**AWS Certified Developer Associate - Results**

65 questions | 2 hours 10 minutes | 72% correct required to pass

Attempt 1: Passed! (72% required to pass)

72%correct(47/65)

1 hour 23 minutes

**Domains**

**AWS Security, Identity, & Compliance (12 questions)**

83%

17%

**AWS Database (8 questions)**

75%

25%

**AWS Application Integration (10 questions)**

80%

20%

**AWS Management & Governance (5 questions)**

100%

**AWS Networking & Content Delivery (7 questions)**

43%

57%

**AWS Compute (14 questions)**

57%

43%

**AWS Storage (3 questions)**

67%

33%

**AWS Developer Tools (6 questions)**

83%

17%

## AWS Certified Developer Associate - Results

Back to result overview

**New**

[Share feedback](https://survey.alchemer.com/s3/7719142/Practice-tests-learner-csat/?user_id=52182326&course_id=2981704&quiz_id=4852742)

**Attempt 1**

All domains

* 65 all
* 47 correct
* 18 incorrect
* 0 skipped
* 22 marked

Collapse all questions

**Question 1Incorrect**

A three-tier application is being migrated from an on-premises data center. The application includes an Apache Tomcat web tier, an application tier running on Linux, and a MySQL back end. A Developer must refactor the application to run on the AWS cloud. The cloud-based application must be fault tolerant and elastic.

How can the Developer refactor the web tier and application tier? (Select TWO.)

**Create an Amazon CloudFront distribution for the web tier**

**Implement an Elastic Load Balancer for the application tier**

**Your selection is incorrect**

**Use a multi-AZ Amazon RDS database for the back end using the MySQL engine**

**Correct selection**

**Create an Auto Scaling group of EC2 instances for both the web tier and application tier**

**Your selection is correct**

**Implement an Elastic Load Balancer for both the web tier and the application tier**

Overall explanation

The key requirements in this scenario are to add fault tolerances and elasticity to the web tier and application tier. Note that no specific requirements for the back end have been included.

To add elasticity to the web and application tiers the Developer should create Auto Scaling groups of EC2 instances. We know that the application tier runs on Linux and the web tier runs on Apache Tomcat (which could be on Linux or Windows). Therefore, these workloads are suitable for an ASG and this will ensure the number of instances dynamically scales out and in based on actual usage.

To add fault tolerance to the web and application tiers the Developer should add an Elastic Load Balancer. This will ensure that if the number of EC2 instances are changed by the ASG, the load balancer is able to distribute traffic to them. This also assists with elasticity.

**CORRECT:**"Create an Auto Scaling group of EC2 instances for both the web tier and application tier" is a correct answer.

**CORRECT:**"Implement an Elastic Load Balancer for both the web tier and the application tier" is also a correct answer.

**INCORRECT:** "Create an Amazon CloudFront distribution for the web tier" is incorrect as CloudFront is used for performance reasons, not elasticity or fault tolerance. You would use CloudFront to get content closer to end users around the world.

**INCORRECT:** "Use a multi-AZ Amazon RDS database for the back end using the MySQL engine" is incorrect as the question does not ask for fault tolerance of the back end, only the web tier and the application tier.

**INCORRECT:** "Implement an Elastic Load Balancer for the application tier" is incorrect. An Elastic Load Balancer should be implemented for both the web tier and the application tier as that is how we ensure fault tolerance and elasticity for both of those tiers.

**References:**

<https://docs.aws.amazon.com/elasticloadbalancing/latest/userguide/what-is-load-balancing.html>

<https://docs.aws.amazon.com/AmazonECS/latest/developerguide/service-auto-scaling.html>

**Save time with our AWS cheat sheets:**

<https://digitalcloud.training/aws-elastic-load-balancing-aws-elb/>

<https://digitalcloud.training/amazon-ec2-auto-scaling/>

**Domain**

AWS Compute

**Question 2Correct**

An application needs to read up to 100 items at a time from an Amazon DynamoDB. Each item is up to 100 KB in size and all attributes must be retrieved.

What is the BEST way to minimize latency?

**Use a Scan operation with pagination**

**Your answer is correct**

**Use BatchGetItem**

**Use GetItem and use a projection expression**

**Use a Query operation with a FilterExpression**

Overall explanation

The BatchGetItem operation returns the attributes of one or more items from one or more tables. You identify requested items by primary key.

A single operation can retrieve up to 16 MB of data, which can contain as many as 100 items. In order to minimize response latency, BatchGetItem retrieves items in parallel.

By default, BatchGetItem performs eventually consistent reads on every table in the request. If you want strongly consistent reads instead, you can set ConsistentRead to true for any or all tables.

**CORRECT:**"Use BatchGetItem" is the correct answer.

**INCORRECT:** "Use GetItem and use a projection expression" is incorrect as this will limit the attributes returned and will retrieve the items sequentially which results in more latency.

**INCORRECT:** "Use a Scan operation with pagination" is incorrect as a Scan operation is the least efficient way to retrieve the data as all items in the table are returned and then filtered. Pagination just breaks the results into pages.

**INCORRECT:** "Use a Query operation with a FilterExpression" is incorrect as this would limit the results that are returned.

**References:**

<https://docs.aws.amazon.com/amazondynamodb/latest/APIReference/API_BatchGetItem.html>

**Save time with our AWS cheat sheets:**

<https://digitalcloud.training/amazon-dynamodb/>

**Domain**

AWS Database

**Question 3Correct**

A Developer is creating an AWS Lambda function that generates a new file each time it runs. Each new file must be checked into an AWS CodeCommit repository hosted in the same AWS account.

How should the Developer accomplish this?

**Upload the new file to an Amazon S3 bucket. Create an AWS Step Function to accept S3 events. In the Step Function, add the new file to the repository**

**When the Lambda function starts, use the Git CLI to clone the repository. Check the new file into the cloned repository and push the change**

**After the new file is created in Lambda, use cURL to invoke the CodeCommit API. Send the file to the repository**

**Your answer is correct**

**Use an AWS SDK to instantiate a CodeCommit client. Invoke the put\_file method to add the file to the repository**

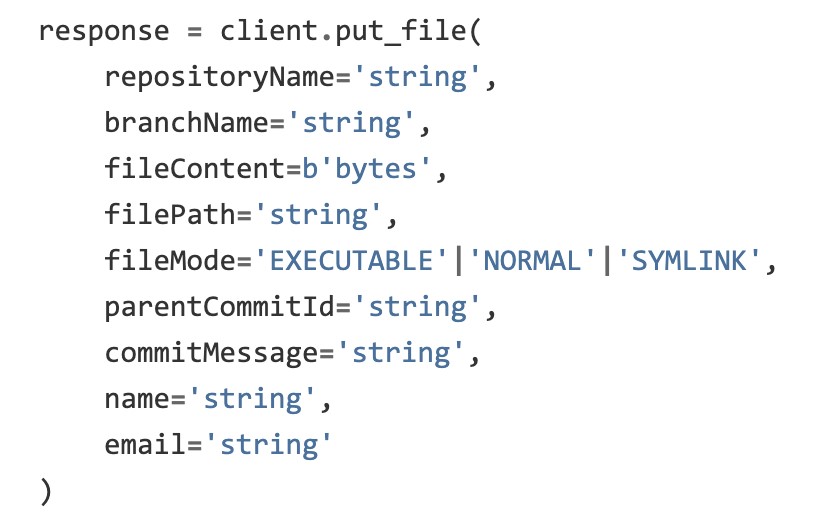
Overall explanation

The Developer can use the AWS SDK to instantiate a CodeCommit client. For instance, the code might include the following:



The client can then be used with put\_file which adds or updates a file in a branch in an AWS CodeCommit repository, and generates a commit for the addition in the specified branch.

The request syntax is as follows:



**CORRECT:**"Use an AWS SDK to instantiate a CodeCommit client. Invoke the put\_file method to add the file to the repository" is the correct answer.

**INCORRECT:** "When the Lambda function starts, use the Git CLI to clone the repository. Check the new file into the cloned repository and push the change" is incorrect as there is no need to clone a repository, a file just needs to be added to an existing repository.

**INCORRECT:** "After the new file is created in Lambda, use cURL to invoke the CodeCommit API. Send the file to the repository" is incorrect as a URL cannot be used to invoke a CodeCommit client and upload and check in the file.

**INCORRECT:** "Upload the new file to an Amazon S3 bucket. Create an AWS Step Function to accept S3 events. In the Step Function, add the new file to the repository" is incorrect as Step Functions is not triggered by S3 events.

**References:**

<https://boto3.amazonaws.com/v1/documentation/api/latest/reference/services/codecommit.html>

**Save time with our AWS cheat sheets:**

<https://digitalcloud.training/aws-developer-tools/>

**Domain**

AWS Developer Tools

**Question 4Correct**

An organization is developing a data processing application that is hosted on AWS Lambda and utilizes a PostgreSQL database on Amazon RDS. The security team mandates a policy that requires rotating database credentials every week.

What strategy should the developer adopt to manage the database credentials for the application?

**Store database credentials in Amazon S3 buckets with versioning enabled, rotating, and uploading new credentials weekly.**

**Enable IAM database authentication and manage weekly rotation manually.**

**Your answer is correct**

**Deploy AWS Secrets Manager to store database credentials and set up automatic weekly rotation.**

**Store the credentials in environment variables of the Lambda function and manually update them every week.**

Overall explanation

AWS Secrets Manager enables the creation, retrieval, and rotation of secrets like database credentials. Configuring automatic rotation weekly satisfies the security team's requirement, with the least manual effort.

**CORRECT:**"Deploy AWS Secrets Manager to store database credentials and set up automatic weekly rotation" is the correct answer (as explained above.)

**INCORRECT:** "Store database credentials in Amazon S3 buckets with versioning enabled, rotating and uploading new credentials weekly" is incorrect.

Storing credentials in Amazon S3 is not a secure practice as it does not have built-in mechanisms for rotating secrets.

**INCORRECT:** "Enable IAM database authentication and manage weekly rotation manually" is incorrect.

IAM database authentication allows you to use IAM users/roles to access your database, but it does not have an automatic mechanism for weekly rotation of IAM credentials.

**INCORRECT:** "Store the credentials in environment variables of the Lambda function and manually update them every week" is incorrect.

Storing credentials as environment variables in AWS Lambda functions could expose them to potential security risks, and manual rotation weekly would be labor-intensive.

**References:**

<https://docs.aws.amazon.com/secretsmanager/latest/userguide/rotate-secrets_managed.html>

**Save time with our AWS cheat sheets:**

<https://digitalcloud.training/aws-secrets-manager/>

**Domain**

AWS Security, Identity, & Compliance

**Question 5Correct**

A large quantity of sensitive data must be encrypted. A Developer will use a custom CMK to generate the encryption key. The key policy currently looks like this:

1. {
2. "Sid": "Allow Key Usage",
3. "Effect": "Allow",
4. "Principal": {"AWS": [
5. "arn:aws:iam::111122223333:user/CMKUser"
6. ]},
7. "Action": [
8. "kms:Encrypt",
9. "kms:Decrypt",
10. "kms:ReEncrypt\*",
11. "kms:DescribeKey"
12. ],
13. "Resource": "\*"
14. }

What API action must be added to the key policy?

**kms:EnableKey**

**kms:CreateKey**

**Your answer is correct**

**kms:GenerateDataKey**

**kms:GetKeyPolicy**

Overall explanation

A key policy is a document that uses [JSON (JavaScript Object Notation)](http://json.org/) to specify permissions. You can work with these JSON documents directly, or you can use the AWS Management Console to work with them using a graphical interface called the default view.

The key policy supplied with this question is missing the GenerateDataKey API action which is a permission that is required to generate a data encryption key. A data encryption key is required to encrypt large amounts of data as a CMK can only encrypt up to 4 KB.

The GenerateDataKey API Generates a unique symmetric data key. This operation returns a plaintext copy of the data key and a copy that is encrypted under a customer master key (CMK) that you specify. You can use the plaintext key to encrypt your data outside of AWS KMS and store the encrypted data key with the encrypted data.

**CORRECT:**"kms:GenerateDataKey" is the correct answer.

**INCORRECT:** "kms:EnableKey" is incorrect as this sets the key state of a customer master key (CMK) to enabled. It allows you to use the CMK for [cryptographic operations](https://docs.aws.amazon.com/kms/latest/developerguide/concepts.html#cryptographic-operations).

**INCORRECT:** "kms:CreateKey" is incorrect as this creates a unique customer managed [customer master key](https://docs.aws.amazon.com/kms/latest/developerguide/concepts.html#master-keys) (CMK) in your AWS account and Region. In this case the CMK already exists, the Developer needs to create a data encryption key.

**INCORRECT:** "kms:GetKeyPolicy" is incorrect as this simply gets a key policy attached to the specified customer master key (CMK).

**References:**

<https://docs.aws.amazon.com/kms/latest/developerguide/key-policies.html>

**Save time with our AWS cheat sheets:**

<https://digitalcloud.training/aws-kms/>

**Domain**

AWS Security, Identity, & Compliance

**Question 6Correct**

A Developer has setup an Amazon Kinesis Data Stream with 6 shards to ingest a maximum of 2000 records per second. An AWS Lambda function has been configured to process these records. In which order will these records be processed?

**The Developer can select exact order or reverse order using the GetRecords API**

**Your answer is correct**

**Lambda will receive each record in the exact order it was placed into the shard. There is no guarantee of order across shards**

**Lambda will receive each record in the exact order it was placed into the stream**

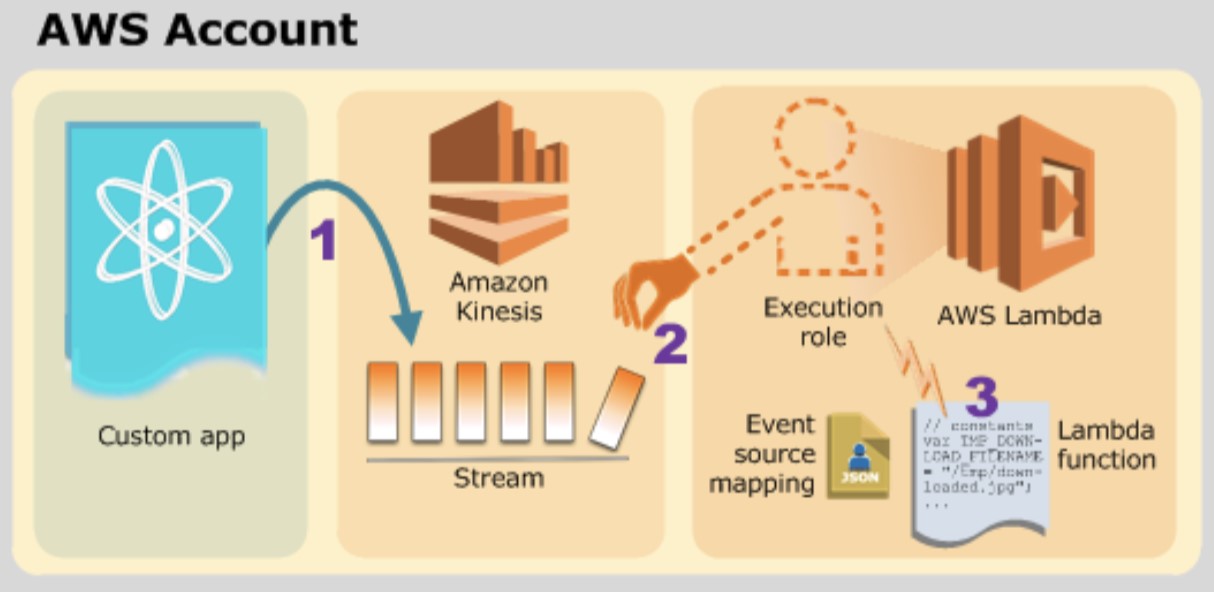
**Lambda will receive each record in the reverse order it was placed into the stream**

Overall explanation

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.

KDS receives data from producers, and the data is stored in shards. Consumers then take the data and process it. In this case the AWS Lambda function is consuming the records from the shards.

In this scenario an application will be producing records and placing them in the stream as in step 1 of the image below. The AWS Lambda function will then consume the records (step 2) and will then execute the function by assuming the execution role specified (step 3).



A shard is an append-only log and a unit of streaming capability. A shard contains an ordered sequence of records ordered by arrival time. The order is guaranteed within a shard but not across shards.

Therefore, the best answer to this question is that AWS Lambda will receive each record in the exact order it was placed into the shard but there is no guarantee of order across shards

**CORRECT:**"Lambda will receive each record in the exact order it was placed into the shard. There is no guarantee of order across shards" is the correct answer.

**INCORRECT:** "Lambda will receive each record in the exact order it was placed into the stream " is incorrect as there are multiple shards in the stream and the order of records is not guaranteed across shards.

**INCORRECT:** "Lambda will receive each record in the reverse order it was placed into the stream" is incorrect as the order is guaranteed within a shard.

**INCORRECT:** "The Developer can select exact order or reverse order using the GetRecords API" is incorrect as you cannot choose the order you receive records with the GetRecords API.

**References:**

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

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

**Save time with our AWS cheat sheets:**

<https://digitalcloud.training/amazon-kinesis/>

**Domain**

AWS Application Integration

**Question 7Correct**

A Developer is looking for a way to use shorthand syntax to express functions, APIs, databases, and event source mappings. The Developer will test using AWS SAM to create a simple Lambda function using Nodejs.12x.

What is the SIMPLEST way for the Developer to get started with a Hello World Lambda function?

**Your answer is correct**

**Install the AWS SAM CLI, run sam init and use one of the AWS Quick Start Templates**

**Use AWS CloudFormation to deploy a Hello World stack using AWS SAM**

**Install the AWS CLI, run aws sam init and use one of the AWS Quick Start Templates**

**Use the AWS Management Console to access AWS SAM and deploy a Hello World function**

Overall explanation

The sam init command initializes a serverless application with an AWS SAM template. The template provides a folder structure for your Lambda functions and is connected to an event source such as APIs, S3 buckets, or DynamoDB tables. This application includes everything you need to get started and to eventually extend it into a production-scale application.

This is the simplest way for the Developer to quickly get started with testing AWS SAM. Before the Developer can use the “sam” commands it is necessary to install the AWS SAM CLI. This is separate to the AWS CLI.

**CORRECT:**"Install the AWS SAM CLI, run sam init and use one of the AWS Quick Start Templates" is the correct answer.

**INCORRECT:** "Install the AWS CLI, run aws sam init and use one of the AWS Quick Start Templates" is incorrect as “sam init” is not an AWS CLI command, therefore you cannot put “aws” in front of “sam”.

**INCORRECT:** "Use the AWS Management Console to access AWS SAM and deploy a Hello World function" is incorrect as you cannot access AWS SAM through the console. You can, however, access the Serverless Application Repository through the console and deploy SAM templates.

**INCORRECT:** "Use AWS CloudFormation to deploy a Hello World stack using AWS SAM" is incorrect as though AWS SAM does use CloudFormation you cannot deploy SAM templates through the AWS CloudFormation console. You must use the SAM CLI or deploy using the Serverless Application Repository.

<https://docs.aws.amazon.com/cli/latest/reference/cloudformation/package.html>

**Save time with our AWS cheat sheets:**

<https://digitalcloud.training/aws-sam/>

**Domain**

AWS Compute

**Question 8Correct**

A Developer wants to find a list of items in a global secondary index from an Amazon DynamoDB table.

Which DynamoDB API call can the Developer use in order to consume the LEAST number of read capacity units?

**Your answer is correct**

**Query operation using eventually-consistent reads**

**Scan operation using eventually-consistent reads**

**Query operation using strongly-consistent reads**

**Scan operation using strongly-consistent reads**

Overall explanation

Amazon DynamoDB provides fast access to items in a table by specifying primary key values. However, many applications might benefit from having one or more secondary (or alternate) keys available, to allow efficient access to data with attributes other than the primary key. To address this, you can create one or more secondary indexes on a table and issue Query or Scan requests against these indexes.

A secondary index is a data structure that contains a subset of attributes from a table, along with an alternate key to support Query operations. You can retrieve data from the index using a Query, in much the same way as you use Query with a table. A table can have multiple secondary indexes, which give your applications access to many different query patterns.

You can also issue scan operations on a global secondary index however it is less efficient as it will return all items in the index.

**CORRECT:**"Query operation using eventually-consistent reads" is the correct answer.

**INCORRECT:** "Query operation using strongly-consistent reads" is incorrect. Strongly consistent reads require more RCUs and also are not supported on a global secondary index (they are supported on local secondary indexes).

**INCORRECT:** "Scan operation using eventually-consistent reads" is incorrect as a scan is less efficient than a query and will therefore use more RCUs.

**INCORRECT:** "Scan operation using strongly-consistent reads" is incorrect as a scan is less efficient than a query and will therefore use more RCUs.

**References:**

<https://docs.amazonaws.cn/en_us/amazondynamodb/latest/developerguide/SecondaryIndexes.html>

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

**Save time with our AWS cheat sheets:**

<https://digitalcloud.training/amazon-dynamodb/>

**Domain**

AWS Database

**Question 9Incorrect**

A Developer created an AWS Lambda function for a serverless application. The Lambda function has been executing for several minutes and the Developer cannot find any log data in CloudWatch Logs.

What is the MOST likely explanation for this issue?

**The Lambda function is missing CloudWatch Logs as a source trigger to send log data**

**Your answer is incorrect**

**The Lambda function does not have any explicit log statements for the log data to send it to CloudWatch Logs**

**The Lambda function is missing a target CloudWatch Logs group**

**Correct answer**

**The execution role for the Lambda function is missing permissions to write log data to the CloudWatch Logs**

Overall explanation

AWS Lambda automatically monitors Lambda functions on your behalf, reporting metrics through Amazon CloudWatch. To help you troubleshoot failures in a function, Lambda logs all requests handled by your function and also automatically stores logs generated by your code through Amazon CloudWatch Logs.

Lambda automatically integrates with CloudWatch Logs and pushes all logs from your code to a CloudWatch Logs group associated with a Lambda function, which is named /aws/lambda/<function name>.

An AWS Lambda function's execution role grants it permission to access AWS services and resources. You provide this role when you create a function, and Lambda assumes the role when your function is invoked. You can create an execution role for development that has permission to send logs to Amazon CloudWatch and upload trace data to AWS X-Ray.

For the lambda function to create log stream and publish logs to cloudwatch, the lambda execution role needs to have the following permissions:



The most likely cause of this issue is that the execution role assigned to the Lambda function does not have the permissions (shown above) to write to CloudWatch Logs.

**CORRECT:**"The execution role for the Lambda function is missing permissions to write log data to the CloudWatch Logs" is the correct answer.

**INCORRECT:** "The Lambda function does not have any explicit log statements for the log data to send it to CloudWatch Logs" is incorrect as this is not required, Lambda automatically logs data to CloudWatch logs and just needs the permissions to do so.

**INCORRECT:** "The Lambda function is missing a target CloudWatch Logs group" is incorrect as the CloudWatch Logs group will be created automatically if the function has sufficient permissions.

**INCORRECT:** "The Lambda function is missing CloudWatch Logs as a source trigger to send log data" is incorrect as CloudWatch Logs is a destination, not a source in this case. However, you do not need to configure CloudWatch Logs as a destination, it is automatic.

**References:**

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

**Save time with our AWS cheat sheets:**

<https://digitalcloud.training/aws-lambda/>

<https://digitalcloud.training/amazon-cloudwatch/>

**Domain**

AWS Compute

**Question 10Incorrect**

A company is running a Docker application on Amazon ECS. The application must scale based on user load in the last 15 seconds.

How should the Developer instrument the code so that the requirement can be met?

**Create a standard-resolution custom Amazon CloudWatch metric for user activity data, then publish data every 30 seconds**

**Your answer is incorrect**

**Create a standard-resolution custom Amazon CloudWatch metric for user activity data, then publish data every 5 seconds**

**Correct answer**

**Create a high-resolution custom Amazon CloudWatch metric for user activity data, then publish data every 5 seconds**

**Create a high-resolution custom Amazon CloudWatch metric for user activity data, then publish data every 30 seconds**

Overall explanation

Metrics produced by AWS services are standard resolution by default. When you publish a custom metric, you can define it as either standard resolution or high resolution. When you publish a high-resolution metric, CloudWatch stores it with a resolution of 1 second, and you can read and retrieve it with a period of 1 second, 5 seconds, 10 seconds, 30 seconds, or any multiple of 60 seconds.

User activity is not a standard CloudWatch metric and as stated above for the resolution we need in this scenario a custom CloudWatch metric is required anyway. Therefore, for this scenario the Developer should create a high-resolution custom Amazon CloudWatch metric for user activity data and publish the data every 5 seconds.

**CORRECT:**"Create a high-resolution custom Amazon CloudWatch metric for user activity data, then publish data every 5 seconds" is the correct answer.

**INCORRECT:** "Create a high-resolution custom Amazon CloudWatch metric for user activity data, then publish data every 30 seconds" is incorrect as the resolution is lower than required which will not provide the granularity required.

**INCORRECT:** "Create a standard-resolution custom Amazon CloudWatch metric for user activity data, then publish data every 30 seconds" is incorrect as standard resolution metrics have a granularity of one minute.

**INCORRECT:** "Create a standard-resolution custom Amazon CloudWatch metric for user activity data, then publish data every 5 seconds" is incorrect as standard resolution metrics have a granularity of one minute.

**References:**

<https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/publishingMetrics.html>

**Save time with our AWS cheat sheets:**

<https://digitalcloud.training/amazon-cloudwatch/>

**Domain**

AWS Compute

**Question 11Correct**

A Development team are deploying an AWS Lambda function that will be used by a production application. The function code will be updated regularly, and new versions will be published. The development team do not want to modify application code to point to each new version.

How can the Development team setup a static ARN that will point to the latest published version?

**Publish a mutable version and point it to the $LATEST version**

**Setup a Route 53 Alias record that points to the published version**

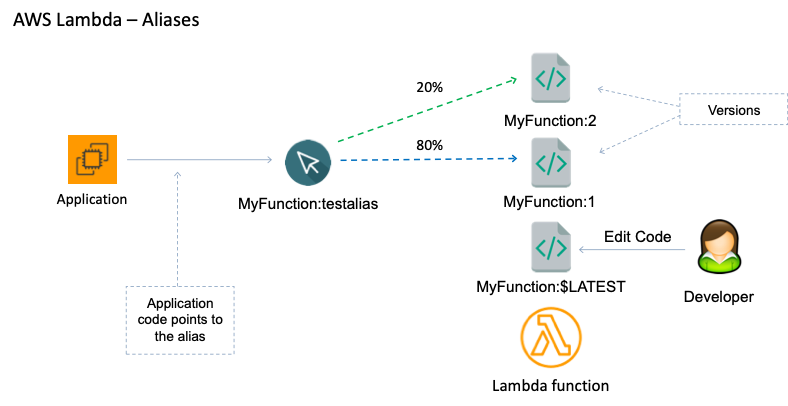
**Use an unqualified ARN**

**Your answer is correct**

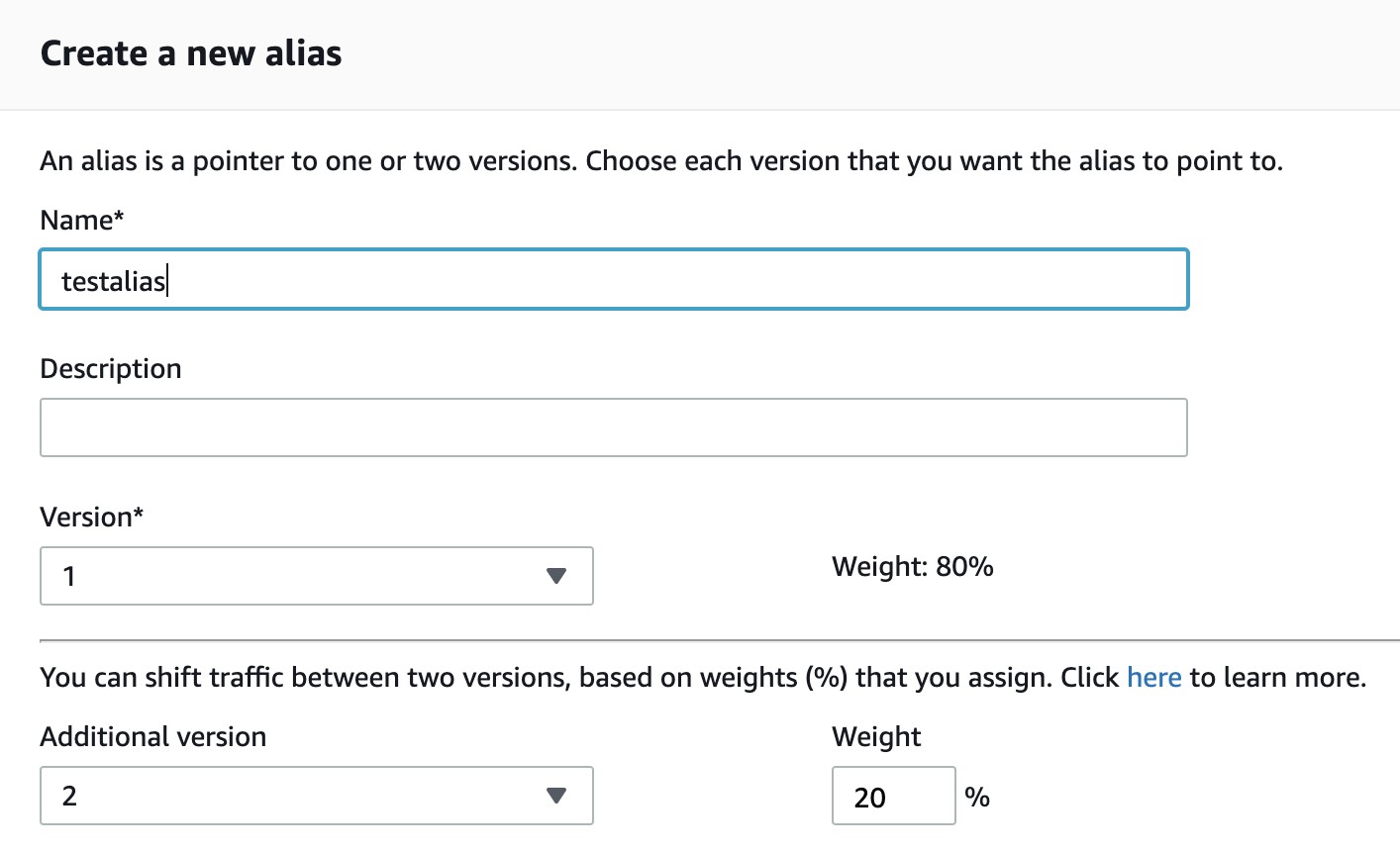
**Setup an Alias that will point to the latest version**

Overall explanation

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. Users can access the function version using the alias ARN.



This is the best way to setup the Lambda function so you don’t need to modify the application code when a new version is published. Instead, the developer will simply need to update the Alias to point to the new version:



As you can see above you can also point to multiple versions and send a percentage of traffic to each. This is great for testing new code.

**CORRECT:**"Setup an Alias that will point to the latest version" is the correct answer.

**INCORRECT:** "Publish a mutable version and point it to the $LATEST version" is incorrect as all published versions are immutable (cannot be modified) and you cannot modify a published version to point to the $LATEST version.

**INCORRECT:** "Use an unqualified ARN" is incorrect as this is an ARN that does not have a version number which means it points to the $LATEST version, not to a published version (as published versions always have version numbers).

**INCORRECT:** "Setup a Route 53 Alias record that points to the published version" is incorrect as you cannot point a Route 53 Alias record to an AWS Lambda function.

**References:**

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

**Save time with our AWS cheat sheets:**

<https://digitalcloud.training/aws-lambda/>

**Domain**

AWS Networking & Content Delivery

**Question 12Incorrect**

A company manages an application that stores data in an Amazon DynamoDB table. The company need to keep a record of all new changes made to the DynamoDB table in another table within the same AWS region. What is the MOST suitable way to deliver this requirement?

**Correct answer**

**Use Amazon DynamoDB streams**

**Use CloudWatch events**

**Use Amazon CloudTrail**

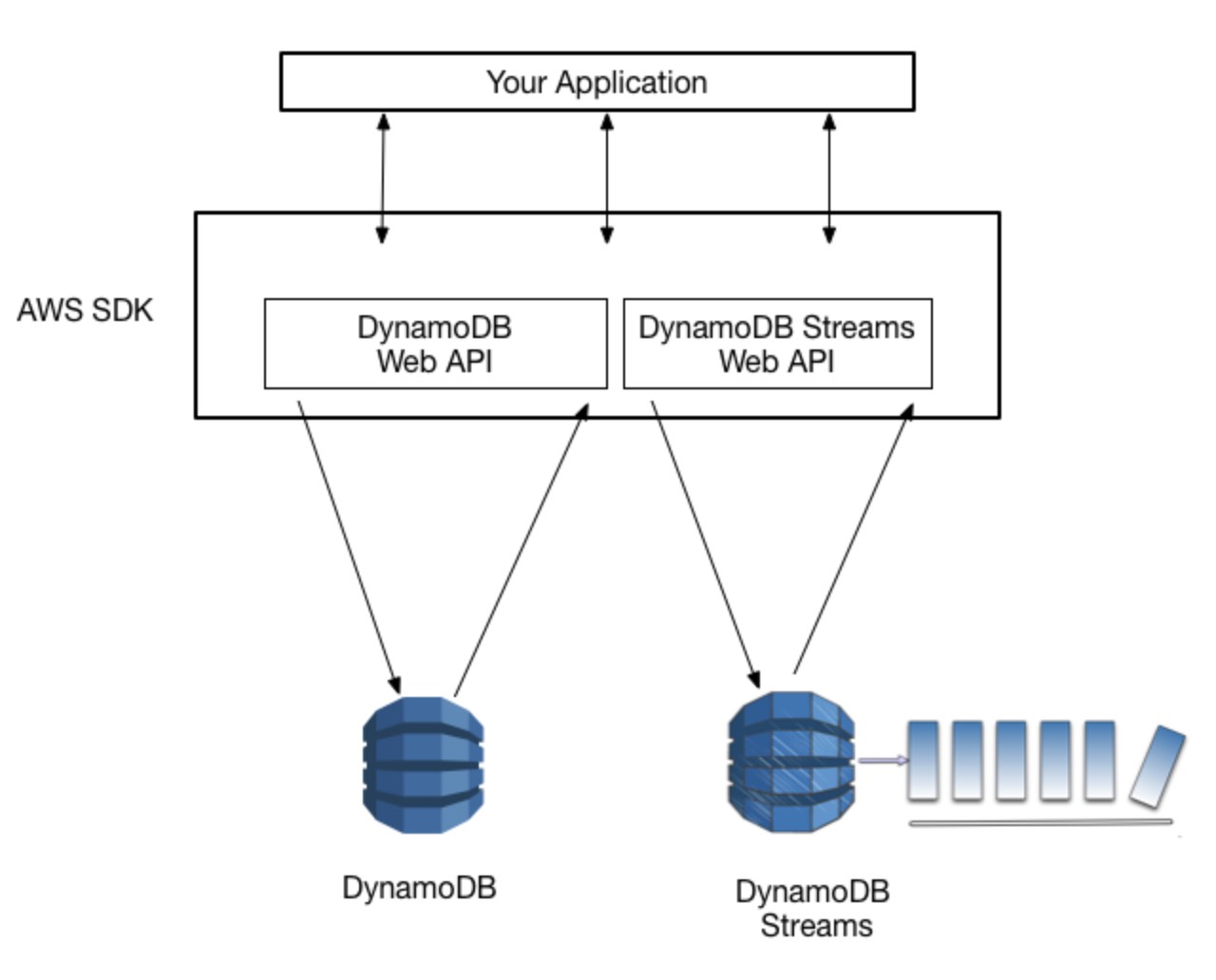
**Your answer is incorrect**

**Use Amazon DynamoDB snapshots**

Overall explanation

A DynamoDB stream is an ordered flow of information about changes to items in a DynamoDB table. When you enable a stream on a table, DynamoDB captures information about every modification to data items in the table.

Whenever an application creates, updates, or deletes items in the table, DynamoDB Streams writes a stream record with the primary key attributes of the items that were modified. A stream record contains information about a data modification to a single item in a DynamoDB table. You can configure the stream so that the stream records capture additional information, such as the "before" and "after" images of modified items.



This is the best way to capture a record of new changes made to the DynamoDB table. Another table can then be populated with this data so the data is stored persistently.

**CORRECT:**"Use Amazon DynamoDB streams" is the correct answer.

**INCORRECT:** "Use CloudWatch events" is incorrect. CloudWatch Events delivers a near real-time stream of system events that describe changes in Amazon Web Services (AWS) resources. However, it does not capture the information that changes in a DynamoDB table so is unsuitable for this purpose.

**INCORRECT:** "Use Amazon CloudTrail" is incorrect as CloudTrail records a history of API calls on your account. It is used for creating an audit trail of events.

**INCORRECT:** "Use Amazon DynamoDB snapshots" is incorrect as snapshots only capture a point in time, they are not used for recording item-level changes.

**References:**

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

**Save time with our AWS cheat sheets:**

<https://digitalcloud.training/amazon-dynamodb/>

**Domain**

AWS Database

**Question 13Correct**

An application uses Amazon Kinesis Data Streams to ingest and process large streams of data records in real time. Amazon EC2 instances consume and process the data using the Amazon Kinesis Client Library (KCL). The application handles the failure scenarios and does not require standby workers. The application reports that a specific shard is receiving more data than expected. To adapt to the changes in the rate of data flow, the “hot” shard is resharded.

Assuming that the initial number of shards in the Kinesis data stream is 6, and after resharding the number of shards increased to 8, what is the maximum number of EC2 instances that can be deployed to process data from all the shards?

**Your answer is correct**

**8**

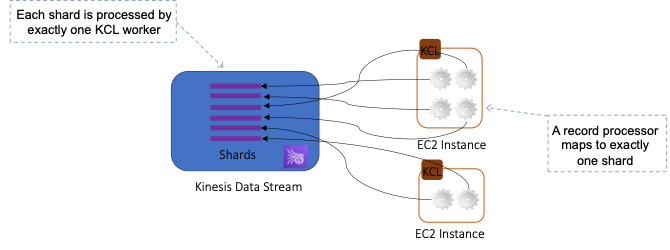
**6**

**1**

**12**

Overall explanation

Typically, when you use the KCL, you should ensure that the number of instances does not exceed the number of shards (except for failure standby purposes). Each shard is processed by exactly one KCL worker and has exactly one corresponding record processor, so you never need multiple instances to process one shard. However, one worker can process any number of shards, so it's fine if the number of shards exceeds the number of instances.



In this scenario, the number of shards has been increased to 8. Therefore, the maximum number of instances that can be deployed is 8 as the number of instances cannot exceed the number of shards.

**CORRECT:**"8" is the correct answer.

**INCORRECT:** "6" is incorrect as this is not the maximum number of instances that can be deployed to process 8 shards. The maximum number of instances should be the same as the number of shards.

**INCORRECT:** "12" is incorrect as the number of instances exceeds the number of shards. You should ensure that the number of instances does not exceed the number of shards

**INCORRECT:** "1" is incorrect as this is not the maximum number of instances that can be deployed to process 8 shards. The maximum number of instances should be the same as the number of shards.

**References:**

<https://docs.aws.amazon.com/streams/latest/dev/kinesis-record-processor-scaling.html>

**Save time with our AWS cheat sheets:**

<https://digitalcloud.training/amazon-kinesis/>

**Domain**

AWS Application Integration

**Question 14Correct**

A Developer needs to run some code using Lambda in response to an event and forward the execution result to another application using a pub/sub notification.

How can the Developer accomplish this?

**Your answer is correct**

**Configure a Lambda “on success” destination and route the execution results to Amazon SNS**

**Configure a Lambda “on success” destination and route the execution results to Amazon SQS**

**Configure a CloudWatch Events alarm the triggers based on Lambda execution success and route the execution results to Amazon SNS**

**Configure a CloudWatch Events alarm the triggers based on Lambda execution success and route the execution results to Amazon SQS**

Overall explanation

With Destinations, you can send asynchronous function execution results to a destination resource without writing code. A function execution result includes version, timestamp, request context, request payload, response context, and response payload. For each execution status (i.e. Success and Failure), you can choose one destination from four options: another [Lambda function](https://docs.aws.amazon.com/lambda/latest/dg/welcome.html), an [SNS topic](https://docs.aws.amazon.com/sns/latest/dg/welcome.html), an [SQS standard queue](https://docs.aws.amazon.com/AWSSimpleQueueService/latest/SQSDeveloperGuide/standard-queues.html), or [EventBridge](https://docs.aws.amazon.com/AmazonCloudWatch/latest/events/WhatIsCloudWatchEvents.html).

For this scenario, the code will be run by Lambda and the execution result will then be sent to the SNS topic. The application that is subscribed to the SNS topics will then receive the notification.

**CORRECT:**"Configure a Lambda “on success” destination and route the execution results to Amazon SNS" is the correct answer.

**INCORRECT:** "Configure a CloudWatch Events alarm the triggers based on Lambda execution success and route the execution results to Amazon SNS" is incorrect as CloudWatch Events is used to track changes in the state of AWS resources. To forward execution results from Lambda a destination should be used.

**INCORRECT:** "Configure a Lambda “on success” destination and route the execution results to Amazon SQS" is incorrect as SQS is a message queue not a pub/sub notification service.

**INCORRECT:** "Configure a CloudWatch Events alarm the triggers based on Lambda execution success and route the execution results to Amazon SQS" is incorrect as CloudWatch Events is used to track changes in the state of AWS resources. To forward execution results from Lambda a destination should be used (with an SNS topic).

**References:**

<https://aws.amazon.com/about-aws/whats-new/2019/11/aws-lambda-supports-destinations-for-asynchronous-invocations/>

**Save time with our AWS cheat sheets:**

<https://digitalcloud.training/aws-lambda/>

**Domain**

AWS Compute

**Question 15Correct**

A media organization utilizes an Amazon API Gateway REST API endpoint to disseminate updates from an internal Content Management System (CMS) to Amazon EventBridge. An EventBridge rule is set up to monitor these updates and control content syndication in a primary AWS account. The organization now wants to extend the reach of these updates across several affiliate AWS accounts.

How can the developer accomplish this without altering the configuration of the CMS?

**Use AWS Lambda in the main account to clone updates and manually invoke it in affiliate accounts.**

**Your answer is correct**

**Implement an EventBridge event bus in the affiliate AWS accounts to create a rule that matches events and forwards them from the main account.**

**Establish multiple API Gateway REST API endpoints in affiliate accounts to directly receive updates from the CMS.**

**Set up multiple instances of the CMS, each linked to a different AWS account.**

Overall explanation

EventBridge allows events to be sent to other AWS accounts, providing a straightforward solution to distribute the updates across multiple accounts. This wouldn't require any changes to the CMS configuration.

**CORRECT:**"Implement an EventBridge event bus in the affiliate AWS accounts to create a rule that matches events and forwards them from the main account" is the correct answer (as explained above.)

**INCORRECT:** "Establish multiple API Gateway REST API endpoints in affiliate accounts to directly receive updates from the CMS" is incorrect.

Setting up multiple API Gateway REST API endpoints in affiliate accounts to directly receive updates from the CMS would not only require changes to the CMS configuration but would also mean maintaining multiple endpoints.

**INCORRECT:** "Use AWS Lambda in the main account to clone updates and manually invoke it in affiliate accounts" is incorrect.

Using AWS Lambda in the main account to clone updates and manually invoke it in affiliate accounts would be an inefficient solution, leading to potential latency and management overhead.

**INCORRECT:** "Set up multiple instances of the CMS, each linked to a different AWS account" is incorrect.

Setting up multiple instances of the CMS, each linked to a different AWS account, would be a significantly complex and resource-intensive solution. It's neither practical nor necessary for the given requirement.

**References:**

<https://docs.aws.amazon.com/eventbridge/latest/userguide/eb-cross-account.html>

**Save time with our AWS cheat sheets:**

<https://digitalcloud.training/amazon-cloudwatch/>

**Domain**

AWS Management & Governance

**Question 16Correct**

What does an Amazon SQS delay queue accomplish?

**Your answer is correct**

**Messages are hidden for a configurable amount of time when they are first added to the queue**

**Message cannot be deleted for a configurable amount of time after they are consumed from the queue**

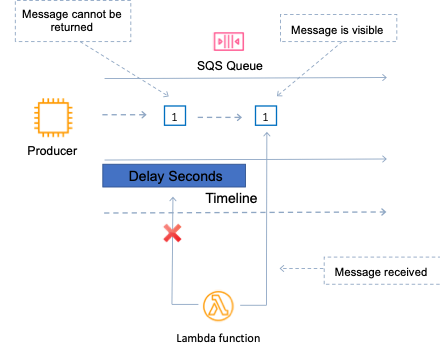
**The consumer can poll the queue for a configurable amount of time before retrieving a message**

**Messages are hidden for a configurable amount of time after they are consumed from the queue**

Overall explanation

Delay queues let you postpone the delivery of new messages to a queue for a number of seconds, for example, when your consumer application needs additional time to process messages.

If you create a delay queue, any messages that you send to the queue remain invisible to consumers for the duration of the delay period. The default (minimum) delay for a queue is 0 seconds. The maximum is 15 minutes.



Therefore, the correct explanation is that with an Amazon SQS Delay Queue messages are hidden for a configurable amount of time when they are first added to the queue

**CORRECT:**"Messages are hidden for a configurable amount of time when they are first added to the queue" is the correct answer.

**INCORRECT:** "Messages are hidden for a configurable amount of time after they are consumed from the queue" is incorrect. They are hidden when they are added to the queue.

**INCORRECT:** "The consumer can poll the queue for a configurable amount of time before retrieving a message" is incorrect. A delay queue simply delays visibility of the message, it does not affect polling behavior.

**INCORRECT:** "Message cannot be deleted for a configurable amount of time after they are consumed from the queue" is incorrect. That is what a visibility timeout achieves.

**References:**

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

**Save time with our AWS cheat sheets:**

<https://digitalcloud.training/aws-application-integration-services/>

**Domain**

AWS Application Integration

**Question 17Incorrect**

A gaming application displays the results of games in a leaderboard. The leaderboard is updated by 4 KB messages that are retrieved from an Amazon SQS queue. The updates are received infrequently but the Developer needs to minimize the time between the messages arriving in the queue and the leaderboard being updated.

Which technique provides the shortest delay in updating the leaderboard?

**Correct answer**

**Retrieve the messages from the queue using long polling every 15 seconds**

**Your answer is incorrect**

**Retrieve the messages from the queue using short polling every 10 seconds**

**Store the message payload in Amazon S3 and use the SQS Extended Client Library for Java**

**Reduce the size of the messages with compression before sending them**

Overall explanation

The process of consuming messages from a queue depends on whether you use short or long polling. By default, Amazon SQS uses short polling, querying only a subset of its servers (based on a weighted random distribution) to determine whether any messages are available for a response.

You can use long polling to reduce your costs while allowing your consumers to receive messages as soon as they arrive in the queue. When the wait time for the ReceiveMessage API action is greater than 0, long polling is in effect. The maximum long polling wait time is 20 seconds.

Long polling helps reduce the cost of using Amazon SQS by eliminating the number of empty responses (when there are no messages available for a [ReceiveMessage](https://docs.aws.amazon.com/AWSSimpleQueueService/latest/APIReference/API_ReceiveMessage.html) request) and false empty responses (when messages are available but aren't included in a response). It also returns messages as soon as they become available.

**CORRECT:**"Retrieve the messages from the queue using long polling every 15 seconds" is the correct answer.

**INCORRECT:** "Retrieve the messages from the queue using short polling every 10 seconds" is incorrect as short polling is configured when the WaitTimeSeconds parameter of a [ReceiveMessage](https://docs.aws.amazon.com/AWSSimpleQueueService/latest/APIReference/API_ReceiveMessage.html) request is set to 0. Any number above zero indicates long polling is in effect.

**INCORRECT:** "Reduce the size of the messages with compression before sending them" is incorrect as this will not mean messages are picked up earlier and there is no reason to compress messages that are 4 KB in size.

**INCORRECT:** "Store the message payload in Amazon S3 and use the SQS Extended Client Library for Java" is incorrect as this is unnecessary for messages of this size and will also not result in the shortest delay when updating the leaderboard.

**References:**

<https://docs.aws.amazon.com/AWSSimpleQueueService/latest/SQSDeveloperGuide/sqs-short-and-long-polling.html>

**Save time with our AWS cheat sheets:**

<https://digitalcloud.training/aws-application-integration-services/>

**Domain**

AWS Application Integration

**Question 18Incorrect**

An application is running on a cluster of Amazon EC2 instances. The application has received an error when trying to read objects stored within an Amazon S3 bucket. The bucket is encrypted with server-side encryption and AWS KMS managed keys (SSE-KMS). The error is as follows:

Service: AWSKMS; Status Code: 400, Error Code: ThrottlingException

Which combination of steps should be taken to prevent this failure? (Select TWO.)

**Import a customer master key (CMK) with a larger key size**

**Your selection is correct**

**Perform error retries with exponential backoff in the application code**

**Your selection is incorrect**

**Contact AWS support to request an S3 rate limit increase**

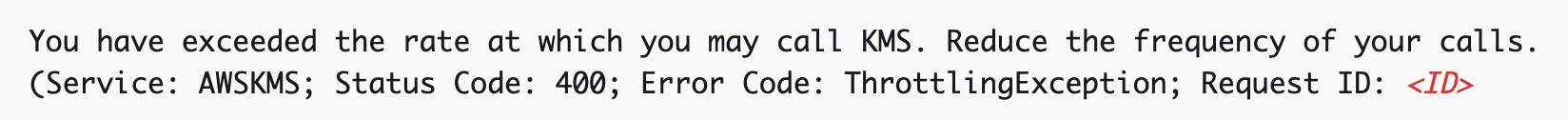
**Correct selection**

**Contact AWS support to request an AWS KMS rate limit increase**

**Use more than once customer master key (CMK) to encrypt S3 data**

Overall explanation

AWS KMS establishes quotas for the number of API operations requested in each second. When you exceed an API request quota, AWS KMS throttles the request, that is, it rejects an otherwise valid request and returns a ThrottlingException error like the following one.



As the error indicates, one of the recommendations is to reduce the frequency of calls which can be implemented by using exponential backoff logic in the application code. It is also possible to contact AWS and request an increase in the quota.

**CORRECT:**"Contact AWS support to request an AWS KMS rate limit increase" is a correct answer.

**CORRECT:**"Perform error retries with exponential backoff in the application code" is a correct answer.

**INCORRECT:** "Contact AWS support to request an S3 rate limit increase" is incorrect as the error indicates throttling in AWS KMS.

**INCORRECT:** "Import a customer master key (CMK) with a larger key size" is incorrect as the key size does not affect the quota for requests to AWS KMS.

**INCORRECT:** "Use more than once customer master key (CMK) to encrypt S3 data" is incorrect as the issue is not the CMK it is the request quota on AWS KMS.

**References:**

<https://docs.aws.amazon.com/kms/latest/developerguide/requests-per-second.html>

**Save time with our AWS cheat sheets:**

<https://digitalcloud.training/aws-kms/>

**Domain**

AWS Security, Identity, & Compliance

**Question 19Correct**

An international research organization holds a wide range of data across several Amazon S3 buckets. They recently received an alert indicating potential exposure of sensitive financial data via a public-facing web portal. The developer's job is to trace all potential data leakage points across their AWS infrastructure.

What is the most effective strategy for this task?

**Manually inspect each S3 bucket and the data tables contained within to identify any potential financial data exposures.**

**Implement Amazon Macie and apply the SensitiveData:S3Object/personal finding type across all S3 buckets. This approach would identify personal data but may not effectively identify all potential exposure of sensitive financial data.**

**Utilize AWS CloudTrail logs to track activity and find potential data exposures, specifically focusing on financial data transactions.**

**Your answer is correct**

**Implement Amazon Macie and apply the SensitiveData:S3Object/financial finding type across all S3 buckets to automatically identify potential exposure of sensitive financial data.**

Overall explanation

Amazon Macie is a fully managed data security and data privacy service that uses machine learning and pattern matching to discover and protect sensitive data in AWS. Using the **SensitiveData:S3Object/financial** finding type within Macie, it is possible to automatically detect and identify sensitive financial data in your S3 buckets.

**CORRECT:**"Implement Amazon Macie and apply the SensitiveData:S3Object/financial finding type across all S3 buckets to automatically identify potential exposure of sensitive financial data" is the correct answer (as explained above.)

**INCORRECT:** "Implement Amazon Macie and apply the SensitiveData:S3Object/personal finding type across all S3 buckets. This approach would identify personal data but may not effectively identify all potential exposure of sensitive financial data" is incorrect.

Using Amazon Macie with the **SensitiveData:S3Object/personal** finding type will only help in identifying potential exposure of sensitive personal data but not specifically financial data. It is not an effective solution for this specific requirement.

**INCORRECT:** "Manually inspect each S3 bucket and the data tables contained within to identify any potential financial data exposures" is incorrect.

Manual inspection of each S3 bucket and data table would be time-consuming and prone to human error, especially when dealing with a large amount of data and numerous S3 buckets.

**INCORRECT:** "Utilize AWS CloudTrail logs to track activity and find potential data exposures, specifically focusing on financial data transactions" is incorrect.

AWS CloudTrail logs would provide data about AWS account activity, but they are not specifically designed to detect sensitive data exposures.

**References:**

<https://docs.aws.amazon.com/macie/latest/user/findings-types.html>

**Domain**

AWS Security, Identity, & Compliance

**Question 20Correct**

The development team is experiencing issues with their application hosted on Amazon EC2 instances, as they are unable to connect to an Amazon S3 bucket during test runs.

What should be the appropriate measures to resolve this issue? (Select TWO.)

**Validate the VPC peering connections to ensure the S3 bucket is reachable.**

**Modify the Amazon S3 bucket to be public to allow EC2 instances to access it.**

**Your selection is correct**

**Verify the IAM roles attached to the EC2 instances and ensure they have the necessary permissions to access the S3 bucket.**

**Your selection is correct**

**Check the bucket policies for the Amazon S3 bucket and confirm that they permit access from the EC2 instances.**

**Check the security groups attached to the EC2 instances to ensure the required inbound rules are in place.**

Overall explanation

IAM roles attached to the EC2 instances allow applications running on these instances to use AWS services without the need to manage and store AWS credentials. If an EC2 instance is unable to access an S3 bucket, one possibility is that the IAM role associated with the EC2 instance does not have the necessary permissions to access the S3 bucket. Checking and updating the IAM roles can resolve this issue.

Amazon S3 bucket policies define who can access the contents of the bucket and what actions they can perform. If the EC2 instances are unable to access the S3 bucket, another possible cause is that the bucket policy does not permit access to the EC2 instances. Checking and modifying the bucket policies to grant access to the EC2 instances can fix this problem.

**CORRECT:**"Verify the IAM roles attached to the EC2 instances and ensure they have the necessary permissions to access the S3 bucket" is a correct answer (as explained above.)

**CORRECT:**"Check the bucket policies for the Amazon S3 bucket and confirm that they permit access from the EC2 instances" is also a correct answer (as explained above.)

**INCORRECT:** "Check the security groups attached to the EC2 instances to ensure the required inbound rules are in place" is incorrect.

Security groups are associated with EC2 instances and act as a firewall that controls the traffic for one or more instances. Inbound rules affect traffic coming towards the EC2 instance and this does not affect access to S3. For S3, the instance needs to have an outbound rule that allows the relevant protocols.

**INCORRECT:** "Validate the VPC peering connections to ensure the S3 bucket is reachable" is incorrect.

Amazon VPC peering connection allows you to route traffic between two VPCs using private IPv4 or IPv6 addresses. However, S3 buckets are not accessed via VPC peering connections; they are accessed using the Amazon S3 service endpoint. Hence, validating VPC peering connections won't solve the issue here.

**INCORRECT:** "Modify the Amazon S3 bucket to be public to allow EC2 instances to access it" is incorrect.

Making an Amazon S3 bucket public can pose serious security risks, as it will allow anyone to access the contents of the bucket. This action is generally not recommended, especially not to solve access issues for EC2 instances. The best practice is to manage access at a more granular level using IAM roles and bucket policies.

**References:**

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

<https://docs.aws.amazon.com/AmazonS3/latest/userguide/Welcome.html>

**Save time with our AWS cheat sheets:**

<https://digitalcloud.training/aws-iam/>

**Domain**

AWS Storage

**Question 21Incorrect**

An organization needs to add encryption in-transit to an existing website running behind an Elastic Load Balancer. The website’s Amazon EC2 instances are CPU-constrained and therefore load on their CPUs should not be increased. What should be done to secure the website? (Select TWO.)

**Configure an Elastic Load Balancer with a KMS CMK**

**Your selection is incorrect**

**Install SSL certificates on the EC2 instances**

**Your selection is correct**

**Configure an Elastic Load Balancer with SSL termination**

**Configure an Elastic Load Balancer with SSL pass-through**

**Correct selection**

**Configure SSL certificates on an Elastic Load Balancer**

Overall explanation

The company need to add security to their website by encrypting traffic in-transit using HTTPS. This requires adding SSL/TLS certificates to enable the encryption. The process of encrypting and decrypting data is CPU intensive and therefore the company need to avoid adding certificates to the EC2 instances as that will place further load on their CPUs.

Therefore, the solution is to configure SSL certificates on the Elastic Load Balancer and then configure SSL termination. This can be done by adding a certificate to a HTTPS listener on the load balancer.

**CORRECT:**"Configure SSL certificates on an Elastic Load Balancer" is a correct answer.

**CORRECT:**"Configure an Elastic Load Balancer with SSL termination" is a correct answer.

**INCORRECT:** "Configure an Elastic Load Balancer with SSL pass-through" is incorrect as with pass-through the SSL session must be terminated on the EC2 instances which should be avoided as they are CPU-constrained.

**INCORRECT:** "Configure an Elastic Load Balancer with a KMS CMK" is incorrect as a KMS CMK is used to encrypt data at rest, it is not used for in-transit encryption.

**INCORRECT:** "Install SSL certificates on the EC2 instances" is incorrect as this would increase the load on the CPUs

**References:**

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

**Save time with our AWS cheat sheets:**

<https://digitalcloud.training/aws-elastic-load-balancing-aws-elb/>

**Domain**

AWS Networking & Content Delivery

**Question 22Correct**

Based on the following AWS CLI command the resulting output, what has happened here?

1. $ aws lambda invoke --function-name MyFunction --payload ewogICJrZXkxIjogInZhbHVlMSIsCiAgImtleTIiOiAidmFsdWUyIiwKICAia2V5MyI6ICJ2YWx1ZTMiCn0= response.json
2. {
3. "StatusCode": 200
4. }

**An AWS Lambda function has been invoked synchronously and has not completed successfully**

**Your answer is correct**

**An AWS Lambda function has been invoked synchronously and has completed successfully**

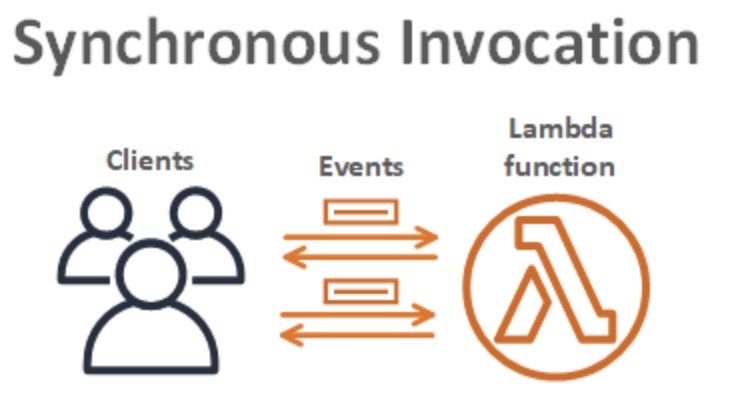
**An AWS Lambda function has been invoked asynchronously and has completed successfully**

**An AWS Lambda function has been invoked asynchronously and has not completed successfully**

Overall explanation

When you invoke a function synchronously, Lambda runs the function and waits for a response. When the function execution ends, Lambda returns the response from the function's code with additional data, such as the version of the function that was executed. To invoke a function synchronously with the AWS CLI, use the invoke command.

The following diagram shows clients invoking a Lambda function synchronously. Lambda sends the events directly to the function and sends the function's response back to the invoker.



We know the function has been run synchronously as the --invocation-type Event parameter has not been included. Also, the status code 200 indicates a successful execution of a synchronous execution.

**CORRECT:**"An AWS Lambda function has been invoked synchronously and has completed successfully" is the correct answer.

**INCORRECT:** "An AWS Lambda function has been invoked synchronously and has not completed successfully" is incorrect as the status code 200 indicates a successful execution.

**INCORRECT:** "An AWS Lambda function has been invoked asynchronously and has completed successfully" is incorrect as the --invocation-type Event has parameter is not included so this is not an asynchronous invocation.

**INCORRECT:** "An AWS Lambda function has been invoked asynchronously and has not completed successfully" is incorrect as the --invocation-type Event has parameter is not included so this is not an asynchronous invocation.

**References:**

<https://docs.aws.amazon.com/lambda/latest/dg/invocation-sync.html>

**Save time with our AWS cheat sheets:**

<https://digitalcloud.training/aws-lambda/>

**Domain**

AWS Compute

**Question 23Correct**

An application is running on a fleet of EC2 instances running behind an Elastic Load Balancer (ELB). The EC2 instances session data in a shared Amazon S3 bucket. Security policy mandates that data must be encrypted in transit.

How can the Developer ensure that all data that is sent to the S3 bucket is encrypted in transit?

**Create an S3 bucket policy that denies traffic where SecureTransport is true**

**Configure HTTP to HTTPS redirection on the Elastic Load Balancer**

**Your answer is correct**

**Create an S3 bucket policy that denies traffic where SecureTransport is false**

**Create an S3 bucket policy that denies any S3 Put request that does not include the x-amz-server-side-encryption**

Overall explanation

At the Amazon S3 bucket level, you can configure permissions through a bucket policy. For example, you can limit access to the objects in a bucket by IP address range or specific IP addresses. Alternatively, you can make the objects accessible only through HTTPS.

The following bucket policy allows access to Amazon S3 objects only through HTTPS (the policy was generated with the AWS Policy Generator).



Here the bucket policy explicitly denies ("Effect": "Deny") all read access ("Action": "s3:GetObject") from anybody who browses ("Principal": "\*") to Amazon S3 objects within an Amazon S3 bucket if they are not accessed through HTTPS ("aws:SecureTransport": "false").

**CORRECT:**"Create an S3 bucket policy that denies traffic where SecureTransport is false" is the correct answer.

**INCORRECT:** "Create an S3 bucket policy that denies traffic where SecureTransport is true" is incorrect. This will not work as it is denying traffic that IS encrypted in transit.

**INCORRECT:** "Create an S3 bucket policy that denies any S3 Put request that does not include the x-amz-server-side-encryption" is incorrect. This will ensure that the data is encrypted at rest, but not in-transit.

**INCORRECT:** "Configure HTTP to HTTPS redirection on the Elastic Load Balancer" is incorrect. This will ensure the client traffic reaching the ELB is encrypted however we need to ensure the traffic from the EC2 instances to S3 is encrypted and the ELB is not involved in this communication.

**References:**

<https://aws.amazon.com/blogs/security/how-to-use-bucket-policies-and-apply-defense-in-depth-to-help-secure-your-amazon-s3-data/>

**Save time with our AWS cheat sheets:**

<https://digitalcloud.training/amazon-s3-and-glacier/>

**Domain**

AWS Security, Identity, & Compliance

**Question 24Correct**

An application runs on Amazon EC2 and generates log files. A Developer needs to centralize the log files so they can be queried and retained. What is the EASIEST way for the Developer to centralize the log files?

**Setup a CloudWatch Events rule to trigger an SNS topic when an application log file is generated**

**Your answer is correct**

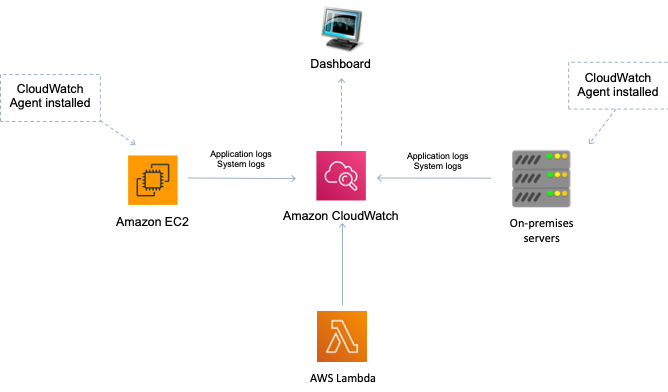
**Install the Amazon CloudWatch Logs agent and collect the logs from the instances**

**Create a script that copies the log files to Amazon S3 and use a cron job to run the script on a recurring schedule**

**Create a script that uses the AWS SDK to collect and send the log files to Amazon CloudWatch Logs**

Overall explanation

You can use Amazon CloudWatch Logs to monitor, store, and access your log files from Amazon Elastic Compute Cloud (Amazon EC2) instances, AWS CloudTrail, Route 53, and other sources.



CloudWatch Logs enables you to centralize the logs from all of your systems, applications, and AWS services that you use, in a single, highly scalable service. You can then easily view them, search them for specific error codes or patterns, filter them based on specific fields, or archive them securely for future analysis.

To collect logs from Amazon EC2 and on-premises instances it is necessary to install an agent. There are two options: the unified CloudWatch Agent which collects logs and advanced metrics (such as memory usage), or the older CloudWatch Logs agent which only collects logs from Linux servers.

**CORRECT:**"Install the Amazon CloudWatch Logs agent and collect the logs from the instances" is the correct answer.

**INCORRECT:** "Create a script that copies the log files to Amazon S3 and use a cron job to run the script on a recurring schedule" is incorrect as the best place to move the log files to for querying and long term retention would be CloudWatch Logs. It is also easier to use the agent than to create and maintain a script.

**INCORRECT:** "Create a script that uses the AWS SDK to collect and send the log files to Amazon CloudWatch Logs" is incorrect as this is not the easiest way to achieve this outcome. It will be easier to use the CloudWatch Logs agent.

**INCORRECT:** "Setup a CloudWatch Events rule to trigger an SNS topic when an application log file is generated" is incorrect as CloudWatch Events does not collect log files, it monitors state changes in resources.

**References:**

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

**Save time with our AWS cheat sheets:**

<https://digitalcloud.training/amazon-cloudwatch/>

**Domain**

AWS Management & Governance

**Question 25Correct**

You run an ad-supported photo sharing website using Amazon S3 to serve photos to visitors of your site. At some point you find out that other sites have been linking to the photos on your site, causing loss to your business.  
What is an effective method to mitigate this?

**Your answer is correct**

**Remove public read access and use signed URLs with expiry dates**

**Use CloudFront distributions for static content**

**Block the IPs of the offending websites in Security Groups**

**Store photos on an EBS volume of the web server**

Overall explanation

When Amazon S3 objects are private, only the object owner has permission to access these objects. However, the object owner can optionally share objects with others by creating a presigned URL, using their own security credentials, to grant time-limited permission to download the objects.

When you create a presigned URL for your object, you must provide your security credentials, specify a bucket name, an object key, specify the HTTP method (GET to download the object) and expiration date and time. The presigned URLs are valid only for the specified duration.

Anyone who receives the presigned URL can then access the object. In this scenario, the photos can be shared with the owner’s website but not with any other 3rd parties. This will stop other sites from linking to the photos as they will not display anywhere else.

**CORRECT:**"Remove public read access and use signed URLs with expiry dates" is the correct answer.

**INCORRECT:** "Store photos on an EBS volume of the web server" is incorrect as this does not add any more control over content visibility in the website.

**INCORRECT:** "Use CloudFront distributions for static content" is incorrect as this alone will not protect the content. You can also use pre-signed URLs with CloudFront, but this isn’t mentioned.

**INCORRECT:** "Block the IPs of the offending websites in Security Groups" is incorrect as you can only configure allow rules in security groups so this would be hard to manage.

**References:**

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

**Save time with our AWS cheat sheets:**

<https://digitalcloud.training/amazon-s3-and-glacier/>

**Domain**

AWS Storage

**Question 26Correct**

A media company uses Amazon EC2 instances managed by AWS Elastic Beanstalk to run its high-traffic website. The engineering team needs to introduce a new feature, which requires upgrading the underlying platform to a newer version of Node.js. The deployment of the new code and the platform upgrade need to happen without causing any downtime.

Which strategy should the team adopt to fulfill these requirements?

**Your answer is correct**

**Implement Blue/Green (CNAME Swap) deployment using Elastic Beanstalk. Prepare a separate environment with the new version of Node.js and the new code, and once testing is complete, swap the CNAMEs of the two environments.**

**Create a duplicate EC2 instance manually with the new version of Node.js and the updated code. Test the new instance and replace one of the instances behind the ELB once the testing is successful.**

**Upgrade the platform version of the current Elastic Beanstalk environment, then deploy the new application code. Monitor the application and quickly roll back changes if any issues occur.**

**Use AWS CodeDeploy to deploy the new application code first, then manually update Node.js on the Elastic Beanstalk environment.**

Overall explanation

The Blue/Green deployment strategy involves creating a separate environment (the 'green' environment) with the new application version and platform updates. It allows full testing and validation. Once you're confident in the 'green' environment, you can swap the environment URLs to redirect traffic to the new environment. This results in zero downtime.

**CORRECT:**"Implement Blue/Green (CNAME Swap) deployment using Elastic Beanstalk. Prepare a separate environment with the new version of Node.js and the new code, and once testing is complete, swap the CNAMEs of the two environments" is the correct answer (as explained above.)

**INCORRECT:** "Upgrade the platform version of the current Elastic Beanstalk environment, then deploy the new application code. Monitor the application and quickly roll back changes if any issues occur" is incorrect.

This approach introduces risks that could result in downtime during the upgrade and deployment process. If any issues occur during the upgrade or after the deployment of the new application code, rolling back these changes could cause further downtime.

**INCORRECT:** "Create a duplicate EC2 instance manually with the new version of Node.js and the updated code. Test the new instance and replace one of the instances behind the ELB once the testing is successful" is incorrect.

Managing the process manually and replacing instances behind the load balancer may result in inconsistencies and is not as streamlined or reliable as using the deployment and environment management capabilities of Elastic Beanstalk.

**INCORRECT:** "Use a rolling deployment for the new application code. Apply the code to a subset of EC2 instances until the tests pass. Redeploy the previous code if the tests fail" is incorrect.

While AWS CodeDeploy is a robust deployment service, performing the platform upgrade manually after the code deployment doesn't guarantee zero downtime, as the platform upgrade might require a restart of the EC2 instances, causing service interruptions.

**References:**

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

**Save time with our AWS cheat sheets:**

<https://digitalcloud.training/aws-elastic-beanstalk/>

**Domain**

AWS Compute

**Question 27Correct**

A development team require a fully-managed source control service that is compatible with Git.

Which service should they use?

**AWS CodeDeploy**

**AWS CodePipeline**

**Your answer is correct**

**AWS CodeCommit**

**AWS Cloud9**

Overall explanation

AWS CodeCommit is a version control service hosted by Amazon Web Services that you can use to privately store and manage assets (such as documents, source code, and binary files) in the cloud. CodeCommit is a fully-managed service that hosts secure Git-based repositories.

**CORRECT:**"AWS CodeCommit" is the correct answer.

**INCORRECT:** "AWS CodeDeploy" is incorrect. CodeDeploy is a deployment service that automates application deployments to Amazon EC2 instances, on-premises instances, serverless Lambda functions, or Amazon ECS services.

**INCORRECT:** "AWS CodePipeline" is incorrect. AWS CodePipeline is a fully managed continuous delivery service that helps you automate your release pipelines for fast and reliable application and infrastructure updates.

**INCORRECT:** "AWS Cloud9" is incorrect. AWS Cloud9 is a cloud-based integrated development environment (IDE) that lets you write, run, and debug your code with just a browser.

**References:**

<https://docs.aws.amazon.com/codecommit/latest/userguide/welcome.html>

**Save time with our AWS cheat sheets:**

<https://digitalcloud.training/aws-developer-tools/>

**Domain**

AWS Developer Tools

**Question 28Correct**

A Developer needs to return a list of items in a global secondary index from an Amazon DynamoDB table.

Which DynamoDB API call can the Developer use in order to consume the LEAST number of read capacity units?

**Your answer is correct**

**Query operation using eventually-consistent reads**

**Query operation using strongly-consistent reads**

**Scan operation using strongly-consistent reads**

**Scan operation using eventually-consistent reads**

Overall explanation

The Query operation finds items based on primary key values. You can query any table or secondary index that has a composite primary key (a partition key and a sort key).

For items up to 4 KB in size, one RCU equals one strongly consistent read request per second or two eventually consistent read requests per second. Therefore, using eventually consistent reads uses fewer RCUs.

**CORRECT:**"Query operation using eventually-consistent reads" is the correct answer.

**INCORRECT:** "Query operation using strongly-consistent reads" is incorrect as strongly-consistent reads use more RCUs than eventually consistent reads.

**INCORRECT:** "Scan operation using eventually-consistent reads" is incorrect. The Scan operation returns one or more items and item attributes by accessing every item in a table or a secondary index and therefore uses more RCUs than a query operation.

**INCORRECT:** "Scan operation using strongly-consistent reads" is incorrect. The Scan operation returns one or more items and item attributes by accessing every item in a table or a secondary index and therefore uses more RCUs than a query operation.

**References:**

<https://docs.aws.amazon.com/amazondynamodb/latest/APIReference/API_Query.html>

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

**Save time with our AWS cheat sheets:**

<https://digitalcloud.training/amazon-dynamodb/>

**Domain**

AWS Database

**Question 29Correct**

A Developer is creating a banking application that will be used to view financial transactions and statistics. The application requires multi-factor authentication to be added to the login protocol.

Which service should be used to meet this requirement?

**Amazon Cognito Identity Pool with MFA**

**Your answer is correct**

**Amazon Cognito User Pool with MFA**

**AWS Directory Service**

**AWS IAM with MFA**

Overall explanation

A user pool is a user directory in Amazon Cognito. With a user pool, your users can sign in to your web or mobile app through Amazon Cognito. Your users can also sign in through social identity providers like Google, Facebook, Amazon, or Apple, and through SAML identity providers.

User pools provide:

• Sign-up and sign-in services.

• A built-in, customizable web UI to sign in users.

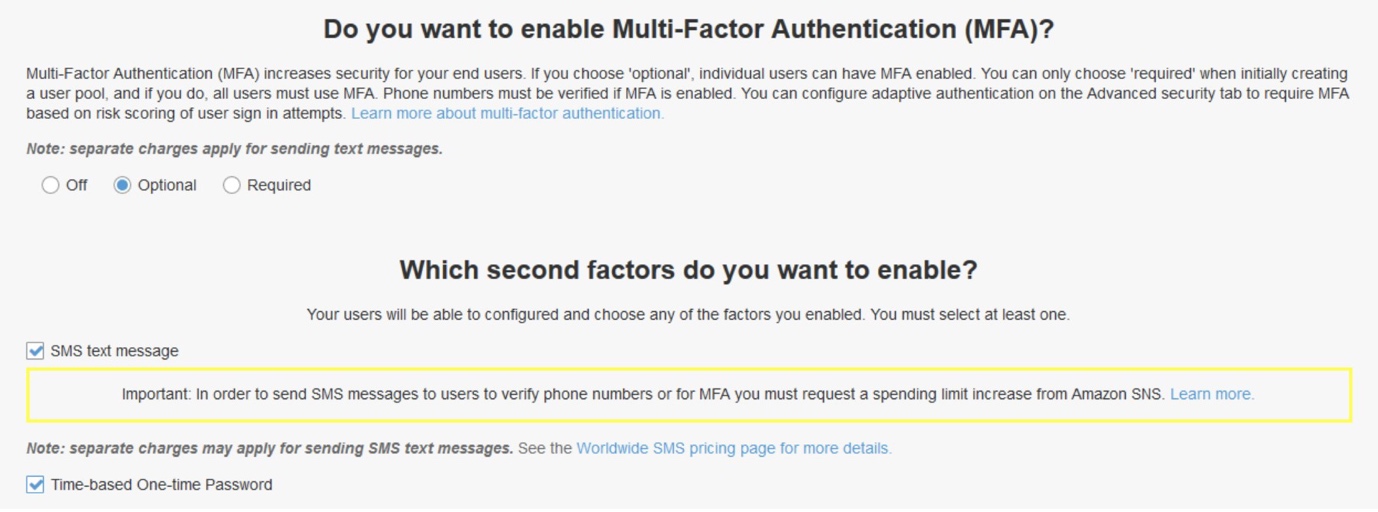
• Social sign-in with Facebook, Google, Login with Amazon, and Sign in with Apple, as well as sign-in with SAML identity providers from your user pool.

• User directory management and user profiles.

• Security features such as multi-factor authentication (MFA), checks for compromised credentials, account takeover protection, and phone and email verification.

• Customized workflows and user migration through AWS Lambda triggers.

Multi-factor authentication (MFA) increases security for your app by adding another authentication method, and not relying solely on username and password. You can choose to use SMS text messages, or time-based one-time (TOTP) passwords as second factors in signing in your users.



For this scenario you would want to set the MFA setting to “Required” as the data is highly secure.

**CORRECT:**"Amazon Cognito User Pool with MFA" is the correct answer.

**INCORRECT:** "Amazon Cognito Identity Pool with MFA" is incorrect

**INCORRECT:** "AWS IAM with MFA" is incorrect. With IAM your user accounts are maintained in your AWS account rather than in a Cognito User Pool. For logging into a web or mobile app it is better to create and manage your users in a Cognito User Pool and add MFA to the User Pool for extra security.

**INCORRECT:** "AWS Directory Service" is incorrect as this is a managed Active Directory service. For a web or mobile application using AWS Cognito User Pools is a better solution for storing your user accounts and authenticating to the application.

**References:**

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

<https://docs.aws.amazon.com/cognito/latest/developerguide/user-pool-settings-mfa.html>

**Save time with our AWS cheat sheets:**

<https://digitalcloud.training/amazon-cognito/>

**Domain**

AWS Security, Identity, & Compliance

**Question 30Correct**

A company needs to encrypt a large quantity of data. The data encryption keys must be generated from a dedicated, tamper-resistant hardware device.

To deliver these requirements, which AWS service should the company use?

**AWS Certificate Manager**

**AWS IAM**

**Your answer is correct**

**AWS CloudHSM**

**AWS KMS**

Overall explanation

The AWS CloudHSM service helps you meet corporate, contractual, and regulatory compliance requirements for data security by using dedicated Hardware Security Module (HSM) instances within the AWS cloud.

A Hardware Security Module (HSM) provides secure key storage and cryptographic operations within a tamper-resistant hardware device. CloudHSM allows you to securely generate, store, and manage cryptographic keys used for data encryption in a way that keys are accessible only by you.

**CORRECT:**"AWS CloudHSM" is the correct answer.

**INCORRECT:** "AWS KMS" is incorrect as it uses shared infrastructure (multi-tenant) and is therefore not a dedicated HSM.

**INCORRECT:** "AWS Certificate Manager" is incorrect as this is used to generate and manage SSL/TLS certificates, it does not generate data encryption keys.

**INCORRECT:** "AWS IAM" is incorrect as this service is not involved with generating encryption keys.

**References:**

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

**Save time with our AWS cheat sheets:**

<https://digitalcloud.training/aws-cloudhsm/>

**Domain**

AWS Security, Identity, & Compliance

**Question 31Correct**

A Developer has created an AWS Lambda function in a new AWS account. The function is expected to be invoked 40 times per second and the execution duration will be around 100 seconds. What MUST the Developer do to ensure there are no errors?

**Implement error handling within the function code**

**Implement tracing with X-Ray**

**Implement a Dead Letter Queue to capture invocation errors**

**Your answer is correct**

**Contact AWS Support to increase the concurrent execution limits**

Overall explanation

Concurrency is the number of requests that your function is serving at any given time. When your function is invoked, Lambda allocates an instance of it to process the event. When the function code finishes running, it can handle another request. If the function is invoked again while a request is still being processed, another instance is allocated, which increases the function's concurrency.

In this scenario the Lambda function will be invoked 40 times per second and run for 100 seconds. Therefore, there can be up to 4,000 executions running concurrently which is above the default per-region limit of 1,000 concurrent executions.

This can be easily rectified by contacting AWS support and requesting the concurrent execution limit to be increased.

**CORRECT:**"Contact AWS Support to increase the concurrent execution limits" is the correct answer.

**INCORRECT:** "Implement error handling within the function code" is incorrect. Though this could be useful it is not something that must be done based on what we know about this scenario.

**INCORRECT:** "Implement a Dead Letter Queue to capture invocation errors" is incorrect as this would be implemented for message handling requirements.

**INCORRECT:** "Implement tracing with X-Ray" is incorrect. X-Ray can be used to analyze and debug distributed applications. We don’t know of any specific issues with this function yet so this is not something that must be done.

**References:**

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

**Domain**

AWS Compute

**Question 32Correct**

An application writes items to an Amazon DynamoDB table. As the application scales to thousands of instances, calls to the DynamoDB API generate occasional ThrottlingException errors. The application is coded in a language that is incompatible with the AWS SDK.

What can be done to prevent the errors from occurring?

**Pass API calls through Amazon API Gateway**

**Use Amazon SQS as an API message bus**

**Your answer is correct**

**Add exponential backoff to the application logic**

**Send the items to DynamoDB through Amazon Kinesis Data Firehose**

Overall explanation

Implementing error retries and exponential backoff is a good way to resolve this issue. Exponential backoff can improve an application's reliability by using progressively longer waits between retries. If you're using an AWS SDK, this logic is built‑in. If you're not using an AWS SDK, consider manually implementing exponential backoff.

Additional options for preventing throttling from occurring include:

• Distribute read and write operations as evenly as possible across your table. A hot partition can degrade the overall performance of your table. For more information, see Designing Partition Keys to Distribute Your Workload Evenly.

• Implement a caching solution. If your workload is mostly read access to static data, then query results can be delivered much faster if the data is in a well‑designed cache rather than in a database. DynamoDB Accelerator (DAX) is a caching service that offers fast in‑memory performance for your application. You can also use Amazon ElastiCache.

**CORRECT:**"Add exponential backoff to the application logic" is the correct answer.

**INCORRECT:** "Use Amazon SQS as an API message bus" is incorrect. SQS is used for decoupling (messages, nut not APIs), however for this scenario it would add extra cost and complexity.

**INCORRECT:** "Pass API calls through Amazon API Gateway" is incorrect. For this scenario we don’t want to add an additional layer in when we can simply configure the application to back off and retry.

**INCORRECT:** "Send the items to DynamoDB through Amazon Kinesis Data Firehose" is incorrect as DynamoDB is not a supported destination for Kinesis Data Firehose.

**References:**

<https://docs.aws.amazon.com/general/latest/gr/api-retries.html>

<https://aws.amazon.com/premiumsupport/knowledge-center/dynamodb-table-throttled/>

**Save time with our AWS cheat sheets:**

<https://digitalcloud.training/amazon-dynamodb/>

**Domain**

AWS Database

**Question 33Correct**

An Amazon Kinesis Data Stream has recently been configured to receive data from sensors in a manufacturing facility. A consumer EC2 instance is configured to process the data every 48 hours and save processing results to an Amazon RedShift data warehouse. Testing has identified a large amount of data is missing. A review of monitoring logs has identified that the sensors are sending data correctly and the EC2 instance is healthy.

What is the MOST likely explanation for this issue?

**Amazon RedShift is not suitable for storing streaming data**

**Your answer is correct**

**Records are retained for 24 hours in the Kinesis Data Stream by default**

**Amazon Kinesis has too many shards provisioned**

**The EC2 instance is failing intermittently**

Overall explanation

Amazon Kinesis Data Streams supports changes to the data record retention period of your stream. A Kinesis data stream is an ordered sequence of data records meant to be written to and read from in real time. Data records are therefore stored in shards in your stream temporarily. The time period from when a record is added to when it is no longer accessible is called the retention period. A Kinesis data stream stores records from 24 hours by default, up to 8760 hours.

You can increase the retention period up to 8760 hours using the [IncreaseStreamRetentionPeriod](https://docs.aws.amazon.com/kinesis/latest/APIReference/API_IncreaseStreamRetentionPeriod.html) operation. You can decrease the retention period down to a minimum of 24 hours using the [DecreaseStreamRetentionPeriod](https://docs.aws.amazon.com/kinesis/latest/APIReference/API_DecreaseStreamRetentionPeriod.html) operation. The request syntax for both operations includes the stream name and the retention period in hours. Finally, you can check the current retention period of a stream by calling the [DescribeStream](https://docs.aws.amazon.com/kinesis/latest/APIReference/API_DescribeStream.html) operation.

Both operations are easy to use. The following is an example of changing the retention period using the AWS CLI:



Therefore, the most likely explanation is that the message retention period is set at the 24-hour default.

**CORRECT:**"Records are retained for 24 hours in the Kinesis Data Stream by default" is the correct answer.

**INCORRECT:** "Amazon RedShift is not suitable for storing streaming data" is incorrect. In this architecture Amazon Kinesis is responsible for receiving streaming data and storing it in a stream. The EC2 instances can then process and store the data in a number of different destinations including Amazon RedShift.

**INCORRECT:** "The EC2 instance is failing intermittently" is incorrect as the question states that a review of monitoring logs indicates that the EC2 instance is healthy. If it was failing intermittently this should be recorded in the logs.

**INCORRECT:** "Amazon Kinesis has too many shards provisioned" is incorrect as this would just mean that the Kinesis Stream has more capacity, not less.

**References:**

<https://docs.aws.amazon.com/streams/latest/dev/kinesis-extended-retention.html>

**Save time with our AWS cheat sheets:**

<https://digitalcloud.training/amazon-kinesis/>

**Domain**

AWS Application Integration

**Question 34Incorrect**

Data must be loaded into an application each week for analysis. The data is uploaded to an Amazon S3 bucket from several offices around the world. Latency is slowing the uploads and delaying the analytics job. What is the SIMPLEST way to improve upload times?

**Upload to Amazon CloudFront and then download from the local cache to the S3 bucket**

**Correct answer**

**Upload using Amazon S3 Transfer Acceleration**

**Upload via a managed AWS VPN connection**

**Your answer is incorrect**

**Upload to a local Amazon S3 bucket within each region and enable Cross-Region Replication (CRR)**

Overall explanation

Amazon S3 Transfer Acceleration enables fast, easy, and secure transfers of files over long distances between your client and an S3 bucket. Transfer Acceleration takes advantage of Amazon CloudFront’s globally distributed edge locations. As the data arrives at an edge location, data is routed to Amazon S3 over an optimized network path.

You might want to use Transfer Acceleration on a bucket for various reasons, including the following:

• You have customers that upload to a centralized bucket from all over the world.

• You transfer gigabytes to terabytes of data on a regular basis across continents.

• You are unable to utilize all of your available bandwidth over the Internet when uploading to Amazon S3.

**CORRECT:**"Upload using Amazon S3 Transfer Acceleration" is the correct answer.

**INCORRECT:** "Upload to a local Amazon S3 bucket within each region and enable Cross-Region Replication (CRR)" is incorrect as this would not speed up the upload as the process introduces more latency.

**INCORRECT:** "Upload via a managed AWS VPN connection" is incorrect as this still uses the public Internet and there’s no real latency advantages here.

**INCORRECT:** "Upload to Amazon CloudFront and then download from the local cache to the S3 bucket" is incorrect. This is going to require some time to propagate to the cache and requires some manual work in retrieving the data. The simplest solution is to use S3 Transfer Acceleration which basically does this for you.

**References:**

<https://docs.aws.amazon.com/AmazonS3/latest/dev/transfer-acceleration.html>

**Save time with our AWS cheat sheets:**

<https://digitalcloud.training/amazon-s3-and-glacier/>

**Domain**

AWS Storage

**Question 35Correct**

A software organization has developed a new feature in its serverless application hosted on AWS. This feature involves an AWS Lambda function that gets invoked by an Amazon API Gateway API. Currently, the API uses a specific Lambda alias to invoke the Lambda function. The organization wants to roll out this new feature to a select group of users for beta testing without affecting the application's existing users.

What would be the most efficient approach to meet these requirements?

**Use the AWS CLI to manually switch between the old and new versions of the Lambda function during testing periods.**

**Implement Amazon S3 bucket versioning on the Lambda function code. For testing purposes, link the API Gateway to the new version of the code stored in the S3 bucket.**

**Use AWS CodeDeploy to deploy the updated Lambda function. Split traffic between the new and old versions of the function for testing purposes.**

**Your answer is correct**

**Create a new version of the Lambda function. Build a new stage on API Gateway integrated with this new Lambda version. Utilize this new API Gateway stage for beta testing.**

Overall explanation

In AWS, Lambda versions are an essential part of the deployment process. Once a version is published, the code within that version cannot be changed. It is ideal for ensuring stability and reliability between deployments.

API Gateway Stages can be used to route traffic to different Lambda versions. Hence, creating a new API Gateway stage for beta testing that is linked to the new version of the Lambda function will ensure that existing users (using the old stage) are not impacted by the new feature.

**CORRECT:**"Create a new version of the Lambda function. Build a new stage on API Gateway integrated with this new Lambda version. Utilize this new API Gateway stage for beta testing" is the correct answer (as explained above.)

**INCORRECT:** "Implement Amazon S3 bucket versioning on the Lambda function code. For testing purposes, link the API Gateway to the new version of the code stored in the S3 bucket" is incorrect.

While Amazon S3 versioning allows for keeping multiple variants of an object in the same bucket, it doesn't interact directly with AWS Lambda or API Gateway. Thus, you cannot control API Gateway to invoke different versions of Lambda function code stored in an S3 bucket.

**INCORRECT:** "Use AWS CodeDeploy to deploy the updated Lambda function. Split traffic between the new and old versions of the function for testing purposes" is incorrect.

AWS CodeDeploy can deploy new versions of Lambda functions and shift traffic gradually but using it for testing purposes might risk exposing beta features to the regular users, potentially impacting their experience.

**INCORRECT:** "Use the AWS CLI to manually switch between the old and new versions of the Lambda function during testing periods" is incorrect.

This approach is operationally intensive, prone to human error, and may impact regular users. AWS provides better ways of managing Lambda versions and aliases, which should be used instead of manual operations.

**References:**

<https://docs.aws.amazon.com/apigateway/latest/developerguide/set-up-stages.html>

**Save time with our AWS cheat sheets:**

<https://digitalcloud.training/aws-lambda/>

**Domain**

AWS Compute

**Question 36Incorrect**

An AWS Lambda function must be connected to an Amazon VPC private subnet that does not have Internet access. The function also connects to an Amazon DynamoDB table. What MUST a Developer do to enable access to the DynamoDB table?

**Attach an ENI to the DynamoDB table**

**Correct answer**

**Configure a VPC endpoint**

**Attach an Internet Gateway**

**Your answer is incorrect**

**Create a route table**

Overall explanation

To connect to AWS services from a private subnet with no internet access, use VPC endpoints. A VPC endpoint for DynamoDB enables resources in a VPC to use their private IP addresses to access DynamoDB with no exposure to the public internet.

When you create a VPC endpoint for DynamoDB, any requests to a DynamoDB endpoint within the Region (for example, dynamodb.us-west-2.amazonaws.com) are routed to a private DynamoDB endpoint within the Amazon network.

**CORRECT:**"Configure a VPC endpoint" is the correct answer.

**INCORRECT:** "Attach an Internet Gateway" is incorrect as you do not attach these to a private subnet.

**INCORRECT:** "Create a route table" is incorrect as a route table will exist for all subnets and it does not help to route out from a private subnet via the Internet unless an entry for a NAT Gateway or Instance is added.

**INCORRECT:** "Attach an ENI to the DynamoDB table" is incorrect as you do not attach Elastic Network Interfaces to DynamoDB tables.

**References:**

<https://docs.aws.amazon.com/lambda/latest/dg/troubleshooting-networking.html>

<https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/vpc-endpoints-dynamodb.html>

**Save time with our AWS cheat sheets:**

<https://digitalcloud.training/amazon-vpc/>

**Domain**

AWS Networking & Content Delivery

**Question 37Correct**

A multimedia streaming service wants to migrate its user authentication system to AWS. The system keeps user session data during an active session, which is crucial for seamless user experience. The new system needs to be fault-tolerant, highly scalable natively, and any service disruption must not affect user experience.

What is the best option to store the user session data?

**Store the user session data in Amazon S3.**

**Your answer is correct**

**Store the user session data in Amazon ElastiCache.**

**Enable session stickiness using elastic load balancers.**

**Store the user session data in Amazon CloudFront.**

Overall explanation

ElastiCache is a fully managed in-memory caching service that provides a high-performance, scalable, and cost-effective caching solution, while removing the complexity associated with deploying and managing a distributed cache environment. It is well-suited for storing user session data due to its high-speed and low-latency capabilities.

**CORRECT:**"Store the user session data in Amazon ElastiCache" is the correct answer (as explained above.)

**INCORRECT:** "Store the user session data in Amazon CloudFront" is incorrect.

CloudFront is a fast content delivery network (CDN) service that securely delivers data, videos, applications, and APIs to customers globally with low latency and high transfer speeds. However, it is not designed for session state storage and thus, would not be the best choice for storing user session data.

**INCORRECT:** "Store the user session data in Amazon S3" is incorrect.

Amazon S3 provides durable, scalable object storage but is not optimized for the type of high-speed, read/write operations typically associated with session data. While you could technically use it to store session data, it would likely not provide the same level of performance as an in-memory store like ElastiCache.

**INCORRECT:** "Enable session stickiness using elastic load balancers" is incorrect.

Enabling session stickiness in load balancers can help ensure that a client is consistently directed to the same instance if that instance is healthy. However, this is more of a traffic routing mechanism rather than a storage solution for session data.

**References:**

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

**Save time with our AWS cheat sheets:**

<https://digitalcloud.training/amazon-elasticache/>

**Domain**

AWS Database

**Question 38Correct**

How can a Developer view a summary of proposed changes to an AWS CloudFormation stack without implementing the changes in production?

**Create a StackSet**

**Use drift detection**

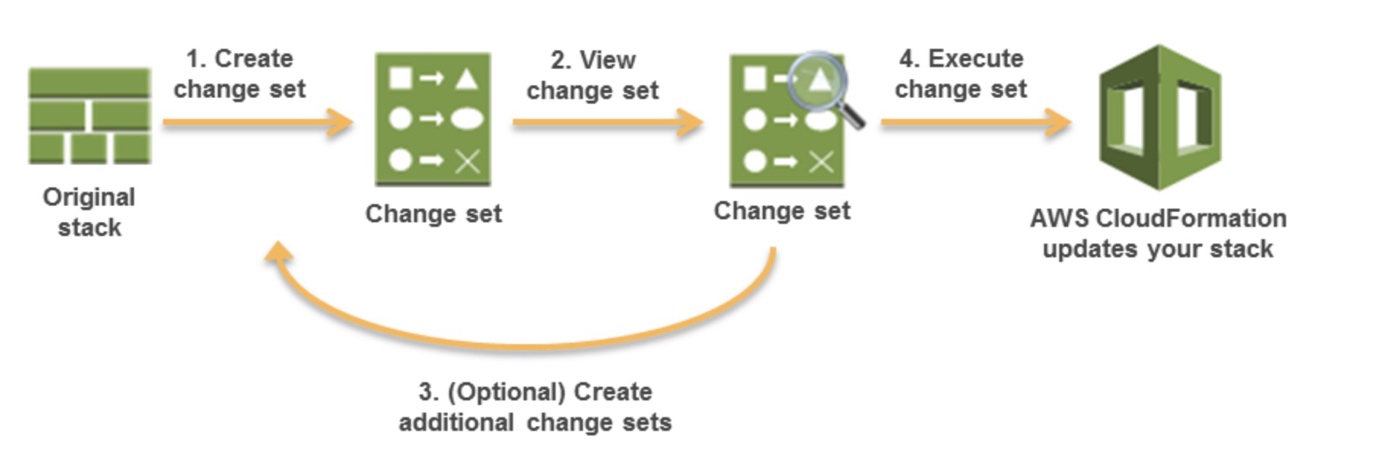
**Your answer is correct**

**Create a Change Set**

**Use a direct update**

Overall explanation

When you need to update a stack, understanding how your changes will affect running resources before you implement them can help you update stacks with confidence. Change sets allow you to preview how proposed changes to a stack might impact your running resources.



AWS CloudFormation makes the changes to your stack only when you decide to execute the change set, allowing you to decide whether to proceed with your proposed changes or explore other changes by creating another change set. You can create and manage change sets using the AWS CloudFormation console, AWS CLI, or AWS CloudFormation API.

**CORRECT:**"Create a Change Set" is the correct answer.

**INCORRECT:** "Create a StackSet" is incorrect as StackSets are used to create, update, or delete stacks across multiple accounts and regions with a single operation.

**INCORRECT:** "Use drift detection" is incorrect as this is used to detect when a configuration deviates from the template configuration.

**INCORRECT:** "Use a direct update" is incorrect as this will directly update the production resources.

**References:**

<https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/using-cfn-updating-stacks-changesets.html>

**Save time with our AWS cheat sheets:**

<https://digitalcloud.training/aws-cloudformation/>

**Domain**

AWS Management & Governance

**Question 39Correct**

A Developer is managing an application that includes an Amazon SQS queue. The consumers that process the data from the queue are connecting in short cycles and the queue often does not return messages. The cost for API calls is increasing. How can the Developer optimize the retrieval of messages and reduce cost?

**Call the ReceiveMessage API with the VisibilityTimeout parameter set to 30**

**Your answer is correct**

**Call the ReceiveMessage API with the WaitTimeSeconds parameter set to 20**

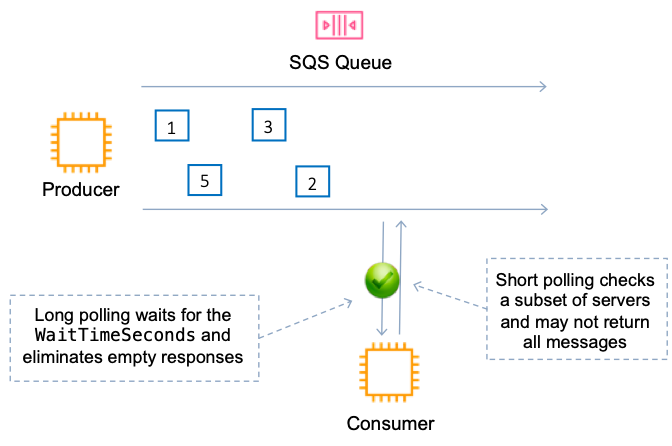
**Call the SetQueueAttributes API with the maxReceiveCount set to 20**

**Call the SetQueueAttributes API with the DelaySeconds parameter set to 900**

Overall explanation

The process of consuming messages from a queue depends on whether you use short or long polling. By default, Amazon SQS uses short polling, querying only a subset of its servers (based on a weighted random distribution) to determine whether any messages are available for a response. You can use long polling to reduce your costs while allowing your consumers to receive messages as soon as they arrive in the queue.

When you consume messages from a queue using short polling, Amazon SQS samples a subset of its servers (based on a weighted random distribution) and returns messages from only those servers. Thus, a particular [ReceiveMessage](https://docs.aws.amazon.com/AWSSimpleQueueService/latest/APIReference/API_ReceiveMessage.html) request might not return all of your messages. However, if you have fewer than 1,000 messages in your queue, a subsequent request will return your messages. If you keep consuming from your queues, Amazon SQS samples all of its servers, and you receive all of your messages.



When the wait time for the ReceiveMessage API action is greater than 0, long polling is in effect. The maximum long polling wait time is 20 seconds. Long polling helps reduce the cost of using Amazon SQS by eliminating the number of empty responses (when there are no messages available for a [ReceiveMessage](https://docs.aws.amazon.com/AWSSimpleQueueService/latest/APIReference/API_ReceiveMessage.html) request) and false empty responses (when messages are available but aren't included in a response)

Therefore, the Developer should call the ReceiveMessage API with the WaitTimeSeconds parameter set to 20 to enable long polling.

**CORRECT:**"Call the ReceiveMessage API with the WaitTimeSeconds parameter set to 20 " is the correct answer.

**INCORRECT:** "Call the ReceiveMessage API with the VisibilityTimeout parameter set to 30" is incorrect

**INCORRECT:** "Call the SetQueueAttributes API with the DelaySeconds parameter set to 900" is incorrect

**INCORRECT:** "Call the SetQueueAttributes API with the maxReceiveCount set to 20" is incorrect

**References:**

<https://docs.aws.amazon.com/AWSSimpleQueueService/latest/SQSDeveloperGuide/sqs-short-and-long-polling.html>

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

**Save time with our AWS cheat sheets:**

<https://digitalcloud.training/aws-application-integration-services/>

**Domain**

AWS Application Integration

**Question 40Correct**

An application scans an Amazon DynamoDB table once per day to produce a report. The scan is performed in non-peak hours when production usage uses around 50% of the provisioned throughput.

How can you MINIMIZE the time it takes to produce the report without affecting production workloads? (Select TWO.)

**Use a Sequential Scan API operation**

**Your selection is correct**

**Use the Limit parameter**

**Increase read capacity units during the scan operation**

**Use pagination to divide results into 1 MB pages**

**Your selection is correct**

**Use a Parallel Scan API operation**

Overall explanation

By default, the Scan operation processes data sequentially. Amazon DynamoDB returns data to the application in 1 MB increments, and an application performs additional Scan operations to retrieve the next 1 MB of data.

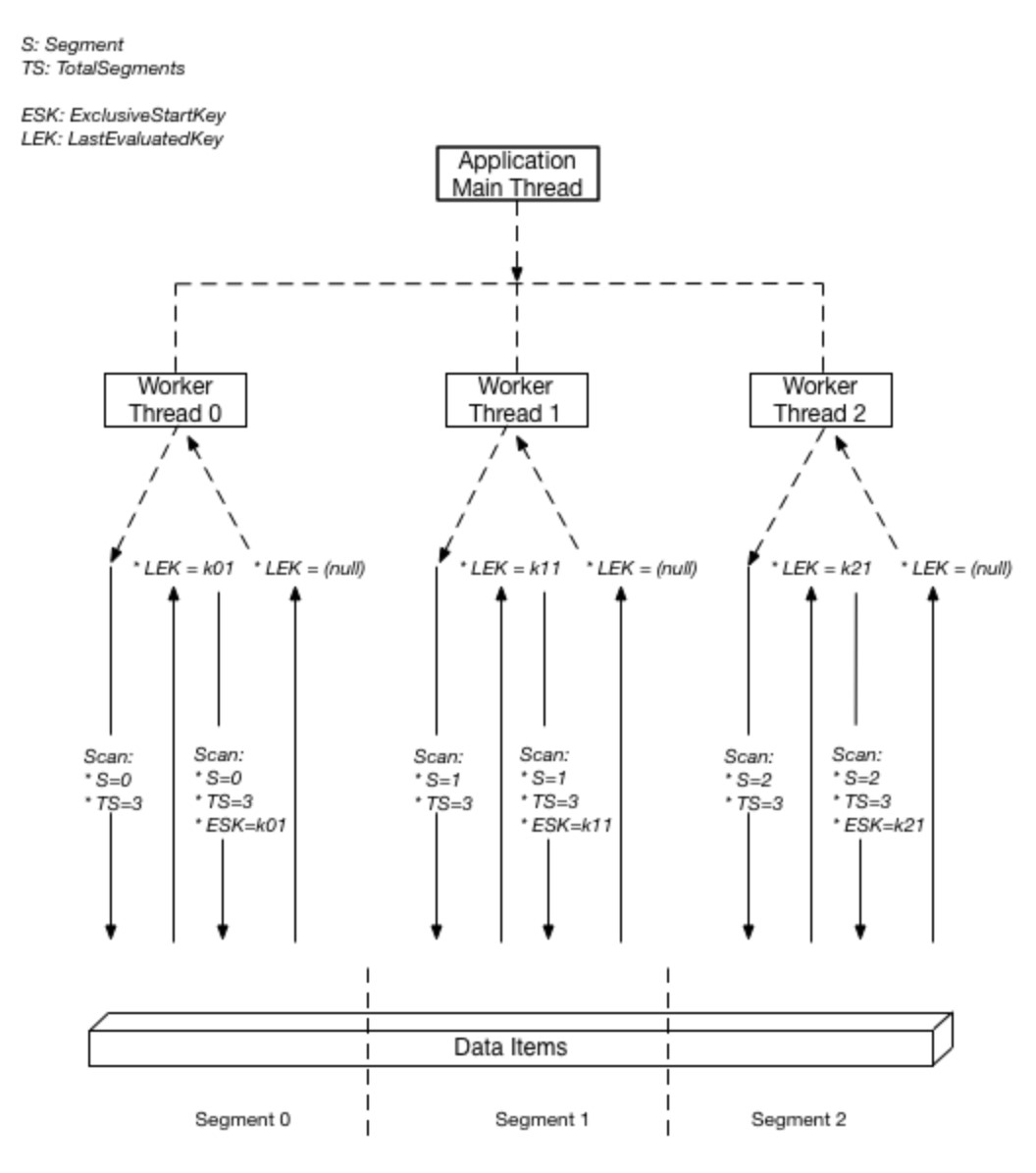
The larger the table or index being scanned, the more time the Scan takes to complete. In addition, a sequential Scan might not always be able to fully use the provisioned read throughput capacity: Even though DynamoDB distributes a large table's data across multiple physical partitions, a Scan operation can only read one partition at a time. For this reason, the throughput of a Scan is constrained by the maximum throughput of a single partition.

To address these issues, the Scan operation can logically divide a table or secondary index into multiple segments, with multiple application workers scanning the segments in parallel. Each worker can be a thread (in programming languages that support multithreading) or an operating system process. To perform a parallel scan, each worker issues its own Scan request with the following parameters:

• Segment — A segment to be scanned by a particular worker. Each worker should use a different value for Segment.

• TotalSegments — The total number of segments for the parallel scan. This value must be the same as the number of workers that your application will use.

The following diagram shows how a multithreaded application performs a parallel Scan with three degrees of parallelism.



To make the most of your table’s provisioned throughput, you’ll want to use the Parallel Scan API operation so that your scan is distributed across your table’s partitions. However, you also need to ensure the scan doesn’t consume your table’s provisioned throughput and cause the critical parts of your application to be throttled.

To control the amount of data returned per request, use the Limit parameter. This can help prevent situations where one worker consumes all of the provisioned throughput, at the expense of all other workers.

Therefore, the best solution to this problem is to use a parallel scan API operation with the Limit parameter.

**CORRECT:**"Use a Parallel Scan API operation " is the correct answer.

**CORRECT:** "Use the Limit parameter" is also a correct answer.

**INCORRECT:** "Use a Sequential Scan API operation" is incorrect as this would take more time and the question requests that we minimize the time it takes to complete the scan.

**INCORRECT:** "Increase read capacity units during the scan operation" is incorrect as this would increase cost and we still need a solution to ensure we maximize usage of available throughput without affecting production workloads.

**INCORRECT:** "Use pagination to divide results into 1 MB pages" is incorrect as this does only divides the results into pages, it does not segment and limit the amount of throughput used.

**References:**

<https://aws.amazon.com/blogs/developer/rate-limited-scans-in-amazon-dynamodb/>

<https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/Scan.html#Scan.ParallelScan>

**Save time with our AWS cheat sheets:**

<https://digitalcloud.training/amazon-dynamodb/>

**Domain**

AWS Database

**Question 41Correct**

A Developer must run a shell script on Amazon EC2 Linux instances each time they are launched by an Amazon EC2 Auto Scaling group. What is the SIMPLEST way to run the script?

**Run the script using the AWS Systems Manager Run Command**

**Package the script in a zip file with some AWS Lambda source code. Upload to Lambda and run the function when instances are launched**

**Configure Amazon CloudWatch Events to trigger the AWS CLI when an instance is launched and run the script**

**Your answer is correct**

**Add the script to the user data when creating the launch configuration**

Overall explanation

The simplest option is to add the script to the user data when creating the launch configuration. User data is information that is parsed when the EC2 instances are launched. When you add a script to the user data in a launch configuration all instances that are launched by that Auto Scaling group will run the script.

**CORRECT:**"Add the script to the user data when creating the launch configuration" is the correct answer.

**INCORRECT:** "Configure Amazon CloudWatch Events to trigger the AWS CLI when an instance is launched and run the script" is incorrect as you cannot trigger the AWS CLI using CloudWatch Events and the script may not involve AWS CLI commands.

**INCORRECT:** "Package the script in a zip file with some AWS Lambda source code. Upload to Lambda and run the function when instances are launched" is incorrect as Lambda does not run shell scripts. You could program the requirements into the function code however you still need a trigger which is not mentioned in this option.

**INCORRECT:** "Run the script using the AWS Systems Manager Run Command" is incorrect as this is not the simplest method. For most Linux AMIs (except Amazon Linux) the developer’s would need to install the agent on the operating system. They would also then need to create a mechanism of triggering the run command.

**References:**

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

**Save time with our AWS cheat sheets:**

<https://digitalcloud.training/amazon-ec2/>

**Domain**

AWS Compute

**Question 42Correct**

A Development team have moved their continuous integration and delivery (CI/CD) pipeline into the AWS Cloud. The team is leveraging AWS CodeCommit for management of source code. The team need to compile their source code, run tests, and produce software packages that are ready for deployment.

Which AWS service can deliver these outcomes?

**Your answer is correct**

**AWS CodeBuild**

**AWS CodeCommit**

**AWS CodePipeline**

**AWS Cloud9**

Overall explanation

AWS CodeBuild is a fully managed build service in the cloud. CodeBuild compiles your source code, runs unit tests, and produces artifacts that are ready to deploy. CodeBuild eliminates the need to provision, manage, and scale your own build servers. It provides prepackaged build environments for popular programming languages and build tools such as Apache Maven, Gradle, and more.

You can also customize build environments in CodeBuild to use your own build tools. CodeBuild scales automatically to meet peak build requests.

CodeBuild provides these benefits:

• **Fully managed** – CodeBuild eliminates the need to set up, patch, update, and manage your own build servers.

• **On demand** – CodeBuild scales on demand to meet your build needs. You pay only for the number of build minutes you consume.

• **Out of the box** – CodeBuild provides preconfigured build environments for the most popular programming languages. All you need to do is point to your build script to start your first build.

Therefore, AWS CodeBuild is the best service to use to compile the Development team’s source code, run tests, and produce software packages that are ready for deployment.

**CORRECT:**"AWS CodeBuild" is the correct answer.

**INCORRECT:** "AWS CodeCommit" is incorrect. The team are already using CodeCommit for its correct purpose, which is to manage source code. CodeCommit cannot perform compiling of source code, testing, or package creation.

**INCORRECT:** "AWS CodePipeline" is incorrect. AWS CodePipeline is a fully managed continuous delivery service that helps you automate your release pipelines for fast and reliable application and infrastructure updates.

**INCORRECT:** "AWS Cloud9" is incorrect. AWS Cloud9 is a cloud-based integrated development environment (IDE) that lets you write, run, and debug your code with just a browser.

**References:**

<https://docs.aws.amazon.com/codebuild/latest/userguide/welcome.html>

**Save time with our AWS cheat sheets:**

<https://digitalcloud.training/aws-developer-tools/>

**Domain**

AWS Developer Tools

**Question 43Incorrect**

An online multiplayer game employs Amazon API Gateway WebSocket APIs with an HTTP backend. The game developer needs to add a feature that identifies players with unstable connections who repeatedly join and leave the game. The developer also wants the ability to disconnect such players from the game.

What two modifications should the developer implement in the game to fulfill these requirements? (Select TWO.)

**Your selection is correct**

**Add logic to track the player's connection status using Amazon DynamoDB in the backend service.**

**Switch to REST APIs in the backend service.**

**Switch to AWS App Runner for the backend service.**

**Correct selection**

**Implement $connect and $disconnect routes in the backend service.**

**Your selection is incorrect**

**Implement AWS Cognito for player authentication in the backend service.**

Overall explanation

By implementing $connect and $disconnect routes in the backend service, you can capture events when a player connects and disconnects from the WebSocket API. This enables tracking and managing of players' connection statuses.

Using Amazon DynamoDB (or another database service) allows you to persist and track the connection status of each player in real-time. This enables you to identify players who connect and disconnect frequently.

**CORRECT:**"Implement $connect and $disconnect routes in the backend service" is a correct answer (as explained above.)

**CORRECT:**"Add logic to track the player's connection status using Amazon DynamoDB in the backend service" is also a correct answer (as explained above.)

**INCORRECT:** "Switch to REST APIs in the backend service" is incorrect.

Switching to REST APIs won't inherently provide the ability to manage or track unstable connections. WebSocket APIs are more suited for applications requiring real-time, two-way communication.

**INCORRECT:** "Implement AWS Cognito for player authentication in the backend service" is incorrect.

While AWS Cognito provides authentication and user management, it doesn't offer features for tracking and managing unstable connections.

**INCORRECT:** "Switch to AWS App Runner for the backend service" is incorrect.

AWS App Runner is a service that makes it easy to build, deploy, and scale containerized applications quickly, but it doesn't directly address the specific requirement of tracking and managing unstable connections.

**References:**

<https://docs.aws.amazon.com/apigateway/latest/developerguide/apigateway-websocket-api-route-keys-connect-disconnect.html>

**Save time with our AWS cheat sheets:**

<https://digitalcloud.training/amazon-api-gateway/>

**Domain**

AWS Networking & Content Delivery

**Question 44Correct**

A developer has a user account in the Development AWS account. He has been asked to modify resources in a Production AWS account. What is the MOST secure way to provide temporary access to the developer?

**Generate an access key on the second account using the root account and share the access keys with the developer for API access**

**Use AWS KMS to generate cross-account customer master keys and use those get short-lived credentials**

**Add the user to a group in the second account that has a role attached granting the necessary permissions**

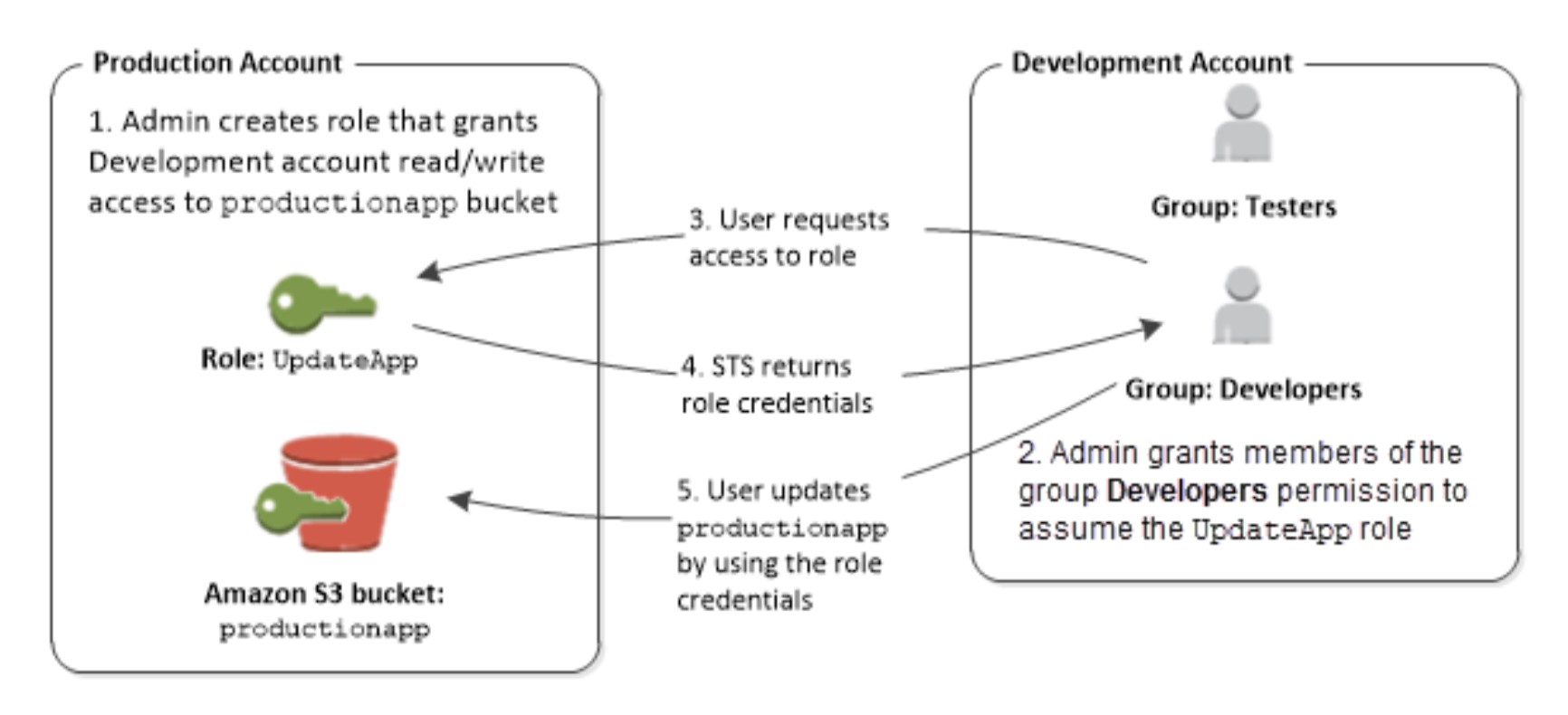
**Your answer is correct**

**Create a cross-account access role, and use sts:AssumeRole API to get short-lived credentials**

Overall explanation

This should be implemented using a role in the Production account and a group in the Development account. The developer in the Development account would then be added to the group. The role in the Production account would provide the necessary access and would allow the group in the Development account to assume the role.

The following image depicts this setup:



Therefore, the most secure way to achieve the required access is to use a role in the Production account that the user is able to assume and then the user can request short-lived credentials from the Security Token Service (STS).

**CORRECT:**"Create a cross-account access role, and use sts:AssumeRole API to get short-lived credentials" is the correct answer.

**INCORRECT:** "Generate an access key on the second account using the root account and share the access keys with the developer for API access" is incorrect as this is highly insecure. You should never share access keys across user accounts, and you should especially not use access keys associated with the root account.

**INCORRECT:** "Add the user to a group in the second account that has a role attached granting the necessary permissions" is incorrect as you cannot add a user to a group in a different AWS account.

**INCORRECT:** "Use AWS KMS to generate cross-account customer master keys and use those get short-lived credentials" is incorrect as you do not use AWS KMS CMKs for obtaining short-lived credentials from the STS service. CMKs are used for encrypting data.

**References:**

<https://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles_common-scenarios_aws-accounts.html>

**Save time with our AWS cheat sheets:**

<https://digitalcloud.training/aws-iam/>

**Domain**

AWS Security, Identity, & Compliance

**Question 45Incorrect**

A Developer needs to setup a new serverless application that includes AWS Lambda and Amazon API Gateway as part of a single stack. The Developer needs to be able to locally build and test the serverless applications before deployment on AWS.

Which service should the Developer use?

**AWS CloudFormation**

**Correct answer**

**AWS Serverless Application Model (SAM)**

**Your answer is incorrect**

**AWS CodeBuild**

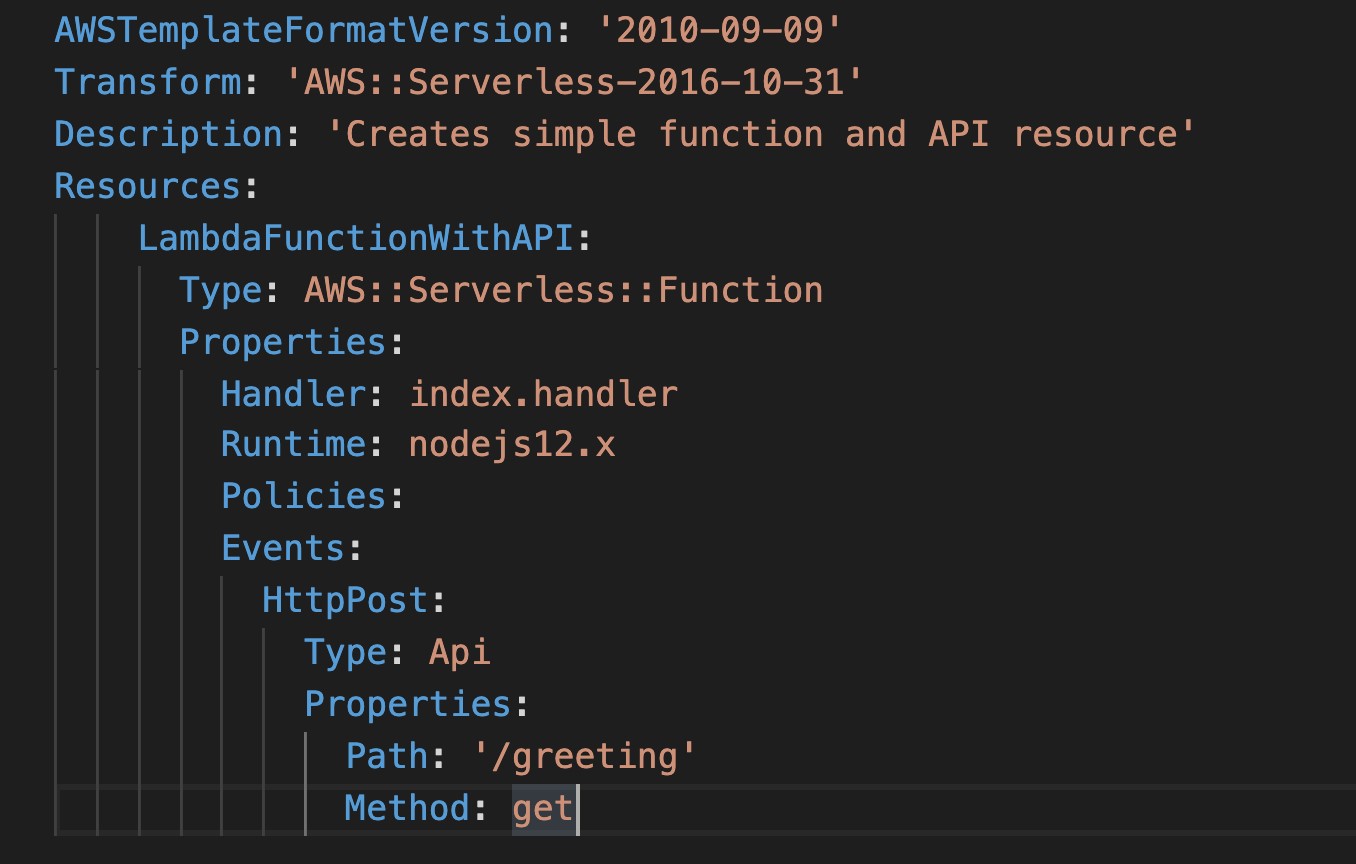
**AWS Elastic Beanstalk**

Overall explanation

The AWS Serverless Application Model (AWS SAM) is an open-source framework that you can use to build [serverless applications](https://aws.amazon.com/serverless/) on AWS. A **serverless application** is a combination of Lambda functions, event sources, and other resources that work together to perform tasks.

AWS SAM provides you with a simple and clean syntax to describe the functions, APIs, permissions, configurations, and events that make up a serverless application.

The example AWS SAM template file below creates an AWS Lambda function and a simple Amazon API Gateway API with a Get method and a /greeting resource:



The AWS SAM CLI lets you locally build, test, and debug serverless applications that are defined by AWS SAM templates. The CLI provides a Lambda-like execution environment locally. It helps you catch issues upfront by providing parity with the actual Lambda execution environment.

**CORRECT:**"AWS Serverless Application Model (SAM)" is the correct answer.

**INCORRECT:** "AWS CloudFormation" is incorrect as you cannot perform local build and test with AWS CloudFormation.

**INCORRECT:** "AWS Elastic Beanstalk" is incorrect as you cannot deploy serverless applications or perform local build and test with Elastic Beanstalk.

**INCORRECT:** "AWS CodeBuild" is incorrect as you cannot perform local build and test with AWS CodeBuild.

**References:**

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

**Save time with our AWS cheat sheets:**

<https://digitalcloud.training/aws-sam/>

**Domain**

AWS Compute

**Question 46Correct**

A Developer manages a website running behind an Elastic Load Balancer in the us-east-1 region. The Developer has recently deployed an identical copy of the website in us-west-1 and needs to send 20% of the traffic to the new site.

How can the Developer achieve this requirement?

**Your answer is correct**

**Use an Amazon Route 53 Weighted Routing Policy**

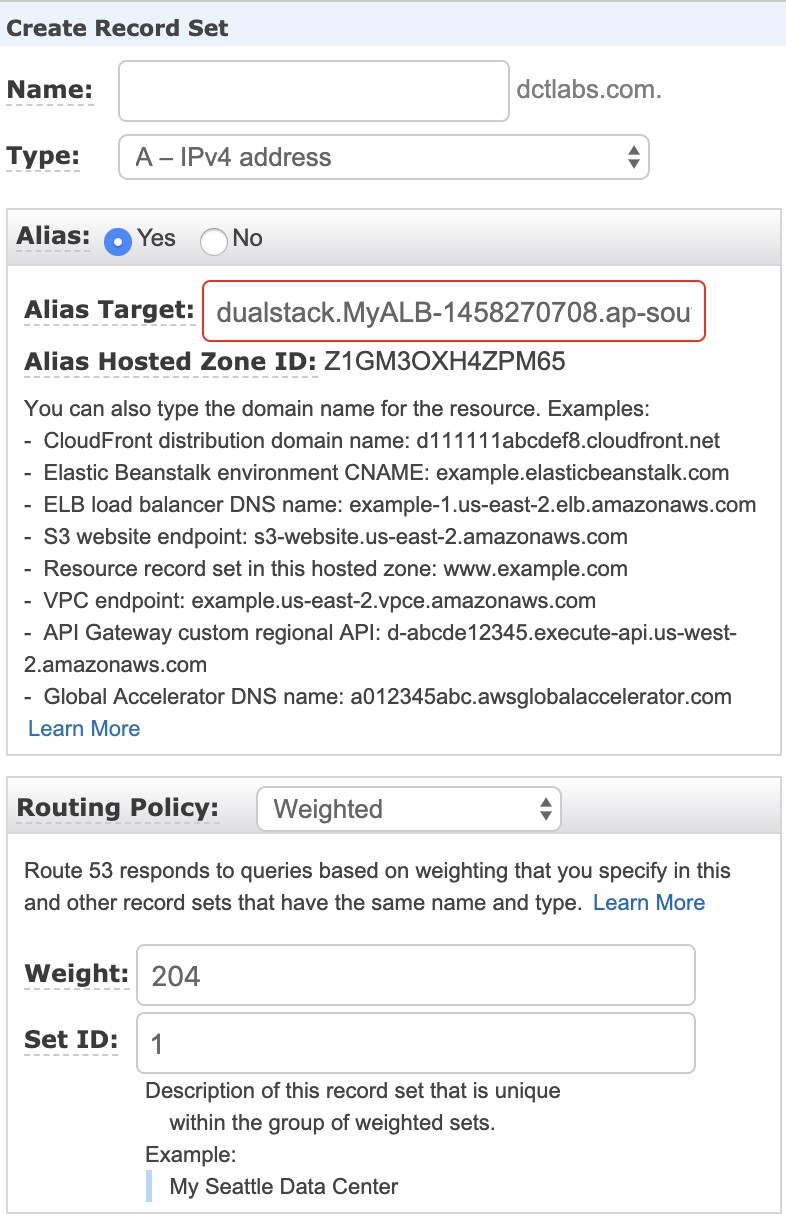
**Use a blue/green deployment with Amazon Elastic Beanstalk**

**Use a blue/green deployment with Amazon CodeDeploy**

**Use an Amazon Route 53 Geolocation Routing Policy**

Overall explanation

Weighted routing lets you associate multiple resources with a single domain name (example.com) or subdomain name (acme.example.com) and choose how much traffic is routed to each resource. This can be useful for a variety of purposes, including load balancing and testing new versions of software.



In this case the Developer can use a weighted routing policy to direct 20% of the incoming traffic to the new site as required.

**CORRECT:**"Use an Amazon Route 53 Weighted Routing Policy" is the correct answer.

**INCORRECT:** "Use an Amazon Route 53 Geolocation Routing Policy" is incorrect as the Developer should use a weighted routing policy for this requirement as a specified percentage of traffic needs to be directed to the new website.

**INCORRECT:** "Use a blue/green deployment with Amazon Elastic Beanstalk" is incorrect as the question does not state that Elastic Beanstalk is being used and the new website has already been deployed.

**INCORRECT:** "Use a blue/green deployment with Amazon CodeDeploy" is incorrect as the question does not state that Amazon CodeDeploy is being used and the website has already been deployed.

**References:**

<https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/routing-policy.html#routing-policy-weighted>

**Save time with our AWS cheat sheets:**

<https://digitalcloud.training/amazon-route-53/>

**Domain**

AWS Networking & Content Delivery

**Question 47Correct**

A Developer has created a task definition that includes the following JSON code:

1. "placementConstraints": [
2. {
3. "expression": "task:group == databases",
4. "type": "memberOf"
5. }
6. ]

What will be the effect for tasks using this task definition?

**They will not be placed on container instances in the “databases” task group**

**They will not be allowed to run unless they have the “databases” tag assigned**

**They will become members of a task group called “databases”**

**Your answer is correct**

**They will be placed on container instances in the “databases” task group**

Overall explanation

A task placement constraint is a rule that is considered during task placement. Task placement constraints can be specified when either running a task or creating a new service.

The memberOf task placement constraint places tasks on container instances that satisfy an expression.

The memberOf task placement constraint can be specified with the following actions:

• Running a task

• Creating a new service

• Creating a new task definition

• Creating a new revision of an existing task definition

The example JSON code uses the memberOf constraint to place tasks on instances in the databases task group. It can be specified with the following actions: [CreateService](https://docs.aws.amazon.com/AmazonECS/latest/APIReference/API_CreateService.html), [UpdateService](https://docs.aws.amazon.com/AmazonECS/latest/APIReference/API_UpdateService.html), [RegisterTaskDefinition](https://docs.aws.amazon.com/AmazonECS/latest/APIReference/API_RegisterTaskDefinition.html), and [RunTask](https://docs.aws.amazon.com/AmazonECS/latest/APIReference/API_RunTask.html).

**CORRECT:**"They will be placed on container instances in the “databases” task group" is the correct answer.

**INCORRECT:** "They will become members of a task group called “databases”" is incorrect. They will be placed on container instances in the “databases” task group.

**INCORRECT:** "They will not be placed on container instances in the “databases” task group" is incorrect. This statement ensures the tasks ARE placed on the container instances in the “databases” task group.

**INCORRECT:** "They will not be allowed to run unless they have the “databases” tag assigned" is incorrect. This JSON code is not related to tagging of the tasks.

**References:**

<https://docs.aws.amazon.com/AmazonECS/latest/developerguide/task-placement-constraints.html>

**Save time with our AWS cheat sheets:**

<https://digitalcloud.training/amazon-ecs-and-eks/>

**Domain**

AWS Compute

**Question 48Incorrect**

A developer is creating a serverless application that will use a DynamoDB table. The average item size is 9KB. The application will make 4 strongly consistent reads/sec, and 2 standard write/sec. How many RCUs/WCUs are required?

**Your answer is incorrect**

**24 RCU and 18 WCU**

**6 RCU and 18 WCU**

**12 RCU and 36 WCU**

**Correct answer**

**12 RCU and 18 WCU**

Overall explanation

With provisioned capacity mode, you specify the number of data reads and writes per second that you require for your application.

**Read capacity unit (RCU):**

• Each API call to read data from your table is a read request.

• Read requests can be strongly consistent, eventually consistent, or transactional.

• For items up to 4 KB in size, one RCU can perform one strongly consistent read request per  second.

• Items larger than 4 KB require additional RCUs.

For items up to 4 KB in size, one RCU can perform two eventually consistent read requests per second.

Transactional read requests require two RCUs to perform one read per second for items up to 4 KB.

For example, a strongly consistent read of an 8 KB item would require two RCUs, an eventually consistent read of an 8 KB item would require one RCU, and a transactional read of an 8 KB item would require four RCUs.

**Write capacity unit (WCU):**

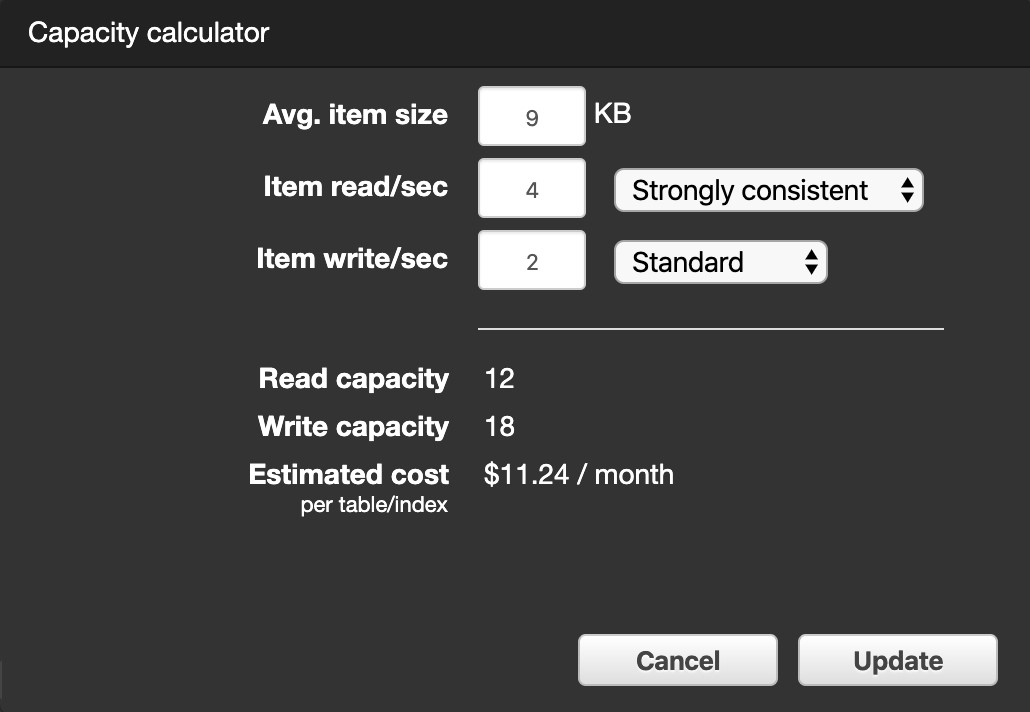
• Each API call to write data to your table is a write request.

For items up to 1 KB in size, one WCU can perform one standard write request per second.

Items larger than 1 KB require additional WCUs.

• Transactional write requests require two WCUs to perform one write per second for items up to 1 KB.

• For example, a standard write request of a 1 KB item would require one WCU, a standard write request of a 3 KB item would require three WCUs, and a transactional write request of a 3 KB item would require six WCUs.



**To determine the number of RCUs required to handle 4 strongly consistent reads per/second with an average item size of 9KB, perform the following steps:**

**1. Determine the average item size by rounding up the next multiple of 4KB (9KB rounds up to 12KB).**

**2. Determine the RCU per item by dividing the item size by 4KB (12KB/4KB = 3).**

**3. Multiply the value from step 2 with the number of reads required per second (3x4 = 12).**

**To determine the number of WCUs required to handle 2 standard writes per/second with an average item size of 9KB, simply multiply the average item size by the number of writes required (9x2=18).**

**CORRECT:**"12 RCU and 18 WCU" is the correct answer.

**INCORRECT:** "24 RCU and 18 WCU" is incorrect. This would be the correct answer for transactional reads and standard writes.

**INCORRECT:** "12 RCU and 36 WCU" is incorrect. This would be the correct answer for strongly consistent reads and transactional writes.

**INCORRECT:** "6 RCU and 18 WCU" is incorrect. This would be the correct answer for eventually consistent reads and standard writes

**References:**

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

**Save time with our AWS cheat sheets:**

<https://digitalcloud.training/amazon-dynamodb/>

**Domain**

AWS Database

**Question 49Correct**

A monitoring application that keeps track of a large eCommerce website uses Amazon Kinesis for data ingestion. During periods of peak data rates, the Kinesis stream cannot keep up with the incoming data.  
What step will allow Kinesis data streams to accommodate the traffic during peak hours?

**Install the Kinesis Producer Library (KPL) for ingesting data into the stream**

**Create an SQS queue and decouple the producers from the Kinesis data stream**

**1. Ingest multiple records into the stream in a single call using PutRecords**

**Your answer is correct**

**Increase the shard count of the stream using UpdateShardCount**

Overall explanation

The UpdateShardCount API action updates the shard count of the specified stream to the specified number of shards.

Updating the shard count is an asynchronous operation. Upon receiving the request, Kinesis Data Streams returns immediately and sets the status of the stream to UPDATING. After the update is complete, Kinesis Data Streams sets the status of the stream back to ACTIVE.

Depending on the size of the stream, the scaling action could take a few minutes to complete. You can continue to read and write data to your stream while its status is UPDATING.

To update the shard count, Kinesis Data Streams performs splits or merges on individual shards. This can cause short-lived shards to be created, in addition to the final shards. These short-lived shards count towards your total shard limit for your account in the Region.

When using this operation, we recommend that you specify a target shard count that is a multiple of 25% (25%, 50%, 75%, 100%). You can specify any target value within your shard limit. However, if you specify a target that isn't a multiple of 25%, the scaling action might take longer to complete.

This operation has the following default limits. By default, you cannot do the following:

• Scale more than ten times per rolling 24-hour period per stream

• Scale up to more than double your current shard count for a stream

• Scale down below half your current shard count for a stream

• Scale up to more than 500 shards in a stream

• Scale a stream with more than 500 shards down unless the result is less than 500 shards

Scale up to more than the shard limit for your account

Note that the question specifically states that the Kinesis data stream cannot keep up with incoming data. This indicates that the producers are attempting to add records to the stream but there are not enough shards to keep up with demand. Therefore, we need to add additional shards and can do this using the UpdateShardCount API action.

**CORRECT:**"Increase the shard count of the stream using UpdateShardCount" is the correct answer.

**INCORRECT:** "Install the Kinesis Producer Library (KPL) for ingesting data into the stream" is incorrect as that will help the producers to be more efficient and increase write throughput to a Kinesis data stream. However, this will not help as the Kinesis data stream already cannot keep up with the incoming demand.

**INCORRECT:** "Create an SQS queue and decouple the producers from the Kinesis data stream " is incorrect. You cannot decouple a Kinesis producer from a Kinesis data stream using SQS. Kinesis is more than capable of keeping up with demand, it just needs more shards in this case.

**INCORRECT:** "Ingest multiple records into the stream in a single call using PutRecords" is incorrect as the stream is already overloaded, we need more shards, not more data to be written.

**References:**

<https://docs.aws.amazon.com/kinesis/latest/APIReference/API_UpdateShardCount.html>

**Save time with our AWS cheat sheets:**

<https://digitalcloud.training/amazon-kinesis/>

**Domain**

AWS Application Integration

**Question 50Incorrect**

A company currently runs a number of legacy automated batch processes for system update management and operational activities. The company are looking to refactor these processes and require a service that can coordinate multiple AWS services into serverless workflows.

What is the MOST suitable service for this requirement?

**Your answer is incorrect**

**Amazon SWF**

**AWS Lambda**

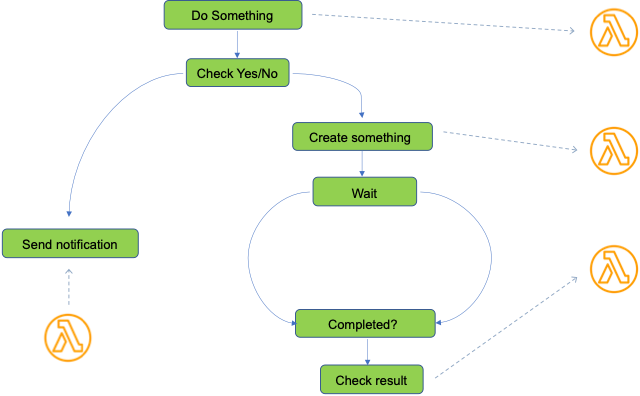
**AWS Batch**

**Correct answer**

**AWS Step Functions**

Overall explanation

AWS Step Functions is a web service that enables you to coordinate the components of distributed applications and microservices using visual workflows. You build applications from individual components that each perform a discrete function, or task, allowing you to scale and change applications quickly.



Step Functions provides a reliable way to coordinate components and step through the functions of your application. Step Functions offers a graphical console to visualize the components of your application as a series of steps. It automatically triggers and tracks each step, and retries when there are errors, so your application executes in order and as expected, every time. Step Functions logs the state of each step, so when things go wrong, you can diagnose and debug problems quickly.

**CORRECT:**"AWS Step Functions" is the correct answer.

**INCORRECT:** "Amazon SWF" is incorrect. You can think of Amazon SWF as a fully-managed state tracker and task coordinator in [the Cloud.](https://aws.amazon.com/what-is-cloud-computing/) It does not coordinate serverless workflows.

**INCORRECT:** "AWS Batch" is incorrect as this is used to run batch computing jobs on Amazon EC2 and is therefore not serverless.

**INCORRECT:** "AWS Lambda" is incorrect as though it is serverless, it does not provide a native capability to coordinate multiple AWS services.

**References:**

<https://docs.aws.amazon.com/step-functions/latest/dg/welcome.html>

**Save time with our AWS cheat sheets:**

<https://digitalcloud.training/aws-application-integration-services/>

**Domain**

AWS Application Integration

**Question 51Incorrect**

A Developer is deploying an Amazon ECS update using AWS CodeDeploy. In the appspec.yaml file, which of the following is a valid structure for the order of hooks that should be specified?

**Your answer is incorrect**

**BeforeBlockTraffic > AfterBlockTraffic > BeforeAllowTraffic > AfterAllowTraffic**

**BeforeInstall > AfterInstall > ApplicationStart > ValidateService**

**BeforeAllowTraffic > AfterAllowTraffic**

**Correct answer**

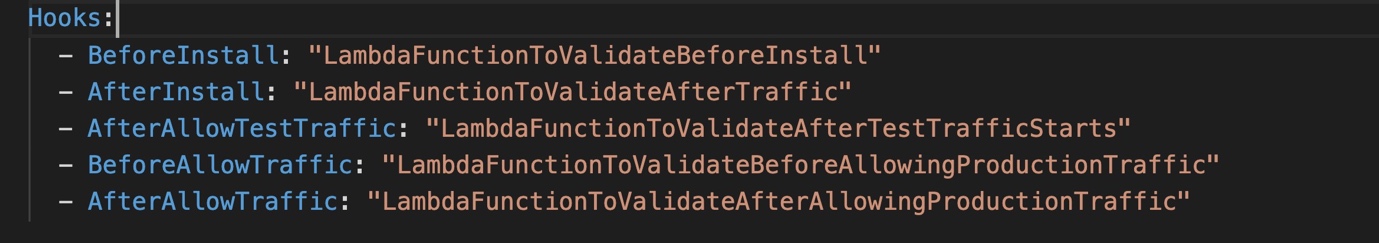
**BeforeInstall > AfterInstall > AfterAllowTestTraffic > BeforeAllowTraffic > AfterAllowTraffic**

Overall explanation

The content in the 'hooks' section of the AppSpec file varies, depending on the compute platform for your deployment. The 'hooks' section for an EC2/On-Premises deployment contains mappings that link deployment lifecycle event hooks to one or more scripts.

The 'hooks' section for a Lambda or an Amazon ECS deployment specifies Lambda validation functions to run during a deployment lifecycle event. If an event hook is not present, no operation is executed for that event. This section is required only if you are running scripts or Lambda validation functions as part of the deployment.

The following code snippet shows a valid example of the structure of hooks for an Amazon ECS deployment:



Therefore, in this scenario a valid structure for the order of hooks that should be specified in the appspec.yml file is: BeforeInstall > AfterInstall > AfterAllowTestTraffic > BeforeAllowTraffic > AfterAllowTraffic

**CORRECT:**"BeforeInstall > AfterInstall > AfterAllowTestTraffic > BeforeAllowTraffic > AfterAllowTraffic" is the correct answer.

**INCORRECT:** "BeforeInstall > AfterInstall > ApplicationStart > ValidateService" is incorrect as this would be valid for Amazon EC2.

**INCORRECT:** "BeforeAllowTraffic > AfterAllowTraffic" is incorrect as this would be valid for AWS Lambda.

**INCORRECT:** "BeforeBlockTraffic > AfterBlockTraffic > BeforeAllowTraffic > AfterAllowTraffic" is incorrect as this is a partial listing of hooks for Amazon EC2 but is incomplete.

**References:**

<https://docs.aws.amazon.com/codedeploy/latest/userguide/reference-appspec-file-structure-hooks.html>

**Save time with our AWS cheat sheets:**

<https://digitalcloud.training/aws-developer-tools/>

**Domain**

AWS Compute

**Question 52Correct**

A Developer has created a serverless function that processes log files. The function should be invoked once every 15 minutes. How can the Developer automatically invoke the function using serverless services?

**Create an Amazon SNS rule to send a notification to Lambda to instruct it to run**

**Configure the Lambda scheduler to run based on recurring time value**

**Your answer is correct**

**Create an Amazon CloudWatch Events rule that is scheduled to run and invoke the function**

**Launch an EC2 Linux instance and add a command to periodically invoke the function to its /etc/crontab file**

Overall explanation

Amazon CloudWatch Events delivers a near real-time stream of system events that describe changes in Amazon Web Services (AWS) resources. Using simple rules that you can quickly set up, you can match events and route them to one or more target functions or streams.

You can use Amazon CloudWatch Events to invoke the Lambda function on a recurring schedule of 15 minutes. This solution is entirely automated and serverless.

**CORRECT:**"Create an Amazon CloudWatch Events rule that is scheduled to run and invoke the function" is the correct answer.

**INCORRECT:** "Launch an EC2 Linux instance and add a command to periodically invoke the function to its /etc/crontab file " is incorrect as this is automatic but it is not serverless.

**INCORRECT:** "Configure the Lambda scheduler to run based on recurring time value" is incorrect as there is no Lambda scheduler that can be used.

**INCORRECT:** "Create an Amazon SNS rule to send a notification to Lambda to instruct it to run" is incorrect as you cannot invoke a function by sending a notification to it from Amazon SNS.

**References:**

<https://docs.aws.amazon.com/AmazonCloudWatch/latest/events/RunLambdaSchedule.html>

**Save time with our AWS cheat sheets:**

<https://digitalcloud.training/amazon-cloudwatch/>

<https://digitalcloud.training/aws-lambda/>

**Domain**

AWS Management & Governance

**Question 53Correct**

A Developer needs to access AWS CodeCommit over SSH. The SSH keys configured to access AWS CodeCommit are tied to a user with the following permissions:

1. {
2. "version": "2012-10-17"
3. "Statement": [
4. {
5. "Effect": "Allow",
6. "Action": [
7. "codecommit:BatchGetRepositories",
8. "codecommit:Get\*"
9. "codecommit:List\*",
10. "codecommit:GitPull"
11. ],
12. "Resource": "\*"
13. }
14. ]
15. }

The Developer needs to create/delete branches.

Which specific IAM permissions need to be added based on the principle of least privilege?

**“codecommit:Update\*”**

**Your answer is correct**

**“codecommit:CreateBranch” and “codecommit:DeleteBranch”**

**“codecommit:\*”**

**“codecommit:Put\*:”**

Overall explanation

The permissions assigned to the user account are missing the privileges to create and delete branches in AWS CodeCommit. The Developer needs to be assigned these permissions but according to the principal of least privilege it’s important to ensure no additional permissions are assigned.

The following API actions can be used to work with branches:

• CreateBranch , which creates a branch in a specified repository.

• DeleteBranch , which deletes the specified branch in a repository unless it is the default branch.

• GetBranch , which returns information about a specified branch.

• ListBranches , which lists all branches for a specified repository.

• UpdateDefaultBranch , which changes the default branch for a repository.

Therefore, the best answer is to add the “codecommit:CreateBranch” and “codecommit:DeleteBranch” permissions to the permissions policy.

**CORRECT:**"codecommit:CreateBranch” and “codecommit:DeleteBranch" is the correct answer.

**INCORRECT:** "codecommit:Put\*:" is incorrect. The wildcard (\*) will allow any API action starting with “Put”, however the only options are put-file and put-repository-triggers, neither of which is related to branches.

**INCORRECT:** "codecommit:Update\*" is incorrect. The wildcard (\*) will allow any API action starting with “Update”, however none of the options available are suitable for working with branches.

**INCORRECT:** "codecommit:\*" is incorrect as this would allow any API action which does not follow the principal of least privilege.

**References:**

<https://docs.aws.amazon.com/cli/latest/reference/codecommit/index.html#cli-aws-codecommit>

**Save time with our AWS cheat sheets:**

<https://digitalcloud.training/aws-developer-tools/>

**Domain**

AWS Developer Tools

**Question 54Correct**

An application is being instrumented to send trace data using AWS X-Ray. A Developer needs to upload segment documents using JSON-formatted strings to X-Ray using the API. Which API action should the developer use?

**The PutTelemetryRecords API action**

**Your answer is correct**

**The PutTraceSegments API action**

**The GetTraceSummaries API action**

**The UpdateGroup API action**

Overall explanation

You can send trace data to X-Ray in the form of segment documents. A segment document is a JSON formatted string that contains information about the work that your application does in service of a request. Your application can record data about the work that it does itself in segments, or work that uses downstream services and resources in subsegments.

Segments record information about the work that your application does. A segment, at a minimum, records the time spent on a task, a name, and two IDs. The trace ID tracks the request as it travels between services. The segment ID tracks the work done for the request by a single service.

Example Minimal complete segment:



You can upload segment documents with the [PutTraceSegments](https://docs.aws.amazon.com/xray/latest/api/API_PutTraceSegments.html) API. The API has a single parameter, TraceSegmentDocuments, that takes a list of JSON segment documents.

Therefore, the Developer should use the PutTraceSegments API action.

**CORRECT:**"The PutTraceSegments API action" is the correct answer.

**INCORRECT:** "The PutTelemetryRecords API action" is incorrect as this is used by the AWS X-Ray daemon to upload telemetry.

**INCORRECT:** "The UpdateGroup API action" is incorrect as this updates a group resource.

**INCORRECT:** "The GetTraceSummaries API action" is incorrect as this retrieves IDs and annotations for traces available for a specified time frame using an optional filter.

**References:**

<https://docs.aws.amazon.com/xray/latest/devguide/xray-api-sendingdata.html>

**Save time with our AWS cheat sheets:**

<https://digitalcloud.training/aws-developer-tools/>

**Domain**

AWS Developer Tools

**Question 55Correct**

A Developer has joined a team and needs to connect to the AWS CodeCommit repository using SSH. What should the Developer do to configure access using Git?

**Create an account on Github and user those login credentials to login to AWS CodeCommit**

**On the Developer’s IAM account, under security credentials, choose to create an access key and secret ID**

**On the Developer’s IAM account, under security credentials, choose to create HTTPS Git credentials for AWS CodeCommit**

**Your answer is correct**

**Generate an SSH public and private key. Upload the public key to the Developer’s IAM account**

Overall explanation

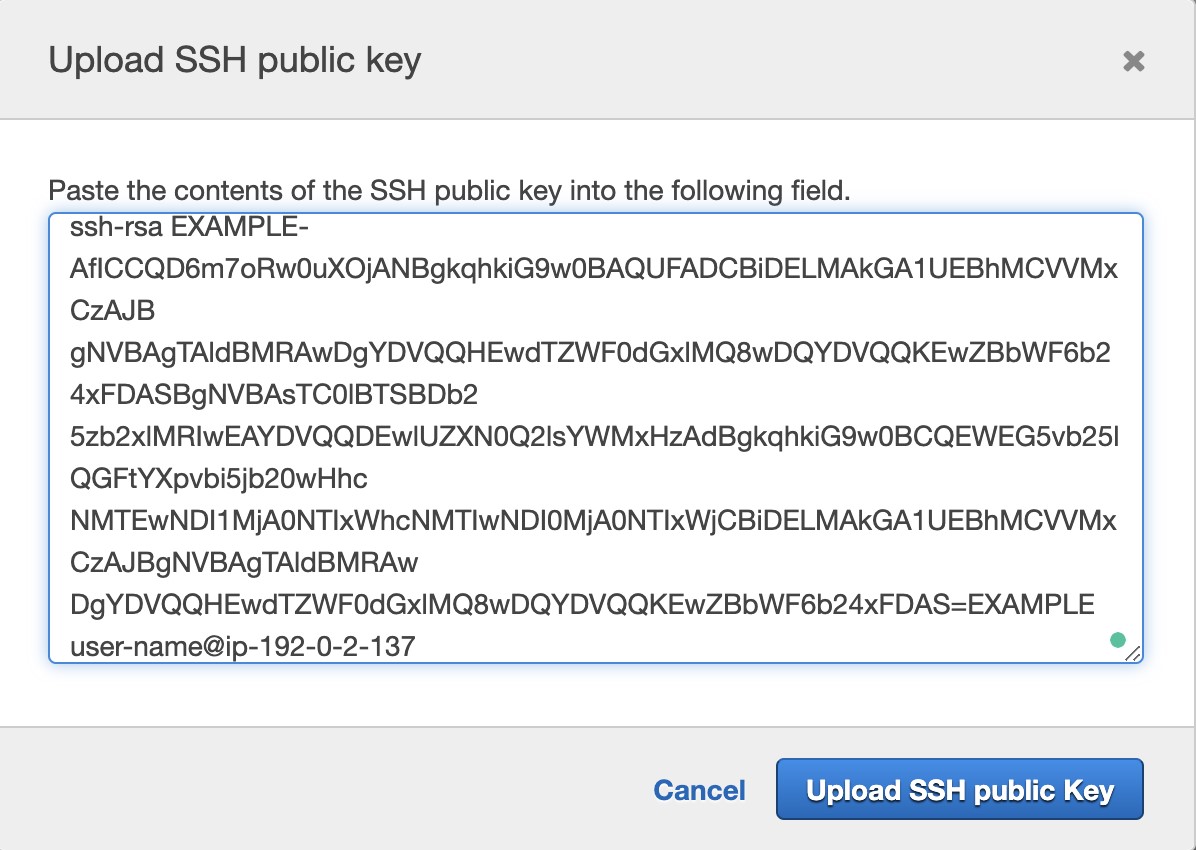
You need to configure your Git client to communicate with CodeCommit repositories. As part of this configuration, you provide IAM credentials that CodeCommit can use to authenticate you. IAM supports CodeCommit with three types of credentials:

• Git credentials, an IAM -generated user name and password pair you can use to communicate with CodeCommit repositories over HTTPS.

• SSH keys, a locally generated public-private key pair that you can associate with your IAM user to communicate with CodeCommit repositories over SSH.

• [AWS access keys](https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_access-keys.html), which you can use with the credential helper included with the AWS CLI to communicate with CodeCommit repositories over HTTPS.

As the Developer is going to use SSH, he first needs to generate an SSH private and public key. These can then be used for authentication. The method of creating these depends on the operating system the Developer is using. Then, the Developer can upload the public key (by copying the contents of the file) into his IAM account under security credentials.



**CORRECT:**"Generate an SSH public and private key. Upload the public key to the Developer’s IAM account" is the correct answer.

**INCORRECT:** "On the Developer’s IAM account, under security credentials, choose to create HTTPS Git credentials for AWS CodeCommit" is incorrect as this method is used for creating credentials when you want to connect to CodeCommit using HTTPS.

**INCORRECT:** "Create an account on Github and user those login credentials to login to AWS CodeCommit" is incorrect as you cannot login to AWS CodeCommit using credentials from Github.

**INCORRECT:** "On the Developer’s IAM account, under security credentials, choose to create an access key and secret ID" is incorrect as though you can use access keys to authenticated to CodeCommit, this requires the credential helper, and enables access over HTTPS.

**References:**

<https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_ssh-keys.html>

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

**Save time with our AWS cheat sheets:**

<https://digitalcloud.training/aws-developer-tools/>

**Domain**

AWS Security, Identity, & Compliance

**Question 56Incorrect**

A Developer is migrating Docker containers to Amazon ECS. A large number of containers will be deployed across some newly deployed ECS containers instances using the same instance type. High availability is provided within the microservices architecture. Which task placement strategy requires the LEAST configuration for this scenario?

**Your answer is incorrect**

**Fargate**

**spread**

**Correct answer**

**random**

**binpack**

Overall explanation

When a task that uses the EC2 launch type is launched, Amazon ECS must determine where to place the task based on the requirements specified in the task definition, such as CPU and memory. Similarly, when you scale down the task count, Amazon ECS must determine which tasks to terminate. You can apply task placement strategies and constraints to customize how Amazon ECS places and terminates tasks. Task placement strategies and constraints are not supported for tasks using the Fargate launch type. By default, Fargate tasks are spread across Availability Zones.

A task placement strategy is an algorithm for selecting instances for task placement or tasks for termination. For example, Amazon ECS can select instances at random, or it can select instances such that tasks are distributed evenly across a group of instances.

Amazon ECS supports the following task placement strategies:

• binpack

Place tasks based on the least available amount of CPU or memory. This minimizes the number of instances in use.

• random

Place tasks randomly.

• spread

Place tasks evenly based on the specified value. Accepted values are instanceId (or host, which has the same effect), or any platform or custom attribute that is applied to a container instance, such as attribute:ecs.availability-zone. Service tasks are spread based on the tasks from that service. Standalone tasks are spread based on the tasks from the same task group.

Therefore, for this scenario the random task placement strategy is most suitable as it requires the least configuration.

**CORRECT:**"random" is the correct answer.

**INCORRECT:** "spread" is incorrect. As high availability is taken care of within the containers there is no need to use a spread strategy to ensure HA.

**INCORRECT:** "binpack" is incorrect as there is no need to pack the containers onto the fewest instances based on CPU or memory.

**INCORRECT:** "Fargate" is incorrect as this is not a task placement strategy, it is a serverless service for running containers.

**References:**

<https://docs.aws.amazon.com/AmazonECS/latest/developerguide/task-placement-strategies.html>

**Save time with our AWS cheat sheets:**

<https://digitalcloud.training/amazon-ecs-and-eks/>

**Domain**

AWS Compute

**Question 57Incorrect**

A company provides a large number of services on AWS to customers. The customers connect to one or more services directly and the architecture is becoming complex. How can the architecture be refactored to provide a single interface for the services?

**Correct answer**

**Amazon API Gateway**

**AWS X-Ray**

**AWS Single Sign On (SSO)**

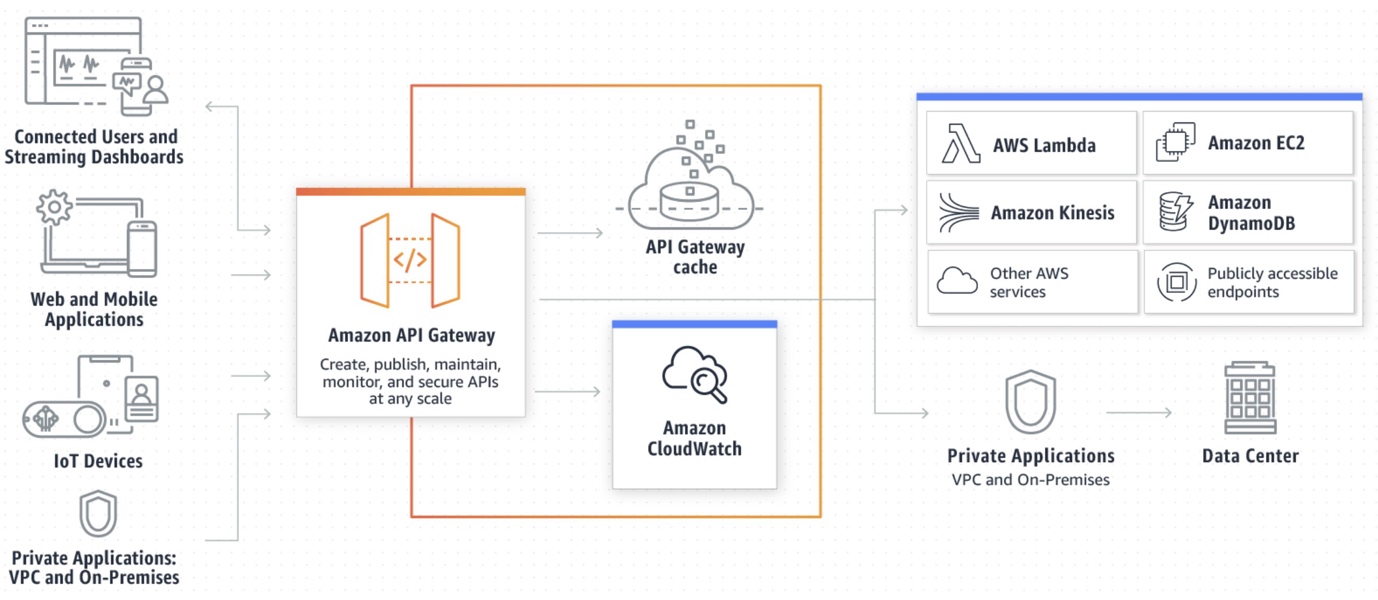
**Your answer is incorrect**

**AWS Cognito**

Overall explanation

Amazon API Gateway is a fully managed service that makes it easy for developers to create, publish, maintain, monitor, and secure APIs at any scale. APIs act as the "front door" for applications to access data, business logic, or functionality from your backend services.

Using API Gateway, you can create RESTful APIs and WebSocket APIs that enable real-time two-way communication applications. API Gateway supports containerized and serverless workloads, as well as web applications.



API Gateway can be used as the single interface for consumers of the services provided by the organization in this scenario. This solution will simplify the architecture.

**CORRECT:**"Amazon API Gateway" is the correct answer.

**INCORRECT:** "AWS X-Ray" is incorrect. AWS X-Ray is used for analyzing and debugging applications.

**INCORRECT:** "AWS Cognito" is incorrect. AWS Cognito is used for adding sign-up, sign-in and access control to web and mobile apps.

**INCORRECT:** "AWS Single Sign On (SSO)" is incorrect. AWS SSO is used to provide central management of multiple AWS accounts and business applications and to provide single sign-on to accounts.

**References:**

<https://aws.amazon.com/api-gateway/features/>

**Save time with our AWS cheat sheets:**

<https://digitalcloud.training/amazon-api-gateway/>

**Domain**

AWS Networking & Content Delivery

**Question 58Correct**

A healthcare service wants to exchange patient data securely with a partner organization through an HTTP API endpoint provided by the partner. The healthcare service has the requisite API key for accessing the HTTP API. The service needs a solution to manage the API key through code.

Which method will fulfill these requirements with maximum security?

**Store the API key in the application code.**

**Use Amazon DynamoDB to store the API key and retrieve it when needed.**

**Your answer is correct**

**Use AWS Secrets Manager to store and retrieve the API key.**

**Store the API key in an environment variable on the application server.**

Overall explanation

AWS Secrets Manager is a secrets management service that helps you protect access to your applications, services, and IT resources. This service enables you to easily rotate, manage, and retrieve API keys, database credentials, and other secrets throughout their lifecycle, making it the most secure option.

**CORRECT:**"Use AWS Secrets Manager to store and retrieve the API key" is the correct answer (as explained above.)

**INCORRECT:** "Store the API key in the application code" is incorrect.

Storing the API key directly in the application code is a security risk and could lead to unintended exposure of the key.

**INCORRECT:** "Use Amazon DynamoDB to store the API key and retrieve it when needed" is incorrect.

While Amazon DynamoDB could be used to store the API key, it does not provide the same level of security management and key rotation capabilities that AWS Secrets Manager offers.

**INCORRECT:** "Store the API key in an environment variable on the application server" is incorrect.

Storing the API key in an environment variable on the application server could lead to unintentional exposure of the key if the environment variables were logged or leaked in some way.

**References:**

<https://aws.amazon.com/secrets-manager/>

**Save time with our AWS cheat sheets:**

<https://digitalcloud.training/aws-secrets-manager/>

**Domain**

AWS Security, Identity, & Compliance

**Question 59Incorrect**

A company needs a fully-managed source control service that will work in AWS. The service must ensure that revision control synchronizes multiple distributed repositories by exchanging sets of changes peer-to-peer. All users need to work productively even when not connected to a network.

Which source control service should be used?

**AWS CodeBuild**

**Correct answer**

**AWS CodeCommit**

**Your answer is incorrect**

**AWS CodeStar**

**Subversion**

Overall explanation

AWS CodeCommit is a version control service hosted by Amazon Web Services that you can use to privately store and manage assets (such as documents, source code, and binary files) in the cloud.

A repository is the fundamental version control object in CodeCommit. It's where you securely store code and files for your project. It also stores your project history, from the first commit through the latest changes. You can share your repository with other users so you can work together on a project. If you add AWS tags to repositories, you can set up notifications so that repository users receive email about events (for example, another user commenting on code).

You can also change the default settings for your repository, browse its contents, and more. You can create triggers for your repository so that code pushes or other events trigger actions, such as emails or code functions. You can even configure a repository on your local computer (a local repo) to push your changes to more than one repository.

**CORRECT:**"AWS CodeCommit" is the correct answer.

**INCORRECT:** "Subversion" is incorrect as this is not a fully managed source control system

**INCORRECT:** "AWS CodeBuild" is incorrect as this is a service used for building and testing code.

**INCORRECT:** "AWS CodeStar" is incorrect as this is not a source control system; it integrates with source control systems such as CodeCommit.

**References:**

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

**Save time with our AWS cheat sheets:**

<https://digitalcloud.training/aws-developer-tools/>

**Domain**

AWS Developer Tools

**Question 60Correct**

A Developer has noticed some suspicious activity in her AWS account and is concerned that the access keys associated with her IAM user account may have been compromised. What is the first thing the Developer do in should do in this situation?

**Change her IAM User account password**

**Your answer is correct**

**Delete the compromised access keys**

**Delete her IAM user account**

**Report the incident to AWS Support**

Overall explanation

In this case the Developer’s access keys may have been compromised so the first step would be to invalidate the access keys by deleting them.

The next step would then be to determine if any temporary security credentials have been issued an invalidating those too to prevent any further misuse.

The user account and user account password have not been compromised so they do not need to be deleted / changed as a first step. However, changing the account password would typically be recommended as a best practice in this situation.

**CORRECT:**"Delete the compromised access keys" is the correct answer.

**INCORRECT:** "Delete her IAM user account" is incorrect. This user account has not been compromised based on the available information, just the access keys. Deleting the access keys will prevent further misuse of the AWS account.

**INCORRECT:** "Report the incident to AWS Support" is incorrect is a good practice but not the first step. The Developer should first attempt to mitigate any further misuse of the account by deleting the access keys.

**INCORRECT:** "Change her IAM User account password" is incorrect as she does not have any evidence that the account has been compromised, just the access keys. However, it would be a good practice to change the password, just not the first thing to do.

**References:**

<https://aws.amazon.com/blogs/security/what-to-do-if-you-inadvertently-expose-an-aws-access-key/>

**Save time with our AWS cheat sheets:**

<https://digitalcloud.training/aws-iam/>

**Domain**

AWS Security, Identity, & Compliance

**Question 61Correct**

Every time an Amazon EC2 instance is launched, certain metadata about the instance should be recorded in an Amazon DynamoDB table. The data is gathered and written to the table by an AWS Lambda function.

What is the MOST efficient method of invoking the Lambda function?

**Configure detailed monitoring on Amazon EC2 and create an alarm that triggers the Lambda function in initialization**

**Create a CloudWatch alarm that triggers the Lambda function based on log streams indicating an EC2 state change in CloudWatch logs**

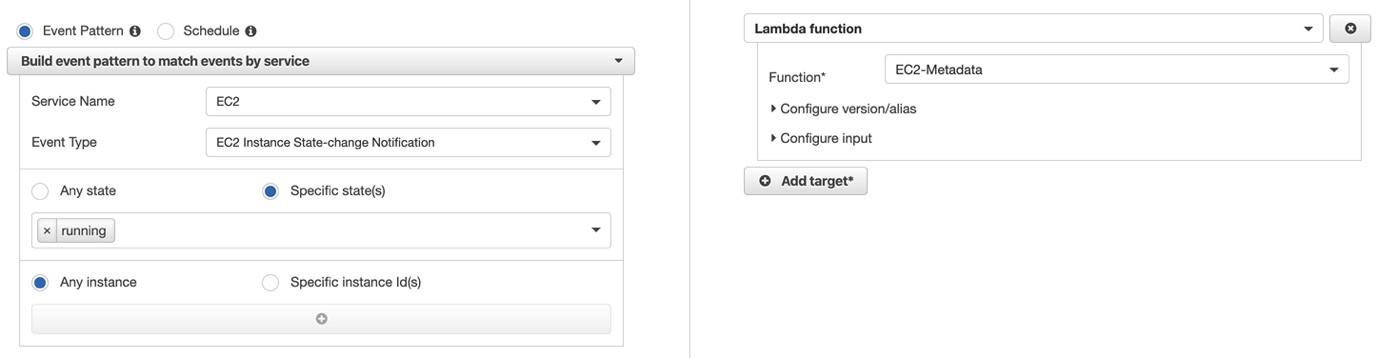
**Create a CloudTrail trail alarm that triggers the Lambda function based on the RunInstances API action**

**Your answer is correct**

**Create a CloudWatch Event with an event pattern looking for EC2 state changes and a target set to use the Lambda function**

Overall explanation

Amazon CloudWatch Events delivers a near real-time stream of system events that describe changes in Amazon Web Services (AWS) resources. Using simple rules that you can quickly set up, you can match events and route them to one or more target functions or streams. CloudWatch Events becomes aware of operational changes as they occur. CloudWatch Events responds to these operational changes and takes corrective action as necessary, by sending messages to respond to the environment, activating functions, making changes, and capturing state information.



In this scenario the only workable solution is to create a CloudWatch Event with an event pattern looking for EC2 state changes and a target set to use the Lambda function.

**CORRECT:**"Create a CloudWatch Event with an event pattern looking for EC2 state changes and a target set to use the Lambda function" is the correct answer.

**INCORRECT:** "Create a CloudWatch alarm that triggers the Lambda function based on log streams indicating an EC2 state change in CloudWatch logs" is incorrect as Amazon EC2 does not create a log group or log stream by default.

**INCORRECT:** "Create a CloudTrail trail alarm that triggers the Lambda function based on the RunInstances API action" is incorrect as you would need to create a CloudWatch alarm for CloudTrail events (CloudTrail does not have its own alarm feature).

**INCORRECT:** "Configure detailed monitoring on Amazon EC2 and create an alarm that triggers the Lambda function in initialization" is incorrect as you cannot trigger a Lambda function on EC2 instances initialization using detailed monitoring (or the EC2 console).

**References:**

<https://docs.aws.amazon.com/AmazonCloudWatch/latest/events/WhatIsCloudWatchEvents.html>

**Save time with our AWS cheat sheets:**

<https://digitalcloud.training/amazon-cloudwatch/>

**Domain**

AWS Management & Governance

**Question 62Correct**

An IT automation architecture uses many AWS Lambda functions invoking one another as a large state machine. The coordination of this state machine is legacy custom code that breaks easily.  
Which AWS Service can help refactor and manage the state machine?

**Your answer is correct**

**AWS Step Functions**

**AWS CodePipeline**

**AWS CloudFormation**

**AWS CodeBuild**

Overall explanation

AWS Step Functions lets you coordinate multiple AWS services into serverless workflows so you can build and update apps quickly. Using Step Functions, you can design and run workflows that stitch together services, such as AWS Lambda, AWS Fargate, and Amazon SageMaker, into feature-rich applications.

Workflows are made up of a series of steps, with the output of one step acting as input into the next. Application development is simpler and more intuitive using Step Functions, because it translates your workflow into a state machine diagram that is easy to understand, easy to explain to others, and easy to change.

Step Functions automatically triggers and tracks each step, and retries when there are errors, so your application executes in order and as expected. With Step Functions, you can craft long-running workflows such as machine learning model training, report generation, and IT automation.

Therefore, AWS Step Functions is the best AWS service to use when refactoring the application away from the legacy code.

**CORRECT:**"AWS Step Functions" is the correct answer.

**INCORRECT:** "AWS CloudFormation" is incorrect as CloudFormation is used for deploying resources no AWS but not for ongoing automation.

**INCORRECT:** "AWS CodePipeline" is incorrect as this is used as part of a continuous integration and delivery (CI/CD) pipeline to deploy software updates to applications.

**INCORRECT:** "AWS CodeBuild" is incorrect as this an AWS build/test service.

**References:**

<https://aws.amazon.com/step-functions/>

**Save time with our AWS cheat sheets:**

<https://digitalcloud.training/aws-application-integration-services/>

**Domain**

AWS Application Integration

**Question 63Incorrect**

A Developer needs to restrict all users and roles from using a list of API actions within a member account in AWS Organizations. The Developer needs to deny access to a few specific API actions.

What is the MOST efficient way to do this?

**Correct answer**

**Create a deny list and specify the API actions to deny**

**Create an allow list and specify the API actions to deny**

**Your answer is incorrect**

**Create an IAM policy that denies the API actions for all users and roles**

**Create an IAM policy that allows only the unrestricted API actions**

Overall explanation

Service control policies (SCPs) are one type of policy that you can use to manage your organization. SCPs offer central control over the maximum available permissions for all accounts in your organization, allowing you to ensure your accounts stay within your organization’s access control guidelines.

You can configure the SCPs in your organization to work as either of the following:

• A [deny list](https://docs.aws.amazon.com/organizations/latest/userguide/SCP_strategies.html#orgs_policies_denylist) – actions are allowed by default, and you specify what services and actions are prohibited

• An [allow list](https://docs.aws.amazon.com/organizations/latest/userguide/SCP_strategies.html#orgs_policies_allowlist) – actions are prohibited by default, and you specify what services and actions are allowed

As there are only a few API actions to restrict the most efficient strategy for this scenario is to create a deny list and specify the specific actions that are prohibited.

**CORRECT:**"Create a deny list and specify the API actions to deny" is the correct answer.

**INCORRECT:** "Create an allow list and specify the API actions to deny" is incorrect as with an allow list you specify the API actions to allow.

**INCORRECT:** "Create an IAM policy that denies the API actions for all users and roles" is incorrect as you cannot create deny policies in IAM. IAM policies implicitly deny access unless you explicitly allow permissions.

**INCORRECT:** "Create an IAM policy that allows only the unrestricted API actions" is incorrect. This will not work for administrative users such as the root account (as they have extra permissions) so an SCP must be used.

**References:**

<https://docs.aws.amazon.com/organizations/latest/userguide/orgs_manage_policies_scp.html>

<https://docs.aws.amazon.com/organizations/latest/userguide/SCP_strategies.html>

**Domain**

AWS Security, Identity, & Compliance

**Question 64Correct**

A Developer is writing an AWS Lambda function that processes records from an Amazon Kinesis Data Stream. The Developer must write the function so that it sends a notice to Administrators if it fails to process a batch of records.

How should the Developer write the function?

**Separate the Lambda handler from the core logic**

**Push the failed records to an Amazon SQS queue**

**Your answer is correct**

**Configure an Amazon SNS topic as an on-failure destination**

**Use Amazon CloudWatch Events to send the processed data**

Overall explanation

With Destinations, you can route asynchronous function results as an execution record to a destination resource without writing additional code. An execution record contains details about the request and response in JSON format including version, timestamp, request context, request payload, response context, and response payload.

For each execution status such as Success or Failure you can choose one of four destinations: another Lambda function, SNS, SQS, or EventBridge. Lambda can also be configured to route different execution results to different destinations.

In this scenario the Developer can publish the processed data to an Amazon SNS topic by configuring an Amazon SNS topic as an on-failure destination.

**CORRECT:**"Configure an Amazon SNS topic as an on-failure destination" is the correct answer.

**INCORRECT:** "Separate the Lambda handler from the core logic" is incorrect as this will not assist with sending a notification to administrators.

**INCORRECT:** "Use Amazon CloudWatch Events to send the processed data" is incorrect as CloudWatch Events is used for tracking state changes, not forwarding execution results

**INCORRECT:** "Push the failed records to an Amazon SQS queue" is incorrect as SQS will not notify the administrators, SNS should be used.

**References:**

<https://aws.amazon.com/blogs/compute/introducing-aws-lambda-destinations/>

**Save time with our AWS cheat sheets:**

<https://digitalcloud.training/aws-lambda/>

**Domain**

AWS Application Integration

**Question 65Correct**

An online retail platform uses the AWS SDK for Python (Boto3) on the frontend to handle user authentication through AWS Security Token Service (AWS STS). The platform stores its digital assets in an Amazon S3 bucket and delivers them using an Amazon CloudFront distribution, which uses the S3 bucket as its origin.

Currently, the application holds its role credentials in plaintext within a Python file in the application code. The platform developers are looking to improve security by creating a mechanism that enables the application to retrieve user credentials without embedding any credentials in the application code.

What solution would meet these requirements?

**Your answer is correct**

**Integrate a Lambda@Edge function with the CloudFront distribution. Trigger the function upon each viewer request. Give the execution role of the function the required permissions to interact with AWS STS. Shift all SDK calls from the frontend to this function.**

**Attach a Lambda@Edge function to the CloudFront distribution. Invoke this function with each viewer request. Transfer the credentials from the Python file into this function and move all SDK calls from the frontend into this function.**

**Configure a CloudFront function within the distribution. Activate the function on viewer request. Grant the function's execution role the necessary permissions to access AWS STS. Relocate all SDK calls from the frontend to this function.**

**Set up a CloudFront function on the distribution. Initiate this function for each viewer request. Shift the credentials from the Python file into this function and relocate all SDK calls from the frontend into the function.**

Overall explanation

The question seeks a solution where the application retrieves user credentials without storing any credentials within the application code. It's a security risk to embed credentials in code; it's safer to employ AWS managed services, like Lambda@Edge, that can securely manage and provide temporary credentials through AWS STS. Lambda@Edge also enables you to run your code closer to your users, reducing latency.

**CORRECT:**"Integrate a Lambda@Edge function with the CloudFront distribution. Trigger the function upon each viewer request. Give the execution role of the function the required permissions to interact with AWS STS. Shift all SDK calls from the frontend to this function" is the correct answer (as explained above.)

**INCORRECT:** "Configure a CloudFront function within the distribution. Activate the function on viewer request. Grant the function's execution role the necessary permissions to access AWS STS. Relocate all SDK calls from the frontend to this function" is incorrect.

CloudFront Functions are primarily used to perform lightweight HTTP(s) transformations and manipulations at AWS edge locations. These functions are designed to execute simple tasks like HTTP header manipulations, URL redirects, or cache key normalizations. They do not have permission to access AWS services like AWS STS, nor can they handle the more complex operations required for this use case. Therefore, this option is incorrect.

**INCORRECT:** "Attach a Lambda@Edge function to the CloudFront distribution. Invoke this function with each viewer request. Transfer the credentials from the Python file into this function and move all SDK calls from the frontend into this function" is incorrect.

This option suggests transferring the plaintext credentials into a Lambda@Edge function, which is not a secure practice. It implies hardcoding the credentials into the Lambda@Edge function, which is discouraged in AWS (or any other environment). Credentials should not be hardcoded because they could be exposed through logs or version control systems, leading to potential security risks. This option does not comply with the security best practices and thus is incorrect.

**INCORRECT:** "Set up a CloudFront function on the distribution. Initiate this function for each viewer request. Shift the credentials from the Python file into this function and relocate all SDK calls from the frontend into the function" is incorrect.

This option also proposes transferring the plaintext credentials into a CloudFront function, which is an insecure practice. Hardcoding credentials is considered risky as they could be unintentionally exposed, causing potential security vulnerabilities. Moreover, CloudFront Functions cannot access other AWS services and cannot perform complex tasks. Therefore, this option does not meet the security requirements and is incorrect.

**References:**

<https://aws.amazon.com/lambda/edge/>

**Save time with our AWS cheat sheets:**

<https://digitalcloud.training/amazon-cloudfront/>

**Domain**

AWS Networking & Content Delivery