EXAMTOPICS

Amazon Discussions

Exam AWS Certified Developer - Associate DVA-C02 All Questions

View all questions & answers for the AWS Certified Developer - Associate DVA-C02 exam

Go to Exam

EXAM AWS CERTIFIED DEVELOPER - ASSOCIATE DVA-C02 TOPIC 1 QUESTION 116 DISC...

Exam question from Amazon's AWS Certified Developer - Associate DVA-C02
Question #: 116
Topic #: 1
[All AWS Certified Developer - Associate DVA-C02 Questions]

A company has hundreds of AWS Lambda functions that the company's QA team needs to test by using the Lambda function URLs. A developer needs to configure the authentication of the Lambda functions to allow access so that the QA IAM group can invoke the Lambda functions by using the public URLs.

Which solution will meet these requirements?

A. Create a CLI script that loops on the Lambda functions to add a Lambda function URL with the AWS_IAM auth type. Run another script to create an IAM identity-based policy that allows the lambda:InvokeFunctionUrl action to all the Lambda function Amazon Resource Names (ARNs). Attach the policy to the QA IAM group.

B. Create a CLI script that loops on the Lambda functions to add a Lambda function URL with the NONE auth type. Run another script to create an IAM resource-based policy that allows the lambda:InvokeFunctionUrl action to all the Lambda function Amazon Resource Names (ARNs). Attach the policy to the QA IAM group.

C. Create a CLI script that loops on the Lambda functions to add a Lambda function URL with the AWS_IAM auth type. Run another script to loop on the Lambda functions to create an IAM identity-based policy that allows the lambda:InvokeFunctionUrl action from the QA IAM group's Amazon Resource Name (ARN).

D. Create a CLI script that loops on the Lambda functions to add a Lambda function URL with the NONE auth type. Run another script to loop on the Lambda functions to create an IAM resource-based policy that allows the lambda:InvokeFunctionUrl action from the QA IAM group's Amazon Resource Name (ARN).

Show Suggested Answer

by MrTee at *April 22, 2023, 11:08 p.m.*

Disclaimers:
- ExamTopics website is **not** related to, affiliated with, endorsed or authorized by Amazon.
- Trademarks, certification & product names are used for reference only and belong to Amazon.

Comments

MrTee Highly Voted 2 years ago

Selected Answer: A

Option A meets these requirements?

upvoted 16 times

ppardav 1 year, 10 months ago

https://docs.aws.amazon.com/lambda/latest/dg/urls-auth.html

upvoted 4 times

jipark 1 year, 9 months ago

create 'AWS_IAM auth type' -> Attach the policy to the QA IAM group

upvoted 5 times

sumanshu Most Recent 4 months, 3 weeks ago

Selected Answer: A

A) Correct - Setting the AWS_IAM authentication type ensures that only IAM users or roles with the right permissions can invoke the Lambda function URLs.

C) Eliminated - While this approach is secure, creating individual resource-based policies for hundreds of Lambda functions is unnecessarily complex and hard to manage. An identity-based policy (used in Option A) is simpler because it applies to the entire QA IAM group at once.

B/D - Eliminated - Setting NONE as the auth type makes the Lambda function URLs publicly accessible without authentication.

upvoted 1 times

Anandesh 10 months, 1 week ago

Selected Answer: A

Apologies again, please refer to the youtube link I shared earlier..correct ans is A

upvoted 1 times

Anandesh 10 months, 2 weeks ago

I think the answer is B here, reason being the function should be invoked using public urls upvoted 1 times

65703c1 12 months ago

Selected Answer: A

A is the correct answer.

upvoted 1 times

SerialiDr 1 year, 2 months ago

Selected Answer: A

This approach leverages AWS IAM authentication (AWS_IAM auth type) for Lambda function URLs, ensuring that only authenticated and authorized IAM entities can invoke the Lambda functions. By creating an IAM policy that specifies the lambda:InvokeFunctionUrl action and attaching it to the QA IAM group, you provide the necessary permissions for the QA team to invoke the Lambda functions securely. This method aligns with AWS best practices for security and access control, allowing for scalable and manageable access management across multiple Lambda functions.

upvoted 1 times

CrescentShared 1 year, 2 months ago

Selected Answer: C

I don't know why so much A, but ins't A giving the access to all the lambda?

upvoted 1 times

SD_CS 1 year, 3 months ago

Selected Answer: A

I have to go for A even though it appears both should suffice. I took this from AWS Documentation

If you choose the AWS_IAM auth type, users who need to invoke your Lambda function URL must have the lambda:InvokeFunctionUrl permission. Depending on who makes the invocation request, you may have to grant this permission using a resource-based policy.

If the principal making the request is in the same AWS account as the function URL, then the principal must either have lambda:InvokeFunctionUrl permissions in their identity-based policy, OR have permissions granted to them in the function's resource-based policy.

AWS clearly states both should be good. The reason for selecting A is the wording is clear, loop on to lambda function to provide the permission was bit of confusing to me.

upvoted 1 times

konieczny69 1 year, 3 months ago

Selected Answer: C

I don't get all A answers. This is typical resource based policy that allows invoking a function by concrete principal - in this case its the QA role.

For all those who vote for A - go ahead and create simple API Gateway with a lambda integration type. Then look at the resource based policy - lambda:InvokeFunction allowed by apigateway.amazonaws.com with ArnLike condition.

ChatGTP also says C.

upvoted 2 times

love777 1 year, 8 months ago

Selected Answer: C

Explanation:

In this scenario, the QA team needs to test AWS Lambda functions using Lambda function URLs while ensuring proper authentication and access control. Here's why option C is the appropriate solution:

Authentication Type: Using the AWS_IAM auth type for the Lambda function URLs ensures that the Lambda functions can be invoked only by users and roles that have the necessary IAM permissions.

Identity-Based Policy: By creating an IAM identity-based policy, you grant permissions directly to the QA IAM group to invoke the Lambda functions using the Lambda function URLs. This provides fine-grained control over which IAM entities can access the functions.

Option A uses the AWS_IAM auth type and creates a policy for the QA IAM group, which is a good direction. However, the creation of a policy that allows lambda:InvokeFunctionUrl for all Lambda function ARNs might grant excessive permissions.

upvoted 4 times

[Removed] 1 year, 5 months ago

pay attention to the wording of the answers:
A - Run another script to create an IAM identity-based policy that allows the lambda:InvokeFunctionUrl action to all the Lambda function Amazon Resource Names (ARNs).
*This option is very clear. You are creating an IAM identity based policy allowing access to invoke the function and then attaching this policy to the QA IAM group.

C - Run another script to loop on the Lambda functions to create an IAM identity-based policy that allows the lambda:InvokeFunctionUrl action from the QA IAM group's Amazon Resource Name (ARN).
*What does "Run another script to loop on the Lambda functions" What does this even mean?? are we doing some sort of while loop here? Wording for this option is very confusing and makes no sense to me. I go with A

upvoted 4 times

Manel87 1 year, 4 months ago

good thought!

upvoted 1 times

dezoito 1 year, 7 months ago

Why A grant excessive permissions? The policy will contain only the Lambda's ARNs wich the QA group should have access to.

upvoted 2 times