nextwork.org

Fetch Data with AWS Lambda

ME Melvin J Bonner

ME
Melvin J Bonner
NextWork Student

Introducing Today's Project!

In this project, I will demonstrate how to store and manage data relevant to this application. I'm ding this project to learn how to connect DynamoDB to a Lambda Function

Tools and concepts

The services I used were DyanmoDB and Lambda. The key concepts I learned included Lambda functions, IAM permission policies, and DynamoDB tables.

Project reflection

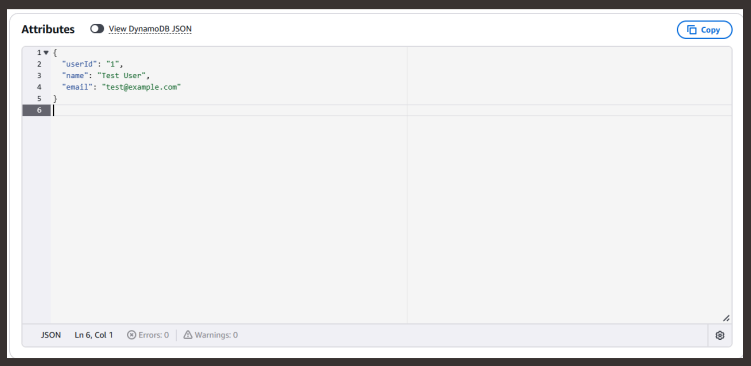
This project took me approximately 3 hours. The most challenging part was updating the IAM permissions and testing the Lambda function. It was most rewarding to get the positive results on the Lamda function test.

I did this project to complete the four part three-tier architecture series. This project met my goals.

Project Setup

To set up my project, I created a database using DynamoDB. The partition key is the heart of how DynamoDB organizes data. Think of the partition key as a label that you can use to group similar items.

In my DynamoDB table, I added a new item. DynamoDB is shemaless, which means you can add attributes as you need, and every item in your database can have a different set of attributes.



AWS Lambda

AWS Lambda is a service that lets you run code without needing to manage any servers. I'm using Lambda in this project to process data and respond to events.

ME
Melvin J Bonner
NextWork Student

AWS Lambda Function

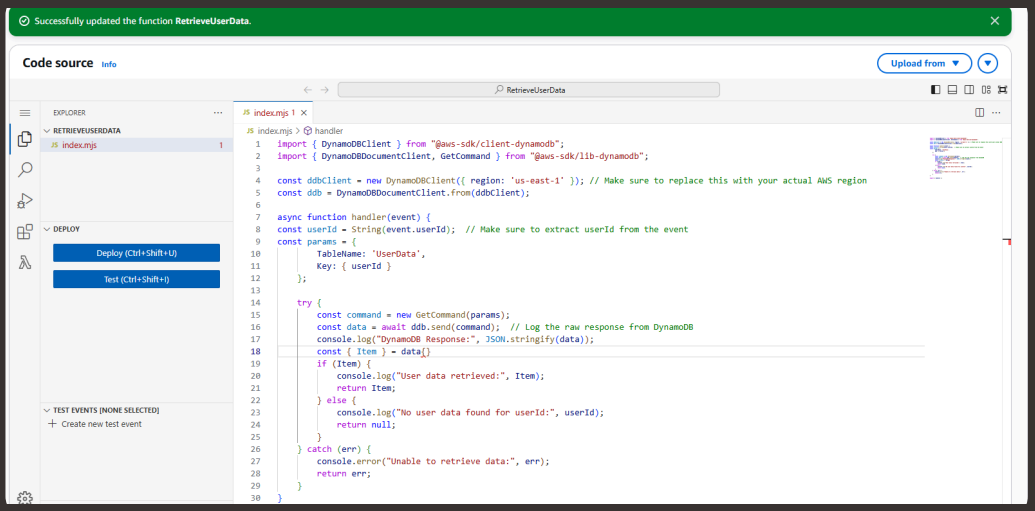
My Lambda functions has an execution role, which is create a new new role with basic Lambda permissions. By default, the role grants basic permissions for writing logs to CloudWatch.

My Lambda function will use the AWS SDK for JavaScript to interact with DynamoDB and handle potential errors during the database operation.

The code uses AWS SDK, which is is a set of tools that let developers build apps that interact with AWS. My code uses SDK to use prewritten functions for communicating with DynamoDB and getting data from a table.

ME

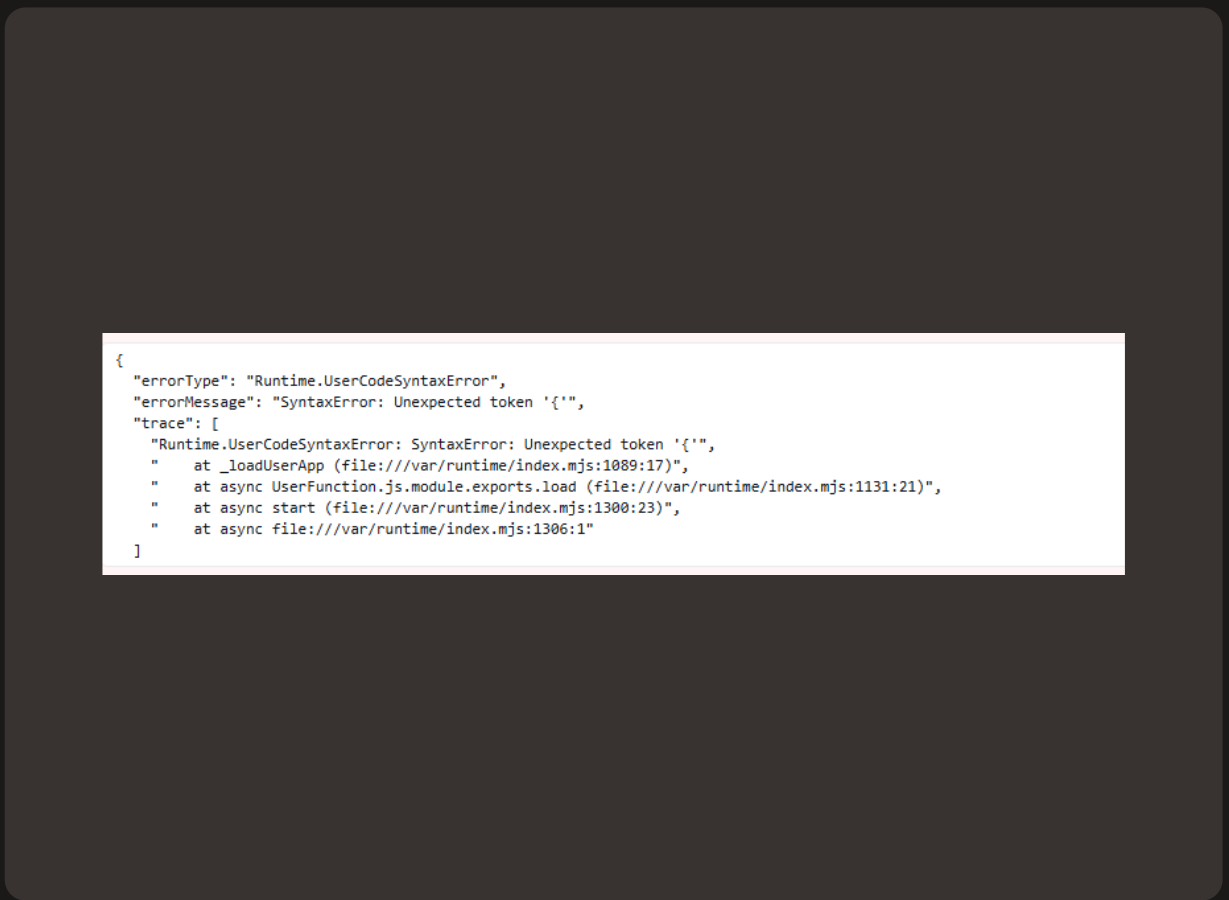
Melvin J Bonner
NextWork Student



Function Testing

To test whether my Lambda function works, I will be writing a test that will show me how the function would react if a data request was made. The test is written in Json. If the test is successful, I will see Executing function succeeded.

This test displayed a 'success' because the function itself could run. The function's response was a failed test because Lambda doesn't have permission to access the DynamoDB table.

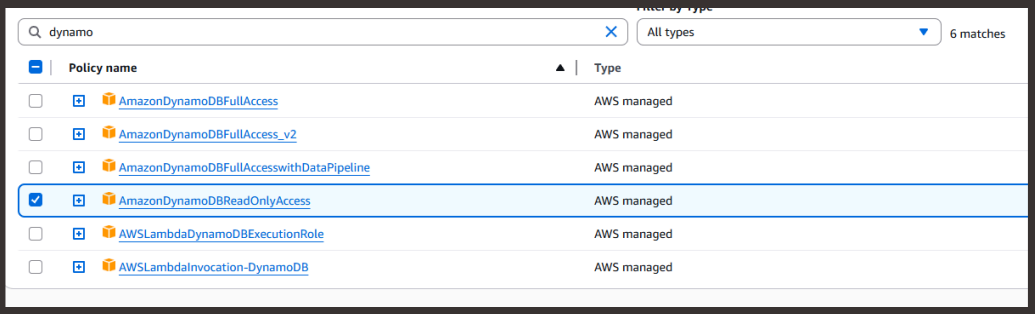


Function Permissions

To resolve the AccessDenied error I added one permission policy, DynamoDBReadOnlyAccess, it contains the GetItem permission that is needed for Lambda.

There were 6 DynamoDB permission policies I could have chosen from, but I didn't pick FullAccess, FullAccess v2, FullAccess with Data Pipeline, DBExecution, or Invocation access because Lambda needs the GetItem permission.

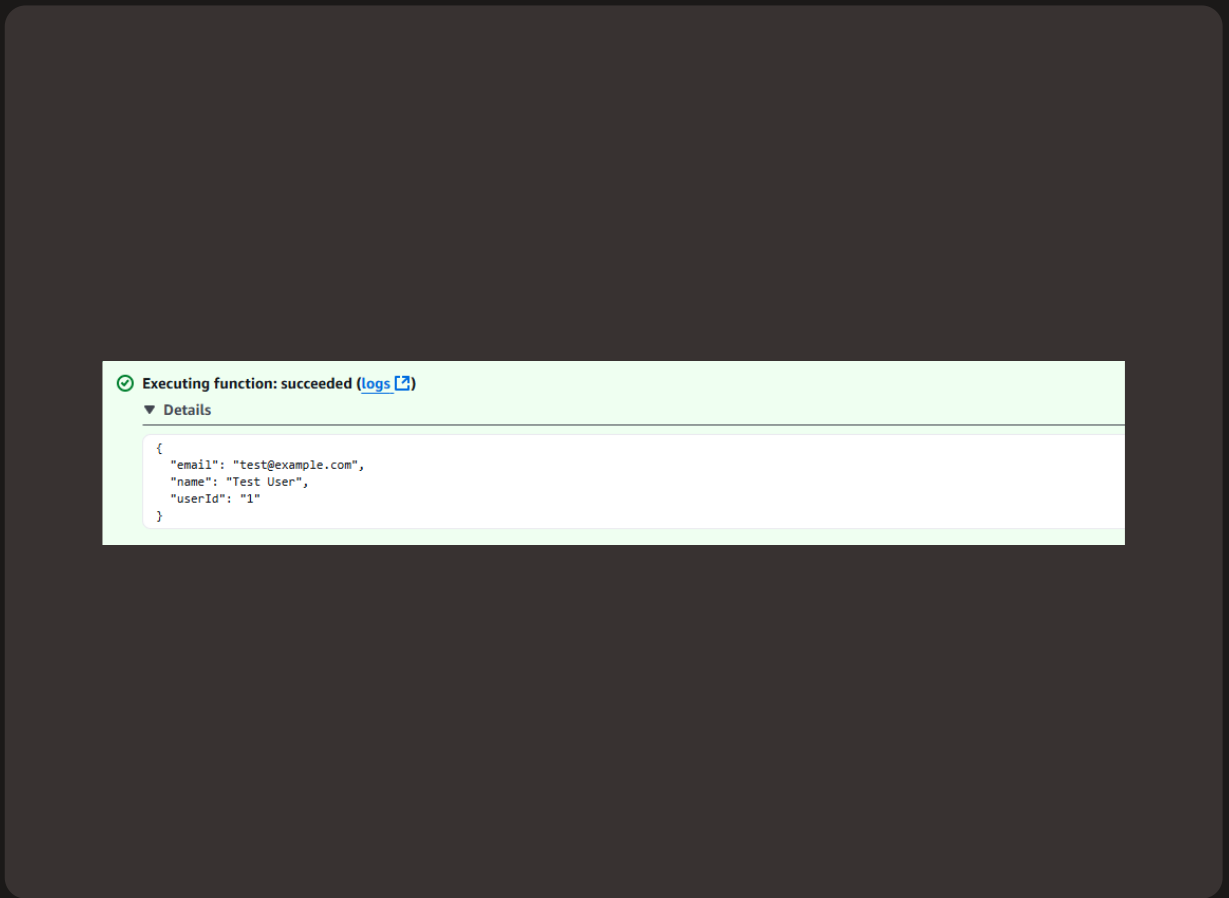
I also didn't pick FullAccess, FullAccess v2, FullAccess with Data Pipeline, DBExecution, or Invocation because AmazonDynamoDBReadOnlyAccess was the write choice because it contained the GetItem that was needed for Lambda.


dynamo
All types
6 matches
Policy name
Type
AmazonDynamoDBFullAccess
AWS managed
AmazonDynamoDBFullAccess_v2
AWS managed
AmazonDynamoDBFullAccesswithDataPipeline
AWS managed
AmazonDynamoDBReadOnlyAccess
AWS managed
AWSLambdaDynamoDBExecutionRole
AWS managed
AWSLambdaInvocation-DynamoDB
AWS managed

Final Testing and Reflection

To validate my permission settings, I reran the test event. The results were successful because my Lambda function now has the necessary permissions to access DynamoDB.

Web apps are a popular use case fo using Lambda and DynamoDB. For example, I could use Lambda to help customers find products, get product information or see their order history by fetching data from DynamoDB.



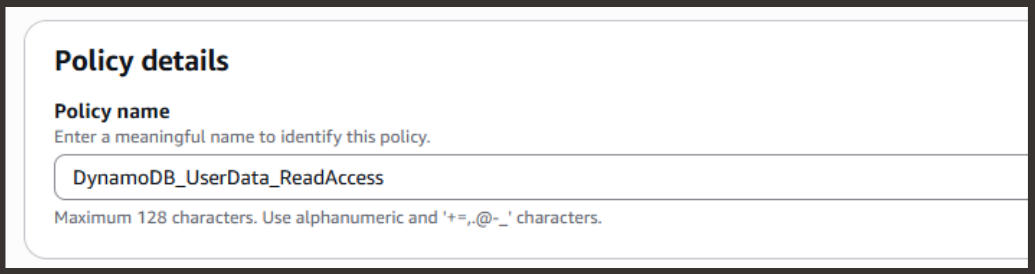
ME
Melvin J Bonner
NextWork Student

Enahancing Security

For my project extension, I callenged myself to create an inline policy. This will allow for more granular control, I want my Lambda function to only have access to my UserData table.

To create the permission policy, I used JSON because the process is quicker when you have the JSON code.

When updating a Lambda function's permission policies, I could risk granting broader permissions than necessary and allowing unintended access to other AWS services. I validated that my Lamba function still works by testing the function and receiving positive results.


Policy details
Policy name
Enter a meaningful name to identify this policy.
DynamoDB_UserData_ReadAccess
Maximum 128 characters. Use alphanumeric and '+=,.@-_' characters.

nextwork.org

The place to learn & showcase your skills

Check out nextwork.org for more projects