**AWS Certified Developer Associate - Results**

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

Attempt 2: Passed! (72% required to pass)

**86%correct(56/65)**

50 minutes

**Domains**

**AWS Application Integration (7 questions)**

71%

29%

**AWS Database (9 questions)**

89%

11%

**AWS Compute (15 questions)**

87%

13%

**AWS Storage (4 questions)**

100%

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

92%

8%

**AWS Developer Tools (6 questions)**

100%

**AWS Management & Governance (6 questions)**

83%

17%

**AWS Networking & Content Delivery (6 questions)**

67%

33%

## 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=4852740)

**Attempt 2**

All domains

* 65 all
* 56 correct
* 9 incorrect
* 0 skipped
* 23 marked

Collapse all questions

**Question 1Correct**

An Amazon DynamoDB table will store authentication credentials for a mobile app. The table must be secured so only a small group of Developers are able to access it.

How can table access be secured according to this requirement and following AWS best practice?

**Create a shared user account and attach a permissions policy granting access to the table. Instruct the Developer’s to login with the user account**

**Your answer is correct**

**Attach a permissions policy to an IAM group containing the Developer’s IAM user accounts that grants access to the table**

**Create an AWS KMS resource-based policy to a CMK and grant the developer’s user accounts the permissions to decrypt data in the table using the CMK**

**Attach a resource-based policy to the table and add an IAM group containing the Developer’s IAM user accounts as a Principal in the policy**

Overall explanation

Amazon DynamoDB supports identity-based policies only. The best practice method to assign permissions to the table is to create a permissions policy that grants access to the table and assigning that policy to an IAM group that contains the Developer’s user accounts.

This will provide all users with accounts in the IAM group with the access required to access the DynamoDB table.

**CORRECT:**"Attach a permissions policy to an IAM group containing the Developer’s IAM user accounts that grants access to the table" is the correct answer.

**INCORRECT:** "Attach a resource-based policy to the table and add an IAM group containing the Developer’s IAM user accounts as a Principal in the policy" is incorrect as you cannot assign resource-based policies to DynamoDB tables.

**INCORRECT:** "Create an AWS KMS resource-based policy to a CMK and grant the developer’s user accounts the permissions to decrypt data in the table using the CMK" is incorrect as the questions requires that the Developers can access the table, not to be able to decrypt data.

**INCORRECT:** "Create a shared user account and attach a permissions policy granting access to the table. Instruct the Developer’s to login with the user account" is incorrect as this is against AWS best practice. You should never share user accounts.

**References:**

<https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/using-identity-based-policies.html>

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

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

**Domain**

AWS Database

**Question 2Correct**

A company is developing a game for the Android and iOS platforms. The mobile game will securely store user game history and other data locally on the device. The company would like users to be able to use multiple mobile devices and synchronize data between devices.

Which service can be used to synchronize the data across mobile devices without the need to create a backend application?

**Amazon DynamoDB**

**Your answer is correct**

**Amazon Cognito**

**Amazon API Gateway**

**AWS Lambda**

Overall explanation

Amazon Cognito lets you save end user data in datasets containing key-value pairs. This data is associated with an Amazon Cognito identity, so that it can be accessed across logins and devices. To sync this data between the Amazon Cognito service and an end user’s devices, invoke the synchronize method. Each dataset can have a maximum size of 1 MB. You can associate up to 20 datasets with an identity.

The Amazon Cognito Sync client creates a local cache for the identity data. Your app talks to this local cache when it reads and writes keys. This guarantees that all of your changes made on the device are immediately available on the device, even when you are offline. When the synchronize method is called, changes from the service are pulled to the device, and any local changes are pushed to the service. At this point the changes are available to other devices to synchronize.

**CORRECT:**"Amazon Cognito" is the correct answer.

**INCORRECT:** "AWS Lambda" is incorrect. AWS Lambda provides serverless functions that run your code, it is not used for mobile client data synchronization.

**INCORRECT:** "Amazon API Gateway" is incorrect as API Gateway provides APIs for traffic coming into AWS. It is not used for mobile client data synchronization.

**INCORRECT:** "Amazon DynamoDB" is incorrect as DynamoDB is a NoSQL database. It is not used for mobile client data synchronization.

**References:**

<https://docs.aws.amazon.com/cognito/latest/developerguide/synchronizing-data.html>

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

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

**Domain**

AWS Security, Identity, & Compliance

**Question 3Correct**

A development team have deployed a new application and users have reported some performance issues. The developers need to enable monitoring for specific metrics with a data granularity of one second. How can this be achieved?

**Do nothing, CloudWatch uses standard resolution metrics by default**

**Create custom metrics and enable detailed monitoring**

**Create custom metrics and configure them as standard resolution**

**Your answer is correct**

**Create custom metrics and configure them as high resolution**

Overall explanation

You can publish your own metrics to CloudWatch using the AWS CLI or an API. You can view statistical graphs of your published metrics with the AWS Management Console.

CloudWatch stores data about a metric as a series of data points. Each data point has an associated time stamp. You can even publish an aggregated set of data points called a statistic set.

Each metric is one of the following:

• Standard resolution, with data having a one-minute granularity

• High resolution, with data at a granularity of one second

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.

High-resolution metrics can give you more immediate insight into your application's sub-minute activity. Keep in mind that every PutMetricData call for a custom metric is charged, so calling PutMetricData more often on a high-resolution metric can lead to higher charges.

Therefore, the best action to take is to Create custom metrics and configure them as high resolution. This will ensure that granularity can be down to 1 second.

**CORRECT:**"Create custom metrics and configure them as high resolution" is the correct answer.

**INCORRECT:** "Do nothing, CloudWatch uses standard resolution metrics by default" is incorrect as standard resolution has a granularity of one-minute.

**INCORRECT:** "Create custom metrics and configure them as standard resolution" is incorrect as standard resolution has a granularity of one-minute.

**INCORRECT:** "Create custom metrics and enable detailed monitoring" is incorrect as detailed monitoring has 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 Management & Governance

**Question 4Correct**

A team of Developers have been assigned to a new project. The team will be collaborating on the development and delivery of a new application and need a centralized private repository for managing source code. The repository should support updates from multiple sources. Which AWS service should the development team use?

**Your answer is correct**

**AWS CodeCommit**

**AWS CodeDeploy**

**AWS CodeBuild**

**AWS CodePipeline**

Overall explanation

CodeCommit is a secure, highly scalable, managed source control service that hosts private Git repositories. CodeCommit eliminates the need for you to manage your own source control system or worry about scaling its infrastructure.

You can use CodeCommit to store anything from code to binaries. It supports the standard functionality of Git, so it works seamlessly with your existing Git-based tools.

With CodeCommit, you can:

• **Benefit from a fully managed service hosted by AWS**. CodeCommit provides high service availability and durability and eliminates the administrative overhead of managing your own hardware and software. There is no hardware to provision and scale and no server software to install, configure, and update.

• **Store your code securely**. CodeCommit repositories are encrypted at rest as well as in transit.

• **Work collaboratively on code.** CodeCommit repositories support pull requests, where users can review and comment on each other's code changes before merging them to branches; notifications that automatically send emails to users about pull requests and comments; and more.

• **Easily scale your version control projects**. CodeCommit repositories can scale up to meet your development needs. The service can handle repositories with large numbers of files or branches, large file sizes, and lengthy revision histories.

• **Store anything, anytime**. CodeCommit has no limit on the size of your repositories or on the file types you can store.

• **Integrate with other AWS and third-party services**. CodeCommit keeps your repositories close to your other production resources in the AWS Cloud, which helps increase the speed and frequency of your development lifecycle. It is integrated with IAM and can be used with other AWS services and in parallel with other repositories. **Easily migrate files from other remote repositories**. You can migrate to CodeCommit from any Git-based repository.

• **Use the Git tools you already know**. CodeCommit supports Git commands as well as its own AWS CLI commands and APIs.

Therefore, the development team should select AWS CodeCommit as the repository they use for storing code related to the new project.

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

**INCORRECT:** "AWS CodeBuild" is incorrect. AWS CodeBuild is a fully managed continuous integration (CI) service that compiles source code, runs tests, and produces software packages that are ready to deploy.

**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.

**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 5Correct**

A Developer is deploying an update to a serverless application that includes AWS Lambda using the AWS Serverless Application Model (SAM). The traffic needs to move from the old Lambda version to the new Lambda version gradually, within the shortest period of time.

Which deployment configuration is MOST suitable for these requirements?

**Your answer is correct**

**CodeDeployDefault.LambdaCanary10Percent5Minutes**

**CodeDeployDefault.LambdaLinear10PercentEvery2Minutes**

**CodeDeployDefault.LambdaLinear10PercentEvery1Minute**

**CodeDeployDefault.HalfAtATime**

Overall explanation

If you use AWS SAM to create your serverless application, it comes built-in with [CodeDeploy](https://docs.aws.amazon.com/codedeploy/latest/userguide/welcome.html) to provide gradual Lambda deployments. With just a few lines of configuration, AWS SAM does the following for you:

• Deploys new versions of your Lambda function, and automatically creates aliases that point to the new version.

• Gradually shifts customer traffic to the new version until you're satisfied that it's working as expected, or you roll back the update.

• Defines pre-traffic and post-traffic test functions to verify that the newly deployed code is configured correctly and your application operates as expected.

• Rolls back the deployment if CloudWatch alarms are triggered.

There are several options for how CodeDeploy shifts traffic to the new Lambda version. You can choose from the following:

• **Canary**: Traffic is shifted in two increments. You can choose from predefined canary options. The options specify the percentage of traffic that's shifted to your updated Lambda function version in the first increment, and the interval, in minutes, before the remaining traffic is shifted in the second increment.

• **Linear**: Traffic is shifted in equal increments with an equal number of minutes between each increment. You can choose from predefined linear options that specify the percentage of traffic that's shifted in each increment and the number of minutes between each increment.

**All-at-once**: All traffic is shifted from the original Lambda function to the updated Lambda function version at once.

Therefore CodeDeployDefault.LambdaCanary10Percent5Minutes is the best answer as this will shift 10 percent of the traffic and then after 5 minutes shift the remainder of the traffic. The entire deployment will take 5 minutes to cut over.

**CORRECT:**"CodeDeployDefault.LambdaCanary10Percent5Minutes" is the correct answer.

**INCORRECT:** "CodeDeployDefault.HalfAtATime" is incorrect as this is a CodeDeploy traffic shifting strategy that is not applicable to AWS Lambda. You can use Half at a Time with EC2 and on-premises instances.

**INCORRECT:** "CodeDeployDefault.LambdaLinear10PercentEvery1Minute" is incorrect as this option will take longer. CodeDeploy will shift 10 percent every 1 minute and therefore the deployment time will be 10 minutes.

**INCORRECT:** "CodeDeployDefault.LambdaLinear10PercentEvery2Minutes" is incorrect as this option will take longer. CodeDeploy will shift 10 percent every 2 minutes and therefore the deployment time will be 20 minutes.

**References:**

<https://docs.aws.amazon.com/serverless-application-model/latest/developerguide/automating-updates-to-serverless-apps.html>

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

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

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

**Domain**

AWS Developer Tools

**Question 6Correct**

A serverless application uses Amazon API Gateway an AWS Lambda function and a Lambda authorizer function. There is a failure with the application and a developer needs to trace and analyze user requests that pass through API Gateway through to the back end services.

Which AWS service is MOST suitable for this purpose?

**Amazon CloudWatch**

**Your answer is correct**

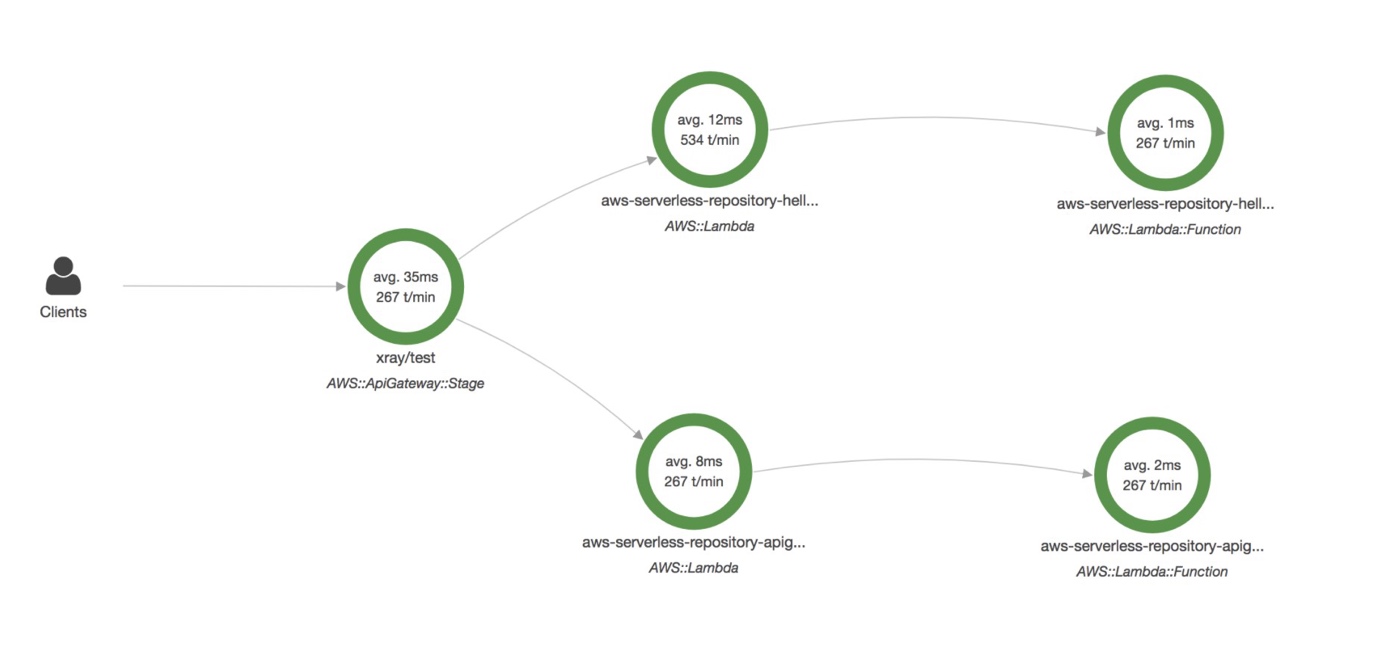
**AWS X-Ray**

**VPC Flow Logs**

**Amazon Inspector**

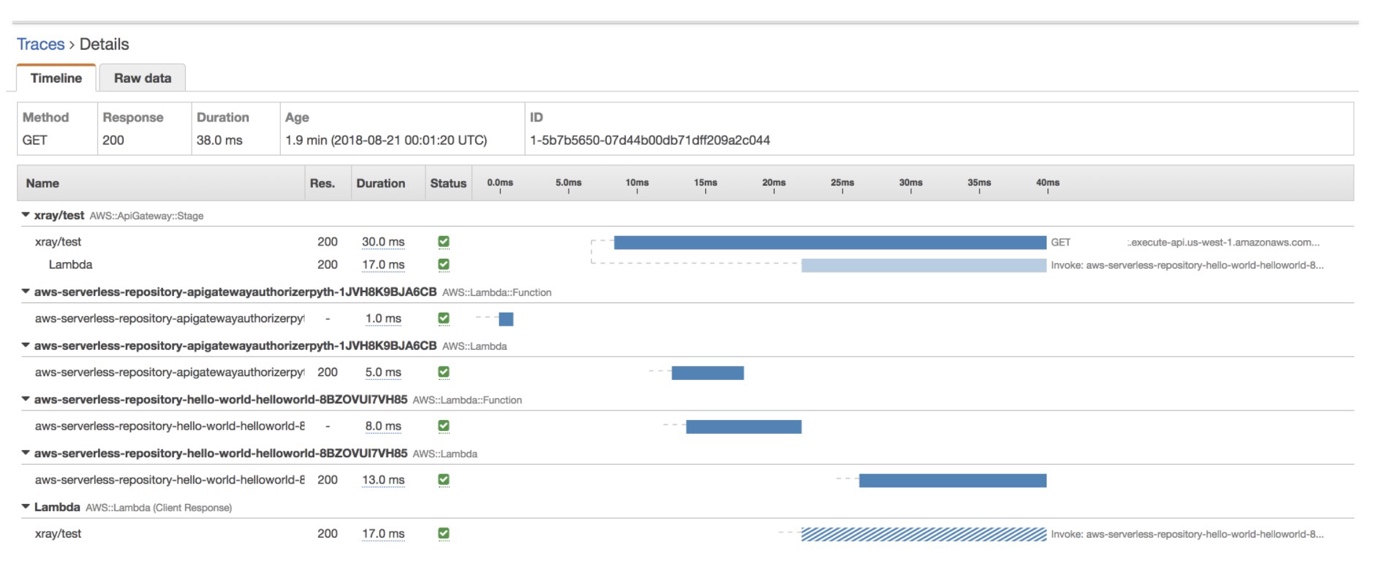
Overall explanation

You can use [AWS X-Ray](https://docs.aws.amazon.com/xray/latest/devguide/xray-services-apigateway.html) to trace and analyze user requests as they travel through your Amazon API Gateway APIs to the underlying services. API Gateway supports X-Ray tracing for all API Gateway endpoint types: Regional, edge-optimized, and private. You can use X-Ray with Amazon API Gateway in all AWS Regions where X-Ray is available.



Because X-Ray gives you an end-to-end view of an entire request, you can analyze latencies in your APIs and their backend services. You can use an X-Ray service map to view the latency of an entire request and that of the downstream services that are integrated with X-Ray. You can also configure sampling rules to tell X-Ray which requests to record and at what sampling rates, according to criteria that you specify.

The following diagram shows a trace view generated for the example API described above, with a Lambda backend function and a Lambda authorizer function. A successful API method request is shown with a response code of 200.



**CORRECT:**"AWS X-Ray" is the correct answer.

**INCORRECT:** "Amazon CloudWatch" is incorrect as it is used to collect metrics and logs. You can use these for troubleshooting however it will be more effective to use AWS X-Ray for analyzing and tracing a distributed application such as this one.

**INCORRECT:** "Amazon Inspector" is incorrect as this is an automated security assessment service. It is not used for analyzing and tracing serverless applications.

**INCORRECT:** "VPC Flow Logs" is incorrect as this is a feature that captures information about TCP/IP traffic related to network interfaces in a VPC.

**References:**

<https://docs.aws.amazon.com/xray/latest/devguide/aws-xray.html>

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

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

**Domain**

AWS Developer Tools

**Question 7Correct**

A company is running an order processing system on AWS. Amazon SQS is used to queue orders and an AWS Lambda function processes them. The company recently started noticing a lot of orders are failing to process.

How can a Developer MOST effectively manage these failures to debug the failed orders later and reprocess them, as necessary?

**Publish failed orders from the order queue to an Amazon SNS topic**

**Your answer is correct**

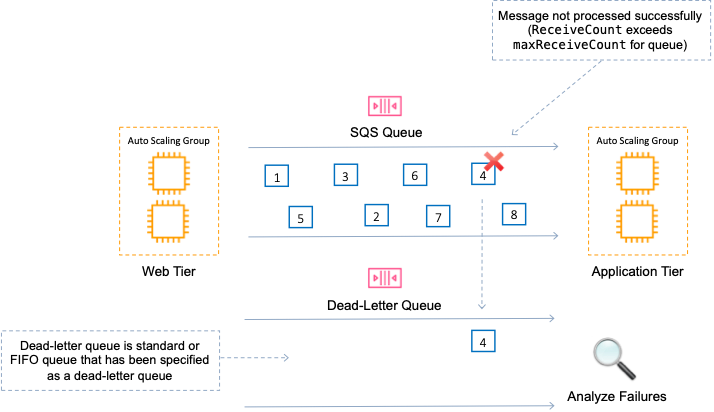
**Implement dead-letter queues for failed orders from the order queue**

**Send failed orders from the order queue to AWS CloudTrail logs**

**Log the failed orders from the order queue using Amazon CloudWatch Logs**

Overall explanation

Amazon SQS supports dead-letter queues, which other queues (source queues) can target for messages that can't be processed (consumed) successfully. Dead-letter queues are useful for debugging your application or messaging system because they let you isolate problematic messages to determine why their processing doesn't succeed.



The Developer should therefore implement dead-letter queues for failed orders from the order queue. This will allow full debugging as the entire message is available for analysis.

**CORRECT:**"Implement dead-letter queues for failed orders from the order queue" is the correct answer.

**INCORRECT:** "Publish failed orders from the order queue to an Amazon SNS topic" is incorrect as there is no way to isolate messages that have failed to process when subscribing an SQS queue to an SNS topic.

**INCORRECT:** "Log the failed orders from the order queue using Amazon CloudWatch Logs" is incorrect as SQS does not publish message success/failure to CloudWatch Logs.

**INCORRECT:** "Send failed orders from the order queue to AWS CloudTrail logs" is incorrect as CloudTrail records API activity not performance metrics or logs.

**References:**

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

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

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

**Domain**

AWS Application Integration

**Question 8Correct**

A serverless application is used to process customer information and outputs a JSON file to an Amazon S3 bucket. AWS Lambda is used for processing the data. The data is sensitive and should be encrypted.

How can a Developer modify the Lambda function to ensure the data is encrypted before it is uploaded to the S3 bucket?

**Your answer is correct**

**Use the GenerateDataKey API, then use the data key to encrypt the file using the Lambda code**

**Use the default KMS key for S3 and encrypt the file using the Lambda code**

**Use the S3 managed key and call the GenerateDataKey API to encrypt the file**

**Enable server-side encryption on the S3 bucket and create a policy to enforce encryption**

Overall explanation

The GenerateDataKey API is used with the AWS KMS services and 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.

For this scenario we can use GenerateDataKey to obtain an encryption key from KMS that we can then use within the function code to encrypt the file. This ensures that the file is encrypted BEFORE it is uploaded to Amazon S3.

**CORRECT:**"Use the GenerateDataKey API, then use the data key to encrypt the file using the Lambda code" is the correct answer.

**INCORRECT:** "Enable server-side encryption on the S3 bucket and create a policy to enforce encryption" is incorrect. This would not encrypt data before it is uploaded as S3 would only encrypt the data as it is written to storage.

**INCORRECT:** "Use the S3 managed key and call the GenerateDataKey API to encrypt the file" is incorrect as you do not use an encryption key to call KMS. You call KMS with the GenerateDataKey API to obtain an encryption key. Also, the S3 managed key can only be used within the S3 service.

**INCORRECT:** "Use the default KMS key for S3 and encrypt the file using the Lambda code" is incorrect. You cannot use the default KMS key for S3 within the Lambda code as it can only be used within the S3 service.

**References:**

<https://docs.aws.amazon.com/kms/latest/APIReference/API_GenerateDataKey.html>

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

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

**Domain**

AWS Security, Identity, & Compliance

**Question 9Incorrect**

A company is using Amazon CloudFront to provide low-latency access to a web application to its global users. The organization must encrypt all traffic between users and CloudFront, and all traffic between CloudFront and the web application.

How can these requirements be met? (Select TWO.)

**Your selection is correct**

**Set the Viewer Protocol Policy to “HTTPS Only” or “Redirect HTTP to HTTPS”**

**Your selection is incorrect**

**Use AWS KMS to encrypt traffic between CloudFront and the web application**

**Enable the CloudFront option Restrict Viewer Access**

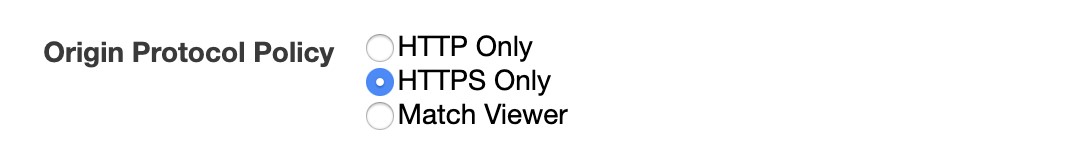
**Set the Origin’s HTTP Port to 443**

**Correct selection**

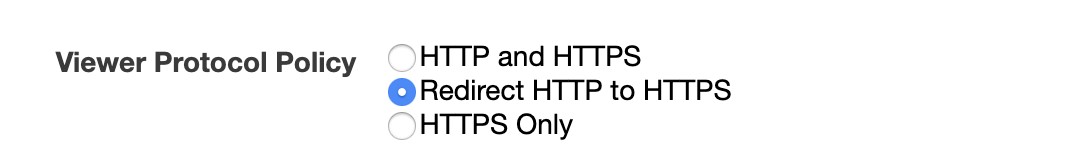
**Set the Origin Protocol Policy to “HTTPS Only”**

Overall explanation

This scenario requires encryption of in-flight data which can be done by implementing HTTPS. To do this the organization must configure the Origin Protocol Policy and the Viewer Protocol Policy on the CloudFront Distribution.



The Origin Protocol Policy can be used to select whether you want CloudFront to connect to your origin using only HTTP, only HTTPS, or to connect by matching the protocol used by the viewer. For example, if you select Match Viewer for the Origin Protocol Policy, and if the viewer connects to CloudFront using HTTPS, CloudFront will connect to your origin using HTTPS.



If you want CloudFront to allow viewers to access your web content using either HTTP or HTTPS, specify HTTP and HTTPS. If you want CloudFront to redirect all HTTP requests to HTTPS, specify Redirect HTTP to HTTPS. If you want CloudFront to require HTTPS, specify HTTPS Only.

**CORRECT:**“Set the Origin Protocol Policy to “HTTPS Only”” is a correct answer.

**CORRECT:**“Set the Viewer Protocol Policy to “HTTPS Only” or “Redirect HTTP to HTTPS”” is also a correct answer.

**INCORRECT:** “Use AWS KMS to encrypt traffic between CloudFront and the web application” is incorrect as KMS is used for encrypting data at rest.

**INCORRECT:** “Set the Origin’s HTTP Port to 443” is incorrect as you must configure the origin protocol policy to HTTPS. The HTTPS port should be set to 443.

**INCORRECT:** “Enable the CloudFront option Restrict Viewer Access” is incorrect as this is used to configure whether you want CloudFront to require users to access your content using a signed URL or a signed cookie.

**References:**

<https://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/distribution-web-values-specify.html>

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

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

**Domain**

AWS Networking & Content Delivery

**Question 10Correct**

An application exports files which must be saved for future use but are not frequently accessed. Compliance requirements necessitate redundant retention of data across AWS regions. Which solution is the MOST cost-effective for these requirements?

**Amazon S3 with Same-Region Replication (CRR)**

**Amazon DynamoDB with Global Tables**

**Your answer is correct**

**Amazon S3 with Cross-Region Replication (CRR)**

**AWS Storage Gateway with a replicated file gateway**

Overall explanation

Replication enables automatic, asynchronous copying of objects across Amazon S3 buckets. Buckets that are configured for object replication can be owned by the same AWS account or by different accounts. You can copy objects between different AWS Regions or within the same Region.

To enable object replication, you add a replication configuration to your source bucket. The minimum configuration must provide the following:

The destination bucket where you want Amazon S3 to replicate objects

An AWS Identity and Access Management (IAM) role that Amazon S3 can assume to replicate objects on your behalf

You can replicate objects between different AWS Regions or within the same AWS Region.

**Cross-Region replication** (CRR) is used to copy objects across Amazon S3 buckets in different AWS Regions.

**Same-Region replication** (SRR) is used to copy objects across Amazon S3 buckets in the same AWS Region.

For this scenario, CRR would be a better fit as the data must be replicated across regions.

**CORRECT:**"Amazon S3 with Cross-Region Replication (CRR)" is the correct answer.

**INCORRECT:** "Amazon S3 with Same-Region Replication (CRR)" is incorrect as the requirement is to replicated data across AWS regions.

**INCORRECT:** "Amazon DynamoDB with Global Tables" is incorrect as this is unlikely to be the most cost-effective solution when data is infrequently accessed. It also may not be possible to store the files in the database, they may need to be referenced from an external location such as S3.

**INCORRECT:** "AWS Storage Gateway with a replicated file gateway" is incorrect. AWS Storage Gateway connects an on-premises software appliance with cloud-based storage to provide seamless integration. This is not used for replicating data within the AWS cloud across regions.

**References:**

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

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

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

**Domain**

AWS Storage

**Question 11Correct**

A set of APIs are exposed to customers using Amazon API Gateway. These APIs have caching enabled on the API Gateway. Customers have asked for an option to invalidate this cache for each of the APIs.

What action can be taken to allow API customers to invalidate the API Cache?

**Ask customers to add a query string parameter called INVALIDATE\_CACHE” when making an API call**

**Ask customers to invoke an AWS API endpoint which invalidates the cache**

**Your answer is correct**

**Ask customers to pass an HTTP header called Cache-Control:max-age=0**

**Ask customers to use AWS credentials to call the InvalidateCache API**

Overall explanation

A client of your API can invalidate an existing cache entry and reload it from the integration endpoint for individual requests. The client must send a request that contains the Cache-Control: max-age=0 header.

The client receives the response directly from the integration endpoint instead of the cache, provided that the client is authorized to do so. This replaces the existing cache entry with the new response, which is fetched from the integration endpoint.

Therefore, the company should ask customers to pass an HTTP header called Cache-Control:max-age=0.

**CORRECT:**"Ask customers to pass an HTTP header called Cache-Control:max-age=0" is the correct answer.

**INCORRECT:** "Ask customers to use AWS credentials to call the InvalidateCache API" is incorrect as this API action is used to invalidate the cache but is not the method the clients use to invalidate the cache.

**INCORRECT:** "Ask customers to invoke an AWS API endpoint which invalidates the cache" is incorrect as you don’t invalidate the cache by invoking an endpoint, the HTTP header mentioned in the explanation is required.

**INCORRECT:** "Ask customers to add a query string parameter called INVALIDATE\_CACHE” when making an API call" is incorrect as this is not a valid method of invalidating an API Gateway cache.

**References:**

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

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

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

**Domain**

AWS Networking & Content Delivery

**Question 12Correct**

A Developer is trying to make API calls using AWS SDK. The IAM user credentials used by the application require multi-factor authentication for all API calls.

Which method should the Developer use to access the multi-factor authentication protected API?

**Your answer is correct**

**GetSessionToken**

**DecodeAuthorizationMessage**

**GetCallerIdentity**

**GetFederationToken**

Overall explanation

The GetSessionToken API call returns a set of temporary credentials for an AWS account or IAM user. The credentials consist of an access key ID, a secret access key, and a security token. Typically, you use GetSessionToken if you want to use MFA to protect programmatic calls to specific AWS API operations

Therefore, the Developer can use GetSessionToken with an MFA device to make secure API calls using the AWS SDK.

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

**INCORRECT:** "GetFederationToken" is incorrect as this is used with federated users to return a set of temporary security credentials (consisting of an access key ID, a secret access key, and a security token).

**INCORRECT:** "GetCallerIdentity" is incorrect as this API action returns details about the IAM user or role whose credentials are used to call the operation.

**INCORRECT:** "DecodeAuthorizationMessage" is incorrect as this API action decodes additional information about the authorization status of a request from an encoded message returned in response to an AWS request.

**References:**

<https://docs.aws.amazon.com/STS/latest/APIReference/API_GetSessionToken.html>

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

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

**Domain**

AWS Security, Identity, & Compliance

**Question 13Correct**

A Developer has written some code that will connect and pull information from several hundred websites. The code needs to run on a daily schedule and execution time will be less than 60 seconds.

Which AWS service will be most suitable and cost-effective?

**Amazon API Gateway**

**Amazon ECS Fargate**

**Your answer is correct**

**AWS Lambda**

**Amazon EC2**

Overall explanation

AWS Lambda is a serverless service with a maximum execution time of 900 seconds. This will be the most suitable and cost-effective option for this use case. You can also schedule Lambda functions to run using Amazon CloudWatch Events.

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

**INCORRECT:** "Amazon ECS Fargate" is incorrect as this is used for running Docker containers and is a better fit for microservices applications rather than running code for a short period of time.

**INCORRECT:** "Amazon EC2" is incorrect as this would require running EC2 instances which would not be cost-effective.

**INCORRECT:** "Amazon API Gateway" is incorrect as this service is used for creating APIs, not running code.

**References:**

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

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

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

**Domain**

AWS Compute

**Question 14Correct**

A security officer has requested that a Developer enable logging for API actions for all AWS regions to a single Amazon S3 bucket.

What is the EASIEST way for the Developer to achieve this requirement?

**Create an AWS CloudTrail trail and apply it to all regions, configure logging to a local bucket, and then use cross-region replication to replicate all logs to a single S3 bucket**

**Create an AWS CloudTrail trail in each region, configure logging to a single S3 bucket**

**Create an AWS CloudTrail trail in each region, configure logging to a local bucket, and then use cross-region replication to replicate all logs to a single S3 bucket**

**Your answer is correct**

**Create an AWS CloudTrail trail and apply it to all regions, configure logging to a single S3 bucket**

Overall explanation

The easiest way to achieve the desired outcome is to create an AWS CloudTrail trail and apply it to all regions and configure logging to a single S3 bucket. This is a supported configuration and will achieve the requirement.

**CORRECT:**"Create an AWS CloudTrail trail and apply it to all regions, configure logging to a single S3 bucket" is the correct answer.

**INCORRECT:** "Create an AWS CloudTrail trail in each region, configure logging to a single S3 bucket" is incorrect. The Developer should apply a trail to all regions. This will be easier.

**INCORRECT:** "Create an AWS CloudTrail trail in each region, configure logging to a local bucket, and then use cross-region replication to replicate all logs to a single S3 bucket" is incorrect. This is unnecessary, the Developer can simply create a trail that is applied to all regions and log to a single bucket.

**INCORRECT:** "Create an AWS CloudTrail trail and apply it to all regions, configure logging to a local bucket, and then use cross-region replication to replicate all logs to a single S3 bucket" is incorrect. This is unnecessary, the Developer can simply create a trail that is applied to all regions and log to a single bucket.

**References:**

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

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

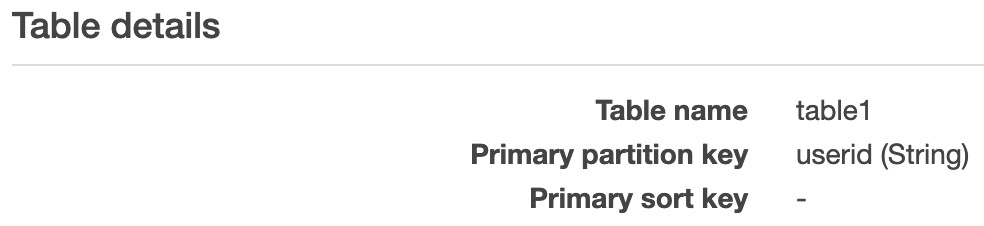
<https://digitalcloud.training/aws-cloudtrail/>

**Domain**

AWS Management & Governance

**Question 15Correct**

A Developer recently created an Amazon DynamoDB table. The table has the following configuration:



The Developer attempted to add two items for userid “user0001” with unique timestamps and received an error for the second item stating: “The conditional request failed”.

What MUST the Developer do to resolve the issue?

**Your answer is correct**

**Recreate the table with a composite key consisting of userid and timestamp**

**Use the SDK to add the items**

**Update the table with a primary sort key for the timestamp attribute**

**Add a local secondary index (LSI) for the timestamp attribute**

Overall explanation

DynamoDB stores and retrieves data based on a Primary key. There are two types of Primary key:

•Partition key – unique attribute (e.g. user ID).

• Value of the Partition key is input to an internal hash function which determines the partition or physical location on which the data is stored.

• If you are using the Partition key as your Primary key, then no two items can have the same partition key.

Composite key – Partition key + Sort key in combination.

• Example is user posting to a forum. Partition key would be the user ID, Sort key would be the timestamp of the post.

• 2 items may have the same Partition key, but they must have a different Sort key.

• All items with the same Partition key are stored together, then sorted according to the Sort key value.

• Allows you to store multiple items with the same partition key.

As stated above, if using a partition key alone as per the configuration provided with the question, then you cannot have two items with the same partition key. The only resolution is to recreate the table with a composite key consisting of the userid and timestamp attributes. In that case the Developer will be able to add multiple items with the same userid as long as the timestamp is unique.

**CORRECT:**"Recreate the table with a composite key consisting of userid and timestamp" is the correct answer.

**INCORRECT:** "Update the table with a primary sort key for the timestamp attribute" is incorrect as you cannot update the table in this case, it must be recreated.

**INCORRECT:** "Add a local secondary index (LSI) for the timestamp attribute" is incorrect as the Developer will still not be able to add multiple entries to the main table for the same userid.

**INCORRECT:** "Use the SDK to add the items" is incorrect as it doesn’t matter whether you use the console, CLI or SDK, the conditional update will still fail with this configuration.

**References:**

<https://aws.amazon.com/blogs/database/choosing-the-right-dynamodb-partition-key/>

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

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

**Domain**

AWS Database

**Question 16Correct**

A software firm is introducing a multimedia application that allows guest users to sample content before deciding to fully register. The company requires a mechanism that can identify users who have already created an account and to monitor the quantity of guest users who ultimately register.

What two actions would best satisfy these needs? (Select TWO.)

**Deploy Amazon S3 for managing user data and monitoring account conversions.**

**Your selection is correct**

**Implement AWS IAM roles to provide distinct permissions for guest users and registered users.**

**Use AWS Glue to oversee user registration and account conversion.**

**Your selection is correct**

**Use Amazon Cognito User Pools for managing user registration and authentication.**

**Use AWS Lambda to track the transitions from guest to full accounts.**

Overall explanation

Amazon Cognito User Pools serve as a user directory that offers the backend capabilities required for user registration, authentication, and account recovery. This makes it the right solution for identifying users who have created accounts.

AWS IAM roles can be used to provide distinct permissions for guest users and registered users. While they do not directly track account conversions, they can help delineate between different user states.

**CORRECT:**"Implement AWS IAM roles to provide distinct permissions for guest users and registered users" is a correct answer (as explained above.)

**CORRECT:**"Use Amazon Cognito User Pools for managing user registration and authentication" is also a correct answer (as explained above.)

**INCORRECT:** "Use AWS Glue to oversee user registration and account conversion" is incorrect.

AWS Glue is an ETL service designed for easy data preparation and loading for analytics, not for managing user registration or authentication.

**INCORRECT:** "Deploy Amazon S3 for managing user data and monitoring account conversions" is incorrect.

Amazon S3 is a storage service and is not designed for managing user registration or tracking account conversions.

**INCORRECT:** "Use AWS Lambda to track the transitions from guest to full accounts" is incorrect.

AWS Lambda is a serverless compute service and is not typically used for managing user registration or tracking conversions.

**References:**

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

<https://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles.html>

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

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

**Domain**

AWS Security, Identity, & Compliance

**Question 17Correct**

A Developer has completed some code updates and needs to deploy the updates to an Amazon Elastic Beanstalk environment. Due to the criticality of the application, the ability to quickly roll back must be prioritized of any other considerations.

Which deployment policy should the Developer choose?

**Your answer is correct**

**Immutable**

**Rolling with additional batch**

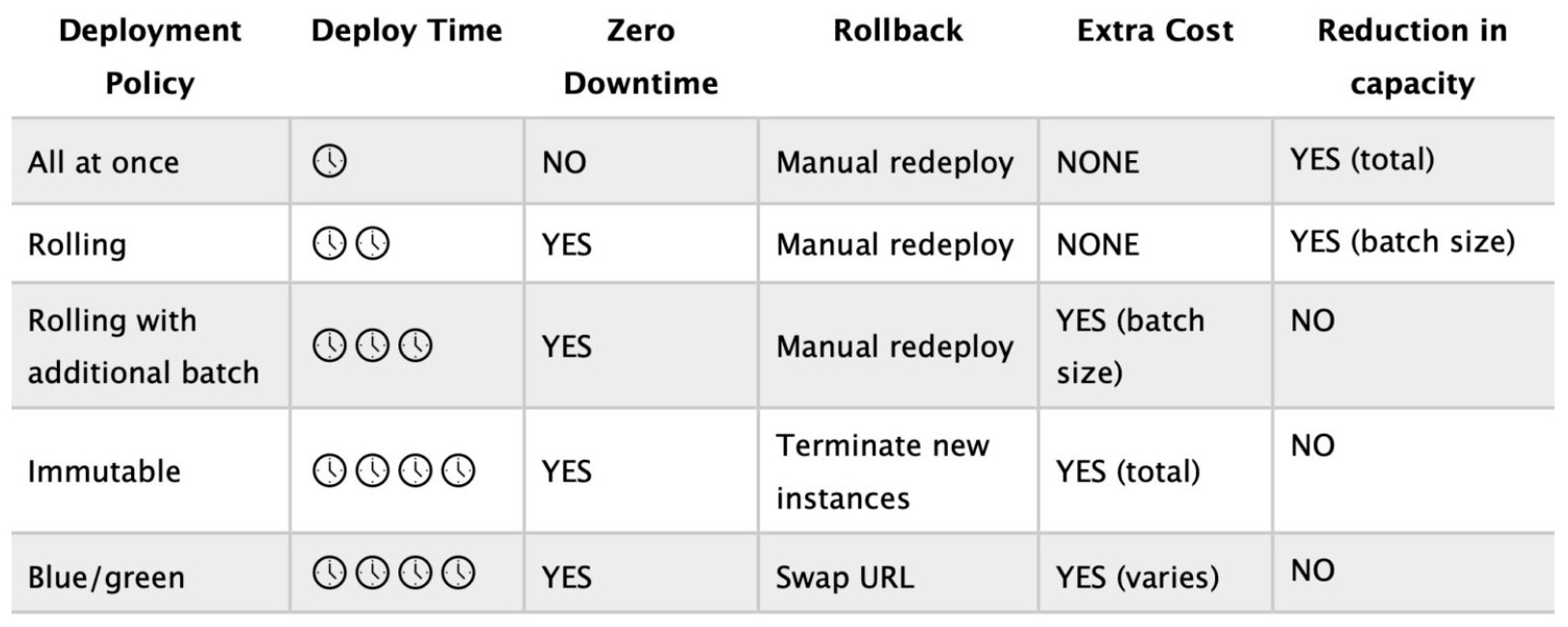
**Rolling**

**All at once**

Overall explanation

AWS Elastic Beanstalk provides several options for how deployments are processed, including deployment policies and options that let you configure batch size and health check behavior during deployments.

Each deployment policy has advantages and disadvantages and it’s important to select the best policy to use for each situation. The following tables summarizes the different deployment policies:



The “immutable” policy will create a new ASG with a whole new set of instances and deploy the updates there.

Immutable:

• Launches new instances in a new ASG and deploys the version update to these instances before swapping traffic to these instances once healthy.

• Zero downtime.

• New code is deployed to new instances using an ASG.

• High cost as double the number of instances running during updates.

• Longest deployment.

• Quick rollback in case of failures.

• Great for production environments.

For this scenario a quick rollback must be prioritized over all other considerations. Therefore, the best choice is “immutable”. This deployment policy is the most expensive and longest (duration) option. However, you can roll back quickly and safely as the original instances are all available and unmodified.

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

**INCORRECT:** "Rolling" is incorrect as this policy requires manual redeployment if there are any issues caused by the update.

**INCORRECT:** "Rolling with additional batch" is incorrect as this policy requires manual redeployment if there are any issues caused by the update.

**INCORRECT:** "All at once" is incorrect as this takes the entire environment down at once and requires manual redeployment if there are any issues caused by the update.

**References:**

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

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

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

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

**Domain**

AWS Compute

**Question 18Correct**

A Developer manages a monitoring service for a fleet of IoT sensors in a major city. The monitoring application uses an Amazon Kinesis Data Stream with a group of EC2 instances processing the data. Amazon CloudWatch custom metrics show that the instances a reaching maximum processing capacity and there are insufficient shards in the Data Stream to handle the rate of data flow.

What course of action should the Developer take to resolve the performance issues?

**Increase the EC2 instance size**

**Your answer is correct**

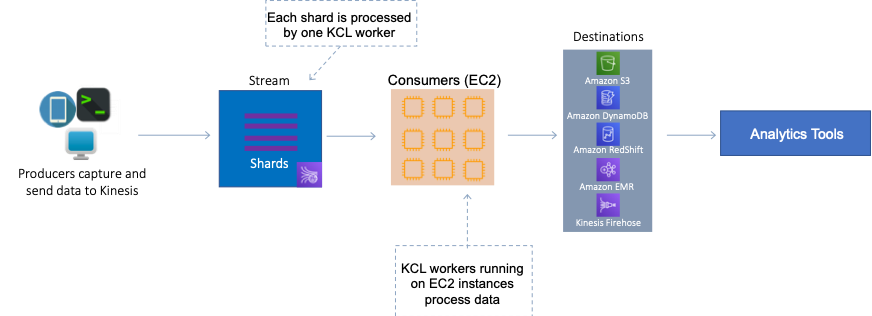
**Increase the EC2 instance size and add shards to the stream**

**Increase the number of EC2 instances to match the number of shards**

**Increase the number of open shards**

Overall explanation

By increasing the instance size and number of shards in the Kinesis stream, the developer can allow the instances to handle more record processors, which are running in parallel within the instance. It also allows the stream to properly accommodate the rate of data being sent in. The data capacity of your stream is a function of the number of shards that you specify for the stream. The total capacity of the stream is the sum of the capacities of its shards.



Therefore, the best answer is to increase both the EC2 instance size and add shards to the stream.

**CORRECT:**"Increase the EC2 instance size and add shards to the stream" is the correct answer.

**INCORRECT:** "Increase the number of EC2 instances to match the number of shards" is incorrect as you can have an individual instance running multiple KCL workers.

**INCORRECT:** "Increase the EC2 instance size" is incorrect as the Developer would also need to add shards to the stream to increase the capacity of the stream.

**INCORRECT:** "Increase the number of open shards" is incorrect as this does not include increasing the instance size or quantity which is required as they are running at capacity.

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

<https://docs.aws.amazon.com/streams/latest/dev/developing-consumers-with-kcl.html>

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

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

**Domain**

AWS Application Integration

**Question 19Correct**

A static website is hosted on Amazon S3 using the bucket name of dctlabs.com. Some HTML pages on the site use JavaScript to download images that are located in the bucket https://dctlabsimages.s3.amazonaws.com/. Users have reported that the images are not being displayed.

What is the MOST likely cause?

**Your answer is correct**

**Cross Origin Resource Sharing is not enabled on the dctlabsimages bucket**

**Amazon S3 Transfer Acceleration should be enabled on the dctlabs.com bucket**

**The dctlabsimages bucket is not in the same region as the dctlabs.com bucket**

**Cross Origin Resource Sharing is not enabled on the dctlabs.com bucket**

Overall explanation

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

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

In this case, you would apply the CORS configuration to the dctlabsimages bucket so that it will allow GET requests from the dctlabs.com origin.

**CORRECT:**"Cross Origin Resource Sharing is not enabled on the dctlabsimages bucket" is the correct answer.

**INCORRECT:** "Cross Origin Resource Sharing is not enabled on the dctlabs.com bucket" is incorrect as in this case the images that are being blocked are located in the dctlabsimages bucket. You need to apply the CORS configuration to the dctlabsimages bucket so it allows requests from the dctlabs.com origin.

**INCORRECT:** "The dctlabsimages bucket is not in the same region as the dctlabs.com bucket" is incorrect as it doesn’t matter what regions the buckets are in.

**INCORRECT:** "Amazon S3 Transfer Acceleration should be enabled on the dctlabs.com bucket" is incorrect as this feature of Amazon S3 is used to speed uploads to S3.

**References:**

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

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

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

**Domain**

AWS Storage

**Question 20Correct**

A Developer is troubleshooting an issue with a DynamoDB table. The table is used to store order information for a busy online store and uses the order date as the partition key. During busy periods writes to the table are being throttled despite the consumed throughput being well below the provisioned throughput.

According to AWS best practices, how can the Developer resolve the issue at the LOWEST cost?

**Add a global secondary index to the table**

**Use an Amazon SQS queue to buffer the incoming writes**

**Your answer is correct**

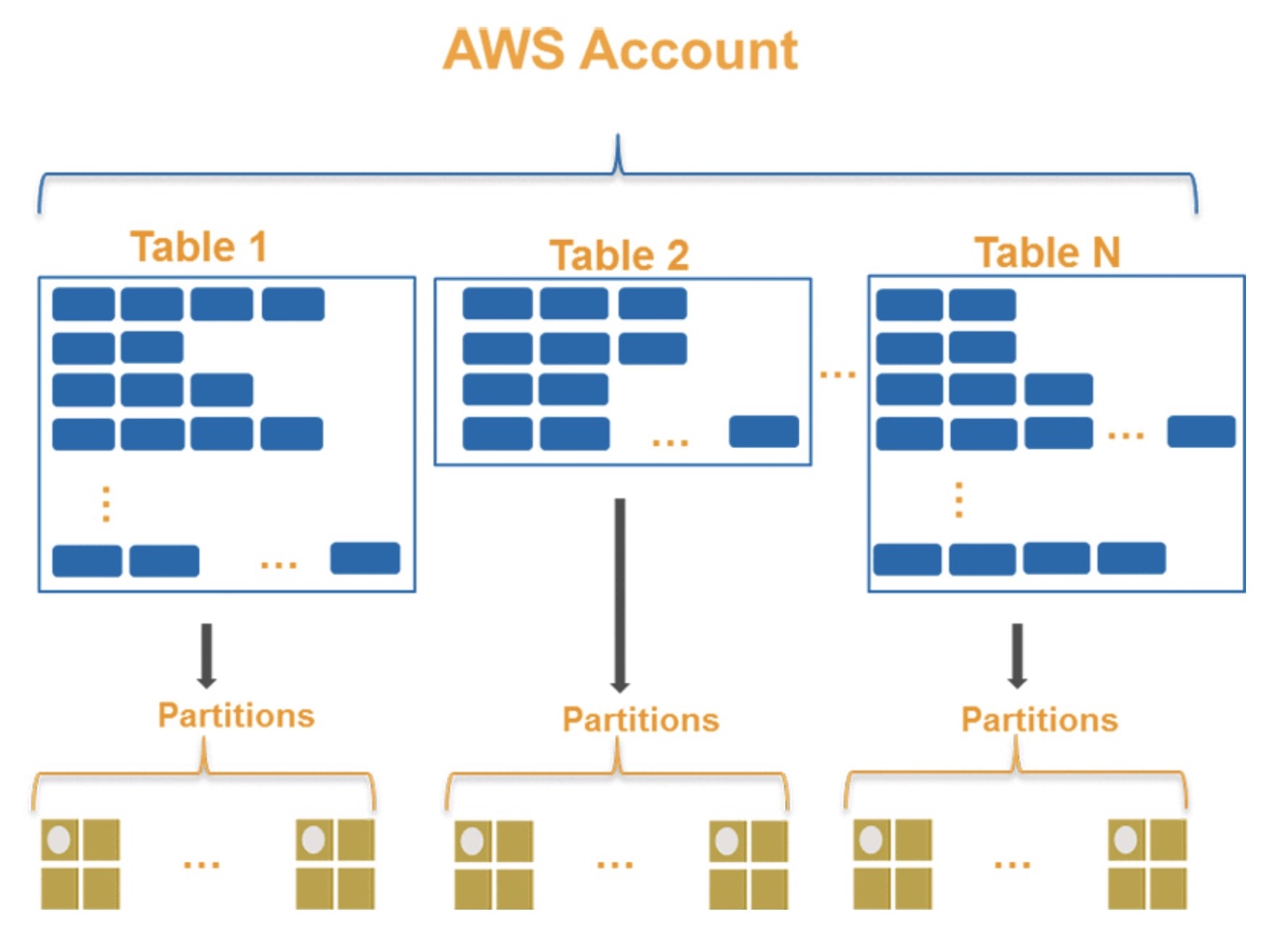
**Add a random number suffix to the partition key values**

**Increase the read and write capacity units for the table**

Overall explanation

DynamoDB stores data as groups of attributes, known as items. Items are similar to rows or records in other database systems. DynamoDB stores and retrieves each item based on the primary key value, which must be unique.

Items are distributed across 10-GB storage units, called partitions (physical storage internal to DynamoDB). Each table has one or more partitions, as shown in the following illustration.



DynamoDB uses the partition key’s value as an input to an internal hash function. The output from the hash function determines the partition in which the item is stored. Each item’s location is determined by the hash value of its partition key.

All items with the same partition key are stored together, and for composite partition keys, are ordered by the sort key value. DynamoDB splits partitions by sort key if the collection size grows bigger than 10 GB.

DynamoDB evenly distributes provisioned throughput—read capacity units (RCUs) and write capacity units (WCUs)—among partitions and automatically supports your access patterns using the throughput you have provisioned. However, if your access pattern exceeds 3000 RCU or 1000 WCU for a single partition key value, your requests might be throttled with a ProvisionedThroughputExceededException error.

To avoid request throttling, design your DynamoDB table with the right partition key to meet your access requirements and provide even distribution of data. Recommendations for doing this include the following:

• Use high cardinality attributes (e.g. email\_id, employee\_no, customer\_id etc.)

• Use composite attributes

• Cache popular items

• Add random numbers or digits from a pre-determined range for write-heavy use cases

In this case there is a hot partition due to the order date being used as the partition key and this is causing writes to be throttled. Therefore, the best solution to ensure the writes are more evenly distributed in this scenario is to add a random number suffix to the partition key values.

**CORRECT:**"Add a random number suffix to the partition key values" is the correct answer.

**INCORRECT:** "Increase the read and write capacity units for the table" is incorrect as this will not solve the hot partition issue and we know that the consumed throughput is lower than provisioned throughput.

**INCORRECT:** "Add a global secondary index to the table" is incorrect as a GSI is used for querying data more efficiently, it will not solve the problem of write performance due to a hot partition.

**INCORRECT:** "Use an Amazon SQS queue to buffer the incoming writes" is incorrect as this is not the lowest cost option. You would need to have producers and consumers of the queue as well as paying for the queue itself.

**References:**

<https://aws.amazon.com/blogs/database/choosing-the-right-dynamodb-partition-key/>

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

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

**Domain**

AWS Database

**Question 21Correct**

A company stores session information for a serverless application in an Amazon DynamoDB table. The company requires an automated process to eliminate outdated items from the table.

What is the most straightforward and lowest cost method to accomplish this?

**Use AWS DMS (Database Migration Service) to purge old items.**

**Implement a periodic AWS Lambda function to scan and delete old items.**

**Use Amazon CloudWatch to monitor and delete old items.**

**Your answer is correct**

**Use Amazon DynamoDB Time to Live (TTL) to automatically delete old items.**

Overall explanation

Amazon DynamoDB Time to Live (TTL) enables you to define a per-item timestamp to determine when an item is no longer needed. Shortly after the date and time of the specified timestamp, DynamoDB deletes the item from your table, making this the simplest automated method to remove old items.

**CORRECT:**"Use Amazon DynamoDB Time to Live (TTL) to automatically delete old items" is the correct answer (as explained above.)

**INCORRECT:** "Implement a periodic AWS Lambda function to scan and delete old items" is incorrect.

Although a periodic AWS Lambda function could potentially scan and delete old items, this would require additional development and maintenance efforts compared to using DynamoDB TTL.

**INCORRECT:** "Use AWS DMS (Database Migration Service) to purge old items" is incorrect.

AWS DMS (Database Migration Service) is used for migrating databases to AWS and is not designed to automatically purge old items from a database.

**INCORRECT:** "Use Amazon CloudWatch to monitor and delete old items" is incorrect.

Amazon CloudWatch is a monitoring service, not a tool designed to delete items from a database based on their age.

**References:**

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

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

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

**Domain**

AWS Database

**Question 22Correct**

A firm intends to utilize AWS CodeDeploy to deploy an application to Amazon Elastic Container Service (Amazon ECS). While deploying an updated version of the application, the company's initial requirement is to direct 10% of active traffic to the updated application version. Following a 15-minute interval, all remaining active traffic must be rerouted to the updated application.

Which predefined CodeDeploy configuration aligns with these needs?

**CodeDeployDefault.ECSAllAtOnce**

**CodeDeployDefault.LambdaLinear10PercentEvery1Minutes**

**Your answer is correct**

**CodeDeployDefault.ECSCanary10Percent15Minutes**

**CodeDeployDefault.ECSLinear10PercentEvery10Minutes**

Overall explanation

CodeDeployDefault.ECSCanary10Percent15Minutes is a predefined CodeDeploy deployment configuration that will first direct 10% of traffic to the newly deployed application and then, after 15 minutes, route the rest of the traffic to the new version. This fits perfectly with the company's requirements.

**CORRECT:**"CodeDeployDefault.ECSCanary10Percent15Minutes" is the correct answer (as explained above.)

**INCORRECT:** "CodeDeployDefault.ECSAllAtOnce" is incorrect.

CodeDeployDefault.ECSAllAtOnce deploys the application revision to all instances at once. This does not meet the requirement to initially expose only 10% of live traffic to the new version.

**INCORRECT:** "CodeDeployDefault.LambdaLinear10PercentEvery1Minutes" is incorrect.

CodeDeployDefault.LambdaLinear10PercentEvery1Minutes is for AWS Lambda deployments, not Amazon ECS deployments.

**INCORRECT:** "CodeDeployDefault.ECSLinear10PercentEvery10Minutes" is incorrect.

CodeDeployDefault.ECSLinear10PercentEvery10Minutes increases the traffic routed to the new version by 10% every 10 minutes, not all at once after 15 minutes as per the requirements.

**References:**

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

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

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

**Domain**

AWS Developer Tools

**Question 23Correct**

A Developer needs to write some code to invoke an AWS Lambda function using the AWS Command Line Interface (CLI). Which option must be specified to cause the function to be invoked asynchronously?

**Set the –invocation-type option to Invoke**

**Your answer is correct**

**Set the –invocation-type option to Event**

**Set the –payload option to Asynchronous**

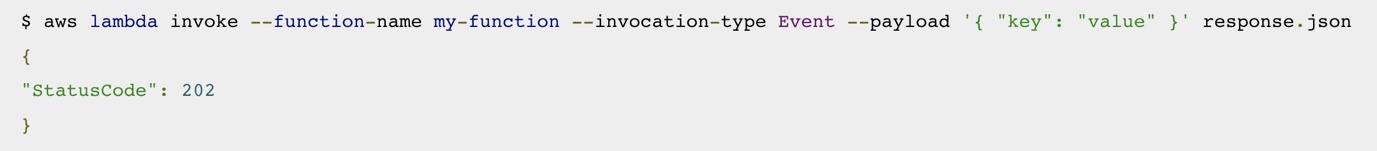
**Set the –qualifier option to Asynchronous**

Overall explanation

Several AWS services, such as Amazon Simple Storage Service (Amazon S3) and Amazon Simple Notification Service (Amazon SNS), invoke functions asynchronously to process events.

When you invoke a function asynchronously, you don't wait for a response from the function code. You hand off the event to Lambda and Lambda handles the rest. You can configure how Lambda handles errors and can send invocation records to a downstream resource to chain together components of your application.

The following code snippet is an example of invoking the “my-function” function asynchronously:



The Developer will therefore need to set the –invocation-type option to Event.

**CORRECT:**"Set the –invocation-type option to Event " is the correct answer.

**INCORRECT:** "Set the –invocation-type option to Invoke" is incorrect as this is not valid value for this option.

**INCORRECT:** "Set the –payload option to Asynchronous" is incorrect as this option is used to provide the JSON blob that you want to provide to your Lambda function as input. You cannot supply “asynchronous” as a value.

**INCORRECT:** "Set the –qualifier option to Asynchronous" is incorrect as this is used to specify a version or alias to invoke a published version of the function. You cannot supply “asynchronous” as a value.

**References:**

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

<https://docs.aws.amazon.com/cli/latest/reference/lambda/invoke.html>

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

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

**Domain**

AWS Compute

**Question 24Correct**

A development team are creating a mobile application that customers will use to receive notifications and special offers. Users will not be required to log in.

What is the MOST efficient method to grant users access to AWS resources?

**Embed access keys in the application that have limited access to resources**

**Your answer is correct**

**Use Amazon Cognito to associate unauthenticated users with an IAM role that has limited access to resources**

**Use Amazon Cognito Federated Identities and setup authentication using a Cognito User Pool**

**Use an IAM SAML 2.0 identity provider to establish trust**

Overall explanation

Amazon Cognito Identity Pools can support unauthenticated identities by providing a unique identifier and AWS credentials for users who do not authenticate with an identity provider. If your application allows users who do not log in, you can enable access for unauthenticated identities.

This is the most efficient and secure way to allow unauthenticated access as the process to set it up is simple and the IAM role can be configured with permissions allowing only the access permitted for unauthenticated users.

**CORRECT:**"Use Amazon Cognito to associate unauthenticated users with an IAM role that has limited access to resources" is the correct answer.

**INCORRECT:** "Use an IAM SAML 2.0 identity provider to establish trust" is incorrect as we need to allow unauthenticated users access to the AWS resources, not those who have been authenticated elsewhere (i.e. Active Directory).

**INCORRECT:** "Use Amazon Cognito Federated Identities and setup authentication using a Cognito User Pool" is incorrect as we need to setup unauthenticated access, not authenticated access through a user pool.

**INCORRECT:** "Embed access keys in the application that have limited access to resources" is incorrect. We should try and avoid embedding access keys in application code, it is better to use the built-in features of Amazon Cognito.

**References:**

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

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

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

**Domain**

AWS Security, Identity, & Compliance

**Question 25Correct**

A website is being delivered using Amazon CloudFront and a Developer recently modified some images that are displayed on website pages. Upon testing the changes, the Developer noticed that the new versions of the images are not displaying.

What should the Developer do to force the new images to be displayed?

**Delete the images from the origin and then save the new version on the origin**

**Your answer is correct**

**Invalidate the old versions of the images on the edge caches**

**Force an update of the cache**

**Invalidate the old versions of the images on the origin**

Overall explanation

If you need to remove a file from CloudFront edge caches before it expires, you can do one of the following:

• Invalidate the file from edge caches. The next time a viewer requests the file, CloudFront returns to the origin to fetch the latest version of the file.

• Use file versioning to serve a different version of the file that has a different name. For more information, see [Updating Existing Files Using Versioned File Names](https://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/UpdatingExistingObjects.html#ReplacingObjects).

To invalidate files, you can specify either the path for individual files or a path that ends with the \* wildcard, which might apply to one file or to many, as shown in the following examples:

• /images/image1.jpg

• /images/image\*

• /images/\*

Therefore, the Developer should invalidate the old versions of the images on the edge cache as this will remove the cached images and the new versions of the images will then be cached when the next request is received.

**CORRECT:**"Invalidate the old versions of the images on the edge caches" is the correct answer.

**INCORRECT:** "Delete the images from the origin and then save the new version on the origin" is incorrect as this will not cause the cache entries to expire. The Developer needs to remove the cached entries to cause a cache miss to occur which will then result in the updated images being cached.

**INCORRECT:** "Invalidate the old versions of the images on the origin" is incorrect as the Developer needs to invalidate the cache entries on the edge caches, not the images on the origin.

**INCORRECT:** "Force an update of the cache" is incorrect as there is no way to directly update the cache. The Developer should invalidate the relevant cache entries and then the cache will be updated next time a request is received for the images.

**References:**

<https://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/Invalidation.html>

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

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

**Domain**

AWS Networking & Content Delivery

**Question 26Correct**

A Developer is working on an AWS Lambda function that accesses Amazon DynamoDB. The Lambda function must retrieve an item and update some of its attributes or create the item if it does not exist. The Lambda function has access to the primary key.

Which IAM permission should the Developer request for the Lambda function to achieve this functionality?

**“dynamodb:GetRecords”, “dynamodb:PutItem”, and “dynamodb:UpdateTable”**

**“dynamodb:UpdateItem”, “dynamodb:GetItem”, and “dynamodb:DescribeTable”**

**“dynamodb:DeleteItem”, “dynamodb:GetItem”, and “dynamodb:PutItem”**

**Your answer is correct**

**“dynamodb:UpdateItem”, “dynamodb:GetItem”, and “dynamodb:PutItem”**

Overall explanation

The Developer needs the permissions to retrieve items, update/modify items, and create items. Therefore permissions for the following API actions are required:

• GetItem - The GetItem operation returns a set of attributes for the item with the given primary key.

• UpdateItem - Edits an existing item's attributes, or adds a new item to the table if it does not already exist. You can put, delete, or add attribute values.

• PutItem - Creates a new item, or replaces an old item with a new item. If an item that has the same primary key as the new item already exists in the specified table, the new item completely replaces the existing item.

**CORRECT:**"“dynamodb:UpdateItem”, “dynamodb:GetItem”, and “dynamodb:PutItem”" is the correct answer.

**INCORRECT:** "“dynamodb:DeleteItem”, “dynamodb:GetItem”, and “dynamodb:PutItem”" is incorrect as the Developer does not need the permission to delete items.

**INCORRECT:** "“dynamodb:UpdateItem”, “dynamodb:GetItem”, and “dynamodb:DescribeTable”" is incorrect as the Developer does not need to return information about the table (DescribeTable) such as the current status of the table, when it was created, the primary key schema, and any indexes on the table.

**INCORRECT:** "“dynamodb:GetRecords”, “dynamodb:PutItem”, and “dynamodb:UpdateTable”" is incorrect as GetRecords is not a valid API action/permission for DynamoDB.

**References:**

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

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

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

**Domain**

AWS Database

**Question 27Correct**

A Developer is creating a service on Amazon ECS and needs to ensure that each task is placed on a different container instance.

How can this be achieved?

**Use a task placement strategy**

**Create a service on Fargate**

**Your answer is correct**

**Use a task placement constraint**

**Create a cluster with multiple container instances**

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.

Amazon ECS supports the following types of task placement constraints:

distinctInstance

Place each task on a different container instance. This task placement constraint can be specified when either running a task or creating a new service.

memberOf

Place tasks on container instances that satisfy an expression. For more information about the expression syntax for constraints, see [Cluster Query Language](https://docs.aws.amazon.com/AmazonECS/latest/developerguide/cluster-query-language.html).

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 following code can be used in a task definition to specify a task placement constraint that ensures that each task will run on a distinct instance:

1. "placementConstraints": [
2. {
3. "type": "distinctInstance"
4. }
5. ]

**CORRECT:**"Use a task placement constraint" is the correct answer.

**INCORRECT:** "Use a task placement strategy" is incorrect as this is used to select instances for task placement using the binpack, random and spread algorithms.

**INCORRECT:** "Create a service on Fargate" is incorrect as Fargate spreads tasks across AZs but not instances.

**INCORRECT:** "Create a cluster with multiple container instances" is incorrect as this will not guarantee that each task runs on a different container instance.

**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 28Incorrect**

A company uses an Amazon Simple Queue Service (SQS) Standard queue for an application. An issue has been identified where applications are picking up messages from the queue that are still being processed causing duplication. What can a Developer do to resolve this issue?

**Correct answer**

**Increase the VisibilityTimeout API action on the queue**

**Your answer is incorrect**

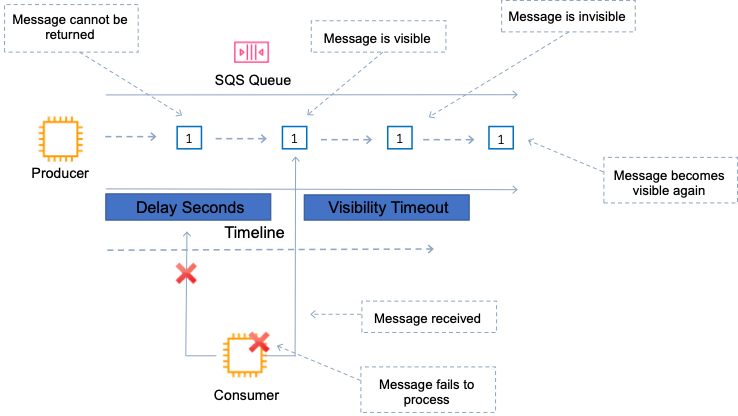
**Increase the DelaySeconds API action on the queue**

**Create a RedrivePolicy for the queue**

**Increase the ReceiveMessageWaitTimeSeconds API action on the queue**

Overall explanation

When a consumer receives and processes a message from a queue, the message remains in the queue. Amazon SQS doesn't automatically delete the message. Because Amazon SQS is a distributed system, there's no guarantee that the consumer actually receives the message (for example, due to a connectivity issue, or due to an issue in the consumer application). Thus, the consumer must delete the message from the queue after receiving and processing it.



Immediately after a message is received, it remains in the queue. To prevent other consumers from processing the message again, Amazon SQS sets a visibility timeout, a period of time during which Amazon SQS prevents other consumers from receiving and processing the message. The default visibility timeout for a message is 30 seconds. The minimum is 0 seconds. The maximum is 12 hours.

Therefore, the best thing the Developer can do in this situation is to increase the VisibilityTimeout API action on the queue

**CORRECT:**"Increase the VisibilityTimeout API action on the queue" is the correct answer.

**INCORRECT:** "Increase the DelaySeconds API action on the queue" is incorrect as this controls the length of time, in seconds, for which the delivery of all messages in the queue is delayed.

**INCORRECT:** "Increase the ReceiveMessageWaitTimeSeconds API action on the queue" is incorrect as this is the length of time, in seconds, for which a [ReceiveMessage](https://docs.aws.amazon.com/AWSSimpleQueueService/latest/APIReference/API_ReceiveMessage.html) action waits for a message to arrive. This is used to configure long polling.

**INCORRECT:** "Create a RedrivePolicy for the queue" is incorrect as this is a string that includes the parameters for the dead-letter queue functionality of the source queue as a JSON object.

**References:**

<https://docs.aws.amazon.com/AWSSimpleQueueService/latest/SQSDeveloperGuide/sqs-visibility-timeout.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 29Correct**

A company has transferred some of its confidential documents to a private Amazon S3 bucket that is not publicly accessible. Now, the company intends to build a serverless application that allows its staff to securely share these files with others.

Which AWS service should the company utilize to ensure secure file sharing and access?

**Your answer is correct**

**S3 presigned URLs**

**Amazon Cognito identity pool**

**S3 Access Control Lists (ACLs)**

**AWS Identity and Access Management (IAM) roles**

Overall explanation

S3 presigned URLs are the correct answer because they provide secure, temporary access to a specific S3 object without requiring AWS security credentials. This allows users to share the files securely with others.

**CORRECT:**"S3 presigned URLs" is the correct answer (as explained above.)

**INCORRECT:** "AWS Identity and Access Management (IAM) roles" is incorrect.

AWS Identity and Access Management (IAM) roles are used for granting applications or AWS services permissions to access AWS resources. They don't directly facilitate the secure sharing of S3 objects with external users.

**INCORRECT:** "Amazon Cognito identity pool" is incorrect.

Amazon Cognito identity pools are used to provide temporary AWS credentials to users of your application, not for directly enabling secure file sharing.

**INCORRECT:** "S3 Access Control Lists (ACLs)" is incorrect.

S3 Access Control Lists (ACLs) can be used to manage permissions at the object level, but they do not provide a method to securely share specific S3 objects with others.

**References:**

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

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

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

**Domain**

AWS Storage

**Question 30Incorrect**

A company uses Amazon SQS to decouple an online application that generates memes. The SQS consumers poll the queue regularly to keep throughput high and this is proving to be costly and resource intensive. A Developer has been asked to review the system and propose changes that can reduce costs and the number of empty responses.

What would be the BEST approach to MINIMIZING cost?

**Your answer is incorrect**

**Set the DelaySeconds parameter of a message to 20 seconds**

**Set the imaging queue MessageRetentionPeriod attribute to 20 seconds**

**Set the imaging queue visibility Timeout attribute to 20 seconds**

**Correct answer**

**Set the Imaging queue ReceiveMessageWaitTimeSeconds attribute to 20 seconds**

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).

Therefore, the best way to optimize resource usage and reduce the number of empty responses (and cost) is to configure long polling by setting the Imaging queue ReceiveMessageWaitTimeSeconds attribute to 20 seconds.

**CORRECT:**"Set the Imaging queue ReceiveMessageWaitTimeSeconds attribute to 20 seconds" is the correct answer.

**INCORRECT:** "Set the imaging queue visibility Timeout attribute to 20 seconds" is incorrect. This attribute configures message visibility which will not reduce empty responses.

**INCORRECT:** "Set the imaging queue MessageRetentionPeriod attribute to 20 seconds" is incorrect. This attribute sets the length of time, in seconds, for which Amazon SQS retains a message.

**INCORRECT:** "Set the DelaySeconds parameter of a message to 20 seconds" is incorrect. This attribute sets the length of time, in seconds, for which the delivery of all messages in the queue is delayed.

**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 31Incorrect**

A Developer has created the code for a Lambda function saved the code in a file named lambda\_function.py. He has also created a template that named template.yaml. The following code is included in the template file:

1. AWSTemplateFormatVersion: '2010-09-09'
2. Transform: 'AWS::Serverless-2016-10-31'
3. Resources:
4. microservicehttpendpointpython3:
5. Type: 'AWS::Serverless::Function'
6. Properties:
7. Handler: lambda\_function.lambda\_handler
8. CodeUri: .

What commands can the Developer use to prepare and then deploy this template? (Select TWO.)

**Run aws cloudformation compile and then aws cloudformation deploy**

**Run aws serverless package and then aws serverless deploy**

**Correct selection**

**Run aws cloudformation package and then aws cloudformation deploy**

**Your selection is correct**

**Run sam package and then sam deploy**

**Run sam build and then sam package**

Overall explanation

The template shown is an AWS SAM template for deploying a serverless application. This can be identified by the template header: Transform: 'AWS::Serverless-2016-10-31'

The Developer will need to package and then deploy the template. To do this the source code must be available in the same directory or referenced using the “codeuri” parameter. Then, the Developer can use the “aws cloudformation package” or “sam package” commands to prepare the local artifacts (local paths) that your AWS CloudFormation template references.

The command uploads local artifacts, such as source code for an AWS Lambda function or a Swagger file for an AWS API Gateway REST API, to an S3 bucket. The command returns a copy of your template, replacing references to local artifacts with the S3 location where the command uploaded the artifacts.

Once that is complete the template can be deployed using the “aws cloudformation deploy” or “sam deploy” commands. Therefore, the developer has two options to prepare and then deploy this package:

1. Run aws cloudformation package and then aws cloudformation deploy

2. Run sam package and then sam deploy

**CORRECT:**"Run aws cloudformation package and then aws cloudformation deploy" is a correct answer.

**INCORRECT:** "Run sam package and then sam deploy" is also a correct answer.

**INCORRECT:** "Run aws cloudformation compile and then aws cloudformation deploy" is incorrect as the “compile” command should be replaced with the “package” command.

**INCORRECT:** "Run sam build and then sam package" is incorrect as the Developer needs to run the “package” command first and then the “deploy” command to actually deploy the function.

**INCORRECT:** "Run aws serverless package and then aws serverless deploy" is incorrect as there is no AWS CLI command named “serverless”.

**References:**

<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 Management & Governance

**Question 32Incorrect**

A Developer received the following error when attempting to launch an Amazon EC2 instance using the AWS CLI.

An error occurred (UnauthorizedOperation) when calling the RunInstances operation: You are not authorized to perform this operation. Encoded authorization failure message: VNVaHFdCohROkbyT\_rIXoRyNTp7vXFJCqnGiwPuyKnsSVf-WSSGK\_06H3vKnrkUa3qx5D40hqj9HEG8kznr04Acmi6lvc8m51tfqtsomFSDylK15x96ZrxMW7MjDJLrMkM0BasPvy8ixo1wi6X2b0C-J1ThyWU9IcrGd7WbaRDOiGbBhJtKs1z01WSn2rVa5\_7sr5PwEK-ARrC9y5Pl54pmeF6wh7QhSv2pFO0y39WVBajL2GmByFmQ4p8s-6Lcgxy23b4NJdJwWOF4QGxK9HcKof1VTVZ2oIpsI-dH6\_0t2DI0BTwaIgmaT7ldontI1p7OGz-3wPgXm67x2NVNgaK63zPxjYNbpl32QuXLKUKNlB9DdkSdoLvsuFIvf-lQOXLPHnZKCWMqrkI87eqKHYpYKyV5c11TIZTAJ3MntTGO\_TJ4U9ySYvTzU2LgswYOtKF\_O76-13fryGG5dhgOW5NxwCWBj6WT2NSJvqOeLykAFjR\_ET4lM6Dl1XYfQITWCqIzlvlQdLmHJ1jqjp4gW56VcQCdqozLv2UAg8IdrZIXd0OJ047RQcvvN1IyZN0ElL7dR6RzAAQrftoKMRhZQng6THZs8PZM6wep6-yInzwfg8J5\_FW6G\_PwYqO-4VunVtJSTzM\_F\_8kojGlRmzqy7eCk5or\_\_bIisUoslw

What action should the Developer perform to make this error more human-readable?

**Correct answer**

**Use the AWS STS decode-authorization-message API to decode the message**

**Make a call to AWS KMS to decode the message**

**Your answer is incorrect**

**Use the AWS IAM decode-authorization-message API to decode this message**

**Use an open source decoding library to decode the message**

Overall explanation

The AWS STS decode-authorization-message API decodes additional information about the authorization status of a request from an encoded message returned in response to an AWS request. The output is then decoded into a more human-readable output that can be viewed in a JSON editor.

The following example is the decoded output from the error shown in the question:

1. {
2. "DecodedMessage": "{\"allowed\":false,\"explicitDeny\":false,\"matchedStatements\":{\"items\":[]},\"failures\":{\"items\":[]},\"context\":{\"principal\":{\"id\":\"AIDAXP4J2EKU7YXXG3EJ4\",\"name\":\"Paul\",\"arn\":\"arn:aws:iam::515148227241:user/Paul\"},\"action\":\"ec2:RunInstances\",\"resource\":\"arn:aws:ec2:ap-southeast-2:515148227241:instance/\*\",\"conditions\":{\"items\":[{\"key\":\"ec2:InstanceMarketType\",\"values\":{\"items\":[{\"value\":\"on-demand\"}]}},{\"key\":\"aws:Resource\",\"values\":{\"items\":[{\"value\":\"instance/\*\"}]}},{\"key\":\"aws:Account\",\"values\":{\"items\":[{\"value\":\"515148227241\"}]}},{\"key\":\"ec2:AvailabilityZone\",\"values\":{\"items\":[{\"value\":\"ap-southeast-2a\"}]}},{\"key\":\"ec2:ebsOptimized\",\"values\":{\"items\":[{\"value\":\"false\"}]}},{\"key\":\"ec2:IsLaunchTemplateResource\",\"values\":{\"items\":[{\"value\":\"false\"}]}},{\"key\":\"ec2:InstanceType\",\"values\":{\"items\":[{\"value\":\"t2.micro\"}]}},{\"key\":\"ec2:RootDeviceType\",\"values\":{\"items\":[{\"value\":\"ebs\"}]}},{\"key\":\"aws:Region\",\"values\":{\"items\":[{\"value\":\"ap-southeast-2\"}]}},{\"key\":\"aws:Service\",\"values\":{\"items\":[{\"value\":\"ec2\"}]}},{\"key\":\"ec2:InstanceID\",\"values\":{\"items\":[{\"value\":\"\*\"}]}},{\"key\":\"aws:Type\",\"values\":{\"items\":[{\"value\":\"instance\"}]}},{\"key\":\"ec2:Tenancy\",\"values\":{\"items\":[{\"value\":\"default\"}]}},{\"key\":\"ec2:Region\",\"values\":{\"items\":[{\"value\":\"ap-southeast-2\"}]}},{\"key\":\"aws:ARN\",\"values\":{\"items\":[{\"value\":\"arn:aws:ec2:ap-southeast-2:515148227241:instance/\*\"}]}}]}}}"
3. }

Therefore, the best answer is to use the AWS STS decode-authorization-message API to decode the message.

**CORRECT:**"Use the AWS STS decode-authorization-message API to decode the message" is the correct answer.

**INCORRECT:** "Make a call to AWS KMS to decode the message" is incorrect as the message is not encrypted, it is base64 encoded.

**INCORRECT:** "Use an open source decoding library to decode the message" is incorrect as you can use the AWS STS decode-authorization-message API.

**INCORRECT:** "Use the AWS IAM decode-authorization-message API to decode this message" is incorrect as the decode-authorization-message API is associated with STS, not IAM.

**References:**

<https://docs.aws.amazon.com/cli/latest/reference/sts/decode-authorization-message.html>

**Domain**

AWS Security, Identity, & Compliance

**Question 33Correct**

A Developer needs to configure an Elastic Load Balancer that is deployed through AWS Elastic Beanstalk. Where should the Developer place the load-balancer.config file in the application source bundle?

**Your answer is correct**

**In the .ebextensions folder**

**In the root of the source code**

**In the bin folder**

**In the load-balancer.config.root**

Overall explanation

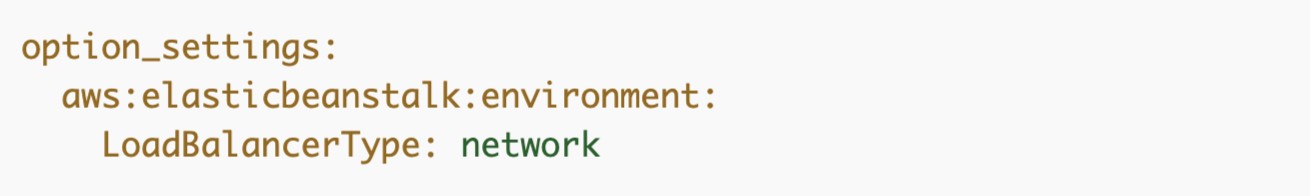
You can add AWS Elastic Beanstalk configuration files (.ebextensions) to your web application's source code to configure your environment and customize the AWS resources that it contains.

Configuration files are YAML- or JSON-formatted documents with a .config file extension that you place in a folder named .ebextensions and deploy in your application source bundle.

For example, you could include a configuration file for setting the load balancer type into:

.ebextensions/load-balancer.config

This example makes a simple configuration change. It modifies a configuration option to set the type of your environment's load balancer to Network Load Balancer:



**Requirements**

•  **Location** – Place all of your configuration files in a single folder, named .ebextensions, in the root of your source bundle. Folders starting with a dot can be hidden by file browsers, so make sure that the folder is added when you create your source bundle.

•  **Naming** – Configuration files must have the .config file extension.

•  **Formatting** – Configuration files must conform to YAML or JSON specifications.

•  **Uniqueness** – Use each key only once in each configuration file.

Therefore, the Developer should place the file in the .ebextensions folder in the application source bundle.

**CORRECT:**"In the .ebextensions folder" is the correct answer.

**INCORRECT:** "In the root of the source code" is incorrect. You need to place .config files in the .ebextensions folder.

**INCORRECT:** "In the bin folder" is incorrect. You need to place .config files in the .ebextensions folder.

**INCORRECT:** "In the load-balancer.config.root" is incorrect. You need to place .config files in the .ebextensions folder.

**References:**

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

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

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

**Domain**

AWS Compute

**Question 34Correct**

A company is creating a REST service using an Amazon API Gateway with AWS Lambda integration. The service must run different versions for testing purposes.

What would be the BEST way to accomplish this?

**Create an API Gateway resource policy to isolate versions and provide context to the Lambda function(s)**

**Your answer is correct**

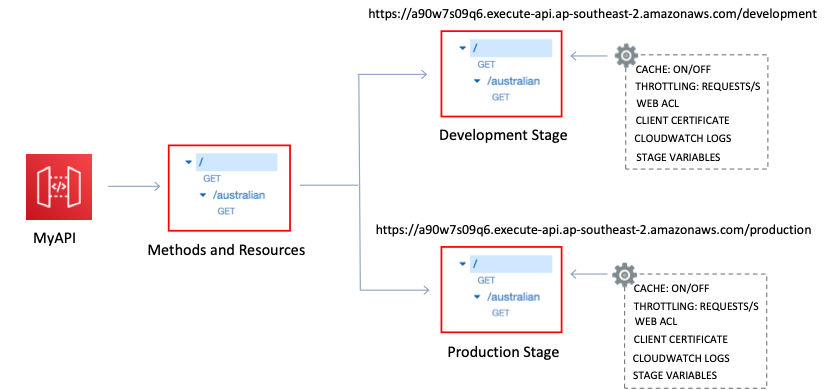
**Deploy the API version as unique stages with unique endpoints and use stage variables to provide further context**

**Use an X-Version header to denote which version is being called and pass that header to the Lambda function(s)**

**Create an API Gateway Lambda authorizer to route API clients to the correct API version**

Overall explanation

A stage is a named reference to a deployment, which is a snapshot of the API. You use a [Stage](https://docs.aws.amazon.com/apigateway/api-reference/resource/stage/) to manage and optimize a particular deployment. For example, you can configure stage settings to enable caching, customize request throttling, configure logging, define stage variables, or attach a canary release for testing.



Stage variables are name-value pairs that you can define as configuration attributes associated with a deployment stage of a REST API. They act like environment variables and can be used in your API setup and mapping templates.

With stages and stage variables, you can configure different settings for different versions of the application and point to different versions of your Lambda function.

**CORRECT:**"Deploy the API version as unique stages with unique endpoints and use stage variables to provide further context" is the correct answer.

**INCORRECT:** "Use an X-Version header to denote which version is being called and pass that header to the Lambda function(s)" is incorrect as you cannot pass a value in a header to a Lambda function and have that determine which version is executed. Versions have unique ARNs and must be connected to separately.

**INCORRECT:** "Create an API Gateway Lambda authorizer to route API clients to the correct API version" is incorrect as a Lambda authorizer is used for authentication, and different versions of an API are created using stages.

**INCORRECT:** "Create an API Gateway resource policy to isolate versions and provide context to the Lambda function(s)" is incorrect as resource policies are not used to isolate versions or provide context. In this scenario, stages and stage variables should be used.

**References:**

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

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

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

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

**Domain**

AWS Networking & Content Delivery

**Question 35Correct**

An online retail application developer is planning to migrate to AWS to accommodate a future surge in traffic. Currently, a web server, which hosts the web application and manages session state in memory, and a separate server hosting a MySQL database for order details, are used.

During peak traffic, memory usage on the web server reaches its limit, leading to considerable slowdowns. As part of the migration plan, the developer intends to use Amazon EC2 instances with an Auto Scaling group and an Application Load Balancer for the web server.

What other changes can the developer implement to enhance application performance?

**Store both the session data and the application data in a MySQL database hosted on an EC2 instance.**

**Leverage the EC2 instance store for managing the session data and Amazon RDS for MySQL DB instance for application data storage.**

**Use Amazon ElastiCache for Memcached to store and manage both the session data and the application data.**

**Your answer is correct**

**Use Amazon ElastiCache for Memcached to store and manage session data, while utilizing Amazon RDS for MySQL DB instance for application data storage.**

Overall explanation

The main performance issue with the current application setup is due to managing additional user sessions, which significantly increases memory usage. To address this, Amazon ElastiCache for Memcached is an ideal service as it is designed to offload the burden of managing session state from web servers, providing fast, in-memory data storage.

For application data, Amazon RDS for MySQL DB instance is a fully managed relational database service, which takes care of database management tasks and provides cost-efficient and resizable capacity.

**CORRECT:**"Use Amazon ElastiCache for Memcached to store and manage session data, while utilizing Amazon RDS for MySQL DB instance for application data storage" is the correct answer (as explained above.)

**INCORRECT:** "Store both the session data and the application data in a MySQL database hosted on an EC2 instance" is incorrect.

Storing both session and application data in a MySQL database hosted on an EC2 instance might overload the database server leading to performance issues.

**INCORRECT:** "Use Amazon ElastiCache for Memcached to store and manage both the session data and the application data" is incorrect.

Amazon ElastiCache for Memcached is not typically used for application data storage but for caching frequently accessed data and offloading the database.

**INCORRECT:** "Leverage the EC2 instance store for managing the session data and Amazon RDS for MySQL DB instance for application data storage" is incorrect.

EC2 instance store provides temporary block-level storage for instances. This storage is ideal for temporary data, but not for session data, as data is lost if the instance is stopped or fails.

**References:**

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

<https://aws.amazon.com/rds/mysql/>

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

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

**Domain**

AWS Database

**Question 36Incorrect**

A company is providing APIs as a web-based service to allow anonymous access to daily updated statistical data, using Amazon API Gateway and AWS Lambda for API development. The service's popularity has grown, and the company aims to improve the API responsiveness.

What measure should the company undertake to fulfill this objective?

**Set up API keys and usage plans in API Gateway.**

**Correct answer**

**Activate caching in API Gateway.**

**Your answer is incorrect**

**Cache API results in an Amazon ElastiCache cluster.**

**Set up API Gateway with an interface VPC endpoint.**

Overall explanation

Enabling caching in API Gateway allows the service to cache the endpoint's responses, which improves the performance of the APIs by reducing the number of calls made to the endpoint and by improving the latency of the requests.

**CORRECT:**"Activate caching in API Gateway" is the correct answer (as explained above.)

**INCORRECT:** "Set up API Gateway with an interface VPC endpoint" is incorrect.

Configuring API Gateway to use an interface VPC endpoint helps in enhancing the security and privacy of the data being exchanged with the service. It does not directly affect the API responsiveness.

**INCORRECT:** "Cache API results in an Amazon ElastiCache cluster" is incorrect.

Amazon ElastiCache can be used for caching data from databases but is not suitable for caching API results for API gateway.

**INCORRECT:** "Set up API keys and usage plans in API Gateway" is incorrect.

Configuring usage plans and API keys in API Gateway can help control and manage usage of the APIs. However, they don't directly improve API responsiveness.

**References:**

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

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

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

**Domain**

AWS Networking & Content Delivery

**Question 37Correct**

A company uses continuous integration and continuous delivery (CI/CD) systems. A Developer needs to automate the deployment of a software package to Amazon EC2 instances as well as to on-premises virtual servers.

Which AWS service can be used for the software deployment?

**Your answer is correct**

**AWS CodeDeploy**

**AWS CloudBuild**

**AWS Elastic Beanstalk**

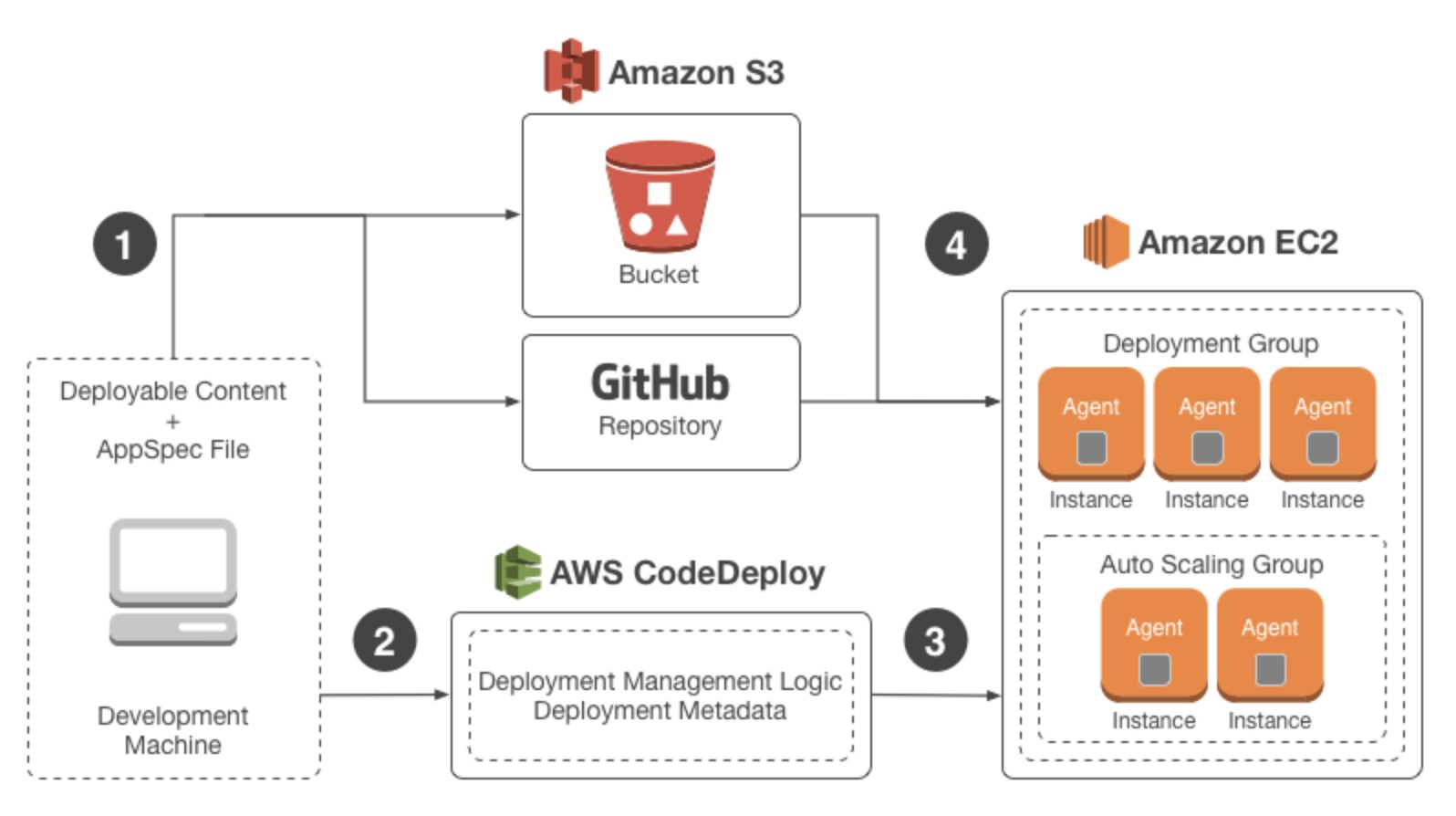
**AWS CodePipeline**

Overall explanation

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

CodeDeploy can deploy application content that runs on a server and is stored in Amazon S3 buckets, GitHub repositories, or Bitbucket repositories. CodeDeploy can also deploy a serverless Lambda function. You do not need to make changes to your existing code before you can use CodeDeploy.

The image below shows the flow of a typical CodeDeploy in-place deployment.



The above deployment could also be directed at on-premises servers. Therefore, the best answer is to use AWS CodeDeploy to deploy the software package to both EC2 instances and on-premises virtual servers.

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

**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. You can use CodeDeploy in a CodePipeline pipeline however it is actually CodeDeploy that deploys the software packages.

**INCORRECT:** "AWS CloudBuild" is incorrect as this is a build tool, not a deployment tool.

**INCORRECT:** "AWS Elastic Beanstalk" is incorrect as you cannot deploy software packages to on-premise virtual servers using Elastic Beanstalk

**References:**

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

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

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

**Domain**

AWS Developer Tools

**Question 38Incorrect**

An application running on a fleet of EC2 instances use the AWS SDK for Java to copy files into several AWS buckets using access keys stored in environment variables. A Developer has modified the instances to use an assumed IAM role with a more restrictive policy that allows access to only one bucket.

However, after applying the change the Developer logs into one of the instances and is still able to write to all buckets. What is the MOST likely explanation for this situation?

**Correct answer**

**The AWS credential provider looks for instance profile credentials last**

**An IAM managed policy is being used on the IAM role**

**The AWS CLI is corrupt and needs to be reinstalled**

**Your answer is incorrect**

**An IAM inline policy is being used on the IAM role**

Overall explanation

When you initialize a new service client without supplying any arguments, the AWS SDK for Java attempts to find AWS credentials by using the default credential provider chain implemented by the [DefaultAWSCredentialsProviderChain](https://docs.aws.amazon.com/AWSJavaSDK/latest/javadoc/index.html?com/amazonaws/auth/DefaultAWSCredentialsProviderChain.html) class. The default credential provider chain looks for credentials in this order:

**1. Environment variables**–AWS\_ACCESS\_KEY\_ID and AWS\_SECRET\_ACCESS\_KEY. The AWS SDK for Java uses the [EnvironmentVariableCredentialsProvider](https://docs.aws.amazon.com/AWSJavaSDK/latest/javadoc/index.html?com/amazonaws/auth/EnvironmentVariableCredentialsProvider.html) class to load these credentials.

**2. Java system properties**–aws.accessKeyId and aws.secretKey. The AWS SDK for Java uses the [SystemPropertiesCredentialsProvider](https://docs.aws.amazon.com/AWSJavaSDK/latest/javadoc/index.html?com/amazonaws/auth/SystemPropertiesCredentialsProvider.html) to load these credentials.

**3. The default credential profiles file**– typically located at ~/.aws/credentials (location can vary per platform) and shared by many of the AWS SDKs and by the AWS CLI. The AWS SDK for Java uses the [ProfileCredentialsProvider](https://docs.aws.amazon.com/AWSJavaSDK/latest/javadoc/index.html?com/amazonaws/auth/profile/ProfileCredentialsProvider.html) to load these credentials.

**4. Amazon ECS container credentials**– loaded from the Amazon ECS if the environment variable AWS\_CONTAINER\_CREDENTIALS\_RELATIVE\_URI is set. The AWS SDK for Java uses the [ContainerCredentialsProvider](https://docs.aws.amazon.com/AWSJavaSDK/latest/javadoc/index.html?com/amazonaws/auth/ContainerCredentialsProvider.html) to load these credentials. You can specify the IP address for this value.

**5. Instance profile credentials**– used on EC2 instances and delivered through the Amazon EC2 metadata service. The AWS SDK for Java uses the [InstanceProfileCredentialsProvider](https://docs.aws.amazon.com/AWSJavaSDK/latest/javadoc/index.html?com/amazonaws/auth/InstanceProfileCredentialsProvider.html) to load these credentials. You can specify the IP address for this value.

Therefore, the AWS SDK for Java will find the credentials stored in environment variables before it checks for instance provide credentials and will allow access to the extra S3 buckets.

**NOTE:** The Default Credential Provider Chain is very similar for other SDKs and the CLI as well. Check the references below for an article showing the steps for the AWS CLI.

**CORRECT:**"The AWS credential provider looks for instance profile credentials last" is the correct answer.

**INCORRECT:** "An IAM inline policy is being used on the IAM role" is incorrect. If an inline policy was also applied to the role with a less restrictive policy it wouldn’t matter, as the most restrictive policy would be applied.

**INCORRECT:** "An IAM managed policy is being used on the IAM role" is incorrect. Though the managed policies are less restrictive by default (read-only or full access), this is not the most likely cause of the situation as we were told the policy is more restrictive and we know the environments variables have access keys in them which will be used before the policy is checked.

**INCORRECT:** "The AWS CLI is corrupt and needs to be reinstalled" is incorrect. There is a plausible explanation for this situation so no reason to suspect a software bug is to blame.

**References:**

<https://docs.aws.amazon.com/sdk-for-java/v1/developer-guide/credentials.html>

<https://docs.aws.amazon.com/cli/latest/userguide/cli-chap-configure.html>

**Domain**

AWS Compute

**Question 39Correct**

A Developer has updated an AWS Lambda function and published a new version. To ensure the code is working as expected the Developer needs to initially direct a percentage of traffic to the new version and gradually increase this over time. It is important to be able to rollback if there are any issues reported.

What is the BEST way the Developer can implement the migration to the new version SAFELY?

**Create an Amazon Route 53 weighted routing policy pointing to the current and new versions, assign a lower weight to the new version**

**Use an immutable update with a new ASG to deploy the new version in parallel, following testing cutover to the new version**

**Use an Amazon Elastic Load Balancer to direct a percentage of traffic to each target group containing the Lambda function versions**

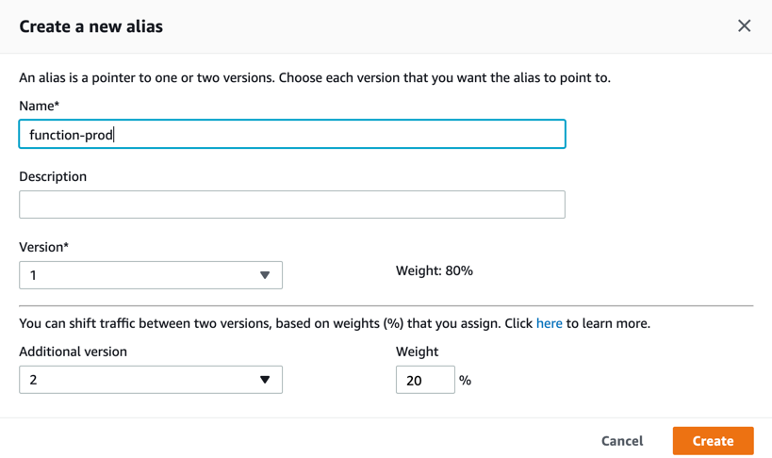
**Your answer is correct**

**Create an Alias, assign the current and new versions and use traffic shifting to assign a percentage of traffic to the new 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.

Each alias has a unique ARN. An alias can only point to a function version, not to another alias. You can update an alias to point to a new version of the function. You can also use traffic shifting to direct a percentage of traffic to a specific version as showing in the image below:



This is the recommended way to direct traffic to multiple function versions and shift traffic when testing code updated. Therefore, the best answer is to create an Alias, assign the current and new versions and use traffic shifting to assign a percentage of traffic to the new version.

**CORRECT:**"Create an Alias, assign the current and new versions and use traffic shifting to assign a percentage of traffic to the new version" is the correct answer.

**INCORRECT:** "Create an Amazon Route 53 weighted routing policy pointing to the current and new versions, assign a lower weight to the new version" is incorrect. AWS Lambda endpoints are not DNS names that you can route to with Route 53. The best way to route traffic to multiple versions is using an alias.

**INCORRECT:** "Use an immutable update with a new ASG to deploy the new version in parallel, following testing cutover to the new version" is incorrect as immutable updates are associated with Amazon Elastic Beanstalk and this service does not deploy updates to AWS Lambda.

**INCORRECT:** "Use an Amazon Elastic Load Balancer to direct a percentage of traffic to each target group containing the Lambda function versions" is incorrect as this introduces an unnecessary layer (complexity and cost) to the architecture. The best choice is to use an alias instead.

**References:**

<https://docs.amazonaws.cn/en_us/lambda/latest/dg/configuration-aliases.html>

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

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

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

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

**Domain**

AWS Compute

**Question 40Correct**

A gaming application stores scores for players in an Amazon DynamoDB table that has four attributes: user\_id, user\_name, user\_score, and user\_rank. The users are allowed to update their names only. A user is authenticated by web identity federation.

Which set of conditions should be added in the policy attached to the role for the dynamodb:PutItem API call?

**"Condition": {**

**"ForAllValues:StringEquals": {**

**"dynamodb:LeadingKeys": [**

**"${www.amazon.com:user\_id}"**

**],**

**"dynamodb:Attributes": [**

**"user\_name", "user\_id"**

**]**

**}**

**}**

**"Condition": {**

**"ForAllValues:StringEquals": {**

**"dynamodb:LeadingKeys": [**

**"${www.amazon.com:user\_name}"**

**],**

**"dynamodb:Attributes": [**

**"user\_name", "user\_id"**

**]**

**}**

**}**

**Your answer is correct**

**"Condition": {**

**"ForAllValues:StringEquals": {**

**"dynamodb:LeadingKeys": [**

**"${www.amazon.com:user\_id}"**

**],**

**"dynamodb:Attributes": [**

**"user\_name"**

**]**

**}**

**}**

**"Condition": {**

**"ForAllValues:StringEquals": {**

**"dynamodb:LeadingKeys": [**

**"${www.amazon.com:user\_name}"**

**],**

**"dynamodb:Attributes": [**

**"user\_id"**

**]**

**}**

**}**

Overall explanation

The users are authenticated by web identity federation. The user\_id value should be used to identify the user in the policy and the policy needs to then allow the user to change the user\_name value when using the dynamodb:PutItem API call.

The key parts of the code to look for are the dynamodb:LeadingKeys which represents the partition key of the table and the dynamodb:Attributes which represents the items that can be changed.

**CORRECT:**The answer that includes dynamodb:LeadingKeys identifying user\_id and dynamodb:Attributes identifying user\_name is the correct answer.

**INCORRECT:** The other answers provide a few incorrect code samples where either the dynamodb:LeadingKeys identifies user\_name (which is incorrect as it is the item to be changed) or dynamodb:Attributes identifying the wrong attributes for modification (should be user\_name).

**References:**

<https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/specifying-conditions.html#FGAC_DDB.ConditionKeys>

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

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

**Domain**

AWS Database

**Question 41Correct**

A small team of Developers require access to an Amazon S3 bucket. An admin has created a resource-based policy. Which element of the policy should be used to specify the ARNs of the user accounts that will be granted access?

**Condition**

**Your answer is correct**

**Principal**

**Sid**

**Id**

Overall explanation

Use the Principal element in a policy to specify the principal that is allowed or denied access to a resource. You cannot use the Principal element in an IAM identity-based policy. You can use it in the trust policies for IAM roles and in resource-based policies. Resource-based policies are policies that you embed directly in an IAM resource.

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

**INCORRECT:** "Condition" is incorrect. The Condition element (or Condition block) lets you specify conditions for when a policy is in effect.

**INCORRECT:** "Sid" is incorrect. The Sid (statement ID) is an optional identifier that you provide for the policy statement.

**INCORRECT:** "Id" is incorrect. The Id element specifies an optional identifier for the policy.

**References:**

<https://docs.aws.amazon.com/IAM/latest/UserGuide/reference_policies_elements_principal.html>

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

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

**Domain**

AWS Security, Identity, & Compliance

**Question 42Correct**

A Developer must deploy a new AWS Lambda function using an AWS CloudFormation template.

Which procedures will deploy a Lambda function? (Select TWO.)

**Your selection is correct**

**Create an AWS::Lambda::Function resource in the template, then write the code directly inside the CloudFormation template**

**1. Upload the function code to a private Git repository, then add a reference to it in an AWS::Lambda::Function resource in the template**

**Upload the code to an AWS CodeCommit repository, then add a reference to it in an AWS::Lambda::Function resource in the template**

**Upload a ZIP file to AWS CloudFormation containing the function code, then add a reference to it in an AWS::Lambda::Function resource in the template**

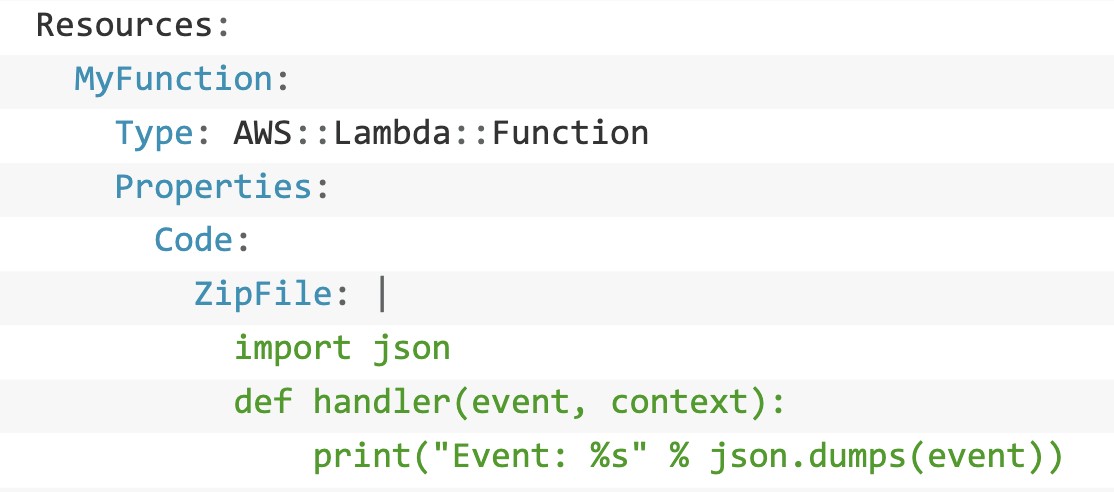
**Your selection is correct**

**Upload a ZIP file containing the function code to Amazon S3, then add a reference to it in an AWS::Lambda::Function resource in the template**

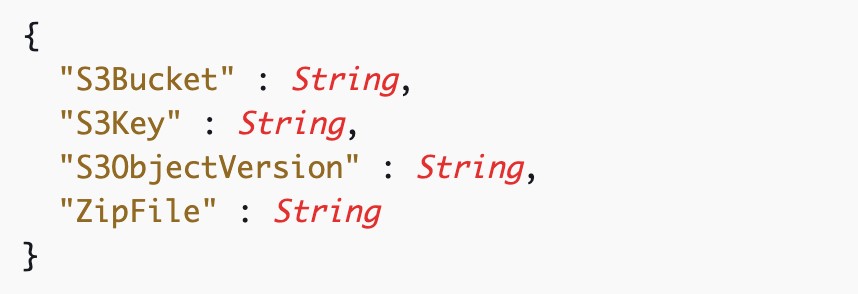
Overall explanation

Of the options presented there are two workable procedures for deploying the Lambda function.

Firstly, you can create an AWS::Lambda::Function resource in the template, then write the code directly inside the CloudFormation template. This is possible for simple functions using Node.js or Python which allow you to declare the code inline in the CloudFormation template. For example:



The other option is to upload a ZIP file containing the function code to Amazon S3, then add a reference to it in an AWS::Lambda::Function resource in the template. To declare this in your AWS CloudFormation template, you can use the following syntax (within AWS::Lambda::Function Code):



**CORRECT:**"Create an AWS::Lambda::Function resource in the template, then write the code directly inside the CloudFormation template" is a correct answer.

**CORRECT:**"Upload a ZIP file containing the function code to Amazon S3, then add a reference to it in an AWS::Lambda::Function resource in the template" is also a correct answer.

**INCORRECT:** "Upload the code to an AWS CodeCommit repository, then add a reference to it in an AWS::Lambda::Function resource in the template" is incorrect as you cannot add a reference to code in a CodeCommit repository.

**INCORRECT:** "Upload a ZIP file to AWS CloudFormation containing the function code, then add a reference to it in an AWS::Lambda::Function resource in the template" is incorrect as you cannot reference a zip file in CloudFormation.

**INCORRECT:** "Upload the function code to a private Git repository, then add a reference to it in an AWS::Lambda::Function resource in the template" is incorrect as you cannot reference the function code in a private Git repository.

**References:**

<https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/aws-properties-lambda-function-code.html>

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

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

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

**Domain**

AWS Management & Governance

**Question 43Correct**

An application is using Amazon DynamoDB as its data store and needs to be able to read 200 items per second as eventually consistent reads. Each item is 12 KB in size.  
What value should be set for the table's provisioned throughput for reads?

**Your answer is correct**

**300 Read Capacity Units**

**1200 Read Capacity Units**

**150 Read Capacity Units**

**600 Read Capacity Units**

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):**

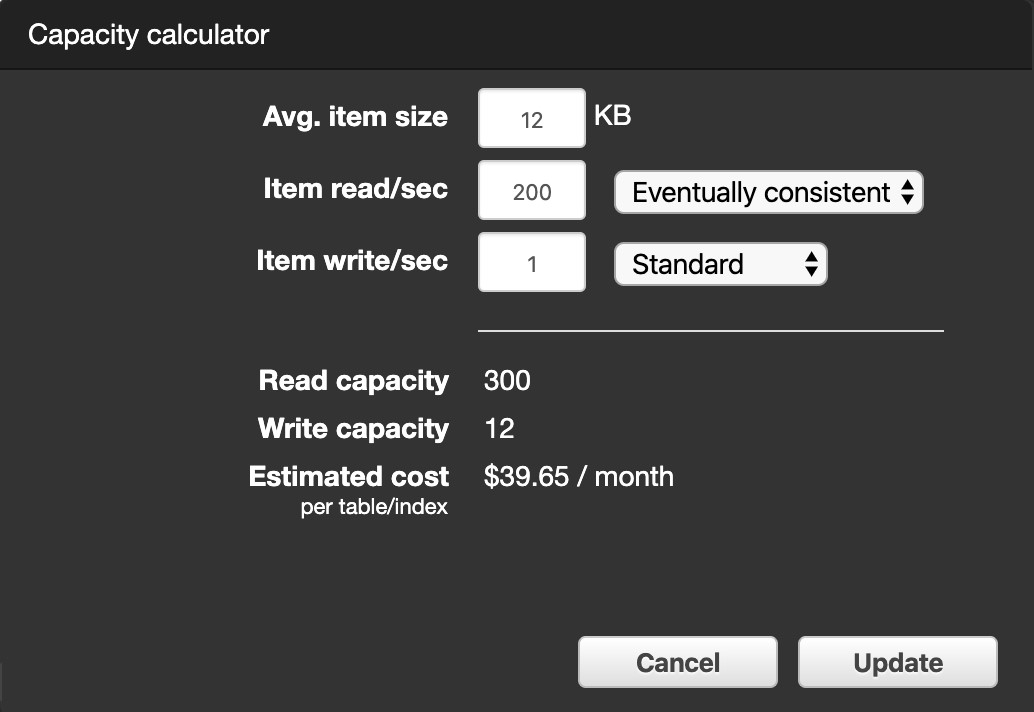
• Each API call to write data to your table is a write request.

• 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 200 eventually consistent reads per/second with an average item size of 12KB, perform the following steps:

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

    2. Determine the RCU per item by dividing the item size by 8KB (12KB/8KB = 1.5).

    3. Multiply the value from step 2 with the number of reads required per second (1.5x200 = 300).

**CORRECT:**"300 Read Capacity Units" is the correct answer.

**INCORRECT:** "600 Read Capacity Units" is incorrect. This would be the value for strongly consistent reads.

**INCORRECT:** "1200 Read Capacity Units" is incorrect. This would be the value for transactional reads.

**INCORRECT:** "150 Read Capacity Units" is incorrect.

**References:**

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

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

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

**Domain**

AWS Database

**Question 44Correct**

A Developer has deployed an AWS Lambda function and an Amazon DynamoDB table. The function code returns data from the DynamoDB table when it receives a request. The Developer needs to implement a front end that can receive HTTP GET requests and proxy the request information to the Lambda function.

What is the SIMPLEST and most COST-EFFECTIVE solution?

**Implement an Amazon Cognito User Pool with a Lambda proxy integration**

**Implement an Elastic Load Balancer with a Lambda function target**

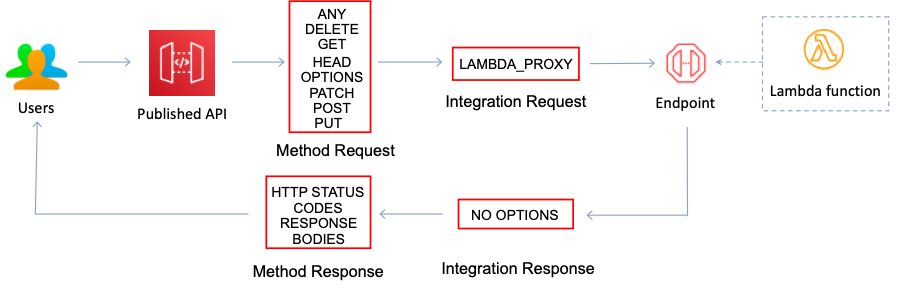
**Your answer is correct**

**Implement an API Gateway API with Lambda proxy integration**

**Implement an API Gateway API with a POST method**

Overall explanation

Amazon API Gateway Lambda proxy integration is a simple, powerful, and nimble mechanism to build an API with a setup of a single API method. The Lambda proxy integration allows the client to call a single Lambda function in the backend. The function accesses many resources or features of other AWS services, including calling other Lambda functions.



In Lambda proxy integration, when a client submits an API request, API Gateway passes to the integrated Lambda function the raw request as-is, except that the order of the request parameters is not preserved. This [request data](https://docs.aws.amazon.com/apigateway/latest/developerguide/set-up-lambda-proxy-integrations.html#api-gateway-simple-proxy-for-lambda-input-format) includes the request headers, query string parameters, URL path variables, payload, and API configuration data.

This solution provides a front end that can listen for HTTP GET requests and then proxy them to the Lambda function and is the simplest option to implement and also the most cost-effective.

**CORRECT:**"Implement an API Gateway API with Lambda proxy integration" is the correct answer.

**INCORRECT:** "Implement an API Gateway API with a POST method" is incorrect as a GET method should be implemented. A GET method is a request for data whereas a POST method is a request to upload data.

**INCORRECT:** "Implement an Elastic Load Balancer with a Lambda function target" is incorrect as though you can do this it is not the simplest or most cost-effective solution.

**INCORRECT:** "Implement an Amazon Cognito User Pool with a Lambda proxy integration" is incorrect as you cannot create Lambda proxy integrations with Cognito.

**References:**

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

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

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

**Domain**

AWS Networking & Content Delivery

**Question 45Correct**

A team of Developers require access to an AWS account that is a member account in AWS Organizations. The administrator of the master account needs to restrict the AWS services, resources, and API actions that can be accessed by the users in the account.

What should the administrator create?

**A Consolidated Billing account**

**A Tag Policy**

**An Organizational Unit**

**Your answer is correct**

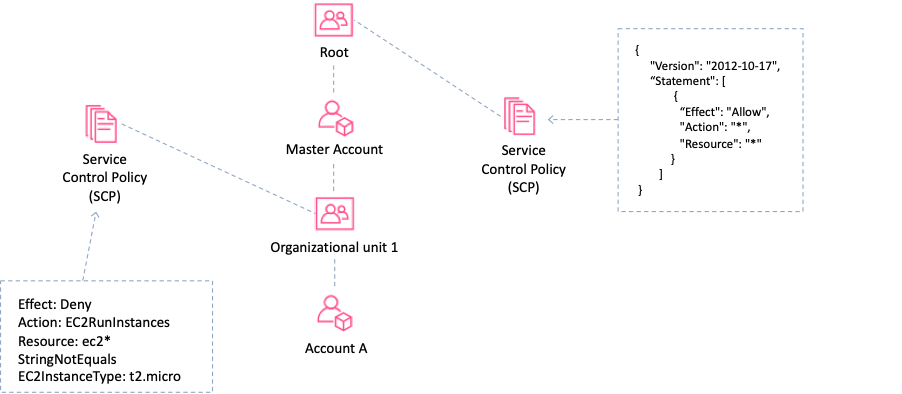
**A Service Control Policy (SCP)**

Overall explanation

As an administrator of the master account of an organization, you can use service control policies (SCPs) to specify the maximum permissions for member accounts in the organization.

In SCPs, you can restrict which AWS services, resources, and individual API actions the users and roles in each member account can access. You can also define conditions for when to restrict access to AWS services, resources, and API actions.

The following example shows how an SCP can be created to restrict the EC2 instance types that any user can run in the account:



These restrictions even override the administrators of member accounts in the organization. When AWS Organizations blocks access to a service, resource, or API action for a member account, a user or role in that account can't access it. This block remains in effect even if an administrator of a member account explicitly grants such permissions in an IAM policy.

**CORRECT:**"A Service Control Policy (SCP)" is the correct answer.

**INCORRECT:** "A Tag Policy" is incorrect as these are used to maintain consistent tags, including the preferred case treatment of tag keys and tag values.

**INCORRECT:** "An Organizational Unit" is incorrect as this is used to group accounts for administration.

**INCORRECT:** "A Consolidated Billing account" is incorrect as consolidated billing is not related to controlling access to resources within an account.

**References:**

<https://docs.aws.amazon.com/organizations/latest/userguide/orgs_introduction.html>

<https://docs.aws.amazon.com/organizations/latest/userguide/orgs_manage_policies_about-scps.html>

**Domain**

AWS Security, Identity, & Compliance

**Question 46Correct**

A company is migrating a stateful web service into the AWS cloud. The objective is to refactor the application to realize the benefits of cloud computing. How can the Developer leading the project refactor the application to enable more elasticity? (Select TWO.)

**Your selection is correct**

**Store the session state in an Amazon DynamoDB table**

**Use Amazon CloudFormation and the Serverless Application Model**

**Use Amazon CloudFront with a Web Application Firewall**

**Store the session state in an Amazon RDS database**

**Your selection is correct**

**Use an Elastic Load Balancer and Auto Scaling Group**

Overall explanation

As this is a stateful application the session data needs to be stored somewhere. Amazon DynamoDB is designed to be used for storing session data and it highly scalable. To add elasticity to the architecture an Amazon Elastic Load Balancer (ELB) and Amazon EC2 Auto Scaling group (ASG) can be used.

With this architecture the web service can scale elastically using the ASG and the ELB will distribute traffic to all new instances that the ASG launches. This is a good example of utilizing some of the key benefits of refactoring applications into the AWS cloud.

**CORRECT:**"Use an Elastic Load Balancer and Auto Scaling Group" is a correct answer.

**CORRECT:**"Store the session state in an Amazon DynamoDB table" is also a correct answer.

**INCORRECT:** "Use Amazon CloudFormation and the Serverless Application Model" is incorrect. AWS SAM is used in CloudFormation templates for expressing serverless applications using a simplified syntax. This application is not a serverless application.

**INCORRECT:** "Use Amazon CloudFront with a Web Application Firewall" is incorrect neither protection from web exploits nor improved performance for content delivery are requirements in this scenario.

**INCORRECT:** "Store the session state in an Amazon RDS database" is incorrect as RDS is not suitable for storing session state data. DynamoDB is a better fit for this use case.

**References:**

<https://docs.aws.amazon.com/aws-sdk-php/v2/guide/feature-dynamodb-session-handler.html>

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

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

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

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

**Domain**

AWS Compute

**Question 47Correct**

An application uses AWS Lambda to process many files. The Lambda function takes approximately 3 minutes to process each file and does not return any important data. A Developer has written a script that will invoke the function using the AWS CLI.

What is the FASTEST way to process all the files?

**Invoke the Lambda function asynchronously with the invocation type RequestResponse and process the files sequentially**

**Your answer is correct**

**Invoke the Lambda function asynchronously with the invocation type Event and process the files in parallel**

**Invoke the Lambda function synchronously with the invocation type Event and process the files in parallel**

**Invoke the Lambda function synchronously with the invocation type RequestResponse and process the files sequentially**

Overall explanation

You can invoke Lambda functions directly with the Lambda console, the Lambda API, the AWS SDK, the AWS CLI, and AWS toolkits.

You can also configure other AWS services to invoke your function, or you can configure Lambda to read from a stream or queue and invoke your function.

When you invoke a function, you can choose to invoke it synchronously or asynchronously.

• Synchronous invocation:

o You wait for the function to process the event and return a response.

o To invoke a function synchronously with the AWS CLI, use the invoke command.

o The Invocation-type can be used to specify a value of “RequestResponse”. This instructs AWS to execute your Lambda function and wait for the function to complete.

• Asynchronous invocation:

o When you invoke a function asynchronously, you don’t wait for a response from the function code.

o For asynchronous invocation, Lambda handles retries and can send invocation records to a destination.

o To invoke a function asynchronously, set the invocation type parameter to Event.

The fastest way to process all the files is to use asynchronous invocation and process the files in parallel. To do this you should specify the invocation type of Event

**CORRECT:**"Invoke the Lambda function asynchronously with the invocation type Event and process the files in parallel" is the correct answer.

**INCORRECT:** "Invoke the Lambda function synchronously with the invocation type Event and process the files in parallel" is incorrect as the invocation type for a synchronous invocation should be RequestResponse.

**INCORRECT:** "Invoke the Lambda function synchronously with the invocation type RequestResponse and process the files sequentially" is incorrect as this is not the fastest way of processing the files as Lambda will wait for completion of once file before moving on to the next one.

**INCORRECT:** "Invoke the Lambda function asynchronously with the invocation type RequestResponse and process the files sequentially" is incorrect as the invocation type RequestResponse is used for synchronous invocations.

**References:**

<https://aws.amazon.com/blogs/architecture/understanding-the-different-ways-to-invoke-lambda-functions/>

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

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

**Domain**

AWS Compute

**Question 48Correct**

A company is in the process of migrating an application from a monolithic architecture to a microservices-based architecture. The developers need to refactor the application so that the many microservices can asynchronously communicate with each other in a decoupled manner.

Which AWS services can be used for asynchronous message passing? (Select TWO.)

**AWS Lambda**

**Your selection is correct**

**Amazon SNS**

**Amazon ECS**

**Your selection is correct**

**Amazon SQS**

**Amazon Kinesis**

Overall explanation

Amazon Simple Queue Service (SQS) is a fully managed message queuing service that enables you to decouple and scale microservices, distributed systems, and serverless applications.

Amazon Simple Notification Service (SNS) is a highly available, durable, secure, fully managed pub/sub messaging service that enables you to decouple microservices, distributed systems, and serverless applications.

These services both enable asynchronous message passing in the form of a message bus (SQS) and notifications (SNS).

**CORRECT:**"Amazon SQS" is the correct answer.

**CORRECT:**"Amazon SNS" is also a correct answer.

**INCORRECT:** "Amazon Kinesis" is incorrect. Kinesis is used for streaming data, it is used for real-time analytics, mobile data capture and IoT and similar use cases.

**INCORRECT:** "Amazon ECS" is incorrect. ECS is a service providing Docker containers on Amazon EC2.

**INCORRECT:** "AWS Lambda" is incorrect. AWS Lambda is a compute service that runs functions in response to triggers.

**References:**

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

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

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

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

**Domain**

AWS Application Integration

**Question 49Correct**

A three tier web application has been deployed on Amazon EC2 instances using Amazon EC2 Auto Scaling. The EC2 instances in the web tier sometimes receive bursts of traffic and the application tier cannot scale fast enough to keep up with messages sometimes resulting in message loss.

How can a Developer decouple the application to prevent loss of messages?

**Migrate the database tier to Amazon DynamoDB and enable scalable session handling**

**Your answer is correct**

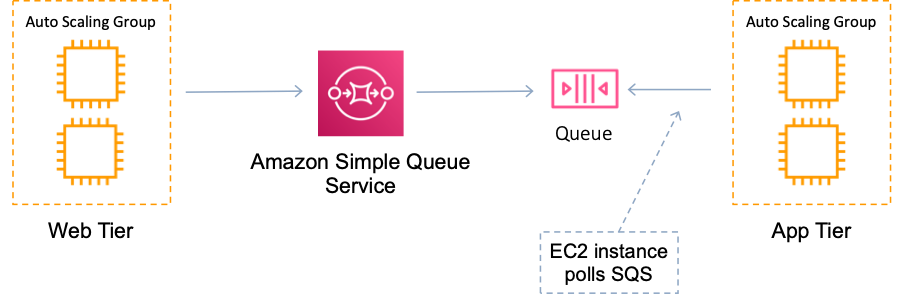
**Add an Amazon SQS queue between the web tier and the application tier**

**Configure the web tier to publish messages to an SNS topic and subscribe the application tier to the SNS topic**

**Add an Amazon SQS queue between the application tier and the database tier**

Overall explanation

Amazon SQS queues messages received from one application component ready for consumption by another component. A queue is a temporary repository for messages that are awaiting processing. The queue acts as a buffer between the component producing and saving data, and the component receiving the data for processing.



With this scenario the best choice for the Developer is to implement an Amazon SQS queue between the web tier and the application tier. This will mean when the web tier receives bursts of traffic the messages will not overburden the application tier. Instead, they will be placed in the queue and can be processed by the app tier.

**CORRECT:**"Add an Amazon SQS queue between the web tier and the application tier" is the correct answer.

**INCORRECT:** "Add an Amazon SQS queue between the application tier and the database tier" is incorrect as the burst of messages are being received by the web tier and it is the application tier that is having difficulty keeping up with demand.

**INCORRECT:** "Configure the web tier to publish messages to an SNS topic and subscribe the application tier to the SNS topic" is incorrect as SNS is used for notifications and those notifications are not queued, they are sent to all subscribers. The messages being passed in this scenario are better suited to being placed in a queue.

**INCORRECT:** "Migrate the database tier to Amazon DynamoDB and enable scalable session handling" is incorrect as this is of no relevance to the situation. We don’t know what type of database is being used and there is not stated issue with the database layer.

**References:**

<https://docs.aws.amazon.com/AWSSimpleQueueService/latest/SQSDeveloperGuide/welcome.html>

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

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

**Domain**

AWS Application Integration

**Question 50Correct**

A Developer is using AWS SAM to create a template for deploying a serverless application. The Developer plans deploy a Lambda function using the template.

Which resource type should the Developer specify?

**AWS::Serverless:API**

**AWS::Serverless:LayerVersion**

**Your answer is correct**

**AWS::Serverless:Function**

**AWS::Serverless::Application**

Overall explanation

A **serverless application** is a combination of Lambda functions, event sources, and other resources that work together to perform tasks. Note that a serverless application is more than just a Lambda function—it can include additional resources such as APIs, databases, and event source mappings.

AWS SAM templates are an extension of AWS CloudFormation templates, with some additional components that make them easier to work with. To create a Lambda function using an AWS SAM template the Developer can use the AWS::Serverless::Function resource type.

The AWS::Serverless::Function resource type can be used to Create a Lambda function, IAM execution role, and event source mappings that trigger the function.

**CORRECT:**"AWS::Serverless:Function" is the correct answer.

**INCORRECT:** "AWS::Serverless::Application" is incorrect as this embeds a serverless application from the [AWS Serverless Application Repository](https://serverlessrepo.aws.amazon.com/applications) or from an Amazon S3 bucket as a nested application.

**INCORRECT:** "AWS::Serverless:LayerVersion" is incorrect as this creates a Lambda LayerVersion that contains library or runtime code needed by a Lambda Function.

**INCORRECT:** "AWS::Serverless:API" is incorrect as this creates a collection of Amazon API Gateway resources and methods that can be invoked through HTTPS endpoints.

**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 51Incorrect**

An AWS Lambda function requires several environment variables with secret values. The secret values should be obscured in the Lambda console and API output even for users who have permission to use the key.

What is the best way to achieve this outcome and MINIMIZE complexity and latency?

**Use an external encryption infrastructure to encrypt the values and add them as environment variables**

**Correct answer**

**Encrypt the secret values client-side using encryption helpers**

**Your answer is incorrect**

**Encrypt the secret values with a customer-managed CMK**

**Store the encrypted values in an encrypted Amazon S3 bucket and reference them from within the code**

Overall explanation

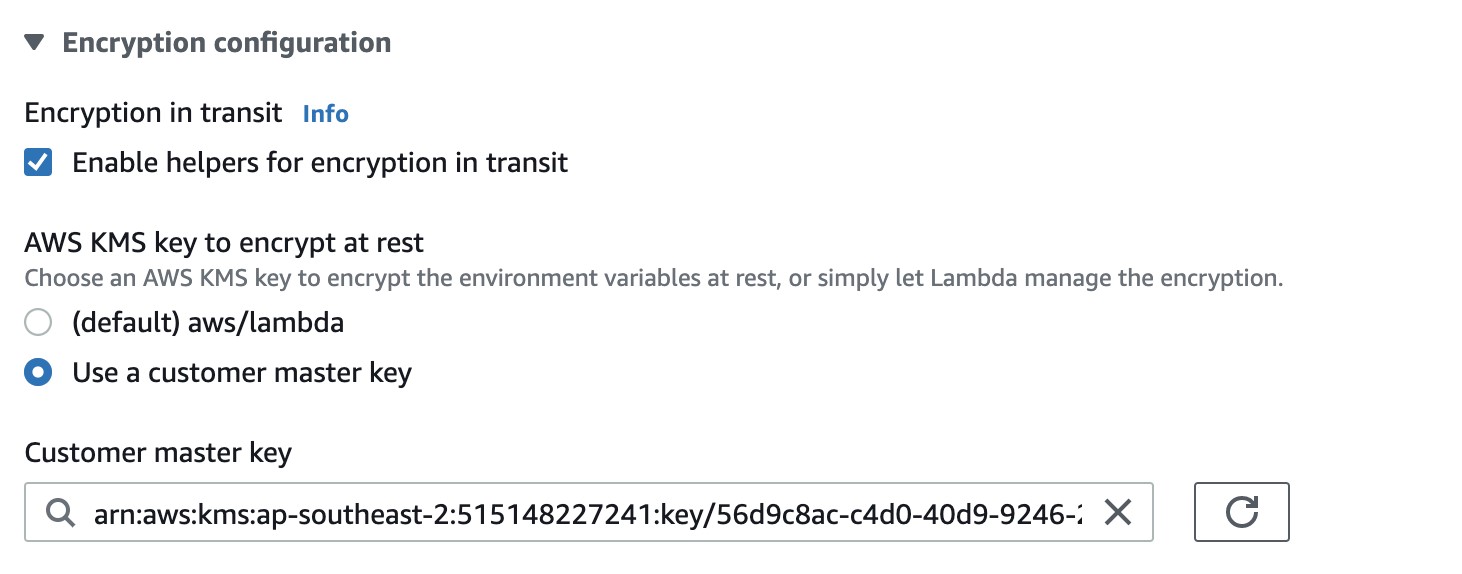
You can use environment variables to store secrets securely for use with Lambda functions. Lambda always encrypts environment variables at rest.

Additionally, you can use the following features to customize how environment variables are encrypted.

• **Key configuration** – On a per-function basis, you can configure Lambda to use an encryption key that you create and manage in AWS Key Management Service. These are referred to as customer managed customer master keys (CMKs) or customer managed keys. If you don't configure a customer managed key, Lambda uses an AWS managed CMK named aws/lambda, which Lambda creates in your account.

• **Encryption helpers** – The Lambda console lets you encrypt environment variable values client side, before sending them to Lambda. This enhances security further by preventing secrets from being displayed unencrypted in the Lambda console, or in function configuration that's returned by the Lambda API. The console also provides sample code that you can adapt to decrypt the values in your function handler.

The configuration for using encryption helps to encrypt data client-side looks like this:



This is the best way to achieve this outcome and minimizes complexity as the encryption infrastructure will still use AWS KMS and be able to decrypt the values during function execution.

**CORRECT:**"Encrypt the secret values client-side using encryption helpers" is the correct answer.

**INCORRECT:** "Encrypt the secret values with a customer-managed CMK" is incorrect as this alone will not achieve the desired outcome as the environment variables should be encrypted client-side with the encryption helper to ensure users cannot see the secret values.

**INCORRECT:** "Store the encrypted values in an encrypted Amazon S3 bucket and reference them from within the code" is incorrect as this would introduce complexity and latency.

**INCORRECT:** "Use an external encryption infrastructure to encrypt the values and add them as environment variables" is incorrect as this would introduce complexity and latency.

**References:**

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

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

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

**Domain**

AWS Compute

**Question 52Correct**

A developer is debugging an application by sifting through log data stored in Amazon CloudWatch Logs. A fresh metric filter has been established to identify exceptions in these logs. Yet, the logs are not returning any results filtered through the new metric.

What could be the reason behind the absence of filtered results?

**Your answer is correct**

**CloudWatch Logs only publish metric data for events that occur after the filter has been established.**

**The application logs have been archived to Amazon S3, making them non-filterable.**

**The time range for the filter hasn't been properly configured.**

**The CloudWatch Logs agent hasn't been installed on the EC2 instances.**

Overall explanation

Amazon CloudWatch Logs starts to publish metric data points only for log events that are ingested after the metric filter is created. This means the filter does not apply retrospectively to log data ingested prior to its creation.

**CORRECT:**"CloudWatch Logs only publish metric data for events that occur after the filter has been established" is the correct answer (as explained above.)

**INCORRECT:** "The CloudWatch Logs agent hasn't been installed on the EC2 instances" is incorrect.

The CloudWatch Logs agent is responsible for sending logs to CloudWatch and does not need to be installed for filtering to work.

**INCORRECT:** "The application logs have been archived to Amazon S3, making them non-filterable" is incorrect.

CloudWatch Logs can still filter logs even if they have been archived to Amazon S3.

**INCORRECT:** "The time range for the filter hasn't been properly configured" is incorrect.

The time range for a metric filter is not relevant as it operates on new logs, regardless of their timestamp.

**References:**

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

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

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

**Domain**

AWS Management & Governance

**Question 53Correct**

A company is running an application built on AWS Lambda functions. One Lambda function has performance issues when it has to download a 50 MB file from the internet every execution. This function is called multiple times a second.

What solution would give the BEST performance increase?

**Increase the Lambda maximum execution time**

**Put an Elastic Load Balancer in front of the Lambda function**

**Cache the file in Amazon S3**

**Your answer is correct**

**Cache the file in the /tmp directory**

Overall explanation

The /tmp directory provides 512 MB of storage space that can be used by a function. When a file is cached by a function in the /tmp directory it is available to be used by subsequent executions of the function which will reduce latency.

**CORRECT:**"Cache the file in the /tmp directory" is the correct answer.

**INCORRECT:** "Increase the Lambda maximum execution time" is incorrect as the function is not timing out.

**INCORRECT:** "Put an Elastic Load Balancer in front of the Lambda function" is incorrect as this would not reduce latency or improve performance.

**INCORRECT:** "Cache the file in Amazon S3" is incorrect as this would not provide better performance as it would still need to be retrieved from S3 for each execution if it is not cached in the /tmp directory.

**References:**

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

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

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

**Domain**

AWS Compute

**Question 54Incorrect**

A developer needs to implement a caching layer in front of an Amazon RDS database. If the caching layer fails, it is time consuming to repopulate cached data so the solution should be designed for maximum uptime. Which solution is best for this scenario?

**Your answer is incorrect**

**Implement Amazon ElastiCache Memcached**

**Migrate the database to Amazon RedShift**

**Implement Amazon DynamoDB DAX**

**Correct answer**

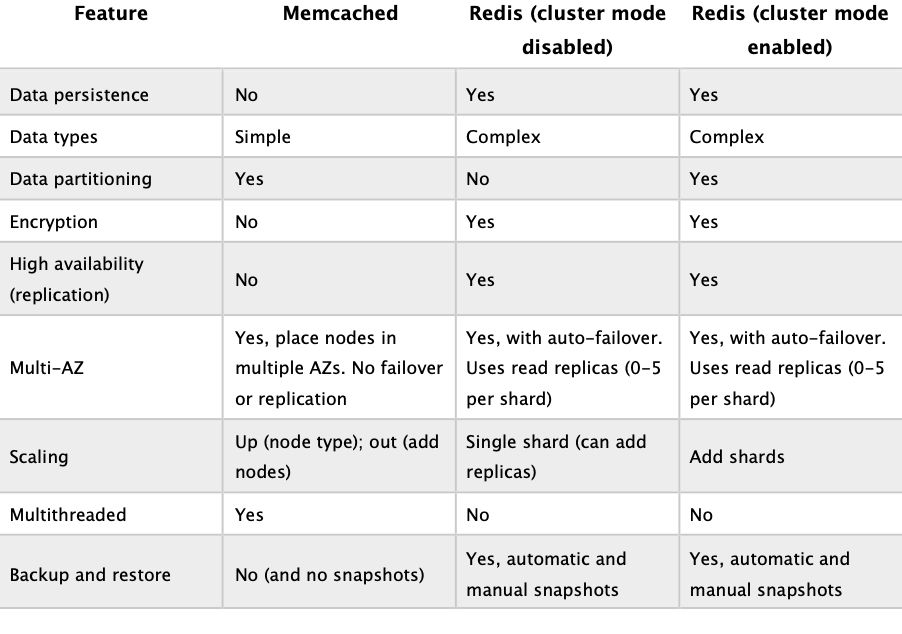
**Implement Amazon ElastiCache Redis**

Overall explanation

Amazon ElastiCache provides fully managed implementations of two popular in-memory data stores – Redis and Memcached. ElastiCache is a web service that makes it easy to deploy and run Memcached or Redis protocol-compliant server nodes in the cloud.

The in-memory caching provided by ElastiCache can be used to significantly improve latency and throughput for many read-heavy application workloads or compute-intensive workloads. It is common to use ElastiCache as a cache in front of databases such as Amazon RDS.

The two implementations, Memcached, and Redis, each offer different capabilities and limitations. As you can see from the table below, only Redis supports read replicas and auto-failover:



The Redis implementation must be used if high availability is required, as is necessary for this scenario. Therefore the correct answer is to use Amazon ElastiCache Redis.

**CORRECT:**"Implement Amazon ElastiCache Redis" is the correct answer.

**INCORRECT:** "Implement Amazon ElastiCache Memcached" is incorrect as Memcached does not offer read replicas or auto-failover and therefore cannot provide high availability.

**INCORRECT:** "Migrate the database to Amazon RedShift" is incorrect as RedShift is a data warehouse for use in online analytics processing (OLAP) use cases. It is not suitable to be used as a caching layer.

**INCORRECT:** "Implement Amazon DynamoDB DAX" is incorrect as DAX is used in front of DynamoDB, not Amazon RDS.

**References:**

<https://aws.amazon.com/elasticache/redis-vs-memcached/>

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

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

**Domain**

AWS Database

**Question 55Correct**

A Developer is writing a web application that allows users to view images from an Amazon S3 bucket. The users will log in with their Amazon login, as well as Facebook and/or Google accounts.

How can the Developer provide this authentication capability?

**Your answer is correct**

**Use Amazon Cognito with web identity federation**

**Use AWS IAM Access/Secret keys in the application code to allow Get\* on the S3 bucket**

**Use Amazon Cognito with SAML-based identity federation**

**Use AWS STS AssumeRole in the application code and assume a role with Get\* permissions on the S3 bucket**

Overall explanation

**Explanation:**

Amazon Cognito identity pools (federated identities) enable you to create unique identities for your users and federate them with identity providers. With an identity pool, you can obtain temporary, limited-privilege AWS credentials to access other AWS services. Amazon Cognito identity pools support the following identity providers:

• Public providers: [Login with Amazon (Identity Pools)](https://docs.aws.amazon.com/cognito/latest/developerguide/amazon.html), [Facebook (Identity Pools)](https://docs.aws.amazon.com/cognito/latest/developerguide/facebook.html), [Google (Identity Pools)](https://docs.aws.amazon.com/cognito/latest/developerguide/google.html) [Sign in with Apple (Identity Pools)](https://docs.aws.amazon.com/cognito/latest/developerguide/apple.html).

• [Amazon Cognito User Pools](https://docs.aws.amazon.com/cognito/latest/developerguide/cognito-user-identity-pools.html)

• [Open ID Connect Providers (Identity Pools)](https://docs.aws.amazon.com/cognito/latest/developerguide/open-id.html)

• [SAML Identity Providers (Identity Pools)](https://docs.aws.amazon.com/cognito/latest/developerguide/saml-identity-provider.html)

• [Developer Authenticated Identities (Identity Pools)](https://docs.aws.amazon.com/cognito/latest/developerguide/developer-authenticated-identities.html)

With the temporary, limited-privilege AWS credentials users will be able to access the images in the S3 bucket. Therefore, the Developer should use Amazon Cognito with web identity federation

**CORRECT:**"Use Amazon Cognito with web identity federation" is the correct answer.

**INCORRECT:** "Use Amazon Cognito with SAML-based identity federation" is incorrect as SAML is used with directory sources such as Microsoft Active Directory, not Facebook or Google.

**INCORRECT:** "Use AWS IAM Access/Secret keys in the application code to allow Get\* on the S3 bucket" is incorrect as this insecure and against best practice. Always try to avoid embedding access keys in application code.

**INCORRECT:** "Use AWS STS AssumeRole in the application code and assume a role with Get\* permissions on the S3 bucket" is incorrect as you cannot do this directly through a Facebook or Google login. For this scenario, a Cognito Identity Pool is required to authenticate the user from the social IdP and provide access to the AWS services.

**References:**

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

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

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

**Domain**

AWS Security, Identity, & Compliance

**Question 56Correct**

A web application is using Amazon Kinesis Data Streams for ingesting IoT data that is then stored before processing for up to 24 hours.  
How can the Developer implement encryption at rest for data stored in Amazon Kinesis Data Streams?

**Encrypt the data once it is at rest with an AWS Lambda function**

**Add a certificate and enable SSL/TLS connections to Kinesis Data Streams**

**Your answer is correct**

**Enable server-side encryption on Kinesis Data Streams with an AWS KMS CMK**

**Use the Amazon Kinesis Consumer Library (KCL) to encrypt the data**

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.

Server-side encryption is a feature in Amazon Kinesis Data Streams that automatically encrypts data before it's at rest by using an AWS KMS customer master key (CMK) you specify. Data is encrypted before it's written to the Kinesis stream storage layer and decrypted after it’s retrieved from storage. As a result, your data is encrypted at rest within the Kinesis Data Streams service. This allows you to meet strict regulatory requirements and enhance the security of your data.

With server-side encryption, your Kinesis stream producers and consumers don't need to manage master keys or cryptographic operations. Your data is automatically encrypted as it enters and leaves the Kinesis Data Streams service, so your data at rest is encrypted. AWS KMS provides all the master keys that are used by the server-side encryption feature. AWS KMS makes it easy to use a CMK for Kinesis that is managed by AWS, a user-specified AWS KMS CMK, or a master key imported into the AWS KMS service.

Therefore, in this scenario the Developer can enable server-side encryption on Kinesis Data Streams with an AWS KMS CMK

**CORRECT:**"Enable server-side encryption on Kinesis Data Streams with an AWS KMS CMK" is the correct answer.

**INCORRECT:** "Add a certificate and enable SSL/TLS connections to Kinesis Data Streams" is incorrect as SSL/TLS is already used with Kinesis (you don’t need to add a certificate) and this only provides encryption in-transit, not encryption at rest.

**INCORRECT:** "Use the Amazon Kinesis Consumer Library (KCL) to encrypt the data" is incorrect. The KCL provides design patterns and code for Amazon Kinesis Data Streams consumer applications. The KCL is not used for adding encryption to the data in a stream.

**INCORRECT:** "Encrypt the data once it is at rest with an AWS Lambda function" is incorrect as this is unnecessary when Kinesis natively supports server-side encryption.

**References:**

<https://docs.aws.amazon.com/streams/latest/dev/what-is-sse.html>

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

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

**Domain**

AWS Application Integration

**Question 57Correct**

An organization has a new AWS account and is setting up IAM users and policies. According to AWS best practices, which of the following strategies should be followed? (Select TWO.)

**Create user accounts that can be shared for efficiency**

**Your selection is correct**

**Create standalone policies instead of using inline policies**

**Your selection is correct**

**Use groups to assign permissions to users**

**Use user accounts to delegate permissions**

**Always use customer managed policies instead of AWS managed policies**

Overall explanation

AWS provide a number of best practices for AWS IAM that help you to secure your resources. The key best practices referenced in this scenario are as follows:

• Use groups to assign permissions to users – this is correct as you should create permissions policies and assign them to groups. Users can be added to the groups to get the permissions they need to perform their jobs.

• Create standalone policies instead of using inline policies ([Use Customer Managed Policies Instead of Inline Policies](https://docs.aws.amazon.com/IAM/latest/UserGuide/best-practices.html#best-practice-managed-vs-inline) in the AWS best practices) – this refers to creating your own policies that are standalone policies which can be reused multiple times (assigned to multiple entities such as groups, and users). This is better than using inline policies which are directly attached to a single entity.

**CORRECT:**"Use groups to assign permissions to users" is the correct answer.

**CORRECT:**"Create standalone policies instead of using inline policies" is the correct answer.

**INCORRECT:** "Use user accounts to delegate permissions" is incorrect as you should use roles to delegate permissions.

**INCORRECT:** "Create user accounts that can be shared for efficiency" is incorrect as you should not share user accounts. Always create individual user accounts.

**INCORRECT:** "Always use customer managed policies instead of AWS managed policies" is incorrect as this is not a best practice. AWS recommend getting started by using AWS managed policies ([Get Started Using Permissions with AWS Managed Policies](https://docs.aws.amazon.com/IAM/latest/UserGuide/best-practices.html#bp-use-aws-defined-policies)).

**References:**

<https://docs.aws.amazon.com/IAM/latest/UserGuide/best-practices.html>

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

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

**Domain**

AWS Security, Identity, & Compliance

**Question 58Correct**

A company is migrating several applications to the AWS cloud. The security team has strict security requirements and mandate that a log of all API calls to AWS resources must be maintained.

Which AWS service should be used to record this information for the security team?

**AWS X-Ray**

**Amazon CloudWatch Logs**

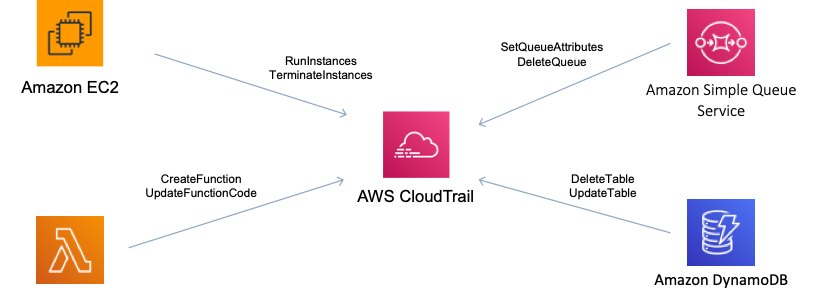
**Your answer is correct**

**AWS CloudTrail**

**Amazon CloudWatch**

Overall explanation

AWS CloudTrail is a web service that records activity made on your account. A CloudTrail trail can be created which delivers log files to an Amazon S3 bucket. CloudTrail is about logging and saves a history of API calls for your AWS account. It enables governance, compliance, and operational and risk auditing of your AWS account.



Therefore, AWS CloudTrail is the best solution for maintaining a log of API calls for the security team.

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

**INCORRECT:** "Amazon CloudWatch" is incorrect as this service records metrics related to performance.

**INCORRECT:** "Amazon CloudWatch Logs" is incorrect as this records log files from services and applications, it does not record a history of API activity.

**INCORRECT:** "AWS X-Ray" is incorrect as this is used for tracing applications to view performance-related statistics.

**References:**

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

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

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

**Domain**

AWS Management & Governance

**Question 59Correct**

A developer is in the process of revising multiple AWS Lambda functions and notes that these functions utilize the same bespoke libraries. The developer intends to centralize these libraries, implement updates with minimal effort, and keep the libraries version controlled.

Which solution aligns with these needs while requiring the least development effort?

**Create a custom Amazon Machine Image (AMI) that includes the custom libraries.**

**Create an AWS CodeCommit repository for storing the custom libraries.**

**Your answer is correct**

**Create a Lambda layer including all the custom libraries.**

**Create an Amazon S3 bucket to store all the custom libraries.**

Overall explanation

Lambda layers allow the sharing of code, libraries, or other resources across multiple Lambda functions, making it easier to manage common code across multiple functions. This enables a developer to maintain versioned libraries that can be easily updated and shared, thus requiring the least development effort.

**CORRECT:**"Create a Lambda layer including all the custom libraries" is the correct answer (as explained above.)

**INCORRECT:** "Create an AWS CodeCommit repository for storing the custom libraries" is incorrect.

AWS CodeCommit is a version control service hosted by AWS; it is not designed to serve as a central library for Lambda functions.

**INCORRECT:** "Create a custom Amazon Machine Image (AMI) that includes the custom libraries" is incorrect.

Creating a custom Amazon Machine Image (AMI) is not related to Lambda functions as Lambda runs without requiring a server setup.

**INCORRECT:** "Create an Amazon S3 bucket to store all the custom libraries" is incorrect.

Amazon S3 buckets are primarily used for storage and not optimal for serving as a centralized library for Lambda functions.

**References:**

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

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

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

**Domain**

AWS Storage

**Question 60Correct**

A Development team is creating a microservices application running on Amazon ECS. The release process workflow of the application requires a manual approval step before the code is deployed into the production environment.  
What is the BEST way to achieve this using AWS CodePipeline?

**Disable a stage just prior the deployment stage**

**Your answer is correct**

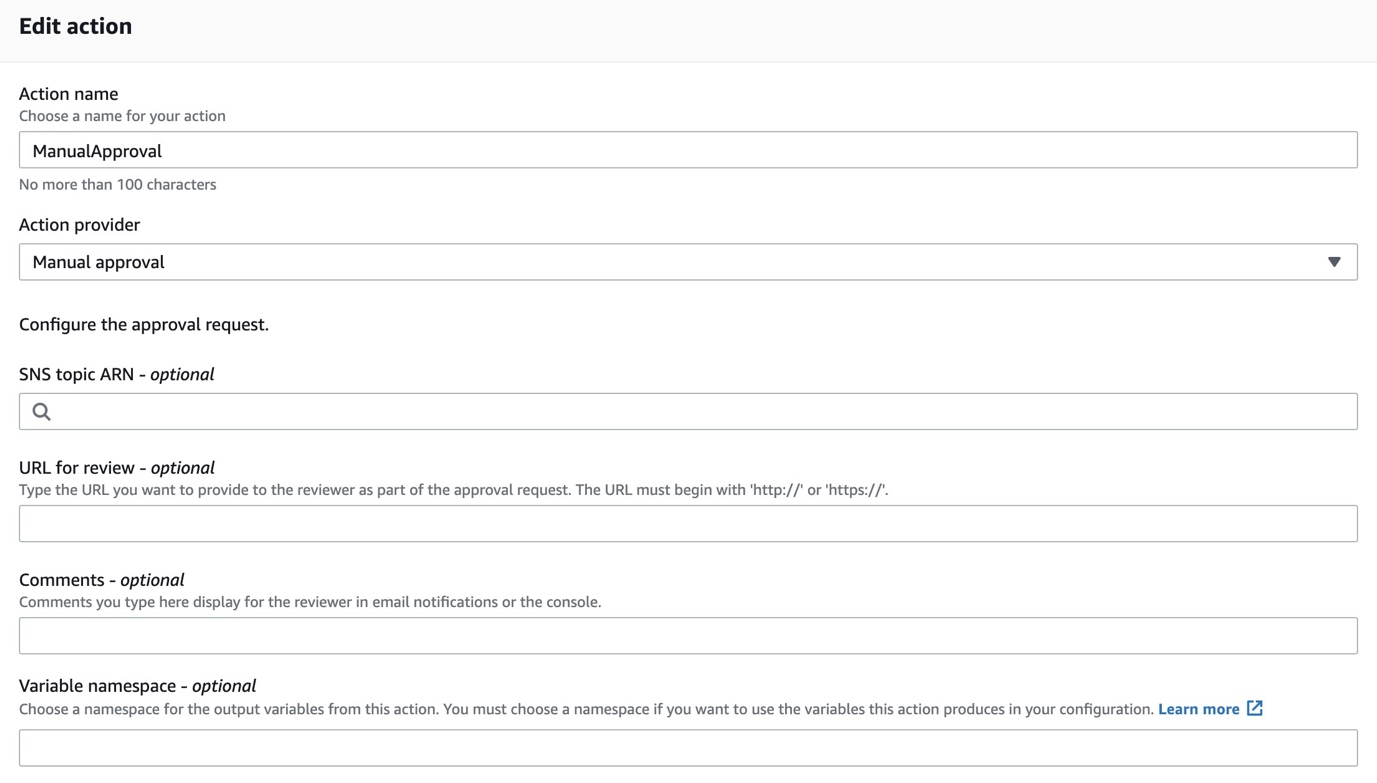
**Use an approval action in a stage before deployment**

**Disable the stage transition to allow manual approval**

**Use an Amazon SNS notification from the deployment stage**

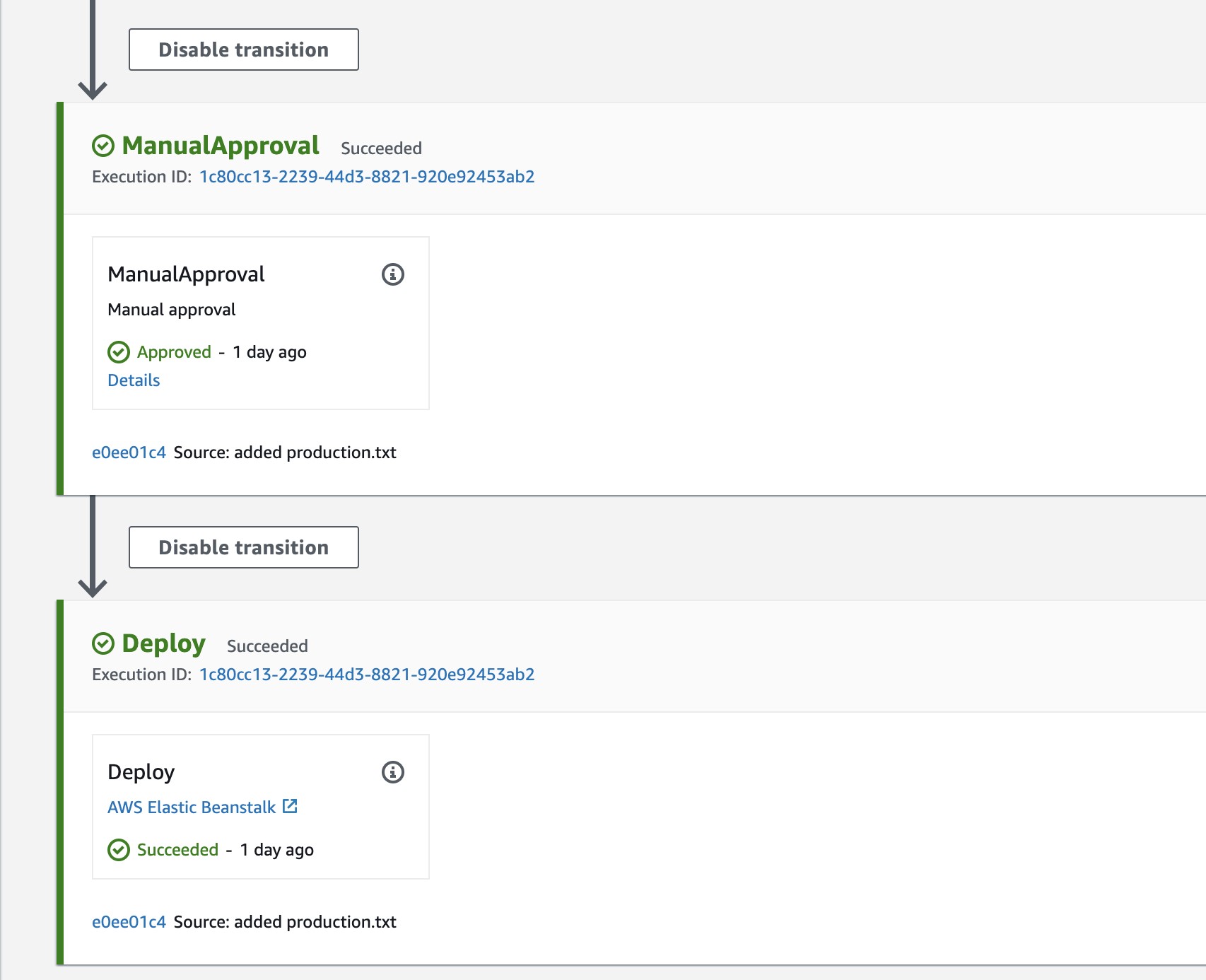
Overall explanation

In AWS CodePipeline, you can add an approval action to a stage in a pipeline at the point where you want the pipeline execution to stop so that someone with the required AWS Identity and Access Management permissions can approve or reject the action.



If the action is approved, the pipeline execution resumes. If the action is rejected—or if no one approves or rejects the action within seven days of the pipeline reaching the action and stopping—the result is the same as an action failing, and the pipeline execution does not continue.

In this scenario, the manual approval stage would be placed in the pipeline before the deployment stage that deploys the application update into production:



Therefore, the best answer is to use an approval action in a stage before deployment to production

**CORRECT:**"Use an approval action in a stage before deployment" is the correct answer.

**INCORRECT:** "Use an Amazon SNS notification from the deployment stage" is incorrect as this would send a notification when the actual deployment is already occurring.

**INCORRECT:** "Disable the stage transition to allow manual approval" is incorrect as this requires manual intervention as could be easily missed and allow the deployment to continue.

**INCORRECT:** "Disable a stage just prior the deployment stage" is incorrect as disabling the stage prior would prevent that stage from running, which may be necessary (could be the build / test stage). It is better to use an approval action in a stage in the pipeline before the deployment occurs

**References:**

<https://docs.aws.amazon.com/codepipeline/latest/userguide/approvals.html>

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

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

**Domain**

AWS Developer Tools

**Question 61Correct**

An AWS developer is building an application that processes sensitive personally identifiable information (PII). The application operates on AWS Lambda and writes diagnostic data to Amazon CloudWatch. However, the developer wants to ensure that PII is not accidentally logged in CloudWatch.

What strategy should the developer adopt to ensure this?

**Incorporate AWS X-Ray and configure it to filter out sensitive PII before logging.**

**Manually insert logging commands into the application code while ensuring PII is not included.**

**Implement AWS Secrets Manager for secure logging of sensitive information.**

**Your answer is correct**

**Use Amazon Macie to regularly scan and identify any PII within the logs.**

Overall explanation

Amazon Macie is a fully managed data privacy and security service that uses machine learning and pattern matching to discover and protect sensitive data in AWS, such as PII. This includes being able to scan CloudWatch logs for PII, making it a good fit for this requirement.

**CORRECT:**"Use Amazon Macie to regularly scan and identify any PII within the logs" is the correct answer (as explained above.)

**INCORRECT:** "Manually insert logging commands into the application code while ensuring PII is not included" is incorrect.

Manually incorporating logging commands could prevent PII from being logged but lacks an automated mechanism for detecting or preventing accidental inclusion of PII.

**INCORRECT:** "Implement AWS Secrets Manager for secure logging of sensitive information" is incorrect.

AWS Secrets Manager is used for managing secrets and does not provide direct capabilities to prevent PII from being logged.

**INCORRECT:** "Incorporate AWS X-Ray and configure it to filter out sensitive PII before logging" is incorrect.

AWS X-Ray provides insights into the behavior of your applications, but it does not inherently include features to detect or filter out PII before logging.

**References:**

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

**Domain**

AWS Security, Identity, & Compliance

**Question 62Correct**

A corporation plans to deploy an application on AWS utilizing an Elastic Load Balancer that operates with HTTP/HTTPS listeners. The application must have the ability to retrieve client IP addresses.

Which load-balancing solution would satisfy these needs?

**Network Load Balancer with Proxy Protocol enabled.**

**Your answer is correct**

**Application Load Balancer with X-Forwarded-For headers enabled.**

**Application Load Balancer with Proxy Protocol enabled.**

**Gateway Load Balancer with X-Forwarded-For headers enabled.**

Overall explanation

Application Load Balancer operates at the application layer (Layer 7), and it supports path-based routing, and it can route requests to one or more ports on each container instance in your cluster. The X-Forwarded-For request header helps to preserve the client-side source IP address which is needed in this scenario.

**CORRECT:**"Application Load Balancer with X-Forwarded-For headers enabled" is the correct answer (as explained above.)

**INCORRECT:** "Network Load Balancer with Proxy Protocol enabled" is incorrect.

Network Load Balancer operates at the transport layer (Layer 4) and handles millions of requests per second. The NLB cannot use HTTP/HTTPS listeners so is not suitable for this solution.

**INCORRECT:** "Gateway Load Balancer with X-Forwarded-For headers enabled" is incorrect.

A gateway load balancer is used for distributing traffic to appliances such as IDS and IPS devices.

**INCORRECT:** "Application Load Balancer with Proxy Protocol enabled" is incorrect.

You cannot use the proxy protocol with a layer 7 load balancer.

**References:**

<https://docs.aws.amazon.com/elasticloadbalancing/latest/userguide/load-balancer-getting-started.html>

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

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

**Domain**

AWS Compute

**Question 63Correct**

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

1. $ aws lambda invoke --function-name MyFunction --invocation-type Event --payload ewogICJrZXkxIjogInZhbHVlMSIsCiAgImtleTIiOiAidmFsdWUyIiwKICAia2V5MyI6ICJ2YWx1ZTMiCn0= response.json
2. {
3. "StatusCode": 202
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 asynchronously and has completed successfully**

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

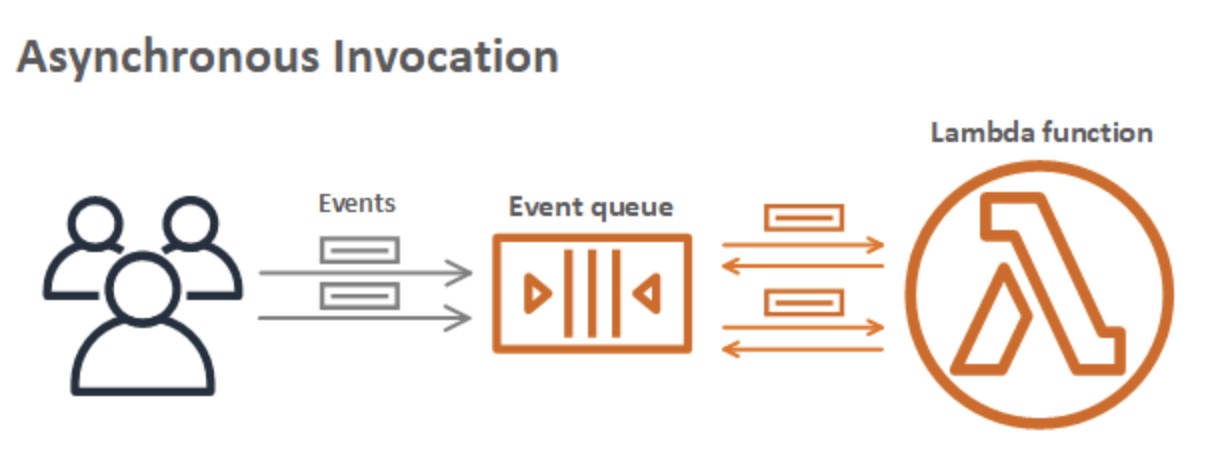
**An AWS Lambda function has been invoked synchronously and has completed successfully**

Overall explanation

Several AWS services, such as Amazon Simple Storage Service (Amazon S3) and Amazon Simple Notification Service (Amazon SNS), invoke functions asynchronously to process events.

When you invoke a function asynchronously, you don't wait for a response from the function code. You hand off the event to Lambda and Lambda handles the rest. You can configure how Lambda handles errors, and can send invocation records to a downstream resource to chain together components of your application.

The following diagram shows clients invoking a Lambda function asynchronously. Lambda queues the events before sending them to the function.



For asynchronous invocation, Lambda places the event in a queue and returns a success response without additional information. A separate process reads events from the queue and sends them to your function. To invoke a function asynchronously, set the invocation type parameter to Event.

In this scenario the Event parameter has been used so we know the function has been invoked asynchronously. For asynchronous invocation the status code 202 indicates a successful execution.

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

**INCORRECT:** "An AWS Lambda function has been invoked synchronously and has completed successfully" is incorrect as the Event parameter indicates an asynchronous invocation.

**INCORRECT:** "An AWS Lambda function has been invoked synchronously and has not completed successfully" is incorrect as the Event parameter indicates an asynchronous invocation (a status code 200 would be a successful execution for a synchronous invocation).

**INCORRECT:** "An AWS Lambda function has been invoked asynchronously and has not completed successfully" is incorrect as the status code 202 indicates a successful execution.

**References:**

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

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

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

**Domain**

AWS Compute

**Question 64Correct**

A mobile application has hundreds of users. Each user may use multiple devices to access the application. The Developer wants to assign unique identifiers to these users regardless of the device they use.

Which of the following methods should be used to obtain unique identifiers?

**Your answer is correct**

**Implement developer-authenticated identities by using Amazon Cognito, and get credentials for these identities**

**Assign IAM users and roles to the users. Use the unique IAM resource ID as the unique identifier**

**Create a user table in Amazon DynamoDB as key-value pairs of users and their devices. Use these keys as unique identifiers**

**Use IAM-generated access key IDs for the users as the unique identifier, but do not store secret keys**

Overall explanation

Amazon Cognito supports developer authenticated identities, in addition to web identity federation through [Facebook (Identity Pools)](https://docs.aws.amazon.com/cognito/latest/developerguide/facebook.html), [Google (Identity Pools)](https://docs.aws.amazon.com/cognito/latest/developerguide/google.html), [Login with Amazon (Identity Pools)](https://docs.aws.amazon.com/cognito/latest/developerguide/amazon.html), and Sign in with Apple (identity Pools).

With developer authenticated identities, you can register and authenticate users via your own existing authentication process, while still using Amazon Cognito to synchronize user data and access AWS resources.

Using developer authenticated identities involves interaction between the end user device, your backend for authentication, and Amazon Cognito.

Therefore, the Developer can implement developer-authenticated identities by using Amazon Cognito, and get credentials for these identities.

**CORRECT:**"Implement developer-authenticated identities by using Amazon Cognito, and get credentials for these identities" is the correct answer.

**INCORRECT:** "Create a user table in Amazon DynamoDB as key-value pairs of users and their devices. Use these keys as unique identifiers" is incorrect as this solution would require additional application logic and would be more complex.

**INCORRECT:** "Use IAM-generated access key IDs for the users as the unique identifier, but do not store secret keys" is incorrect as it is not a good practice to provide end users of mobile applications with IAM user accounts and access keys. Cognito is a better solution for this use case.

**INCORRECT:** "Assign IAM users and roles to the users. Use the unique IAM resource ID as the unique identifier" is incorrect. AWS Cognito is better suited to mobile users and with developer authenticated identities the users can be assigned unique identities.

**References:**

<https://docs.aws.amazon.com/cognito/latest/developerguide/developer-authenticated-identities.html>

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

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

**Domain**

AWS Security, Identity, & Compliance

**Question 65Correct**

A Developer is deploying an application using Docker containers on Amazon ECS. One of the containers runs a database and should be placed on instances in the “databases” task group.

What should the Developer use to control the placement of the database task?

**IAM Group**

**Cluster Query Language**

**ECS Container Agent**

**Your answer is correct**

**Task Placement Constraint**

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 task placement constraints can be updated for existing services as well.

Amazon ECS supports the following types of task placement constraints:

distinctInstance

Place each task on a different container instance. This task placement constraint can be specified when either running a task or creating a new service.

memberOf

Place tasks on container instances that satisfy an expression. For more information about the expression syntax for constraints, see [Cluster Query Language](https://docs.aws.amazon.com/AmazonECS/latest/developerguide/cluster-query-language.html).

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 task placement constraint below 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).

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

The Developer should therefore use task placement constraints as in the above example to control the placement of the database task.

**CORRECT:**"Task Placement Constraint" is the correct answer.

**INCORRECT:** "Cluster Query Language" is incorrect. Cluster queries are expressions that enable you to group objects. For example, you can group container instances by attributes such as Availability Zone, instance type, or custom metadata.

**INCORRECT:** "IAM Group" is incorrect as you cannot control task placement on ECS with IAM Groups. IAM groups are used for organizing IAM users and applying policies to them.

**INCORRECT:** "ECS Container Agent" is incorrect. The Amazon ECS container agent allows container instances to connect to your cluster.

**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