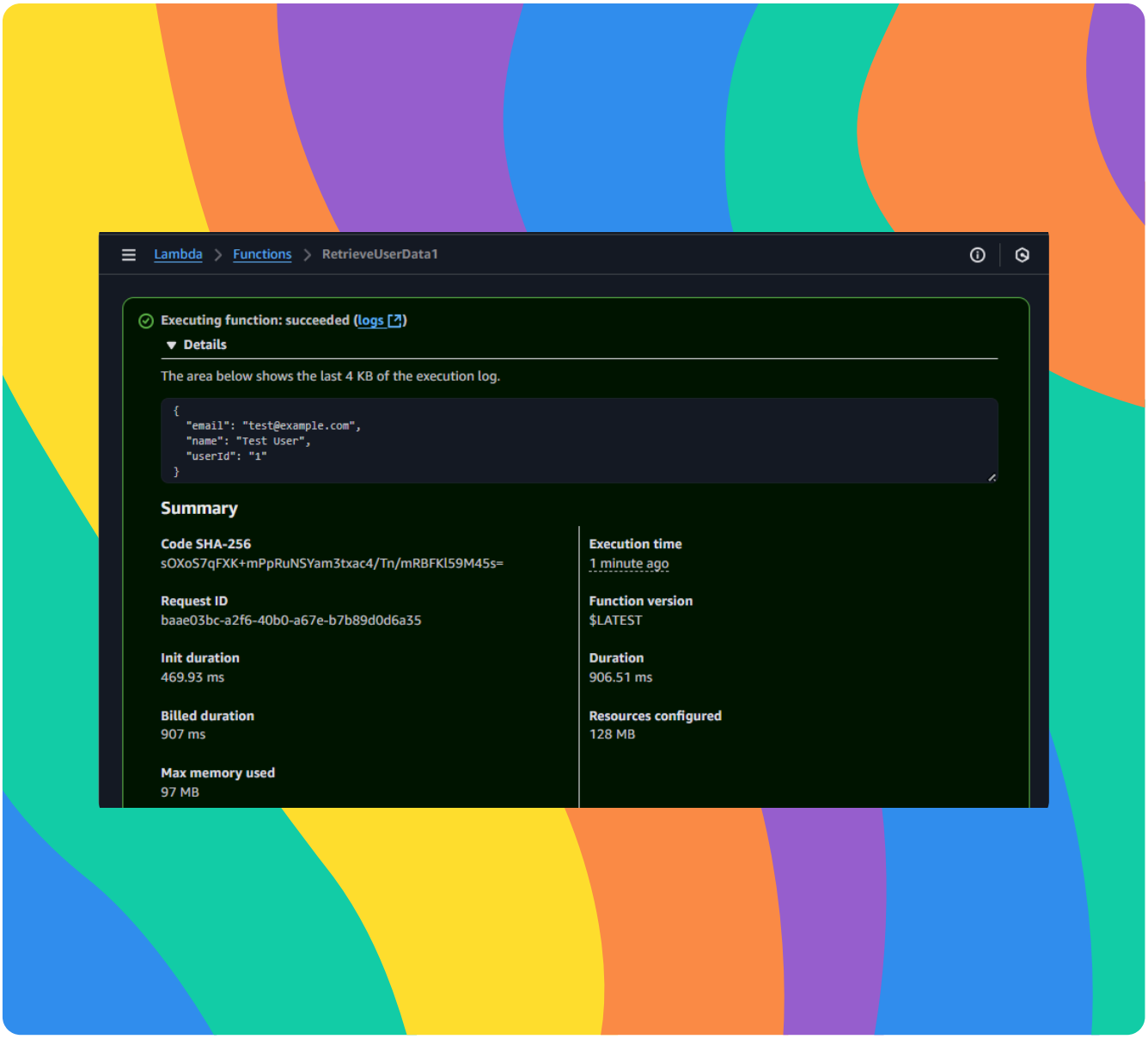
nextwork.org

Fetch Data with AWS Lambda

jyothesh karnam



jyothesh karnam
NextWork Student

Introducing Today's Project!

In this project, I will demonstrate how to use AWS Lambda to retreve data that's in a DynamoDB database. We're doing this project to learn how to set up the data tier of a web app and connect that with a Lambda function too (logic tier).

Tools and concepts

Services I used in this project were Amazon DynamoDB and AWS Lambda. Key concepts I learnt include Lambda functions, adding table items, testing Lambda functions, how to select the right permission policy, and writing custom inline policies.

Project reflection

This project took me approximately 1 hour and 15 minutes including demo time. The most challenging part was writing my own inline policy (super new experience). It was most rewarding to see the right data pop up when I ran a successful function test.

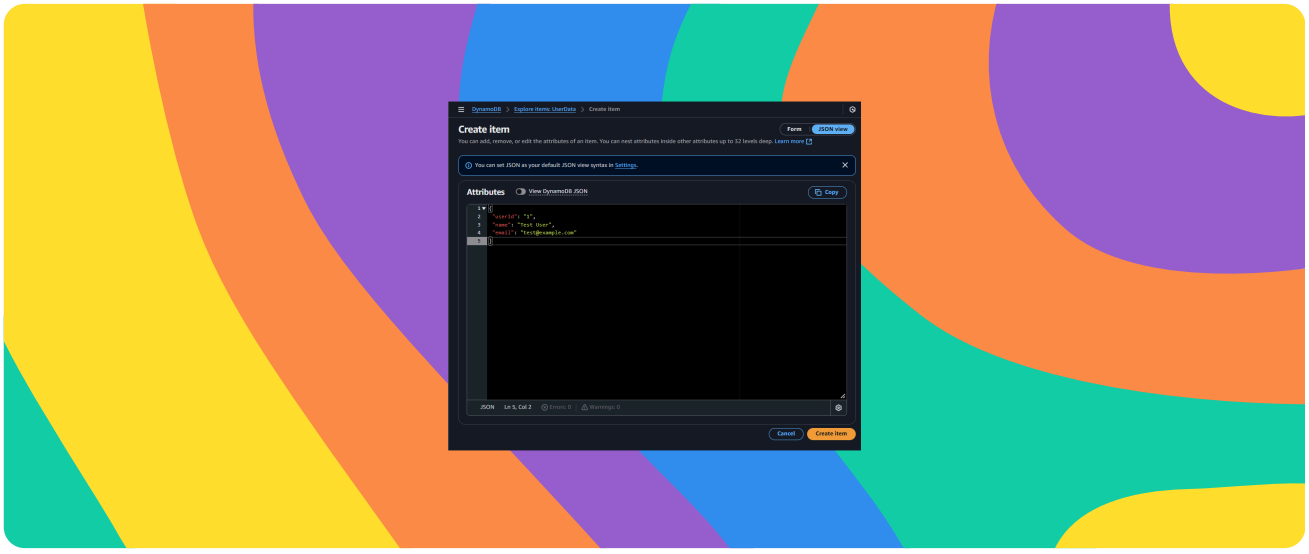
I did this project today to learn about the data tier of a three-tier architecture. This project definitely met my goals as I learnt how to connect it to AWS Lambda (logic tier) as well.

jyothesh karnam
NextWork Student

Project Setup

To set up my project, I created a database using DynamoDB. DynamoDB is a NoSQL database, so it is vert fast and flexible at data retrieval and flexible for data storage. The partition key is userId which means the key identifier for each data is its userId.

In my DynamoDB table, I added a piece of data. DynamoDB is schemaless, which means that the data we add can come with new attributes, and every piece of data can have its set of attributes without affecting other data.



jyothesh karnam
NextWork Student

AWS Lambda

AWS Lambda is a service that lets me run code without having to manage servers. I am usinf Lambda in this project to create a function (i.e a peice of code) that retrieves data from a DynamoDB table.

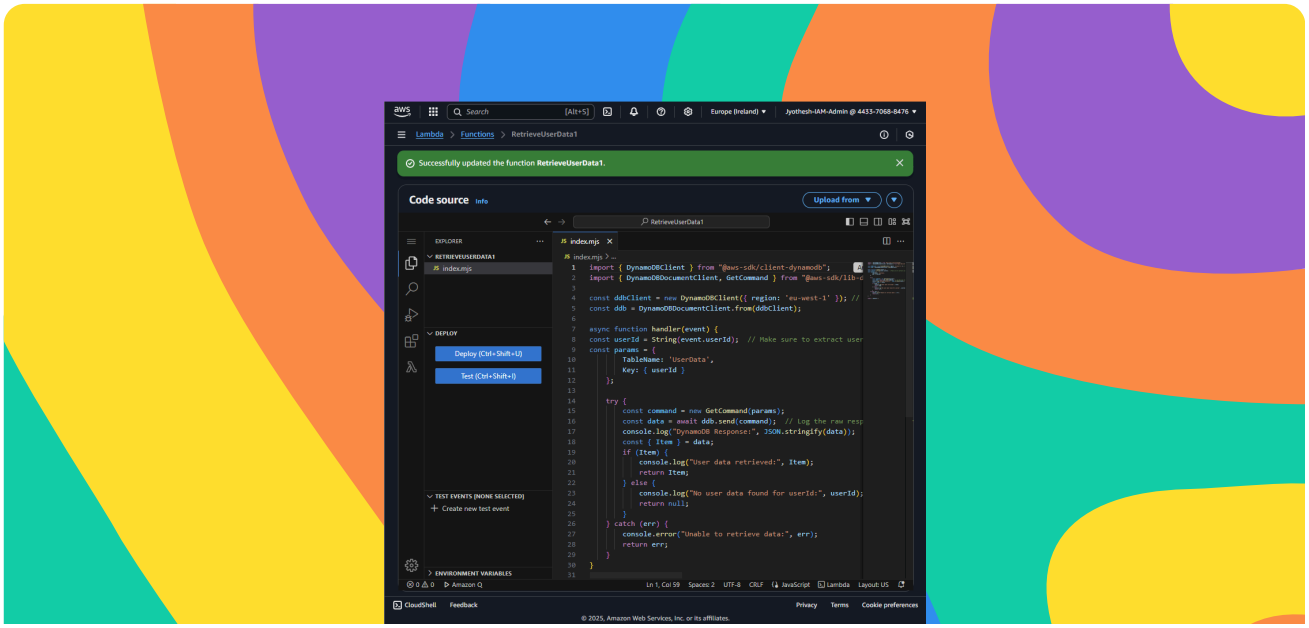
jyothesh karnam
NextWork Student

AWS Lambda Function

My Lambda function has an execution role, which is the setup permissions that it has to interact with other AWS services. By default, the role grants basic Lambda permissions, which are the permissions to write logs with CloudWatch e.g. error logs.

My Lambda function will retrieve data from the DynamoDB database table. The first half of our code is all about the data retrieval + returning it as output. The second half of the code is all about sending responses back. e.g. success messages, item not found, error messages.

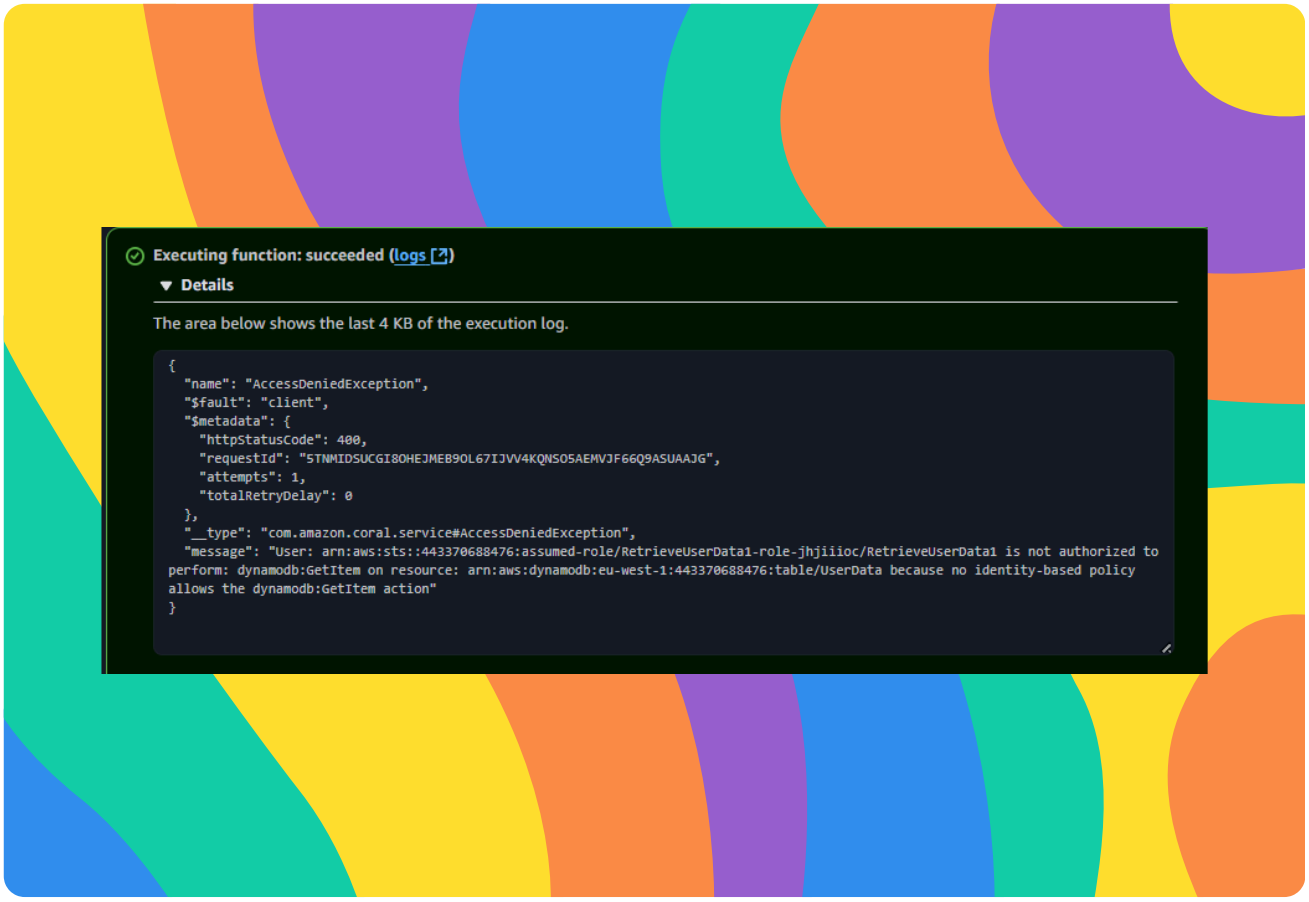
The code uses AWS SDK, which is set of tools that makes it easier for developers to use AWS in their application code. My code uses SDK to get access to DynamoDB actions i.e. read/retrieve data items.



Function Testing

To test whether my Lambda function works, I wrote a function test in the test tab. The test is written in JSON. If the test is successful, I'd see the correct peice of data in JSON when we query for userId=1

The test displayed a 'success' because the function itself ran with no errors i.e. no syntax or dependency issues. But the function's response was actually an error because Lambda function is retrieving data from DynamoDB but does not have permissions to DynamoDB and because of that the access was denied.



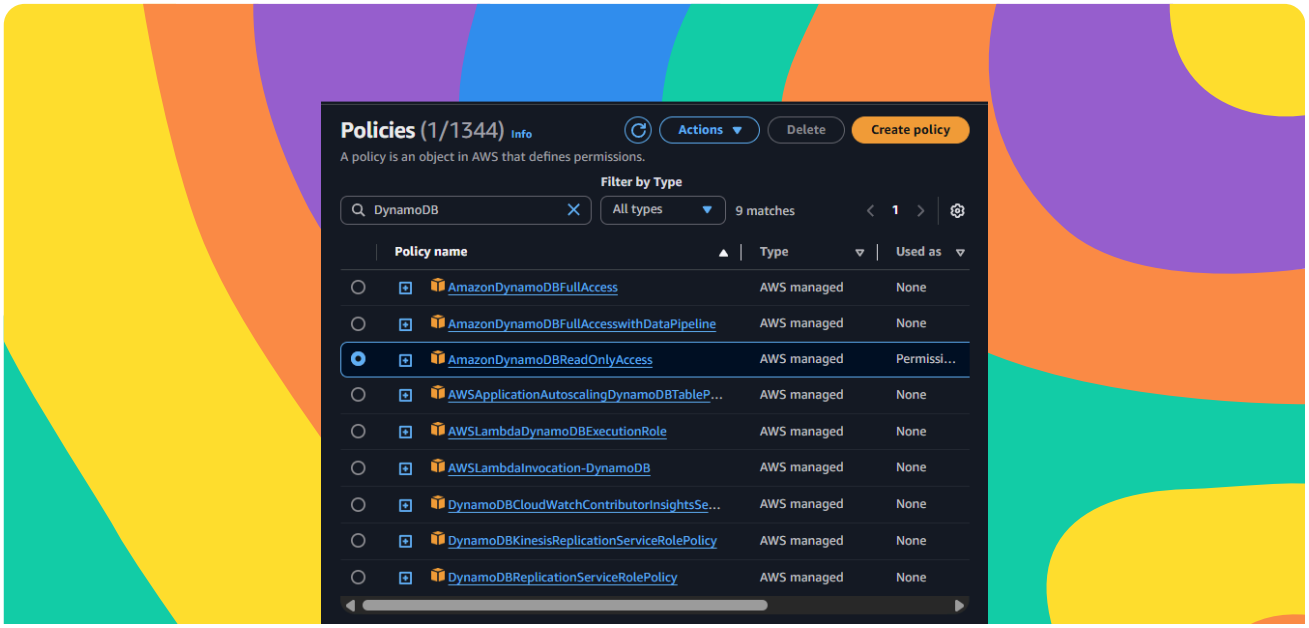
jyothesh karnam
NextWork Student

Function Permissions

To resolve the AccessDenied error, I am providing the Lambda function with the permissions to perform GetItem on a DyamoDB table. This is because this is the action that Lambda is currently blocked from doing.

There were four DynamoDB permission policies I could choose from, but I didn't pick AWSLambdaDynamoDBExecutionRole or AWSLambdaInvocation-DynamoDB because they relate to DynamoDB streams, i.e., live updates of table items that have changed.

I also didn't pick AmazonDynamoDBFullAccess because it would give my Lambda function to many things with the table. AmazonDynamoDBReadOnlyAccess was the right choice because ut secures our table from unauthorised access from the Lambda

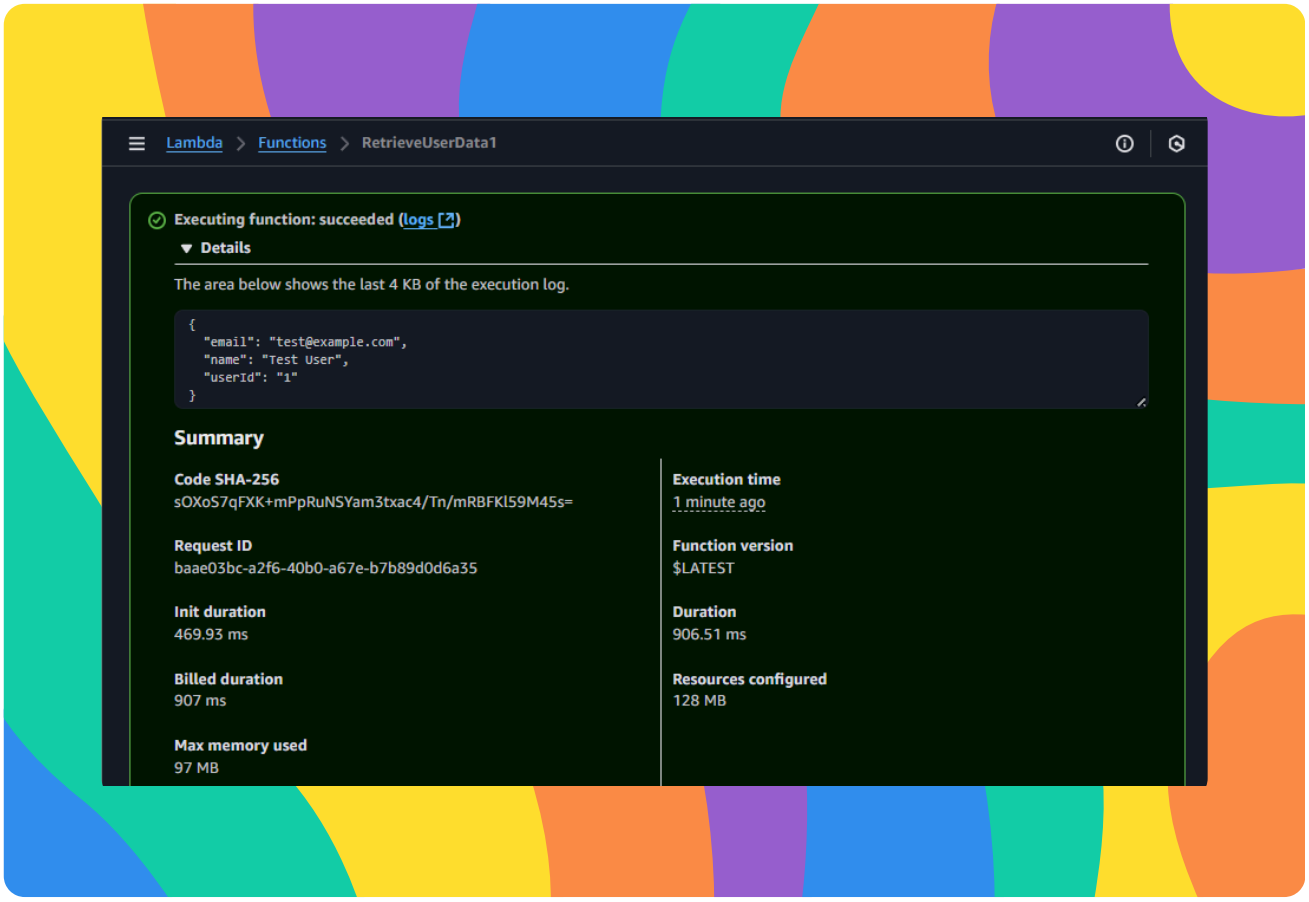


jyothesh karnam
NextWork Student

Final Testing and Reflection

To validate my new permission settings, I retested my Lambda function. The results were a peice of data and three attributes (email, name, userId) because that is the exact data that i had given as an input in my database at the start of this project!

Web apps are a popular use case of using Lambda and DynamoDB. For example, I could use Lambda to retrieve product information, user profiles, content (e.g. blogs or news articles) based on user queries or actions in my web app



jyothesh karnam
NextWork Student

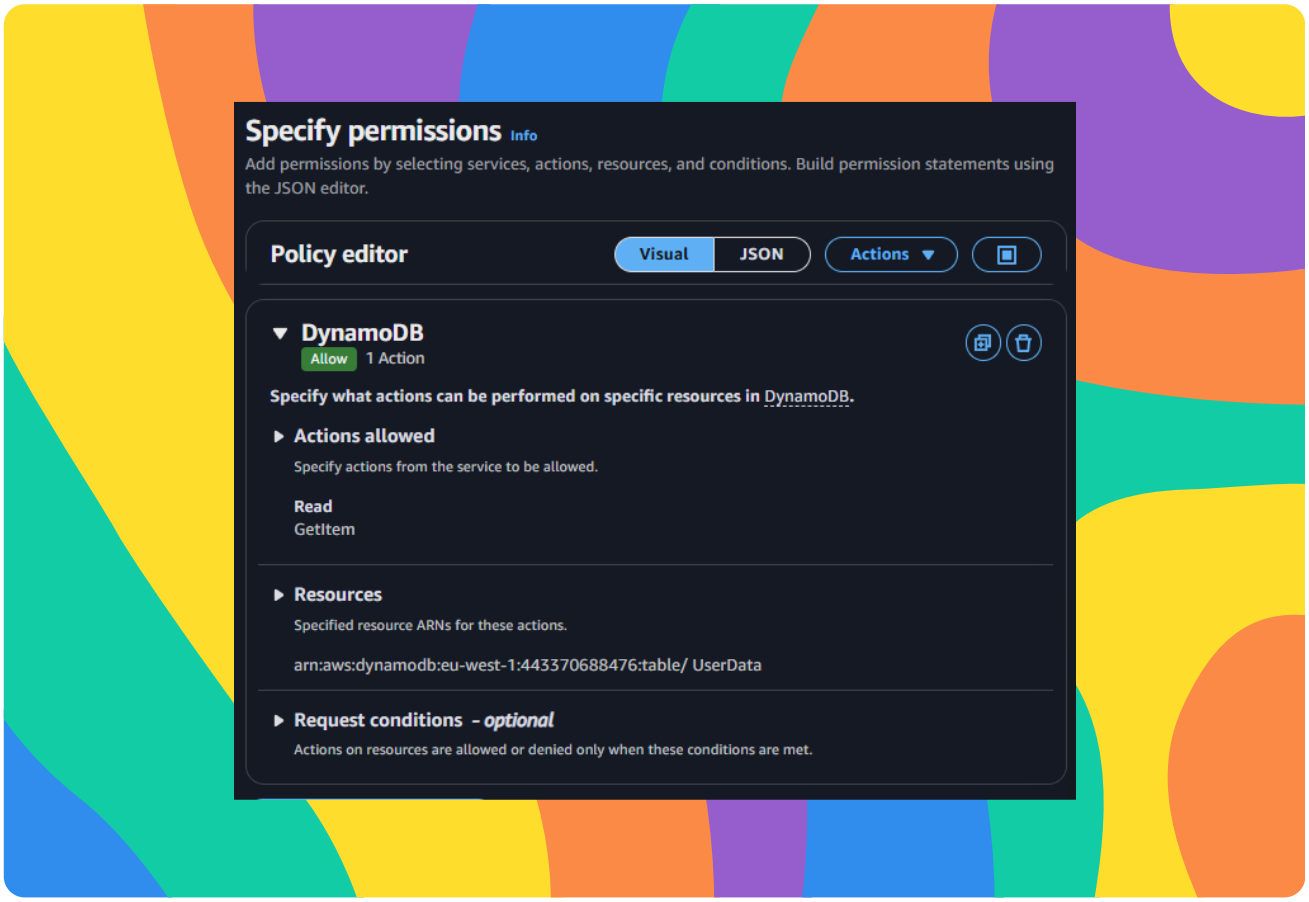
Enahancing Security

For my extension, I challenged myself to replace the permission policy I granted my Lambda function previously. Instead of the AWS managed policy, I'll make sure it can only access UserData. This will enhance DynamoDB's security.

To create the permission policy, I used the create inline policy option and then used the visual editior because this was a more user friendly and guided option for creating our custom policy. This involved checkboxes and dropdowns that we picked, and the policy was written in the background.

When updating a Lambda function's permission policies, I could risk losing access to resources/actions it needs to work. I validated that my Lambda function still works by re-testing it after updating its permission policies – and yes, it still works.

jyothesh karnam
NextWork Student



NextWork.org

Everyone should be in a job they love.

Check out nextwork.org for more projects