DynamoDB table, so this option is incorrect.

**Create an IAM role in Account B with access to DynamoDB. Modify the trust policy of the execution role in Account A to allow the execution role of Lambda to assume the IAM role in Account B. Update the Lambda function code to add the AssumeRole API call** - As mentioned in the explanation above, you need to modify the trust policy of the IAM role in Account B so that it allows the execution role of Lambda function in account A to assume the IAM role in Account B.

Reference: <https://aws.amazon.com/premiumsupport/knowledge-center/lambda-function-assume-iam-role/>

Bottom of Form

Top of Form

Question 51: Skipped

**You are a developer handling a deployment service that automates application deployments to Amazon EC2 instances. Most of the deployments consist of code, but sometimes web and configuration files. One of your deployments failed and was rolled back by AWS CodeDeploy to the last known good application revision.**

**During rollback which of the following instances did AWS CodeDeploy deploy first to?**

* ​

To the failed instances

**(Correct)**

* ​

You cannot rollback a CodeDeploy deployment

* ​

To the non-failed instances

* ​

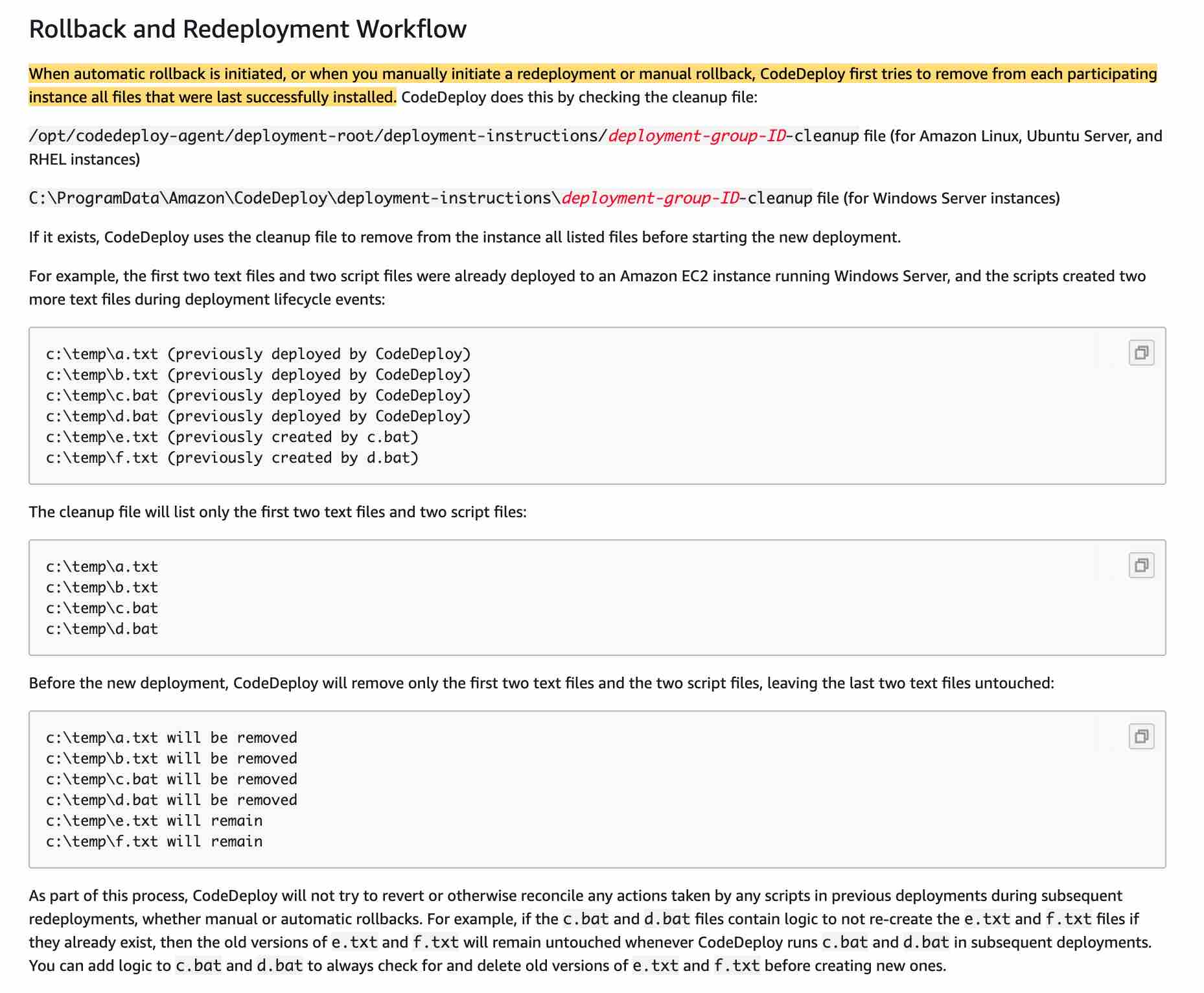
To the new instances

**Explanation**

Correct option:

AWS CodeDeploy is a fully managed deployment service that automates software deployments to a variety of compute services such as Amazon EC2, AWS Fargate, AWS Lambda, and your on-premises servers. AWS CodeDeploy makes it easier for you to rapidly release new features, helps you avoid downtime during application deployment, and handles the complexity of updating your applications.

**To the failed instances**: AWS CodeDeploy rolls back deployments by redeploying a previously deployed revision of an application as a new deployment on the failed instances.

 via - <https://docs.aws.amazon.com/codedeploy/latest/userguide/deployments-rollback-and-redeploy.html>

Incorrect options:

**To the non-failed instances** - Nothing happens to the non-failed instances if any.

**To new instances** - Nothing is deployed to the new instances.

**You cannot rollback a CodeDeploy deployment** - You can rollback a CodeDeploy deployment. This option is incorrect.

Reference:

<https://docs.aws.amazon.com/codedeploy/latest/userguide/deployments-rollback-and-redeploy.html>

Bottom of Form

Top of Form

Question 52: Skipped

**A company has AWS Lambda functions where each is invoked by other AWS services such as Amazon Kinesis Data Firehose, Amazon API Gateway, Amazon Simple Storage Service, or Amazon CloudWatch Events. What these Lambda functions have in common is that they process heavy workloads such as big data analysis, large file processing, and statistical computations.**

**What should you do to improve the performance of your AWS Lambda functions without changing your code?**

* ​

Change the instance type for your Lambda function

* ​

Increase the RAM assigned to your Lambda function

**(Correct)**

* ​

Change your Lambda function runtime to use Golang

* ​

Increase the Lambda function timeout

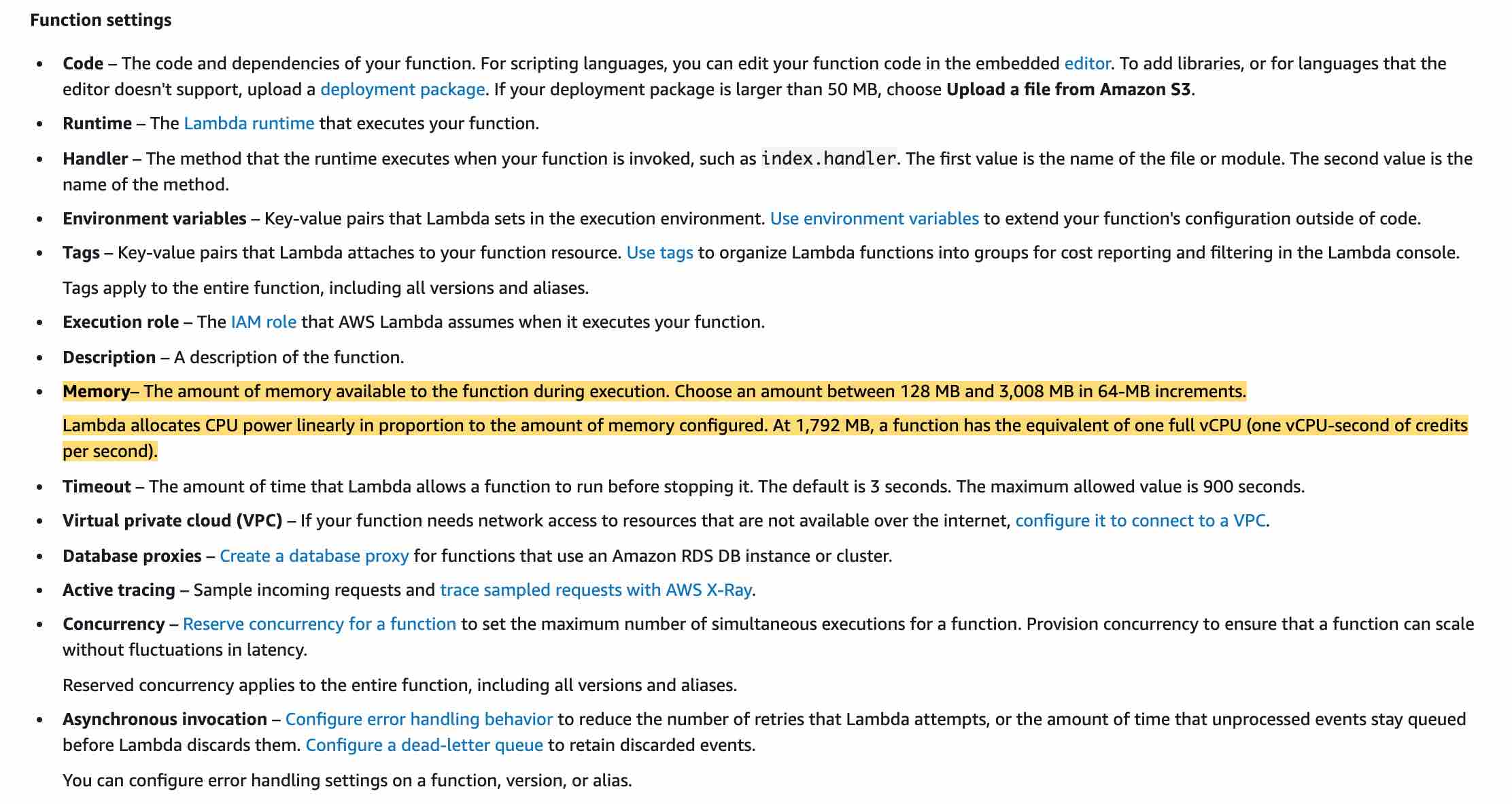
**Explanation**

Correct option:

**Increase the RAM assigned to your Lambda function**

AWS Lambda lets you run code without provisioning or managing servers. You pay only for the compute time you consume.

In the AWS Lambda resource model, you choose the amount of memory you want for your function which allocates proportional CPU power and other resources. This means you will have access to more compute power when you choose one of the new larger settings. You can set your memory in 64MB increments from 128MB to 3008MB. You access these settings when you create a function or update its configuration. The settings are available using the AWS Management Console, AWS CLI, or SDKs.

 via - <https://docs.aws.amazon.com/lambda/latest/dg/configuration-console.html>

Therefore, by increasing the amount of memory available to the Lambda functions, you can run the compute-heavy workflows.

Incorrect options:

**Change the instance type for your Lambda function** - Instance types apply to the EC2 service and not to Lambda function as its a serverless service.

**Change your Lambda function runtime to use Golang** - This changes programming language which requires code changes, so this option is not correct. Besides, changing the runtime may not even address the performance issues.

**Increase the Lambda function timeout** - This option would increase the amount of time for which the Lambda function executes, which may help in case you have some heavy processing, but won't help with the actual performance of your Lambda function.

Reference:

<https://docs.aws.amazon.com/lambda/latest/dg/configuration-console.html>

Bottom of Form

Top of Form

Question 53: Skipped

**The development team at an IT company uses CloudFormation to manage its AWS infrastructure. The team has created a network stack containing a VPC with subnets and a web application stack with EC2 instances and an RDS instance. The team wants to reference the VPC created in the network stack into its web application stack.**

**As a Developer Associate, which of the following solutions would you recommend for the given use-case?**

* ​

Create a cross-stack reference and use the Outputs output field to flag the value of VPC from the network stack. Then use Ref intrinsic function to reference the value of VPC into the web application stack

* ​

Create a cross-stack reference and use the Export output field to flag the value of VPC from the network stack. Then use Ref intrinsic function to reference the value of VPC into the web application stack

* ​

Create a cross-stack reference and use the Outputs output field to flag the value of VPC from the network stack. Then use Fn::ImportValue intrinsic function to import the value of VPC into the web application stack

* ​

Create a cross-stack reference and use the Export output field to flag the value of VPC from the network stack. Then use Fn::ImportValue intrinsic function to import the value of VPC into the web application stack

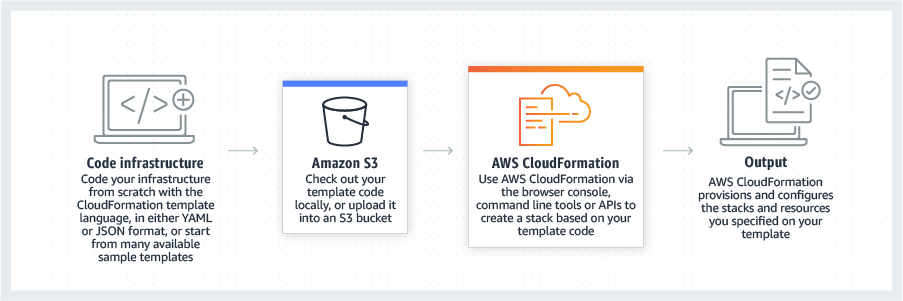
**(Correct)**

**Explanation**

Correct option:

**Create a cross-stack reference and use the Export output field to flag the value of VPC from the network stack. Then use Fn::ImportValue intrinsic function to import the value of VPC into the web application stack**

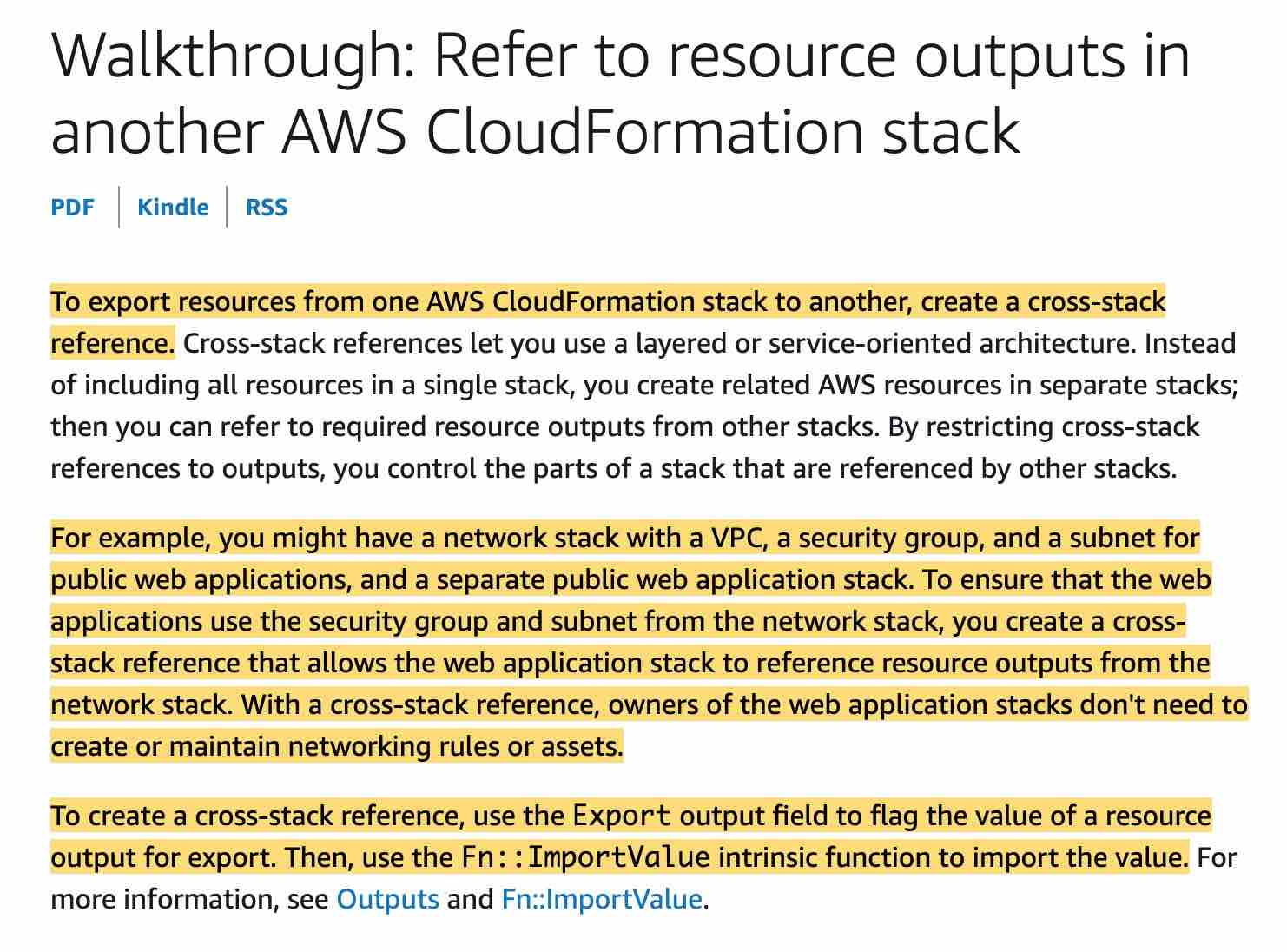
AWS CloudFormation gives developers and businesses an easy way to create a collection of related AWS and third-party resources and provision them in an orderly and predictable fashion.

How CloudFormation Works:  via - <https://aws.amazon.com/cloudformation/>

You can create a cross-stack reference to export resources from one AWS CloudFormation stack to another. For example, you might have a network stack with a VPC and subnets and a separate public web application stack. To use the security group and subnet from the network stack, you can create a cross-stack reference that allows the web application stack to reference resource outputs from the network stack. With a cross-stack reference, owners of the web application stacks don't need to create or maintain networking rules or assets.

To create a cross-stack reference, use the Export output field to flag the value of a resource output for export. Then, use the Fn::ImportValue intrinsic function to import the value.

You cannot use the Ref intrinsic function to import the value.

 via - <https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/walkthrough-crossstackref.html>

Incorrect options:

**Create a cross-stack reference and use the Outputs output field to flag the value of VPC from the network stack. Then use Fn::ImportValue intrinsic function to import the value of VPC into the web application stack**

**Create a cross-stack reference and use the Outputs output field to flag the value of VPC from the network stack. Then use Ref intrinsic function to reference the value of VPC into the web application stack**

**Create a cross-stack reference and use the Export output field to flag the value of VPC from the network stack. Then use Ref intrinsic function to reference the value of VPC into the web application stack**

These three options contradict the details provided in the explanation above, so these options are not correct.

References:

<https://aws.amazon.com/cloudformation/>

<https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/walkthrough-crossstackref.html>

Bottom of Form

Top of Form

Question 54: Skipped

**You are working for a shipping company that is automating the creation of ECS clusters with an Auto Scaling Group using an AWS CloudFormation template that accepts cluster name as its parameters. Initially, you launch the template with input value 'MainCluster', which deployed five instances across two availability zones. The second time, you launch the template with an input value 'SecondCluster'. However, the instances created in the second run were also launched in 'MainCluster' even after specifying a different cluster name.**

**What is the root cause of this issue?**

* ​

The security groups on the EC2 instance are pointing to the wrong ECS cluster

* ​

The ECS agent Docker image must be re-built to connect to the other clusters

* ​

The cluster name Parameter has not been updated in the file /etc/ecs/ecs.config during bootstrap

**(Correct)**

* ​

The EC2 instance is missing IAM permissions to join the other clusters

**Explanation**

Correct option:

**The cluster name Parameter has not been updated in the file /etc/ecs/ecs.config during bootstrap** - In the ecs.config file you have to configure the parameter ECS\_CLUSTER='your\_cluster\_name' to register the container instance with a cluster named 'your\_cluster\_name'.

Sample config for ECS Container Agent:  via - <https://docs.aws.amazon.com/AmazonECS/latest/developerguide/bootstrap_container_instance.html>

Incorrect options:

**The EC2 instance is missing IAM permissions to join the other clusters** - EC2 instances are getting registered to the first cluster, so permissions are not an issue here and hence this statement is an incorrect choice for the current use case.

**The ECS agent Docker image must be re-built to connect to the other clusters** - Since the first set of instances got created from the template without any issues, there is no issue with the ECS agent here.

**The security groups on the EC2 instance are pointing to the wrong ECS cluster** - Security groups govern the rules about the incoming network traffic to your ECS containers. The issue here is not about user access and hence is a wrong choice for the current use case.

References:

<https://docs.aws.amazon.com/AmazonECS/latest/developerguide/bootstrap_container_instance.html>

<https://docs.aws.amazon.com/AmazonECS/latest/developerguide/launch_container_instance.html>

Bottom of Form

Top of Form

Question 55: Skipped

**Your company has embraced cloud-native microservices architectures. New applications must be dockerized and stored in a registry service offered by AWS. The architecture should support dynamic port mapping and support multiple tasks from a single service on the same container instance. All services should run on the same EC2 instance.**

**Which of the following options offers the best-fit solution for the given use-case?**

* ​

Classic Load Balancer + ECS

* ​

Classic Load Balancer + Beanstalk

* ​

Application Load Balancer + Beanstalk

* ​

Application Load Balancer + ECS

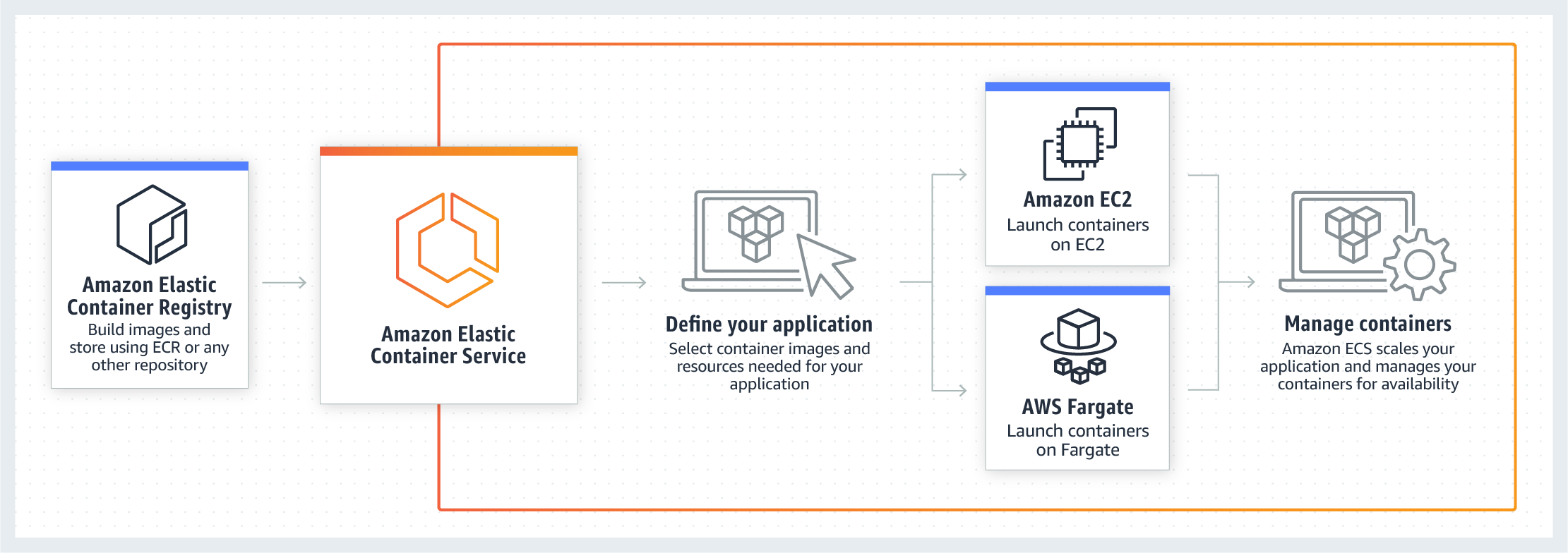
**(Correct)**

**Explanation**

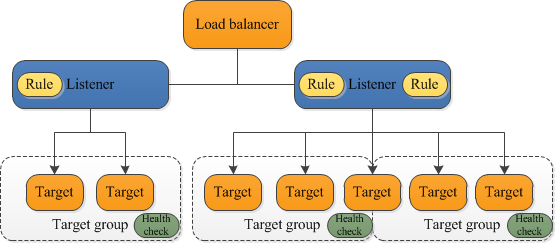
Correct option:

**Application Load Balancer + ECS**

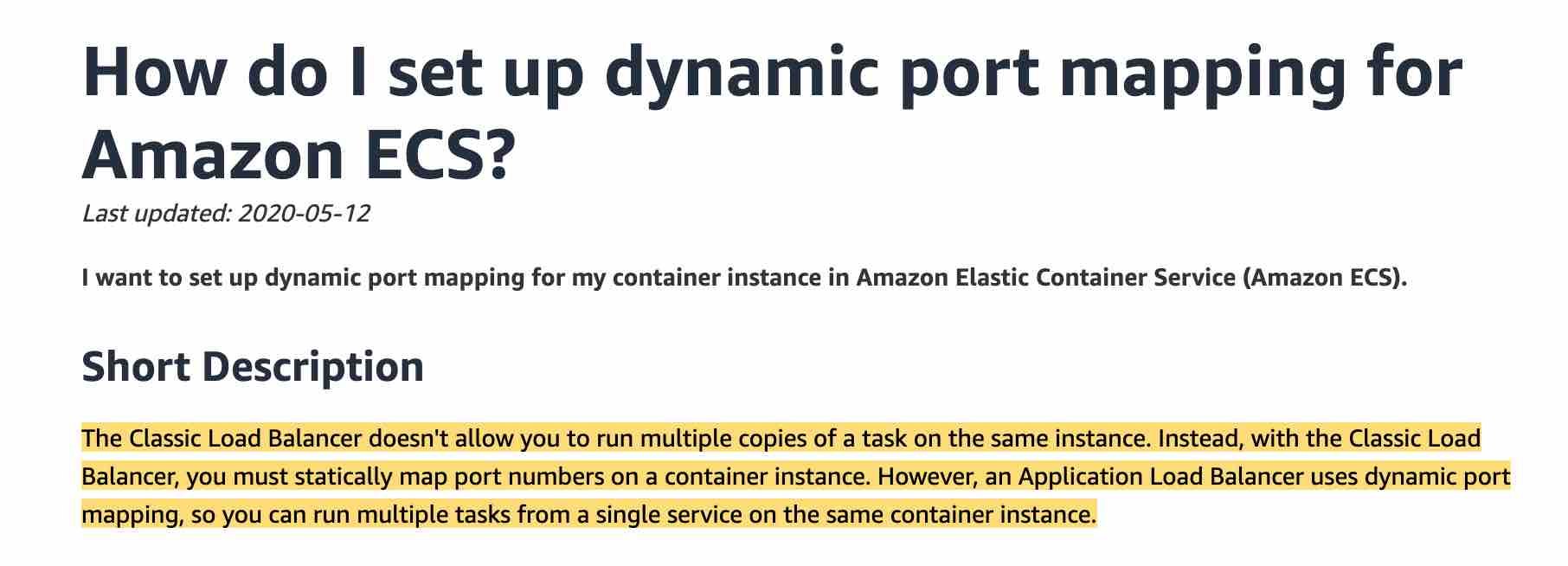
Amazon Elastic Container Service (Amazon ECS) is a highly scalable, fast, container management service that makes it easy to run, stop, and manage Docker containers on a cluster. You can host your cluster on a serverless infrastructure that is managed by Amazon ECS by launching your services or tasks using the Fargate launch type. For more control over your infrastructure, you can host your tasks on a cluster of Amazon Elastic Compute Cloud (Amazon EC2) instances that you manage by using the EC2 launch type.

 via - <https://aws.amazon.com/ecs/>

An Application load balancer distributes incoming application traffic across multiple targets, such as EC2 instances, in multiple Availability Zones. A listener checks for connection requests from clients, using the protocol and port that you configure. The rules that you define for a listener determine how the load balancer routes requests to its registered targets. Each rule consists of a priority, one or more actions, and one or more conditions.

 via - <https://docs.aws.amazon.com/elasticloadbalancing/latest/application/introduction.html>

When you deploy your services using Amazon Elastic Container Service (Amazon ECS), you can use dynamic port mapping to support multiple tasks from a single service on the same container instance. Amazon ECS manages updates to your services by automatically registering and deregistering containers with your target group using the instance ID and port for each container.

 via - <https://aws.amazon.com/premiumsupport/knowledge-center/dynamic-port-mapping-ecs>

Incorrect options:

**Classic Load Balancer + Beanstalk** - The Classic Load Balancer doesn't allow you to run multiple copies of a task on the same instance. Instead, with the Classic Load Balancer, you must statically map port numbers on a container instance. So this option is ruled out.

**Application Load Balancer + Beanstalk** - You can create docker environments that support multiple containers per Amazon EC2 instance with a multi-container Docker platform for Elastic Beanstalk. However, ECS gives you finer control.

**Classic Load Balancer + ECS** - The Classic Load Balancer doesn't allow you to run multiple copies of a task in the same instance. Instead, with the Classic Load Balancer, you must statically map port numbers on a container instance. So this option is ruled out.

References:

<https://aws.amazon.com/premiumsupport/knowledge-center/dynamic-port-mapping-ecs>

<https://docs.aws.amazon.com/elasticloadbalancing/latest/application/tutorial-target-ecs-containers.html>

<https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/create_deploy_docker_ecs.html>

Bottom of Form

Top of Form

Question 56: Skipped

**Your e-commerce company needs to improve its software delivery process and is moving away from the waterfall methodology. You decided that every application should be built using the best CI/CD practices and every application should be packaged and deployed as a Docker container. The Docker images should be stored in ECR and pushed with AWS CodePipeline and AWS CodeBuild.**

**When you attempt to do this, the last step fails with an authorization issue. What is the most likely issue?**

* ​

The IAM permissions are wrong for the CodeBuild service

**(Correct)**

* ​

The ECS instances are misconfigured and must contain additional data in /etc/ecs/ecs.config

* ​

CodeBuild cannot talk to ECR because of security group issues

* ​

The ECR repository is stale, you must delete and re-create it

**Explanation**

Correct option:

**The IAM permissions are wrong for the CodeBuild service**

You can push your Docker or Open Container Initiative (OCI) images to an Amazon ECR repository with the docker push command.

Amazon ECR users require permission to call ecr:GetAuthorizationToken before they can authenticate to a registry and push or pull any images from any Amazon ECR repository. Amazon ECR provides several managed policies to control user access at varying levels

Incorrect options:

**The ECR repository is stale, you must delete and re-create it** - You can delete a repository when you are done using it, stale is not a concept within ECR. This option has been added as a distractor.

**CodeBuild cannot talk to ECR because of security group issues** - A security group acts as a virtual firewall at the instance level and it is not related to pushing Docker images, so this option does not fit the given use-case.

**The ECS instances are misconfigured and must contain additional data in /etc/ecs/ecs.config** - The error Authorization is an indication that there is an access issue, therefore you should not look at your configuration first but rather permissions.

References:

<https://docs.aws.amazon.com/AmazonECR/latest/userguide/docker-push-ecr-image.html>

<https://docs.aws.amazon.com/AmazonECR/latest/userguide/ecr_managed_policies.html>

Bottom of Form

Top of Form

Question 57: Skipped

**A multi-national company maintains separate AWS accounts for different verticals in their organization. The project manager of a team wants to migrate the Elastic Beanstalk environment from Team A's AWS account into Team B's AWS account. As a Developer, you have been roped in to help him in this process.**

**Which of the following will you suggest?**

* ​

Create a saved configuration in Team A's account and configure it to Export. Now, log into Team B's account and choose the Import option. Here, you need to specify the name of the saved configuration and allow the system to create the new application. This takes a little time based on the Regions the two accounts belong to

* ​

Create a saved configuration in Team A's account and download it to your local machine. Make the account-specific parameter changes and upload to the S3 bucket in Team B's account. From Elastic Beanstalk console, create an application from 'Saved Configurations'

**(Correct)**

* ​

It is not possible to migrate Elastic Beanstalk environment from one AWS account to the other

* ​

Create an export configuration from the Elastic Beanstalk console from Team A's account. This configuration has to be shared with the IAM Role of Team B's account. The import option of Team B's account will show the saved configuration, that can be used to create a new Beanstalk application

**Explanation**

Correct option:

**Create a saved configuration in Team A's account and download it to your local machine. Make the account-specific parameter changes and upload to the S3 bucket in Team B's account. From Elastic Beanstalk console, create an application from 'Saved Configurations** - You must use saved configurations to migrate an Elastic Beanstalk environment between AWS accounts. You can save your environment's configuration as an object in Amazon Simple Storage Service (Amazon S3) that can be applied to other environments during environment creation, or applied to a running environment. Saved configurations are YAML formatted templates that define an environment's platform version, tier, configuration option settings, and tags.

Download the saved configuration to your local machine. Change your account-specific parameters in the downloaded configuration file, and then save the changes. For example, change the key pair name, subnet ID, or application name (such as application-b-name). Upload the saved configuration from your local machine to an S3 bucket in Team B's account. From this account, create a new Beanstalk application by choosing 'Saved Configurations' from the navigation panel.

Incorrect options:

**Create a saved configuration in Team A's account and configure it to Export. Now, log into Team B's account and choose the Import option. Here, you need to specify the name of the saved configuration and allow the system to create the new application. This takes a little time based on the Regions the two accounts belong to** - There is no direct Export and Import option for migrating Elastic Beanstalk configurations.

**It is not possible to migrate Elastic Beanstalk environment from one AWS account to the other** - This is an incorrect statement.

**Create an export configuration from the Elastic Beanstalk console from Team A's account. This configuration has to be shared with the IAM Role of Team B's account. The import option of the Team B's account will show the saved configuration, that can be used to create a new Beanstalk application** - This contradicts the explanation provided earlier.

References:

<https://aws.amazon.com/premiumsupport/knowledge-center/elastic-beanstalk-migration-accounts/>

<https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/environment-configuration-savedconfig.html>

Bottom of Form

Top of Form

Question 58: Skipped

**You are a developer working at a cloud company that embraces serverless. You have performed your initial deployment and would like to work towards adding API Gateway stages and associate them with existing deployments. Your stages will include prod, test, and dev and will need to match a Lambda function variant that can be updated over time.**

**Which of the following features must you add to achieve this? (select two)**

* ​

Lambda Aliases

**(Correct)**

* ​

Stage Variables

**(Correct)**

* ​

Mapping Templates

* ​

Lambda X-Ray integration

* ​

Lambda Versions

**Explanation**

Correct options:

**Stage Variables**

Stage variables are name-value pairs that you can define as configuration attributes associated with a deployment stage of an API. They act like environment variables and can be used in your API setup and mapping templates. With deployment stages in API Gateway, you can manage multiple release stages for each API, such as alpha, beta, and production. Using stage variables you can configure an API deployment stage to interact with different backend endpoints.

For example, your API can pass a GET request as an HTTP proxy to the backend web host (for example, http://example.com). In this case, the backend web host is configured in a stage variable so that when developers call your production endpoint, API Gateway calls example.com. When you call your beta endpoint, API Gateway uses the value configured in the stage variable for the beta stage and calls a different web host (for example, beta.example.com).

**Lambda Aliases**

A Lambda alias is like a pointer to a specific Lambda function version. Users can access the function version using the alias ARN.

Lambda Aliases allow you to create a "mutable" Lambda version that points to whatever version you want in the backend. This allows you to have a "dev", "test", prod" Lambda alias that can remain stable over time.

Incorrect options:

**Lambda Versions** - Versions are immutable and cannot be updated over time. So this option is not correct.

**Lambda X-Ray integration** - This is good for tracing and debugging requests so it can be looked at as a good option for troubleshooting issues in the future. This is not the right fit for the given use-case.

**Mapping Templates** - Mapping template overrides provides you with the flexibility to perform many-to-one parameter mappings; override parameters after standard API Gateway mappings have been applied; conditionally map parameters based on body content or other parameter values; programmatically create new parameters on the fly, and override status codes returned by your integration endpoint. This is not the right fit for the given use-case.

References:

<https://docs.aws.amazon.com/apigateway/latest/developerguide/stage-variables.html>

<https://docs.aws.amazon.com/lambda/latest/dg/configuration-aliases.html>

<https://docs.aws.amazon.com/apigateway/latest/developerguide/apigateway-override-request-response-parameters.html>

Bottom of Form

Top of Form

Question 59: Skipped

**A large firm stores its static data assets on Amazon S3 buckets. Each service line of the firm has its own AWS account. For a business use case, the Finance department needs to give access to their S3 bucket's data to the Human Resources department.**

**Which of the below options is NOT feasible for cross-account access of S3 bucket objects?**

* ​

Use Resource-based Access Control List (ACL) and IAM policies for programmatic-only access to S3 bucket objects

* ​

Use IAM roles and resource-based policies delegate access across accounts within different partitions via programmatic access only

**(Correct)**

* ​

Use Resource-based policies and AWS Identity and Access Management (IAM) policies for programmatic-only access to S3 bucket objects

* ​

Use Cross-account IAM roles for programmatic and console access to S3 bucket objects

**Explanation**

Correct option:

**Use IAM roles and resource-based policies delegate access across accounts within different partitions via programmatic access only** - This statement is incorrect and hence the right choice for this question. IAM roles and resource-based policies delegate access across accounts only within a single partition. For example, assume that you have an account in US West (N. California) in the standard aws partition. You also have an account in China (Beijing) in the aws-cn partition. You can't use an Amazon S3 resource-based policy in your account in China (Beijing) to allow access for users in your standard AWS account.

Incorrect options:

**Use Resource-based policies and AWS Identity and Access Management (IAM) policies for programmatic-only access to S3 bucket objects** - Use bucket policies to manage cross-account control and audit the S3 object's permissions. If you apply a bucket policy at the bucket level, you can define who can access (Principal element), which objects they can access (Resource element), and how they can access (Action element). Applying a bucket policy at the bucket level allows you to define granular access to different objects inside the bucket by using multiple policies to control access. You can also review the bucket policy to see who can access objects in an S3 bucket.

**Use Resource-based Access Control List (ACL) and IAM policies for programmatic-only access to S3 bucket objects** - Use object ACLs to manage permissions only for specific scenarios and only if ACLs meet your needs better than IAM and S3 bucket policies. Amazon S3 ACLs allow users to define only the following permissions sets: READ, WRITE, READ\_ACP, WRITE\_ACP, and FULL\_CONTROL. You can use only an AWS account or one of the predefined Amazon S3 groups as a grantee for the Amazon S3 ACL.

**Use Cross-account IAM roles for programmatic and console access to S3 bucket objects** - Not all AWS services support resource-based policies. This means that you can use cross-account IAM roles to centralize permission management when providing cross-account access to multiple services. Using cross-account IAM roles simplifies provisioning cross-account access to S3 objects that are stored in multiple S3 buckets, removing the need to manage multiple policies for S3 buckets. This method allows cross-account access to objects that are owned or uploaded by another AWS account or AWS services. If you don't use cross-account IAM roles, the object ACL must be modified.

References:

<https://docs.aws.amazon.com/AmazonS3/latest/dev/example-walkthroughs-managing-access-example3.html>

<https://docs.aws.amazon.com/IAM/latest/UserGuide/tutorial_cross-account-with-roles.html>

<https://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles_compare-resource-policies.html>

<https://aws.amazon.com/premiumsupport/knowledge-center/cross-account-access-s3/>

Bottom of Form

Top of Form

Question 60: Skipped

**The development team at an e-commerce company wants to run a serverless data store service on two docker containers using shared memory.**

**Which of the following ECS configurations can be used to facilitate this use-case?**

* ​

Put the two containers into two separate task definitions using a Fargate Launch Type

* ​

Put the two containers into two separate task definitions using an EC2 Launch Type

* ​

Put the two containers into a single task definition using an EC2 Launch Type

* ​

Put the two containers into a single task definition using a Fargate Launch Type

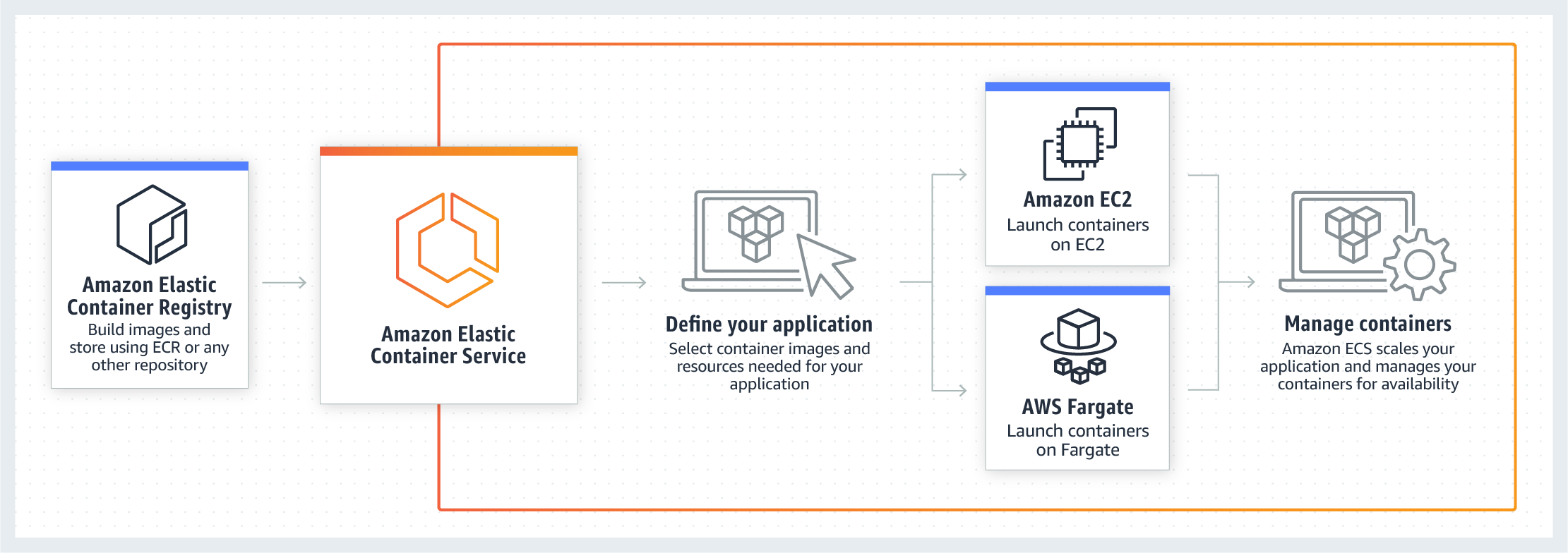
**(Correct)**

**Explanation**

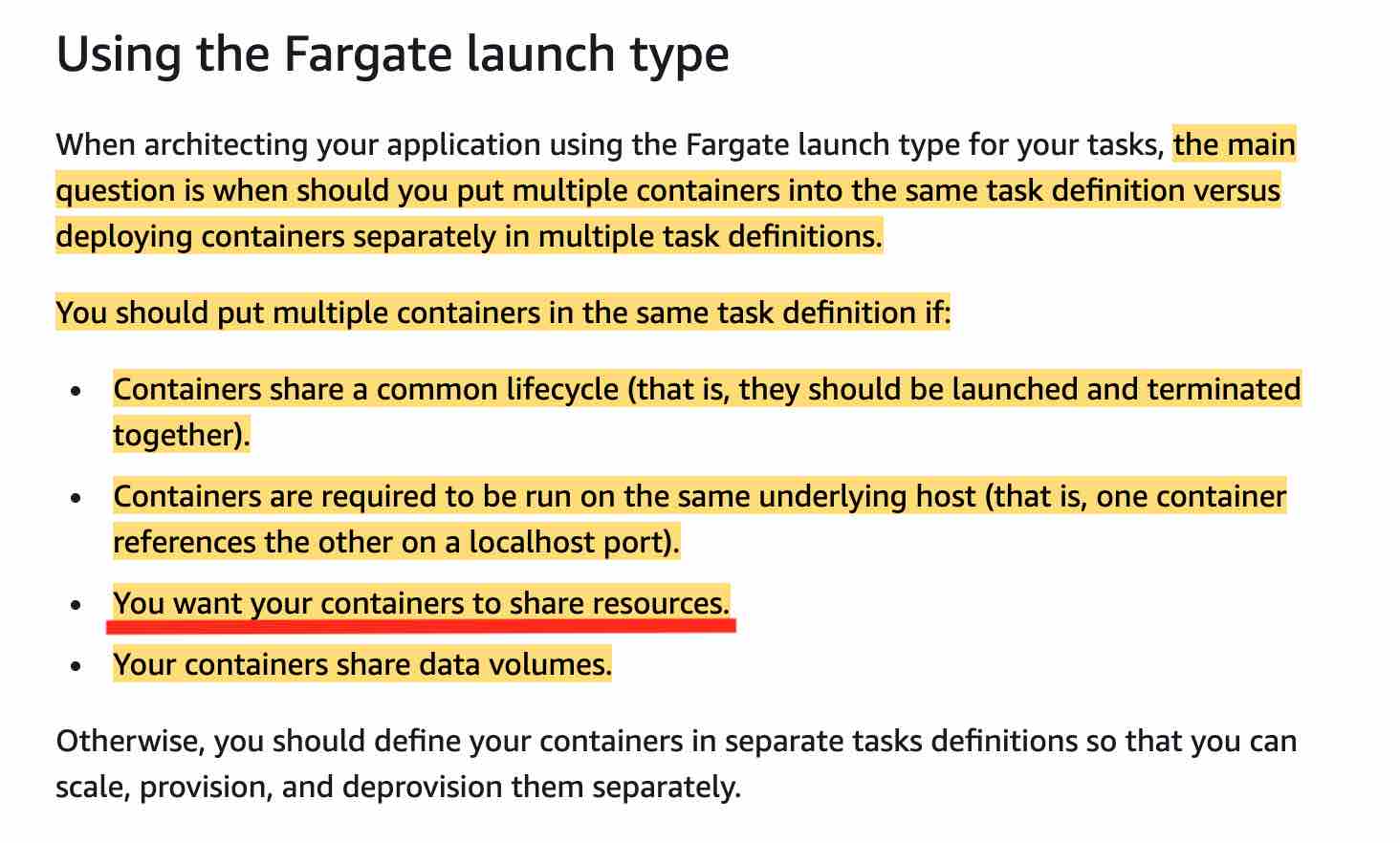
Correct option:

**Put the two containers into a single task definition using a Fargate Launch Type**

Amazon Elastic Container Service (Amazon ECS) is a highly scalable, fast, container management service that makes it easy to run, stop, and manage Docker containers on a cluster. You can host your cluster on a serverless infrastructure that is managed by Amazon ECS by launching your services or tasks using the Fargate launch type. For more control over your infrastructure, you can host your tasks on a cluster of Amazon Elastic Compute Cloud (Amazon EC2) instances that you manage by using the EC2 launch type.

 via - <https://aws.amazon.com/ecs/>

As the development team is looking for a serverless data store service, therefore the two containers should be launched into a single task definition using a Fargate Launch Type. Using a single task definition allows the two containers to share memory. Please see these use-cases for Fargate Launch type when you should put multiple containers into the same task definition:

 via - <https://docs.aws.amazon.com/AmazonECS/latest/developerguide/application_architecture.html>

For a deep-dive on understanding how Amazon ECS manages CPU and memory resources, please review this excellent blog- <https://aws.amazon.com/blogs/containers/how-amazon-ecs-manages-cpu-and-memory-resources/>

Incorrect options:

**Put the two containers into two separate task definitions using a Fargate Launch Type** - This option contradicts the details provided in the explanation above, so this option is ruled out.

**Put the two containers into two separate task definitions using an EC2 Launch Type**

**Put the two containers into a single task definition using an EC2 Launch Type**

As the development team is looking for a serverless data store service, therefore EC2 Launch Type is ruled out. So both these options are incorrect.

References:

<https://docs.aws.amazon.com/AmazonECS/latest/developerguide/application_architecture.html>

<https://aws.amazon.com/blogs/containers/how-amazon-ecs-manages-cpu-and-memory-resources/>

Bottom of Form

Top of Form

Question 61: Skipped

**You have migrated an on-premise SQL Server database to an Amazon Relational Database Service (RDS) database attached to a VPC inside a private subnet. Also, the related Java application, hosted on-premise, has been moved to an Amazon Lambda function.**

**Which of the following should you implement to connect AWS Lambda function to its RDS instance?**

* ​

Use Lambda layers to connect to the internet and RDS separately

* ​

Configure lambda to connect to the public subnet that will give internet access and use Security Group to access RDS inside the private subnet

* ​

Configure Lambda to connect to VPC with private subnet and Security Group needed to access RDS

**(Correct)**

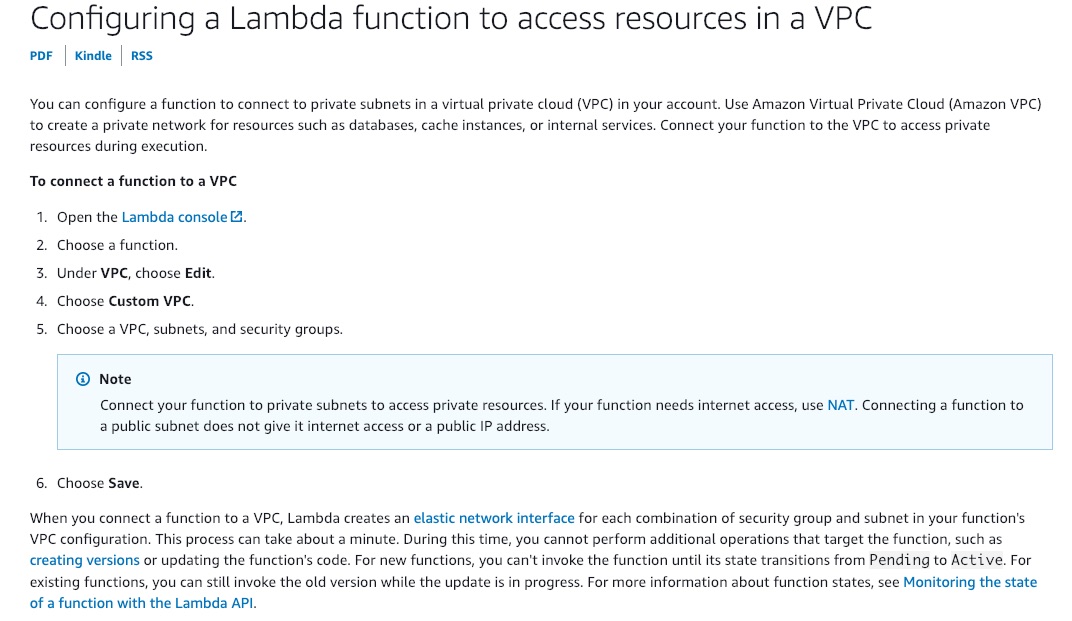
* ​

Use Environment variables to pass in the RDS connection string

**Explanation**

Correct option:

**Configure Lambda to connect to VPC with private subnet and Security Group needed to access RDS** - You can configure a Lambda function to connect to private subnets in a virtual private cloud (VPC) in your account. Use Amazon Virtual Private Cloud (Amazon VPC) to create a private network for resources such as databases, cache instances, or internal services. Connect your lambda function to the VPC to access private resources during execution. When you connect a function to a VPC, Lambda creates an elastic network interface for each combination of the security group and subnet in your function's VPC configuration. This is the right way of giving RDS access to Lambda.

Lambda VPC Config:  via - <https://docs.aws.amazon.com/lambda/latest/dg/configuration-vpc.html>

Incorrect options:

**Use Lambda layers to connect to the internet and RDS separately** - You can configure your Lambda function to pull in additional code and content in the form of layers. A layer is a ZIP archive that contains libraries, a custom runtime, or other dependencies. Layers will not help in configuring access to RDS instance and hence is an incorrect choice.

**Configure lambda to connect to the public subnet that will give internet access and use the Security Group to access RDS inside the private subnet** - This is an incorrect statement. Connecting a Lambda function to a public subnet does not give it internet access or a public IP address. To grant internet access to your function, its associated VPC must have a NAT gateway (or NAT instance) in a public subnet.

**Use Environment variables to pass in the RDS connection string** - You can use environment variables to store secrets securely and adjust your function's behavior without updating code. You can use environment variables to exchange data with RDS, but you will still need access to RDS, which is not possible with just environment variables.

References:

<https://docs.aws.amazon.com/lambda/latest/dg/configuration-vpc.html>

<https://docs.aws.amazon.com/lambda/latest/dg/configuration-layers.html>

<https://docs.aws.amazon.com/lambda/latest/dg/configuration-envvars.html>

<https://aws.amazon.com/premiumsupport/knowledge-center/internet-access-lambda-function/>

Bottom of Form

Top of Form

Question 62: Skipped

**A Developer is configuring Amazon EC2 Auto Scaling group to scale dynamically.**

**Which metric below is NOT part of Target Tracking Scaling Policy?**

* ​

ALBRequestCountPerTarget

* ​

ASGAverageNetworkOut

* ​

ASGAverageCPUUtilization

* ​

ApproximateNumberOfMessagesVisible

**(Correct)**

**Explanation**

Correct option:

**ApproximateNumberOfMessagesVisible** - This is a CloudWatch Amazon SQS queue metric. The number of messages in a queue might not change proportionally to the size of the Auto Scaling group that processes messages from the queue. Hence, this metric does not work for target tracking.

Incorrect options:

With target tracking scaling policies, you select a scaling metric and set a target value. Amazon EC2 Auto Scaling creates and manages the CloudWatch alarms that trigger the scaling policy and calculates the scaling adjustment based on the metric and the target value.

It is important to note that a target tracking scaling policy assumes that it should scale out your Auto Scaling group when the specified metric is above the target value. You cannot use a target tracking scaling policy to scale out your Auto Scaling group when the specified metric is below the target value.

**ASGAverageCPUUtilization** - This is a predefined metric for target tracking scaling policy. This represents the Average CPU utilization of the Auto Scaling group.

**ASGAverageNetworkOut** - This is a predefined metric for target tracking scaling policy. This represents the Average number of bytes sent out on all network interfaces by the Auto Scaling group.

**ALBRequestCountPerTarget** - This is a predefined metric for target tracking scaling policy. This represents the Number of requests completed per target in an Application Load Balancer target group.

Reference:

<https://docs.aws.amazon.com/autoscaling/ec2/userguide/as-scaling-target-tracking.html>

Bottom of Form

Top of Form

Question 63: Skipped

**A Developer at a company is working on a CloudFormation template to set up resources. Resources will be defined using code and provisioned based on certain conditions.**

**Which section of a CloudFormation template does not allow for conditions?**

* ​

Conditions

* ​

Outputs

* ​

Parameters

**(Correct)**

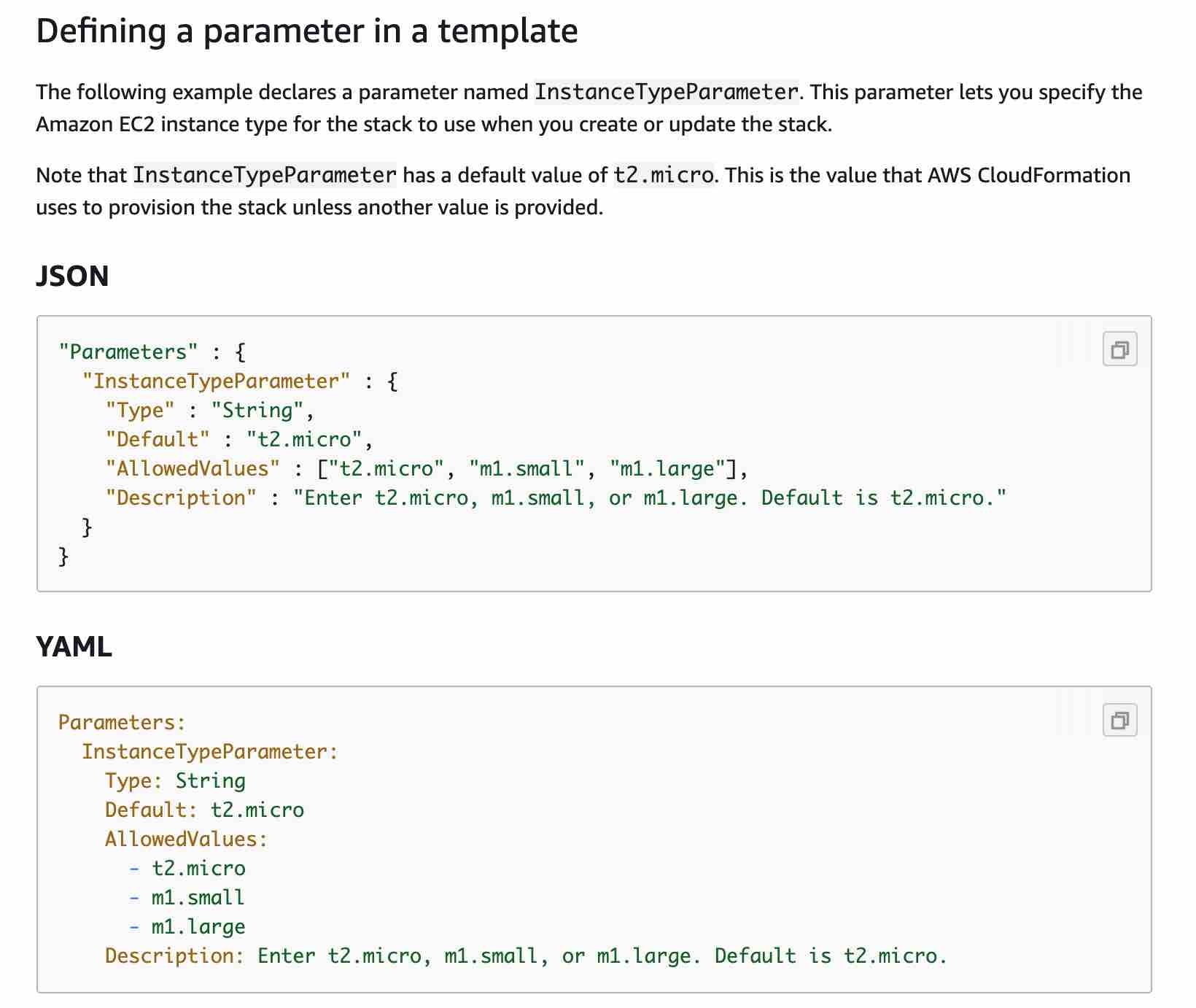
* ​

Resources

**Explanation**

Correct option:

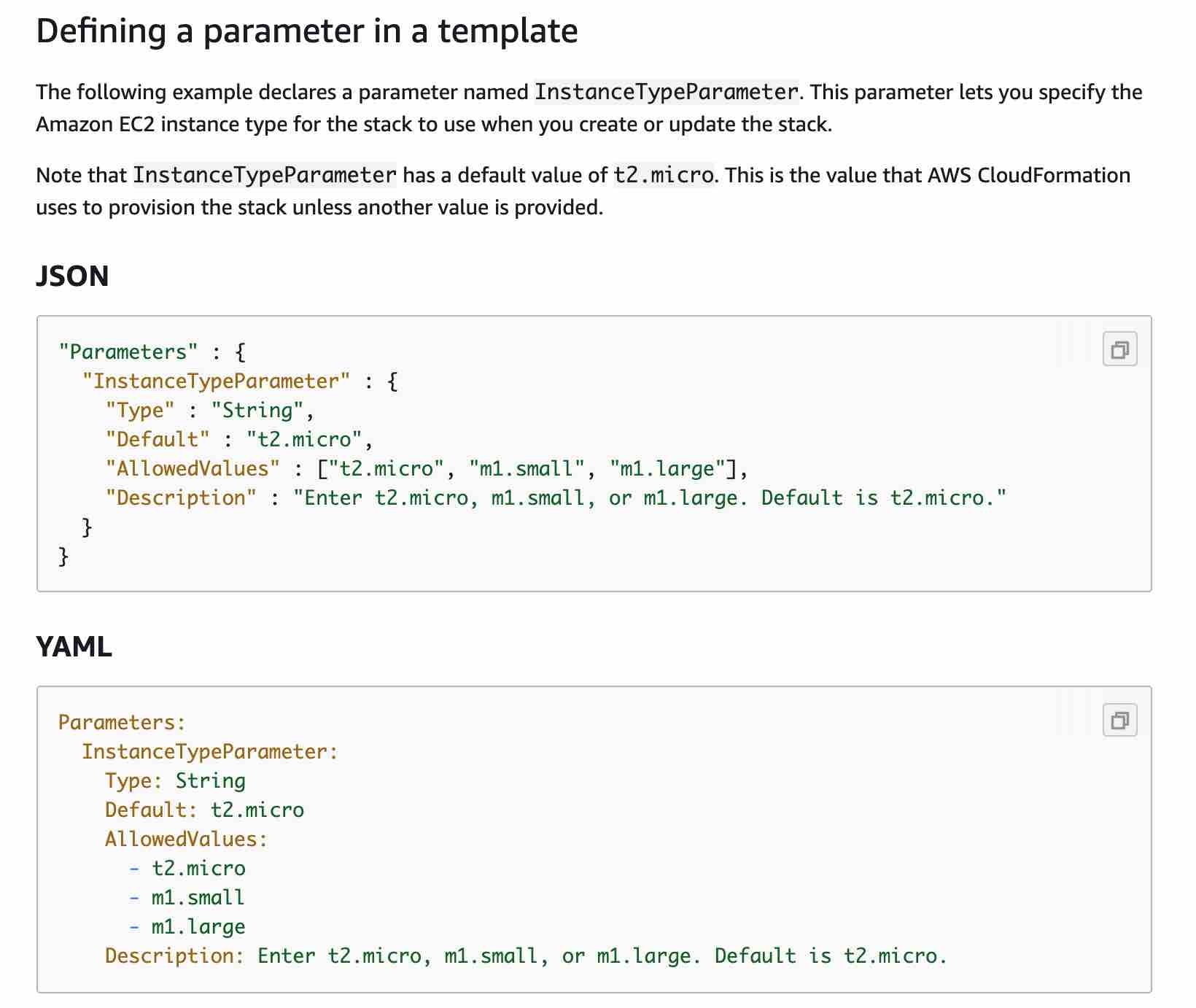
**Parameters**

Parameters enable you to input custom values to your CloudFormation template each time you create or update a stack. Please see this note to understand how to define a parameter in a template:  via - <https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/parameters-section-structure.html>

The optional Conditions section contains statements that define the circumstances under which entities are created or configured. For example, you can create a condition and then associate it with a resource or output so that AWS CloudFormation only creates the resource or output if the condition is true.

You might use conditions when you want to reuse a template that can create resources in different contexts, such as a test environment versus a production environment. In your template, you can add an EnvironmentType input parameter, which accepts either prod or test as inputs. For the production environment, you might include Amazon EC2 instances with certain capabilities; however, for the test environment, you want to use reduced capabilities to save money.

Conditions cannot be used within the Parameters section. After you define all your conditions, you can associate them with resources and resource properties only in the Resources and Outputs sections of a template.

Please review this note for more details:  via - <https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/conditions-section-structure.html>

Please visit https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/parameters-section-structure.html for more information on the parameter structure.

Incorrect options:

**Resources** - Resources section describes the resources that you want to provision in your AWS CloudFormation stacks. You can associate conditions with the resources that you want to conditionally create.

**Conditions** - You actually define conditions in this section of the CloudFormation template

**Outputs** - The optional Outputs section declares output values that you can import into other stacks (to create cross-stack references), return in response (to describe stack calls), or view on the AWS CloudFormation console. For example, you can output the S3 bucket name for a stack to make the bucket easier to find. You can associate conditions with the outputs that you want to conditionally create.

References:

<https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/parameters-section-structure.html>

<https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/conditions-section-structure.html>

Bottom of Form

Top of Form

Question 64: Skipped

**A retail company manages its IT infrastructure on AWS Cloud via Elastic Beanstalk. The development team at the company is planning to deploy the next version with MINIMUM application downtime and the ability to rollback quickly in case deployment goes wrong.**

**As a Developer Associate, which of the following options would you recommend to the development team?**

* ​

Deploy the new application version using 'Rolling with additional batch' deployment policy

* ​

Deploy the new application version using 'All at once' deployment policy

* ​

Deploy the new application version using 'Rolling' deployment policy

* ​

Deploy the new version to a separate environment via Blue/Green Deployment, and then swap Route 53 records of the two environments to redirect traffic to the new version

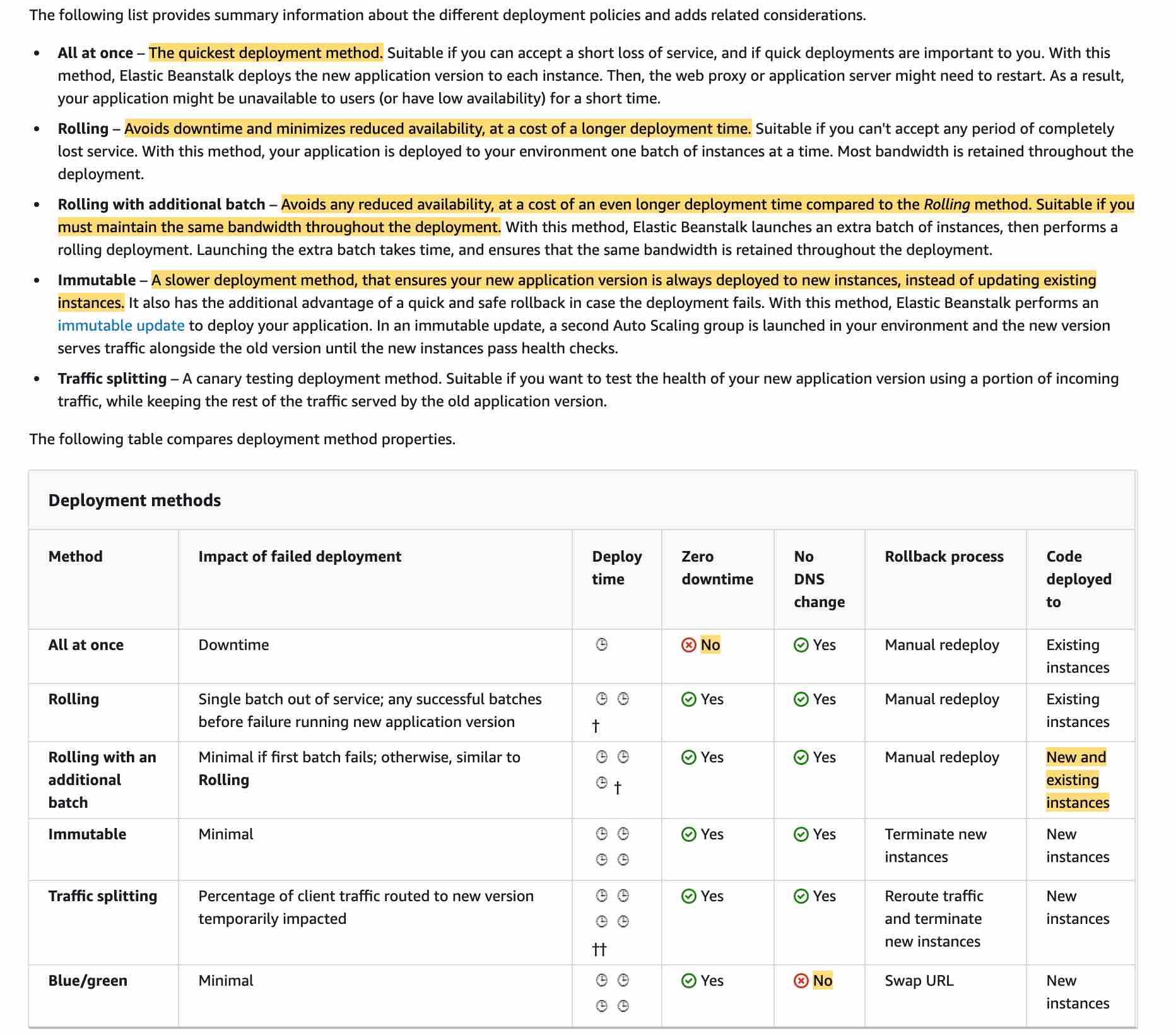
**(Correct)**

**Explanation**

Correct option:

**Deploy the new version to a separate environment via Blue/Green Deployment, and then swap Route 53 records of the two environments to redirect traffic to the new version**

With deployment policies such as 'All at once', AWS Elastic Beanstalk performs an in-place update when you update your application versions and your application can become unavailable to users for a short period of time. You can avoid this downtime by performing a blue/green deployment, where you deploy the new version to a separate environment, and then swap CNAMEs (via Route 53) of the two environments to redirect traffic to the new version instantly. In case of any deployment issues, the rollback process is very quick via swapping the URLs for the two environments.

Overview of Elastic Beanstalk Deployment Policies:  via - <https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/using-features.deploy-existing-version.html>

Incorrect options:

**Deploy the new application version using 'All at once' deployment policy** - Although 'All at once' is the quickest deployment method, but the application may become unavailable to users (or have low availability) for a short time. So this option is not correct.

**Deploy the new application version using 'Rolling' deployment policy** - This policy avoids downtime and minimizes reduced availability, at a cost of a longer deployment time. However rollback process is via manual redeploy, so it's not as quick as the Blue/Green deployment.

**Deploy the new application version using 'Rolling with additional batch' deployment policy** - This policy avoids any reduced availability, at a cost of an even longer deployment time compared to the Rolling method. Suitable if you must maintain the same bandwidth throughout the deployment. However rollback process is via manual redeploy, so it's not as quick as the Blue/Green deployment.

Reference:

<https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/using-features.deploy-existing-version.html>

Bottom of Form

Top of Form

Question 65: Skipped

**You are getting ready for an event to show off your Alexa skill written in JavaScript. As you are testing your voice activation commands you find that some intents are not invoking as they should and you are struggling to figure out what is happening. You included the following code console.log(JSON.stringify(this.event)) in hopes of getting more details about the request to your Alexa skill.**

**You would like the logs stored in an Amazon Simple Storage Service (S3) bucket named MyAlexaLog. How do you achieve this?**

* ​

Use CloudWatch integration feature with S3

**(Correct)**

* ​

Use CloudWatch integration feature with Glue

* ​

Use CloudWatch integration feature with Lambda

* ​

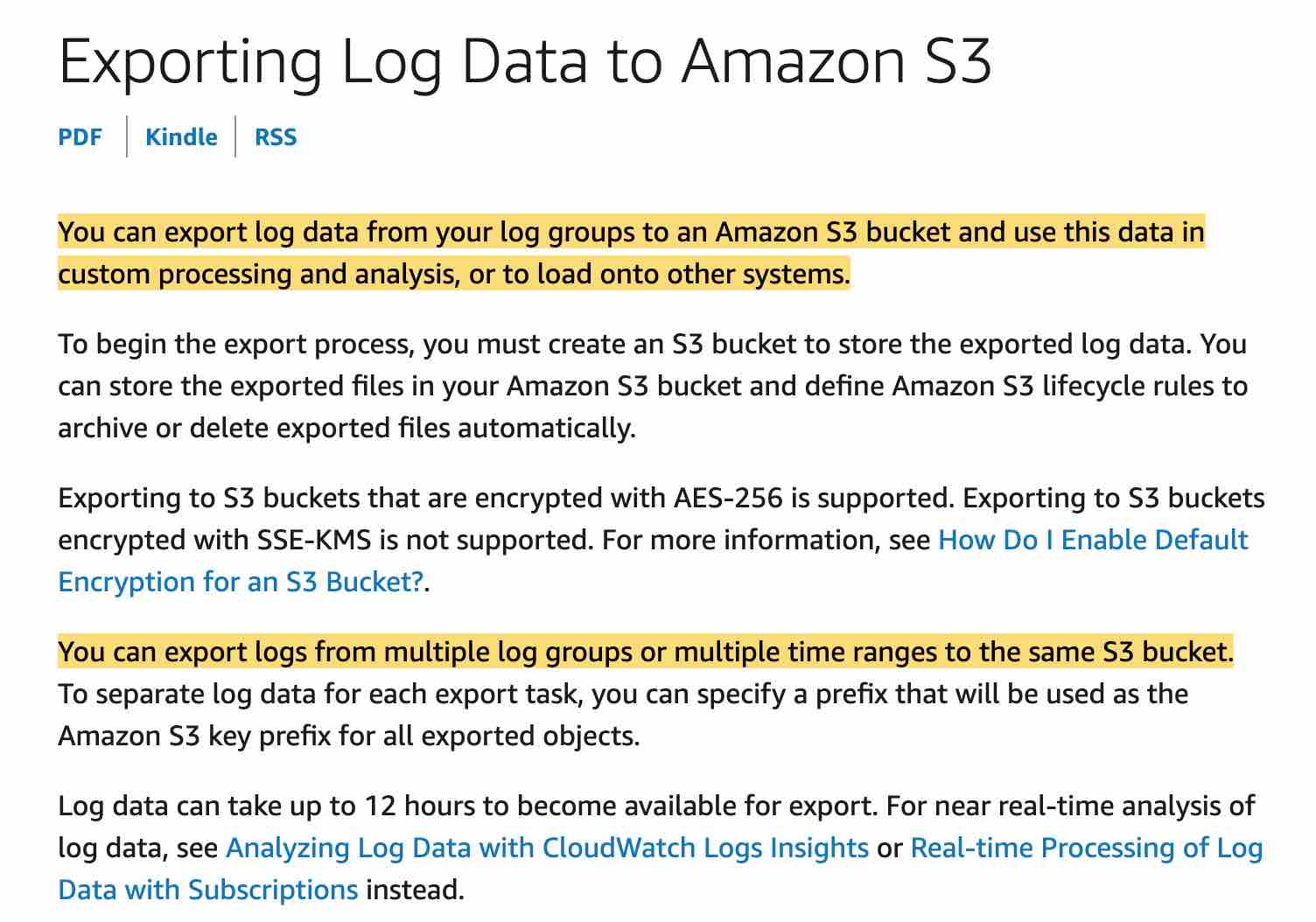
Use CloudWatch integration feature with Kinesis

**Explanation**

Correct option:

**Use CloudWatch integration feature with S3**

You can export log data from your CloudWatch log groups to an Amazon S3 bucket and use this data in custom processing and analysis, or to load onto other systems.

Exporting CloudWatch Log Data to Amazon S3:  via - <https://docs.aws.amazon.com/AmazonCloudWatch/latest/logs/S3Export.html>

Incorrect options:

**Use CloudWatch integration feature with Kinesis** - You can use both to do custom processing or analysis but with S3 you don't have to process anything. Instead, you configure the CloudWatch settings to send logs to S3.

**Use CloudWatch integration feature with Lambda** - You can use both to do custom processing or analysis but with S3 you don't have to process anything. Instead, you configure the CloudWatch settings to send logs to S3.

**Use CloudWatch integration feature with Glue** - AWS Glue is a fully managed extract, transform, and load (ETL) service that makes it easy for customers to prepare and load their data for analytics. Glue is not the right fit for the given use-case.

Reference:

<https://docs.aws.amazon.com/AmazonCloudWatch/latest/logs/S3Export.html>

Bottom of Form

Continue

Retake test