AWS LAMBDA - QUIZ

Attempt	1
Marks Obtained	10 / 25
Your score is	40%
Completed on	Saturday , 12 January 2019 , 04:16 PM
Time Taken	00 H 45 M 55 S
Result	Fail

Domains / Topics wise Quiz Performance Report

S.No.	Topic	Total Questions	Correct	Incorrect	Unattempted
1	Other	25	10	15	0

25 Questions	**10** Correct	**15** Incorrect	**0** Unattempted

Show Answers

QUESTION 1 INCORRECT

You are uploading large files to AWS S3 bucket, ranging from 1GB – 3GB. Your organization has a requirement to calculate the hash checksum of the file by reading entire file so the users can validate the checksum to identify any potential corruptions during downloads. For this, you created a Lambda function and getting it triggered through S3 notifications. However, the request is getting timed out. What could be the reason?

- A. Lambda function is configured with minimal memory of 128 MB.
- B. Lambda function is set to run in a private VPC without NAT Gateway or VPC Endpoint. ✔
- C. You have not setup S3 bucket name in the environment variable.
- D. Lambda function is created in a different region than S3 bucket. ✖

Explanation :

Answer: B

Option A is not correct. If the function reaches the maximum configured memory, in this case 128 MB, the function gets terminated with an error message as below, not as **request timed out**.

REPORT RequestId: xxxxxxxx Duration: xxxxx ms Billed Duration: xxxxx ms

Memory Size: 128 MB Max Memory Used: 129 MB RequestId: xxxxxxx Process exited before completing request

AWS Lambda Limits

AWS Lambda Resource Limits per Invocation

Resource	Limits
Memory allocation range	Minimum = 128 MB / Maximum = 3008 MB (with 64 MB increments). If the maximum memory use is exceeded, function invocation will be terminated.

Option B is correct. AWS Lambda functions can run within a private VPC with the resources allocated inside the subnet provided during configuration.

For the lambda function to access S3 service endpoint from within private VPC, there should be a NAT Gateway or S3 VPC Endpoint configured in the route table associated with the subnet which was chosen during Lambda function setup. If not, the request would get timed out.

- https://aws.amazon.com/premiumsupport/knowledge-center/internet-access-lambda-function/ (https://aws.amazon.com/premiumsupport/knowledge-center/internet-access-lambda-function/)

Option C is not correct. Bucket need not be configured as environment variable.

Lambda function environment variables are used to configure additional parameters that can be passed to lambda function.

- https://docs.aws.amazon.com/lambda/latest/dg/env_variables.html (https://docs.aws.amazon.com/lambda/latest/dg/env_variables.html)

Option D is not correct. As long as Lambda function has internet access, it can access S3 service endpoints irrespective of S3 bucket region.

Ask our Experts

QUESTION 2 CORRECT

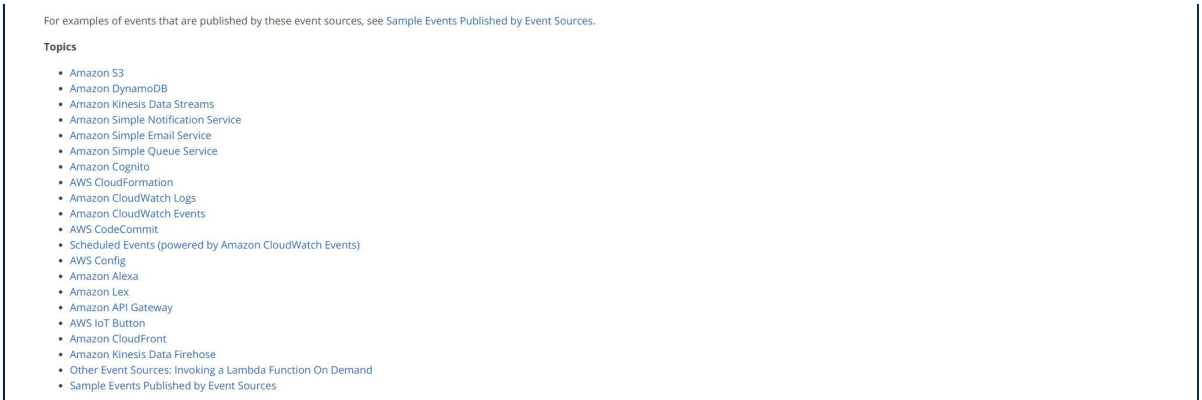
Which of the following are not the supported event sources for AWS Lambda function? (choose 2 options)

- [ ] **A.** AWS S3
- [ ] **B.** AWS IoT
- [x] **C.** AWS CodePipeline ✔
- [ ] **D.** AWS CodeCommit
- [x] **E.** AWS OpsWorks ✔

Explanation :

Answer: C, E

Following are the supported event sources for AWS Lambda function.



For examples of events that are published by these event sources, see Sample Events Published by Event Sources.

Topics

- Amazon S3
- Amazon DynamoDB
- Amazon Kinesis Data Streams
- Amazon Simple Notification Service
- Amazon Simple Email Service
- Amazon Simple Queue Service
- Amazon Cognito
- AWS CloudFormation
- Amazon CloudWatch Logs
- Amazon CloudWatch Events
- AWS CodeCommit
- Scheduled Events (powered by Amazon CloudWatch Events)
- AWS Config
- Amazon Alexa
- Amazon Lex
- Amazon API Gateway
- AWS IoT Button
- Amazon CloudFront
- Amazon Kinesis Data Firehose
- Other Event Sources: Invoking a Lambda Function On Demand
- Sample Events Published by Event Sources

- https://docs.aws.amazon.com/lambda/latest/dg/invoking-lambda-function.html (https://docs.aws.amazon.com/lambda/latest/dg/invoking-lambda-function.html)

Ask our Experts

QUESTION 3 INCORRECT

Your organization is having a requirement to perform big data analysis to transform data and store the result in AWS S3 bucket. They have implemented the solution using AWS Lambda due to its zero-administrative maintenance and cost-effective nature. However, in very few cases, the execution is getting abruptly terminated after 5 minutes. They would like to get a notification in such scenarios. What would you do?

- **A.** Setup timer in the code and send notification when the timer reaches 300 seconds.
- **B.** Configure SES for failures under Configuration option in the lambda function.
- **C.** Setup ERROR_NOTIFY environment variable with email address. Lambda function has inbuilt feature to send email during max memory and time out terminations using this environment variable. ✖
- **D.** Configure Dead-letter Queue and send notification to SNS topic ✔

Explanation :

Answer: D

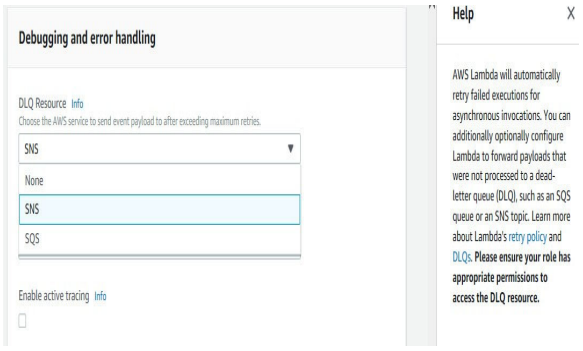
Option A is not correct. Although you can set the timers in the code, it may not be accurate measure to find if the lambda function is terminated after 300 seconds or it just finished executing on 300th second.

Option B is not correct. There is no option to configure AWS SES within Lambda setup.

Option C is not a valid statement.

Option D is correct. You can forward non-processed payloads to Dead Letter Queue (DLQ) using

AWS SQS, AWS SNS.



For more information on Dead Letter Queue (DLQ), refer documentation here.

- https://aws.amazon.com/blogs/compute/robust-serverless-application-design-with-aws-lambda-dlq/ (https://aws.amazon.com/blogs/compute/robust-serverless-application-design-with-aws-lambda-dlq/)

Ask our Experts

QUESTION 4 INCORRECT

Your organization uploads relatively large compressed files ranging between 100MB – 200MB in size to AWS S3 bucket. Once uploaded, they are looking to calculate the total number objects in the compressed file and add the total count as a metadata to the compressed file in AWS S3. They approached you for a cost-effective solution. You have recommended using AWS Lambda through S3 event notifications to perform this operation. However, they were concerned about failures as S3 event notification is an asynchronous one-time trigger and Lambda can fail due to operation time outs, max memory limits, max execution time limits etc. What is the best retry approach you recommend?

- **A.** All the failed events will be logged to CloudWatch. You can manually retrigger failed events.
- **B.** Configure Dead-letter queue with SQS. Configure SQS to trigger Lambda function again. ✔

- **C.** All failures will be caught during exception inside Lambda function. Trigger lambda function inside lambda function code to process failed event. ✖
- **D.** Enable Active tracing using AWS X-Ray. It will automatically retrigger failed events.

Explanation :

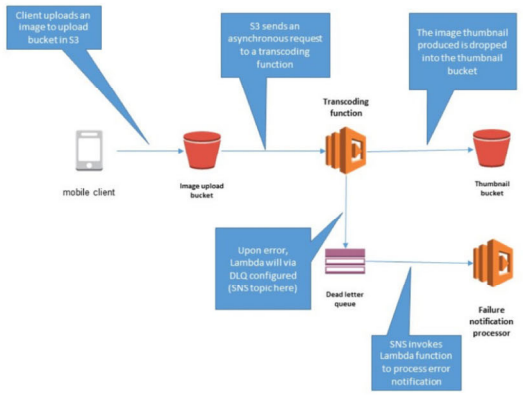
Answer: B

Option A is not recommended approach. Although you can configure logging to CloudWatch, it is difficult to find the specific failure logs. Manual retries are not a best practice in an enterprise level solution designs.

Option B is correct. You can forward non-processed or failed payloads to Dead Letter Queue

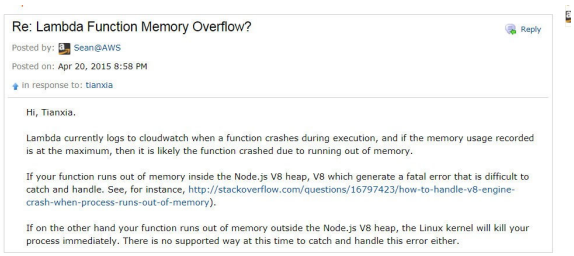
(DLQ) using AWS SQS, AWS SNS.

- **Asynchronous invocation** – Asynchronous events are queued before being used to invoke the Lambda function. If AWS Lambda is unable to fully process the event, it will automatically retry the invocation twice, with delays between retries. If you have specified a Dead Letter Queue for your function, then the failed event is sent to the specified Amazon SQS queue or Amazon SNS topic. If you don't specify a Dead Letter Queue (DLQ), which is not required and is the default setting, then the event will be discarded. For more information, see Dead Letter Queues.



- https://aws.amazon.com/blogs/compute/robust-serverless-application-design-with-aws-lambda-dlq/ (https://aws.amazon.com/blogs/compute/robust-serverless-application-design-with-aws-lambda-dlq/)

Option C is not correct. Max memory limit and max execution time limit gets terminated without being caught in the handler exception.



Option D is not correct. Active tracing option can be used for detailed logging. It will not retry failed events.

- https://docs.aws.amazon.com/lambda/latest/dg/lambda-x-ray.html (https://docs.aws.amazon.com/lambda/latest/dg/lambda-x-ray.html)

Ask our Experts

QUESTION 5 INCORRECT

You have a requirement to create a REST API using AWS API Gateway with Lambda as backend system and Oracle RDS instance as database. You have created API methods, Lambda function code and spinned up Oracle RDS instance in a private VPC with no Internet Gateway. When you are trying to connect to RDS instance from your Lambda, connection getting failed. What could be the reason? (choose 2 options)

- [x] **A.** Lambda execution role does not have policy to access RDS instance. ✖
- [ ] **B.** Lambda function is running in "no VPC" network mode. ✔
- [x] **C.** Lambda is running in same VPC as RDS, but RDS instance security group is not allowing connections from Lambda subnet range. ✔

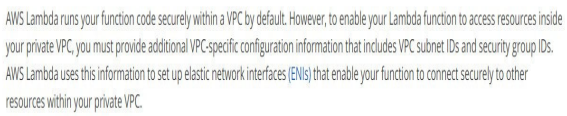
☐ **D.** RDS instance is not configured as destination in Lambda setup.

Explanation :

Answer: B, C

Option A is not correct. A policy on the role can only define access to which API actions can be made on RDS instance such as rds:CreateDBInstance, rds:CreateDBSecurityGroup, rds:CreateDBSnapshot etc. The policy will not define whether a resouce can connect to RDS instance or not.

Option B is correct. When Lambda function is running in "no VPC" network mode, it will not have access to resources running in a private VPC.



- https://docs.aws.amazon.com/lambda/latest/dg/vpc.html (https://docs.aws.amazon.com/lambda/latest/dg/vpc.html)

Option C is correct. Security groups act as a firewall for any resources (such as RDS instance and Lambda in this case) they are connected with. If there is no inbound rule defined to allow connections from Lambda subnet IP range or the Lambda security group, connections will fail.

- https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_SecurityGroups.html (https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_SecurityGroups.html)

Option D is not correct. There is no such configuration for the destination in Lambda setup.

Ask our Experts

QUESTION 6 CORRECT

Which of the following is customer's responsibility with respect to AWS Lambda service? (choose 2 options)

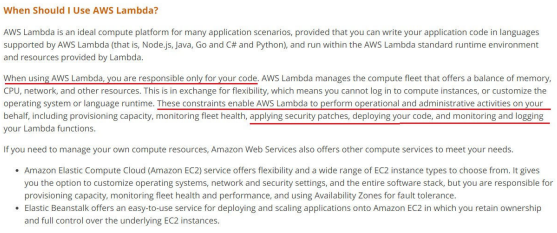
- [x] **A.** Lambda function code. ✔
- [ ] **B.** Monitoring and logging lambda functions.
- [ ] **C.** Security patches.
- [ ] **D.** Installing required libraries in underlying compute instances for Lambda execution.
- [x] **E.** Providing access to AWS resources which triggers Lambda function. ✔

Explanation :

Answer: A, E

Option A is correct.

Options B, C are not correct. Refer below screen shot



When Should I Use AWS Lambda?

AWS Lambda is an ideal compute platform for many application scenarios, provided that you can write your application code in languages supported by AWS Lambda (that is, Node.js, Java, Go and C# and Python), and run within the AWS Lambda standard runtime environment and resources provided by Lambda.

When using AWS Lambda, you are responsible only for your code. AWS Lambda manages the compute fleet that offers a balance of memory, CPU, network, and other resources. This is in exchange for flexibility, which means you cannot log in to compute instances, or customize the operating system or language runtime. These constraints enable AWS Lambda to perform operational and administrative activities on your behalf, including provisioning capacity, monitoring fleet health, applying security patches, deploying your code, and monitoring and logging your Lambda functions.

If you need to manage your own compute resources, Amazon Web Services also offers other compute services to meet your needs.

- Amazon Elastic Compute Cloud (Amazon EC2) service offers flexibility and a wide range of EC2 instance types to choose from. It gives you the option to customize operating systems, network and security settings, and the entire software stack, but you are responsible for provisioning capacity, monitoring fleet health and performance, and using Availability Zones for fault tolerance.
- Elastic Beanstalk offers an easy-to-use service for deploying and scaling applications onto Amazon EC2 in which you retain ownership and full control over the underlying EC2 instances.

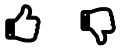
Option D is not correct. You cannot login to underlying compute instances of lambda execution. So, we cannot install any required libraries. However, you can package all the required dependent libraries along with your code.
Refer below documentation for more information on creating deployment package for Lambda functions.

- https://docs.aws.amazon.com/lambda/latest/dg/with-s3-example-deployment-pkg.html (https://docs.aws.amazon.com/lambda/latest/dg/with-s3-example-deployment-pkg.html)

Option E is correct. AWS Lambda assumes the role assigned during setup to access any AWS resources it performs any action on. Policy on the role must grant access on any such resources in order for Lambda to perform operations, for example S3 getobject, Dynamodb GetItem etc.

- https://docs.aws.amazon.com/lambda/latest/dg/intro-permission-model.html#lambda-intro- execution-role (https://docs.aws.amazon.com/lambda/latest/dg/intro-permission-model.html#lambda-intro-%20execution-role)

Ask our Experts



QUESTION 7 CORRECT

Which of the following is not a potential use case for using AWS Lambda?

- **A.** Periodically check the log files for errors in CloudWatch or CloudTrail and send out notifications through SNS.
- **B.** Download S3 bucket objects of size varying between 500 MB-2 GB to a Lambda Ephemeral disk or temp location, read and analyze them for key words and add the key words to the metadata of file object for search purposes. ✔
- **C.** Scheduled job to generate AWS resource usage reports based on certain tags.

○ **D.** A website with highly scalable backend layer which will persist data into RDS or DynamoDB.

Explanation :

Answer: B

Option A is a potential use case for AWS Lambda. You can use Lambda as scheduled event and read log files from AWS CloudWatch or CloudTrail and report any errors through SNS notifications.

Option C is a potential use case.

For more information on scheduling Lambda functions, refer documentation here.

- https://docs.aws.amazon.com/lambda/latest/dg/with-scheduled-events.html?shortFooter=true (https://docs.aws.amazon.com/lambda/latest/dg/with-scheduled-events.html?shortFooter=true)

Option D is a potential use case.

You can host the web frontend on S3, and accelerate content delivery with Cloudfront caching. The web frontend can send requests to Lambda functions via API Gateway HTTPS endpoints. Lambda can handle the application logic, and persist data to a fully managed database service

(RDS for relational, or DynamoDB for non relational database). You can host your Lambda functions and databases within a VPC to isolate them from other networks.
Here is the documentation for building a serverless website.

- https://aws.amazon.com/getting-started/projects/build-serverless-web-app-lambda-apigateway-s3-dynamodb-cognito/ (https://aws.amazon.com/getting-started/projects/build-serverless-web-app-lambda-apigateway-s3-dynamodb-cognito/)

Option B looks like a potential use case. But the scenario will fail due to the /tmp directory space limitation.

Q: What if I need scratch space on disk for my AWS Lambda function?

Each Lambda function receives 500MB of non-persistent disk space in its own /tmp directory.

AWS Lambda Limits

AWS Lambda Resource Limits per Invocation

Resource	Limits
Memory allocation range	Minimum = 128 MB / Maximum = 3008 MB (with 64 MB increments). If the maximum memory use is exceeded, function invocation will be terminated.
Ephemeral disk capacity ("/tmp" space)	512 MB

Ask our Experts

QUESTION 8 CORRECT

You have created a Lambda function for reading data from Kinesis stream of transactions. In the code you were using context logger so that it can log to CloudWatch and you can monitor them at later point of time. Lambda function started running along with Kinesis stream, however, you do not see any log entries for the new Lambda function. What could the reason?

- **A.** Lambda functions with Kinesis stream as event source do not write logs to CloudWatch.
- **B.** Lambda execution role policy does not have access to create CloudWatch logs. ✔
- **C.** Lambda function execution logs will be written to CloudTrail, not to CloudWatch.
- **D.** Active tracing is not enabled on the Lambda function setup configuration.

Explanation :

Answer: B

Option A is not a valid statement. Lambda function will write logs as long as the execution role has access to create and write CloudWatch logs irrespective of source that triggered it.

Option B is correct.

Option C is not correct. AWS CloudTrail is used for logging API calls made to services such as
AWS Lambda, AWS S3 etc.
AWS CloudWatch for Lambda is used for execution logging.

- https://docs.aws.amazon.com/lambda/latest/dg/with-cloudtrail.html (https://docs.aws.amazon.com/lambda/latest/dg/with-cloudtrail.html)

Option D is not correct. AWS X-Ray traces requests made to your serverless applications built using AWS Lambda. This will not be the reason for failing to write logs to CloudWatch.

- https://docs.aws.amazon.com/lambda/latest/dg/lambda-x-ray.html (https://docs.aws.amazon.com/lambda/latest/dg/lambda-x-ray.html)

Ask our Experts

QUESTION 9 CORRECT

When configuring AWS SQS as event source for AWS Lambda function, what is the maximum batch size supported by AWS SQS for ReceiveMessage call?

- A. 20
- B. 40
- C. 10 ✔
- D. 100

Explanation :

Answer: C

- **Enabled**: A flag to signal AWS Lambda that it should start polling your specified Amazon SQS queue.
- **EventSourceArn**: The ARN (Amazon Resource Name) of your Amazon SQS queue that AWS Lambda is monitoring for new messages.
- **FunctionName**: The Lambda function to invoke when AWS Lambda detects new messages on your configured Amazon SQS queue.

Here are other parameters apart from batch size.

Important

Amazon Simple Queue Service supports both Standard and FIFO queues. AWS Lambda supports only standard queues. For more information on the difference, see What Type of Queue Do I Need?

When using these operations to map your Lambda function to an Amazon SQS queue, note the following configuration parameters:

- **BatchSize**: The number of records that AWS Lambda will retrieve from each ReceiveMessage call. Both the default and maximum batch size supported by Amazon Simple Queue Service is up to 10 queue messages per batch.

- https://docs.aws.amazon.com/lambda/latest/dg/with-sqs.html (https://docs.aws.amazon.com/lambda/latest/dg/with-sqs.html)

Ask our Experts

QUESTION 10 CORRECT

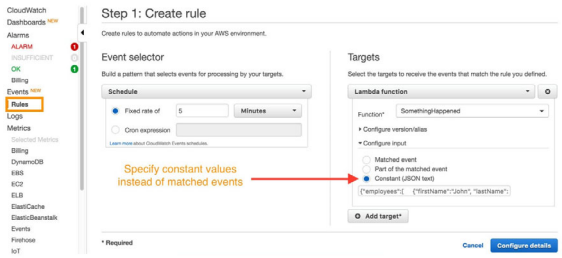
You are to planning to schedule a daily job with AWS CloudWatch scheduled event and AWS Lambda function as trigger to the event which will perform daily health check on your applications running on fleet of EC2 instances. For you to achieve this, you need to provide the EC2 instances' name tags to identify right resources. What is the correct way of passing the inputs in this case?

- **A.** Configure "Variables" option on AWS CloudWatch scheduled event.
- **B.** You can modify the "Matched Event" option while selecting AWS Lambda as trigger for CloudWatch scheduled event.
- **C.** You can set "Constant (JSON text)" option while selecting AWS Lambda as trigger for CloudWatch scheduled event. ✔
- **D.** "Details" object of "Matched Event" can be configured while creating AWS CloudWatch scheduled event.

Explanation :

Answer: C

When using an AWS Cloudwatch rule to trigger a Lambda event, one of the multiple options you have to pass data onto your Lamba function is "Constant (JSON Text)". This handy feature allows you to send static content to your function instead of the matched event.



- https://aws.amazon.com/blogs/compute/simply-serverless-use-constant-values-in-cloudwatch-event-triggered-lambda-functions/ (https://aws.amazon.com/blogs/compute/simply-serverless-use-constant-values-in-cloudwatch-event-triggered-lambda-functions/)

Ask our Experts

QUESTION 11 CORRECT

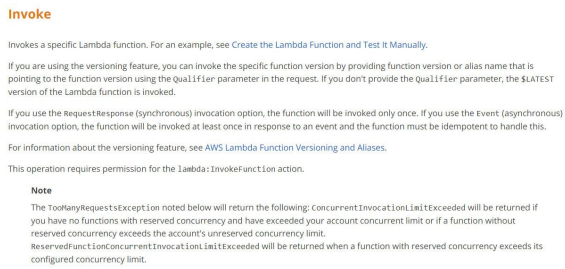
You have an existing AWS setup with DirectConnect. You have migrated certain on premise backend applications to AWS EC2 instances which are having certain processes run based on triggers from other applications. These processes are developed on JAVA programming language. Your organization is looking to migrate these processes to Lambda and reduce the cost incurred on EC2 instances. What would be your recommendation?

- **A.** AWS Lambda cannot be invoked from a custom application. They can only be triggered by AWS supported event sources.
- **B.** Replicate the JAVA code easily onto AWS Lambda function with few modifications and use Lambda Invoke API with input passed as custom event. ✔
- **C.** Trigger Lambda from AWS CloudWatch scheduled event and invoke CloudWatch API from your applications.
- **D.** AWS Lambda is not designed to run backend applications. Better to use EC2 for that purpose.

Explanation :

Answer: B

You can invoke a Lambda function using a custom event through AWS Lambda's invoke API. Only the function's owner or another AWS account that the owner has granted permission can invoke the function.



Invoke

Invokes a specific Lambda function. For an example, see Create the Lambda Function and Test It Manually.

If you are using the versioning feature, you can invoke the specific function version by providing function version or alias name that is pointing to the function version using the `Qualifier` parameter in the request. If you don't provide the `Qualifier` parameter, the `$LATEST` version of the Lambda function is invoked.

If you use the `RequestResponse` (synchronous) invocation option, the function will be invoked only once. If you use the `Event` (asynchronous) invocation option, the function will be invoked at least once in response to an event and the function must be idempotent to handle this.

For information about the versioning feature, see AWS Lambda Function Versioning and Aliases.

This operation requires permission for the `lambda:InvokeFunction` action.

Note

The `TooManyRequestsException` noted below will return the following: `ConcurrentInvocationLimitExceeded` will be returned if you have no functions with reserved concurrency and have exceeded your account concurrent limit or if a function without reserved concurrency exceeds the account's unreserved concurrency limit. `ReservedFunctionConcurrentInvocationLimitExceeded` will be returned when a function with reserved concurrency exceeds its configured concurrency limit.

- https://docs.aws.amazon.com/lambda/latest/dg/API_Invoke.html (https://docs.aws.amazon.com/lambda/latest/dg/API_Invoke.html)
- https://docs.aws.amazon.com/lambda/latest/dg/with-userapp.html (https://docs.aws.amazon.com/lambda/latest/dg/with-userapp.html)

Ask our Experts

QUESTION 12 INCORRECT

Which of the following are AWS CloudFront events that can trigger AWS Lambda@edge function? (choose 3 options)

- [x] **A.** Viewer Request ✔
- [x] **B.** CloudFront Cache ✖
- [ ] **C.** Sender Request

- [x] **D.** Origin Request ✔
- [ ] **E.** Origin Response ✔

Explanation :

Answer: A, D, E