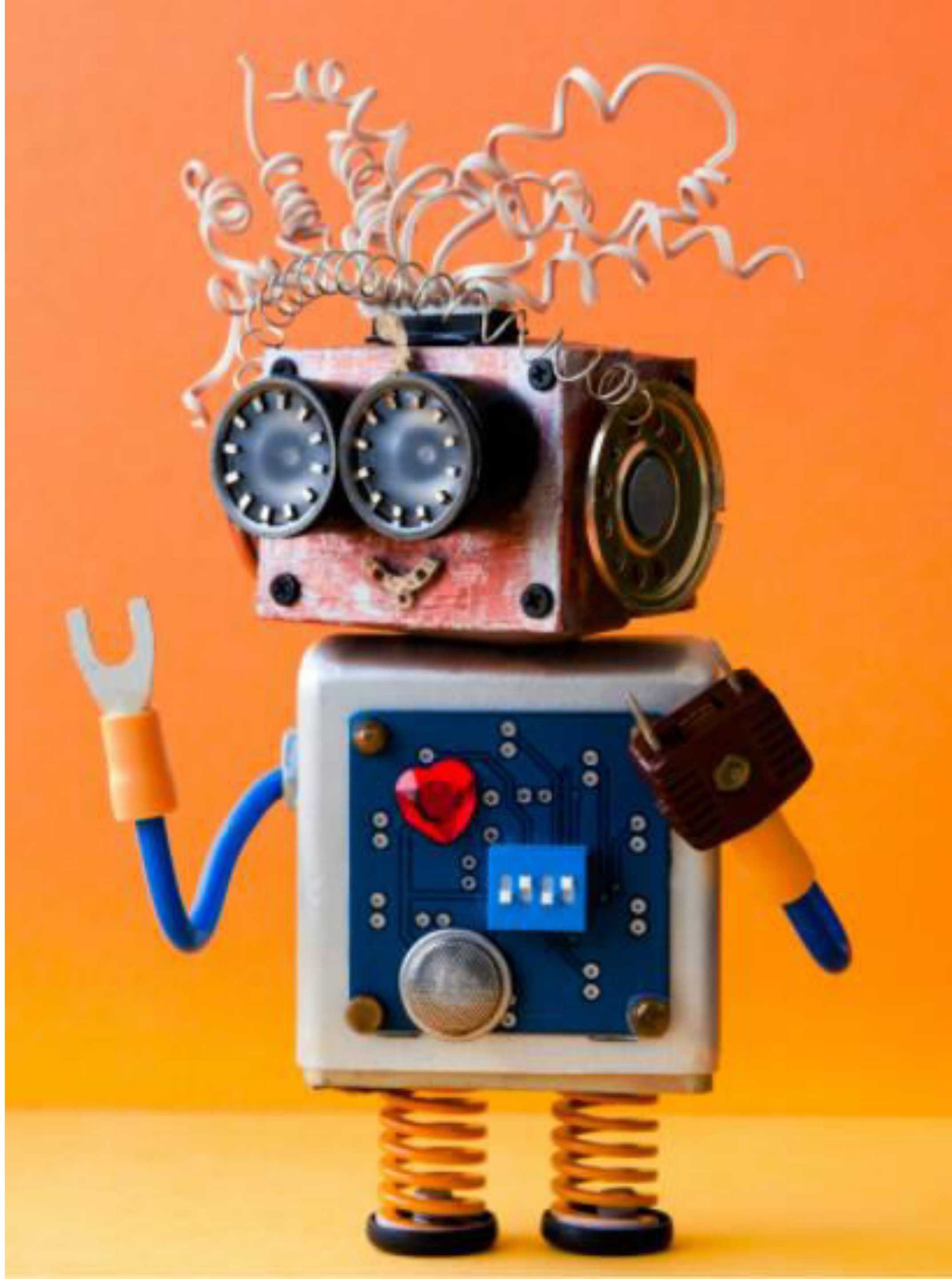
INTRODUCTION

EASY ▲ ADVANCED



INTRODUCTION

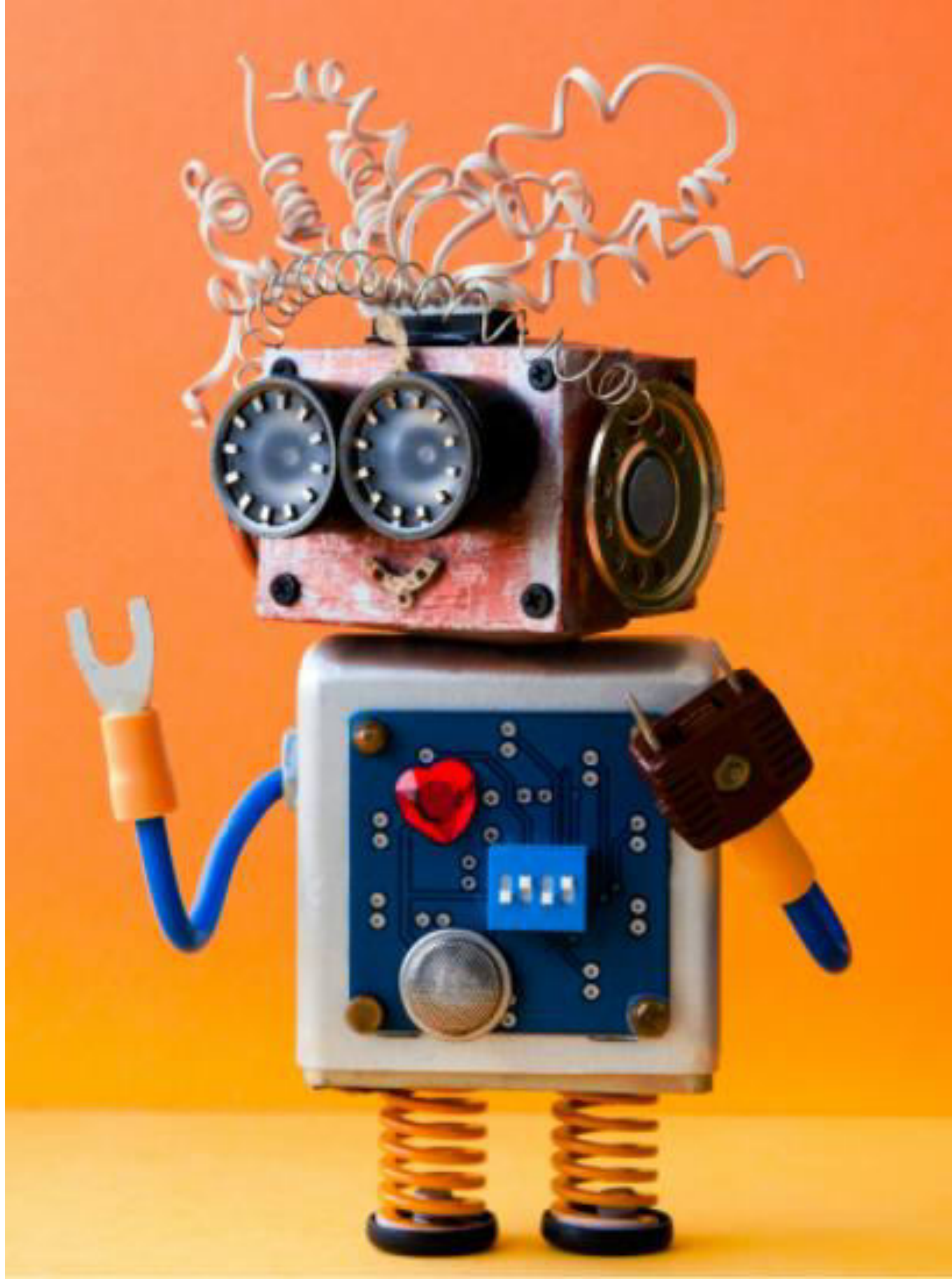
- In this project, we will learn how to define and invoke lambda functions in AWS. Lambda is the most popular and used service in AWS.
- AWS lambda free developers from the worry of provisioning resources, specifying operating systems, managing Hardware, and performing maintenance.
- Simply write your code and run it on Lambda!

KEY LEARNING OUTCOMES

1. Understand Machine Learning workflow automation using AWS Lambda, Step functions and SageMaker Pipelines.
2. Learn how to define a lambda function in AWS management console.
3. Understand the anatomy of Lambda functions.
4. Learn how to configure a test event in Lambda.
5. Monitor Lambda invocations in CloudWatch.

ML WORKFLOWS 101

EASY ADVANCED



ML WORKFLOW AUTOMATION

- So far, we have been manually performing AI/ML model training, testing and deployment.
- Now, we would like to automate the process using Lambda, step functions and SageMaker Pipelines.
- This is critical to improve efficiency, reduce complexity and reduce human error.

AWS Lambda

A **serverless event driven service** that is perfect for **mini tasks** that are **repeated frequently**. AWS lambda empowers anyone to run code without thinking about servers or underlying infrastructure.

AWS Step Functions

allows for creating **serverless workflows** in which the output from a step is fed as an input to the next step.

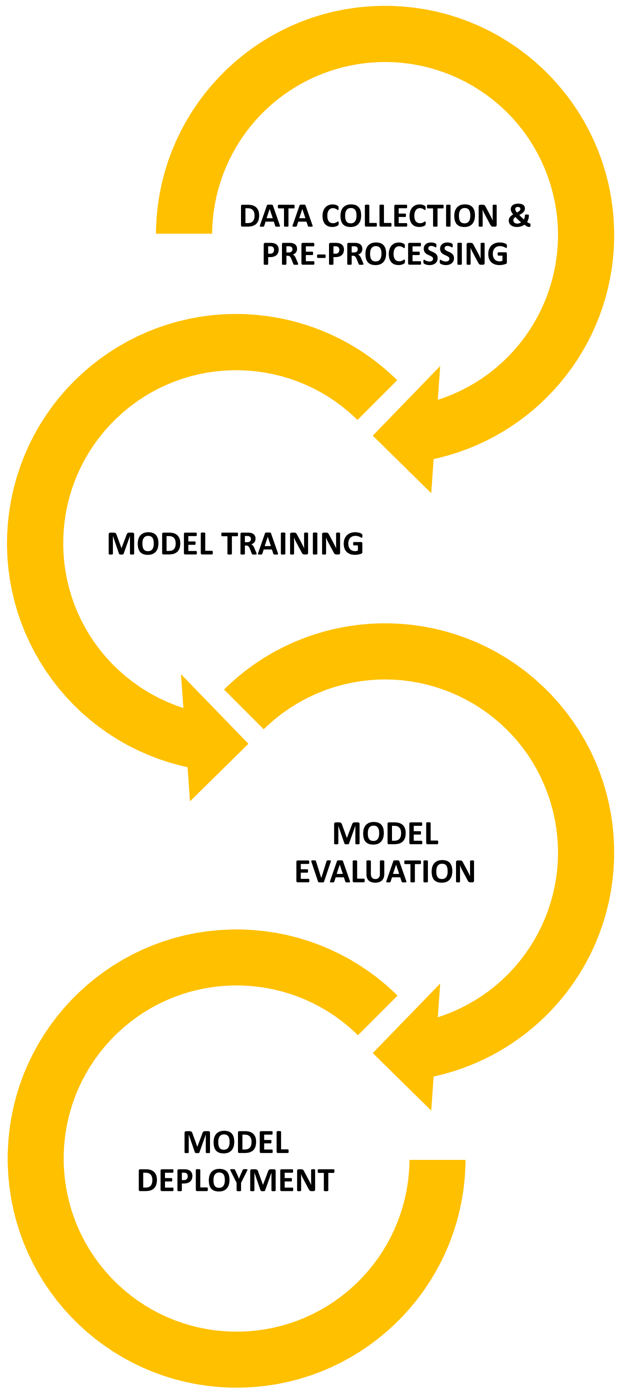
•AWS Step functions converts a workflow into a **state machine diagram** that's easy to debug and understand.

AWS SageMaker Pipelines

Is a specialized continuous integration and continuous delivery (CI/CD) service for machine learning. With SageMaker Pipelines, you can **create, automate, and manage end-to-end ML workflows** at scale.

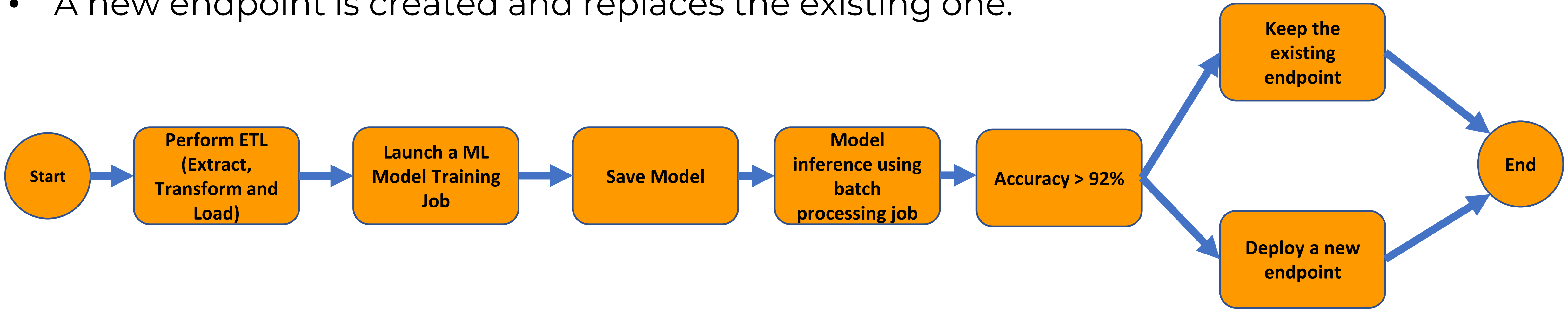
WHAT IS A ML WORKFLOW?

- Machine learning workflows is an orchestrated series of steps that include: (1) data collection and pre-processing, (2) model training and refinement, (3) evaluation, and (4) deployment to production.
- This process needs to be repeated frequently if data becomes available or if a new model is trained and ready for deployment in production.
- The end of each step should kick start the next step! For example, if the trained ML model meets a certain KPI, it should be ready for deployment in production.

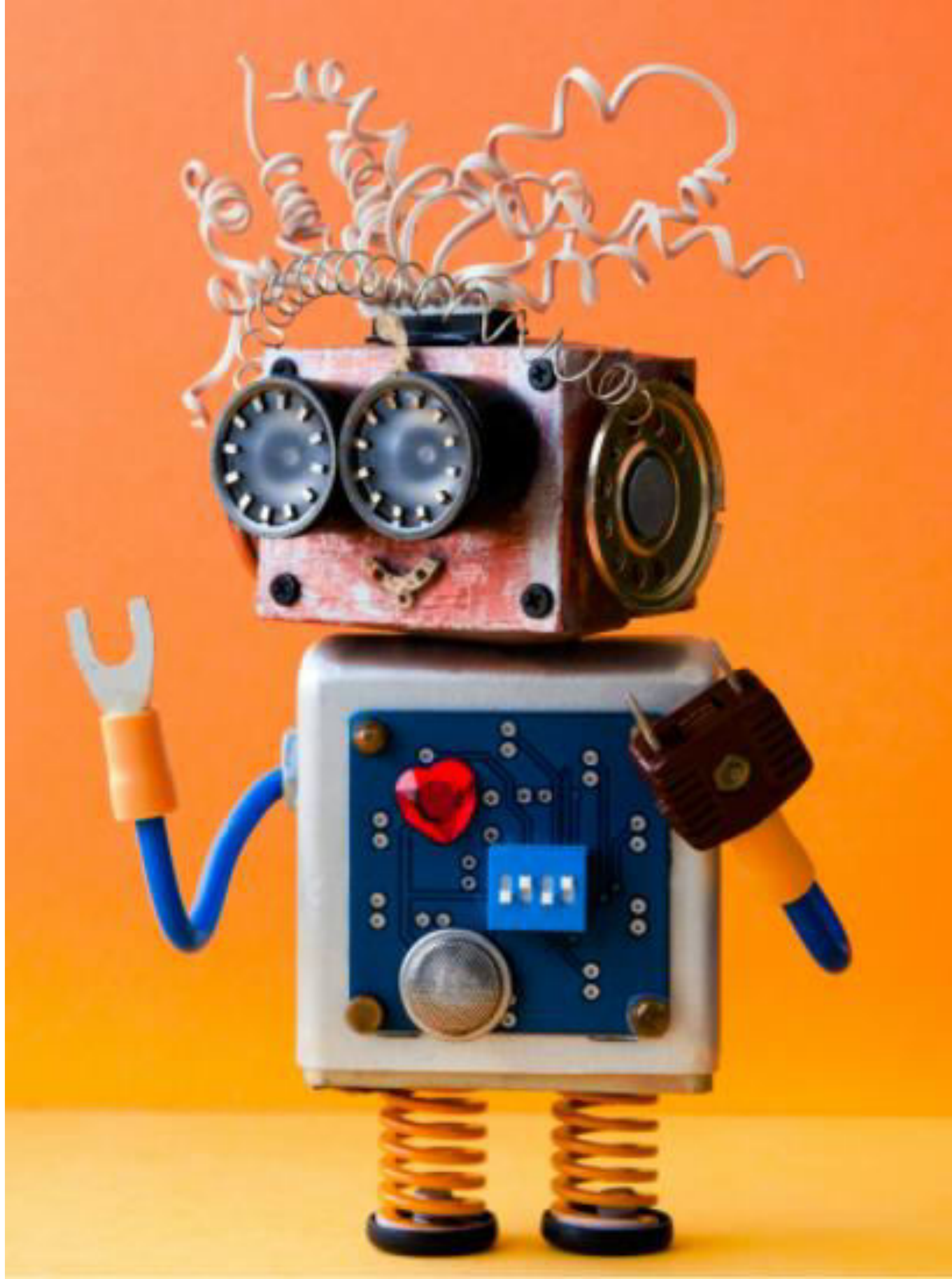


ML WORKFLOW EXAMPLE

- This is an example of an automated ML workflow.
- If data becomes available, an ETL and training jobs are launched.
- Trained model is evaluated and if accuracy exceeds 92%, model is deployed in production.
- A new endpoint is created and replaces the existing one.


Start
Perform ETL (Extract, Transform and Load)
Launch a ML Model Training Job
Save Model
Model inference using batch processing job
Accuracy > 92%
Keep the existing endpoint
Deploy a new endpoint
End

WHAT IS AWS LAMBDA?



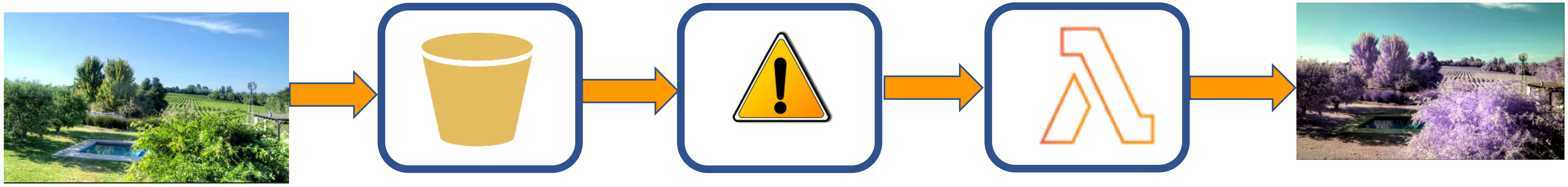
AWS LAMBDA 101

- AWS lambda free developers from the worry of provisioning resources, specifying operating systems, managing Hardware, and performing maintenance.
- Simply write your code and run it on Lambda! You only pay per milliseconds for the compute power so no need to provision infrastructure upfront.
- AWS lambda is one of the most frequently used services in AWS.
- Link to documentation: https://aws.amazon.com/lambda/
- Serverless applications offer the following benefits:

  1. No infrastructure management required
  2. Autoscaling based on demand
  3. Pay for value – due to autoscaling
  4. High Security – AWS manages this on your behalf with shared responsibility model
  5. High availability

AWS LAMBDA 101

- AWS lambda empowers anyone to run code without thinking about servers or underlying infrastructure (you don't have to specify ***t2.medium*** as an example anymore).
- AWS Lambda is **serverless event driven** service that is perfect for mini tasks that are **repeated frequently**.
- AWS Lambda works as an **orchestrator between multiple decoupled AWS Services**.



User took a photo and uploaded it

S3 BUCKET

Photo is uploaded and stored in Amazon S3 Bucket

LAMBDA TRIGGER

Lambda function is triggered once the photo is uploaded to S3

LAMBDA FUNCTION

AWS Lambda consumes photo and runs code to apply a filter to the image

FILTERED PHOTO DISPLAY

Filtered photo is displayed into the web application

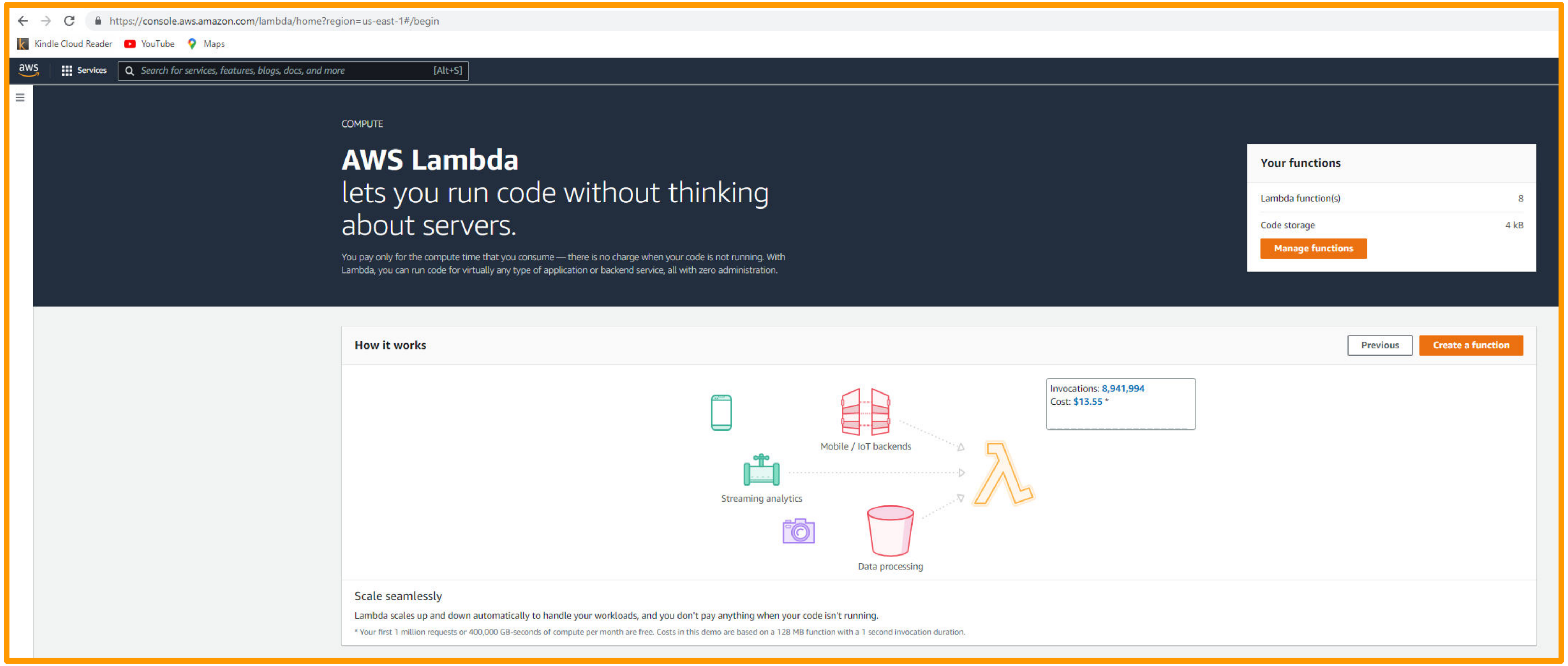
Photo Credit: https://www.flickr.com/photos/trophygeek/20113975429/in/photostream/

AWS LAMBDA 101

- AWS Lambda function could be written in many programming languages such as Python, Java, C#, Go, Ruby, and Node.js
- Lambda functions are event driven, they are triggered when an event takes place such as an object that is uploaded to S3 or request to a given endpoint.
- Lambda functions can be used to invoke other AWS services.
- Note that with AWS Free Tier, you have **1 Million Lambda triggers for free**!
- AWS Pricing is powerful and efficient:
    1. Compute time is charged every 100ms increments
    2. No hourly minimums
    3. No payments for idle resources

AWS LAMBDA 101: DEMO

https://console.aws.amazon.com/lambda/home?region=us-east-1#/begin


https://console.aws.amazon.com/lambda/home?region=us-east-1#/begin

Kindle Cloud Reader YouTube Maps

aws Services Search for services, features, blogs, docs, and more [Alt+S]

COMPUTE

AWS Lambda
lets you run code without thinking about servers.

You pay only for the compute time that you consume — there is no charge when your code is not running. With Lambda, you can run code for virtually any type of application or backend service, all with zero administration.

Your functions

Lambda function(s)	8
Code storage	4 kB

Manage functions

How it works

Previous Create a function

Invocations: 8,941,994
Cost: $13.55 *

Scale seamlessly

Lambda scales up and down automatically to handle your workloads, and you don't pay anything when your code isn't running.

* Your first 1 million requests or 400,000 GB-seconds of compute per month are free. Costs in this demo are based on a 128 MB function with a 1 second invocation duration.

AWS LAMBDA 101: BENEFITS

Enhanced Agility

Reduced Costs

Fast Time to Market

No hassle in Provisioning HW and Managing SW

Allows Developers to Focus on the product

AWS LAMBDA 101: WHAT DOES LAMBDA HANDLE?

Load Balancing by routing requests to available instances

Autoscaling

Failures and retry

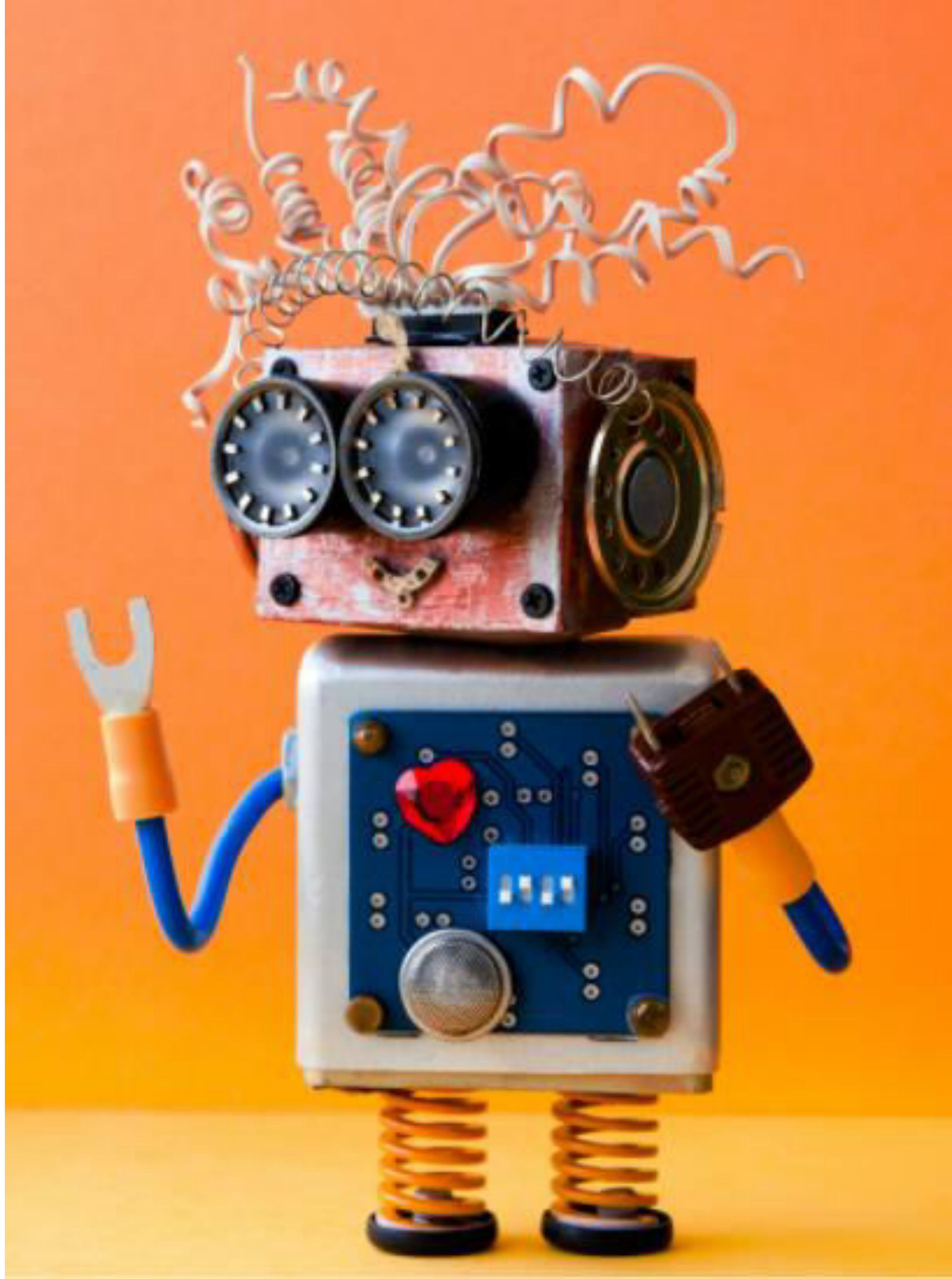
Security and isolation

Operating System Management

Utilization and Billing

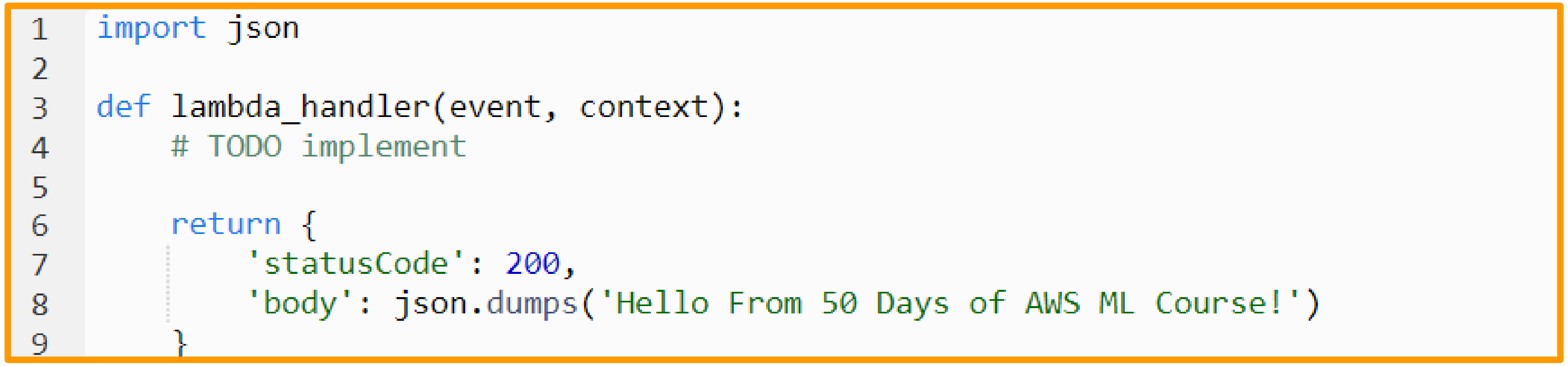
AWS LAMBDA FUNCTION ANATOMY

EASY ▲ ADVANCED



AWS LAMBDA FUNCTION ANATOMY

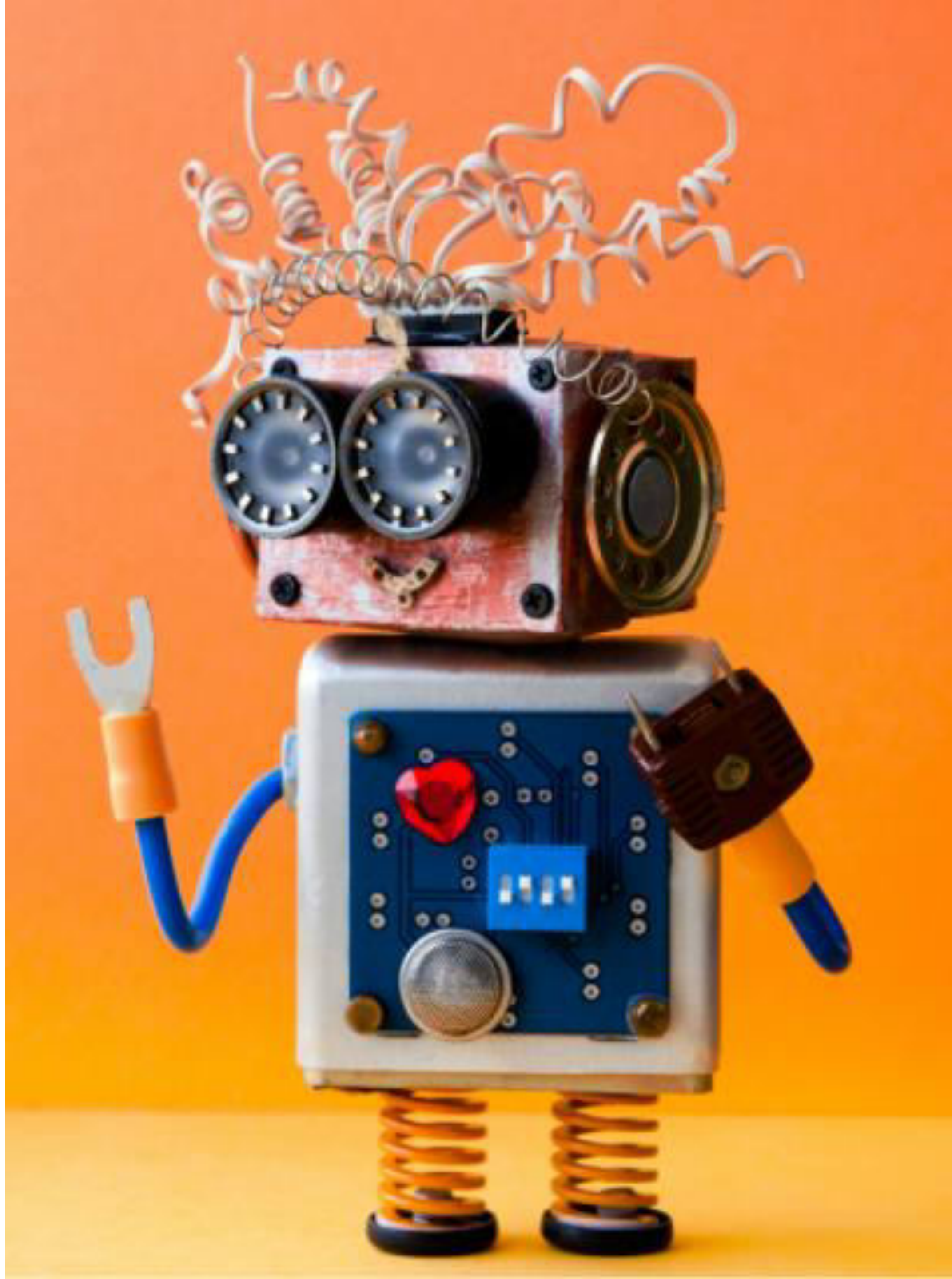
- **Handler() Function:** Function to be executed upon invocation and it requires two arguments “event” and “context”.
- **Event Object:** data sent during lambda function invocation, for example if a request is made from S3, the event object will contain the bucket key and what kind of action has been performed on the bucket.
- **Context object:** this is generated by the platform and contains information about the underlying infrastructure and execution environment such as allowed runtime and memory.



```python
import json

def lambda_handler(event, context):
    # TODO implement

    return {
        'statusCode': 200,
        'body': json.dumps('Hello From 50 Days of AWS ML Course!')
    }
```

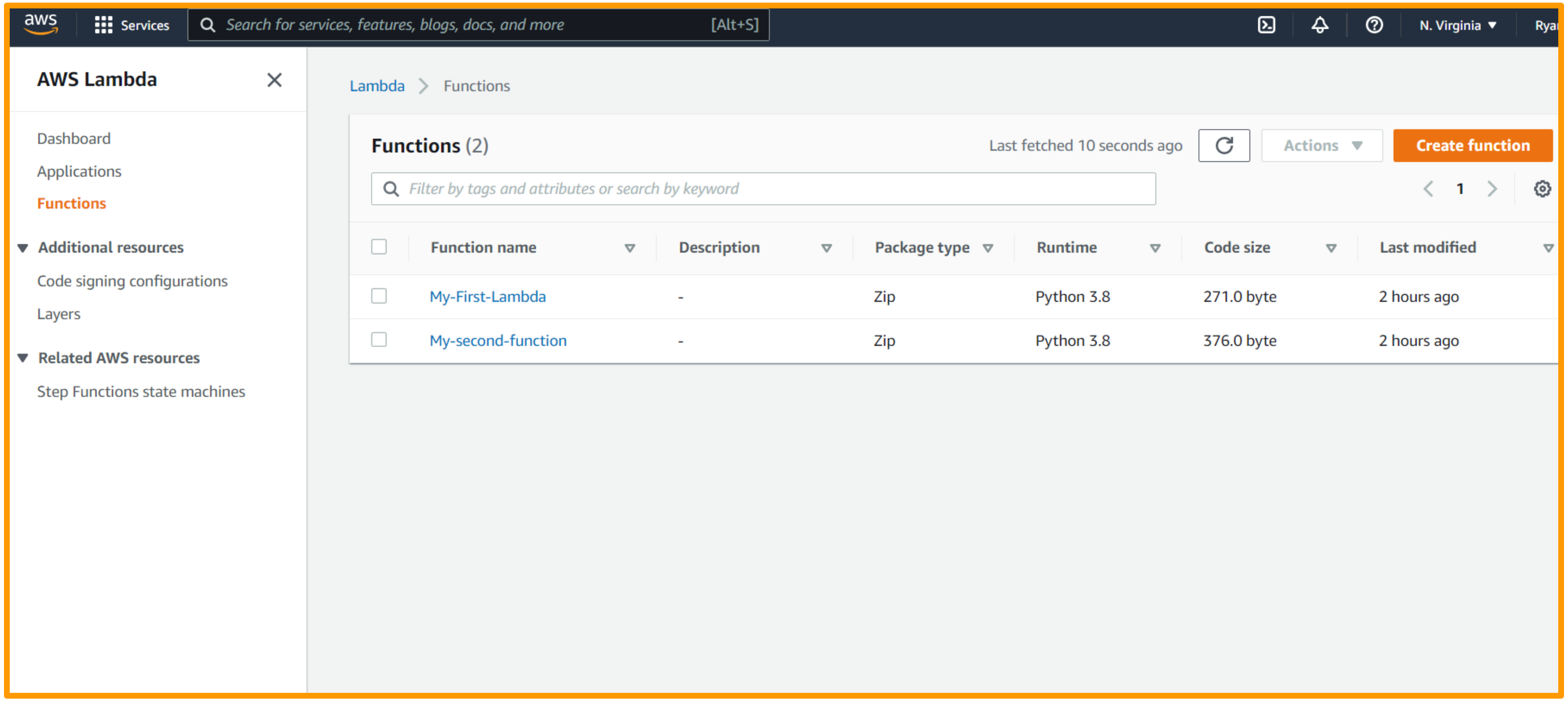
DEMO: DEFINE AN AWS LAMBDA FUNCTION USING CONSOLE – EXAMPLE 1

EASY ADVANCED



DEMO: CREATE AN AWS LAMBDA FUNCTION USING CONSOLE

GO TO AWS LAMBDA AND CLICK ON CREATE FUNCTION

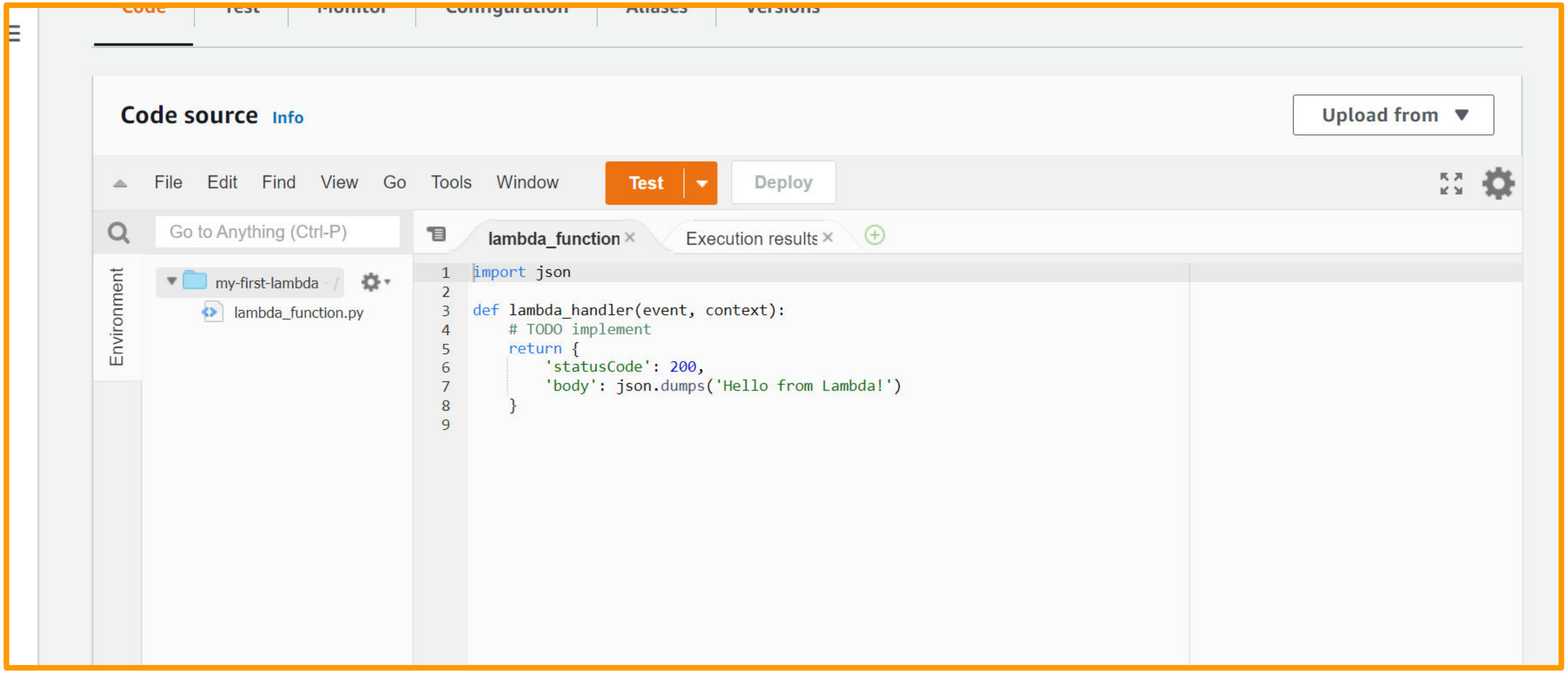

aws
Services
Search for services, features, blogs, docs, and more
[Alt+S]
N. Virginia
Rya
AWS Lambda
Dashboard
Applications
Functions
Additional resources
Code signing configurations
Layers
Related AWS resources
Step Functions state machines
Lambda > Functions
Last fetched 10 seconds ago
Actions
Create function
Filter by tags and attributes or search by keyword
1

Functions (2)

Function name	Description	Package type	Runtime	Code size	Last modified
My-First-Lambda	-	Zip	Python 3.8	271.0 byte	2 hours ago
My-second-function	-	Zip	Python 3.8	376.0 byte	2 hours ago

DEMO: CREATE AN AWS LAMBDA FUNCTION USING CONSOLE

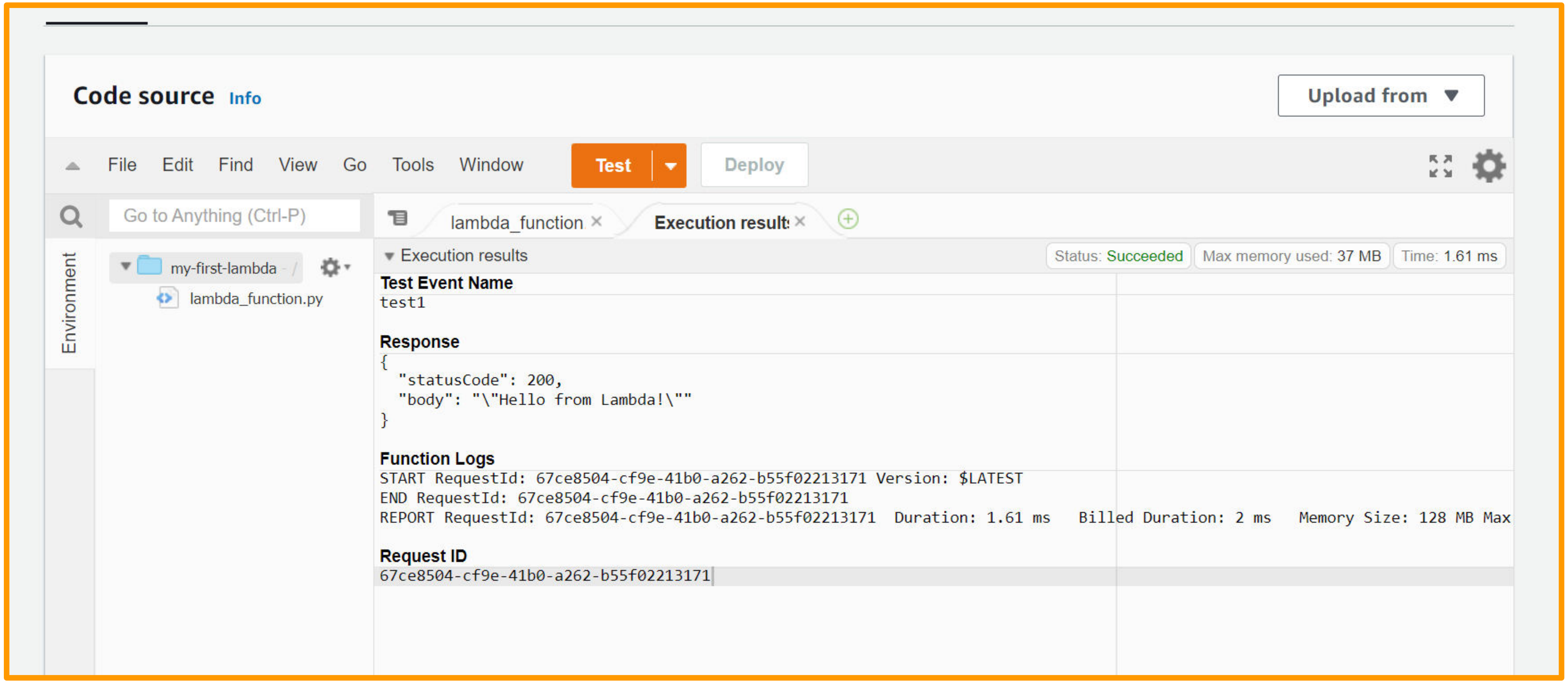
THIS FUNCTION RETURNS STATUS CODE AND “HELLO FROM LAMBDA”



```python
import json

def lambda_handler(event, context):
    # TODO implement
    return {
        'statusCode': 200,
        'body': json.dumps('Hello from Lambda!')
    }

```

DEMO: CREATE AN AWS LAMBDA FUNCTION USING CONSOLE

CONFIGURE A TEST EVENT BELOW. A TEST EVENT IS A JSON OBJECT THAT MOCKS THE STRUCTURE OF REQUESTS EMITTED BY AWS SERVICES TO INVOKE A LAMBDA FUNCTION.



Execution results   Status: Succeeded   Max memory used: 37 MB   Time: 1.61 ms

Test Event Name

```
test1
```

Response

```
{
  "statusCode": 200,
  "body": "\"Hello from Lambda!\""
}
```

Function Logs

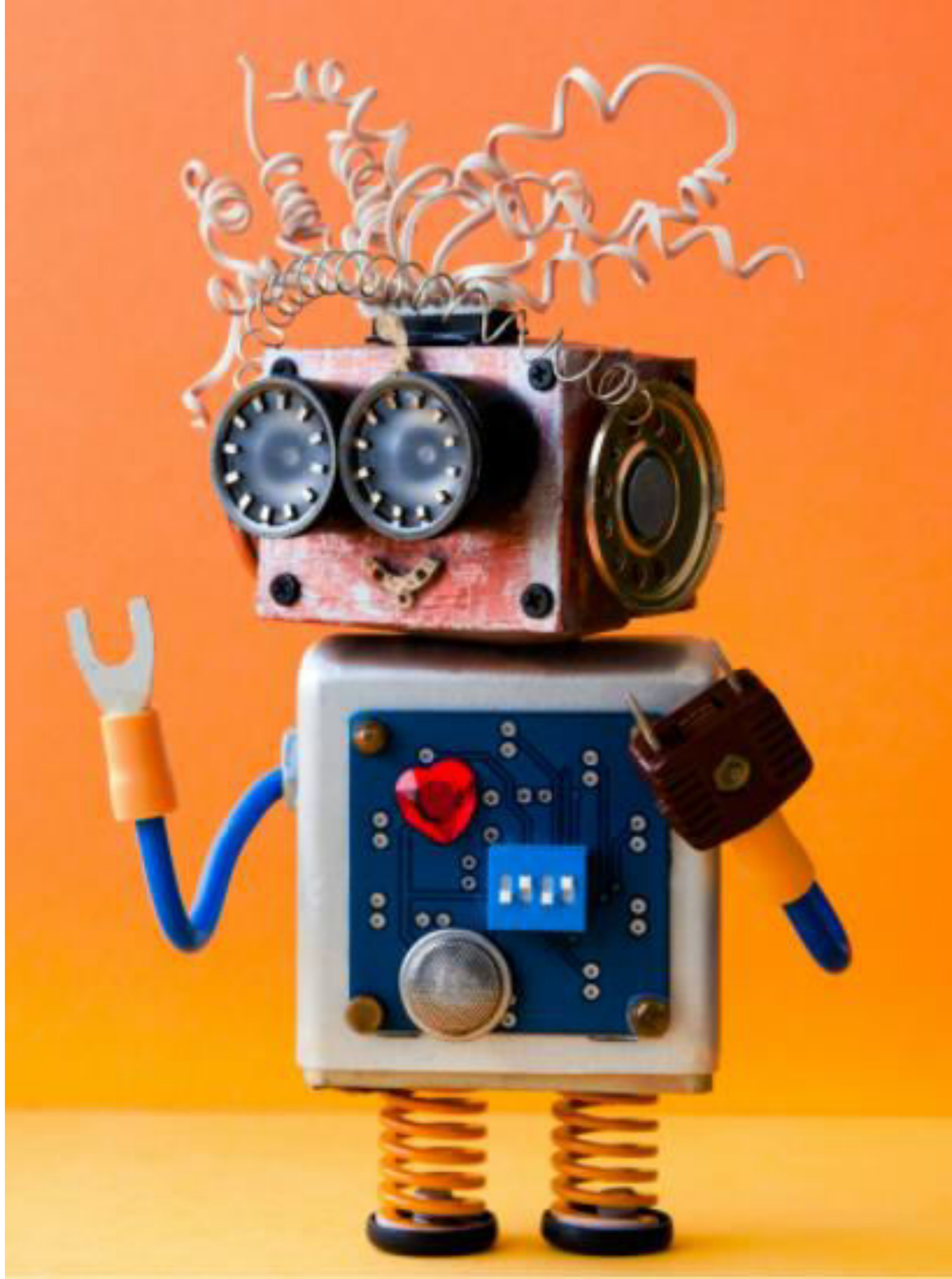
```
START RequestId: 67ce8504-cf9e-41b0-a262-b55f02213171 Version: $LATEST
END RequestId: 67ce8504-cf9e-41b0-a262-b55f02213171
REPORT RequestId: 67ce8504-cf9e-41b0-a262-b55f02213171  Duration: 1.61 ms   Billed Duration: 2 ms   Memory Size: 128 MB Max
```

Request ID

```
67ce8504-cf9e-41b0-a262-b55f02213171
```

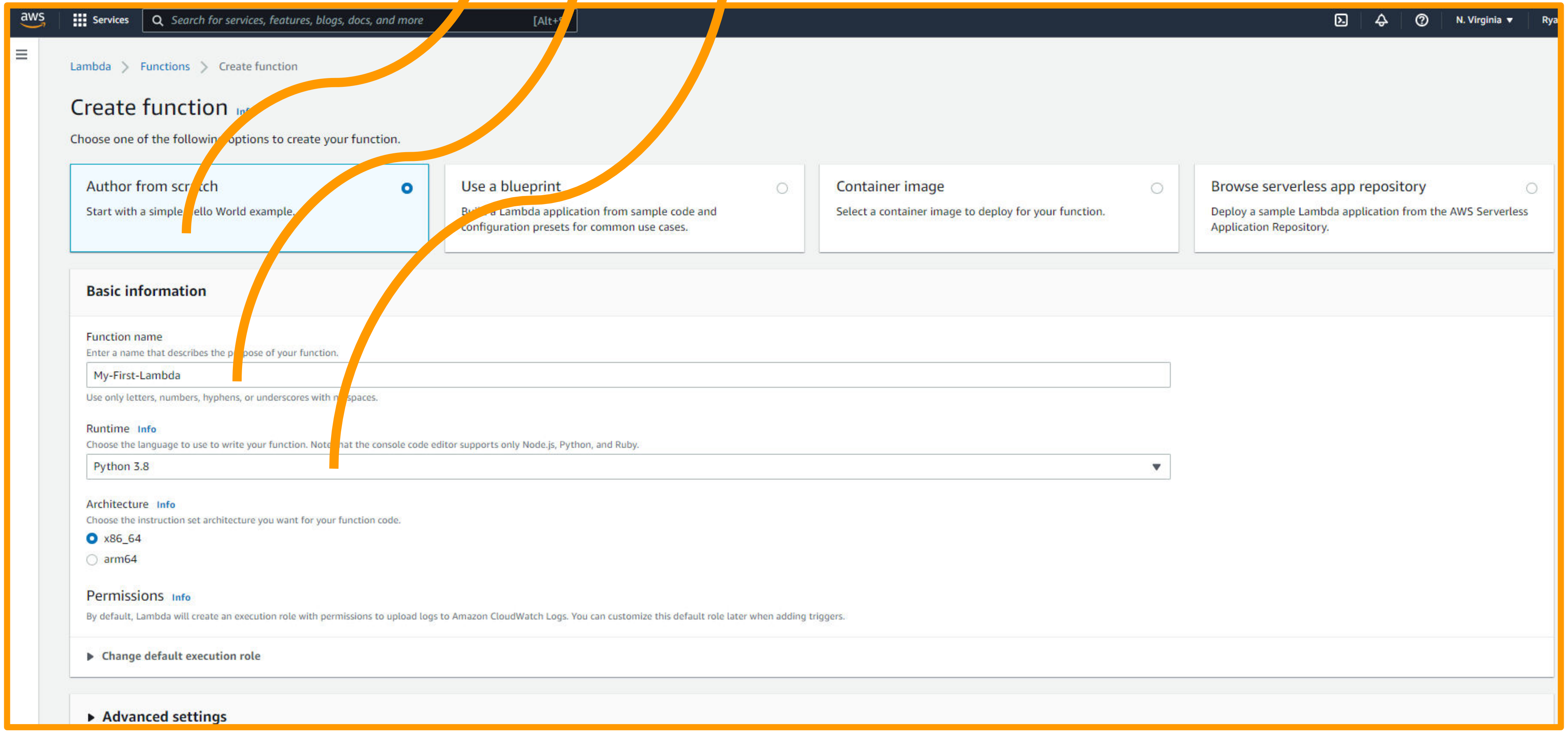
DEMO EXAMPLE 2: DEFINE AN AWS LAMBDA FUNCTION USING CONSOLE

EASY ▲ ADVANCED



DEMO: CREATE AN AWS LAMBDA FUNCTION USING CONSOLE

SELECT AUTHOR FROM SCRATCH, GIVE A UNIQUE LAMBDA FUNCTION NAME, SELECT A RUNTIME AND CLICK CREATE FUNCTION


aws
Services
Search for services, features, blogs, docs, and more
[Alt+S]
N. Virginia ▾
Rya
Lambda > Functions > Create function

Create function Info

Choose one of the following options to create your function.

[✓ Author from scratch] Start with a simple Hello World example.

Use a blueprint — Build a Lambda application from sample code and configuration presets for common use cases.

Container image — Select a container image to deploy for your function.

Browse serverless app repository — Deploy a sample Lambda application from the AWS Serverless Application Repository.

Basic information

Function name
Enter a name that describes the purpose of your function.
My-First-Lambda
Use only letters, numbers, hyphens, or underscores with no spaces.

Runtime Info
Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.
Python 3.8

Architecture Info
Choose the instruction set architecture you want for your function code.
[✓ x86_64]
arm64

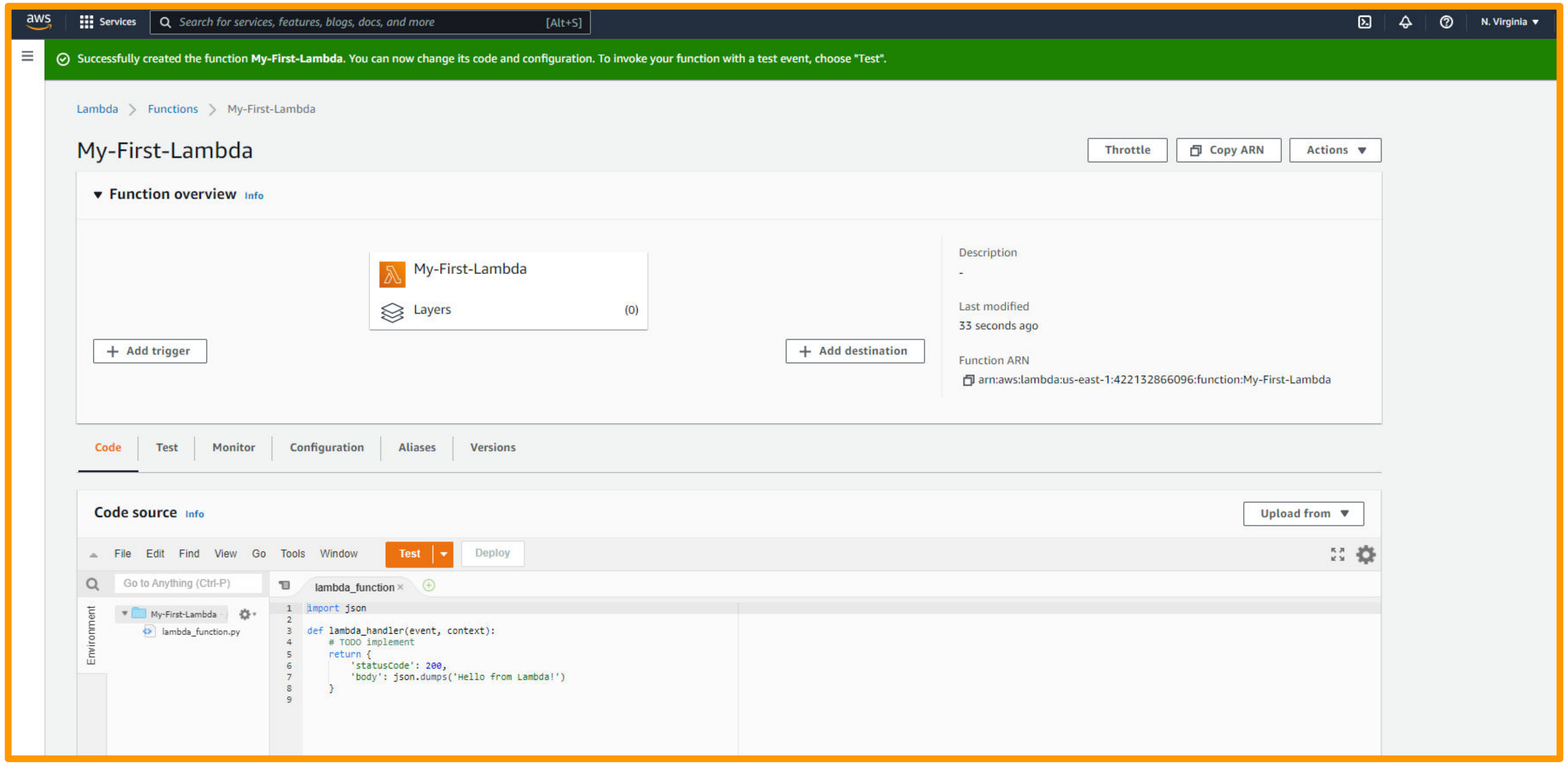
Permissions Info
By default, Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs. You can customize this default role later when adding triggers.

▸ **Change default execution role**

▸ **Advanced settings**

DEMO: CREATE AN AWS LAMBDA FUNCTION USING CONSOLE

WELCOME TO YOUR FIRST LAMBDA FUNCTION!
YOU CAN SEE THE CODE SOURCE BELOW



Successfully created the function **My-First-Lambda**. You can now change its code and configuration. To invoke your function with a test event, choose "Test".

Lambda > Functions > My-First-Lambda

My-First-Lambda

Description
-

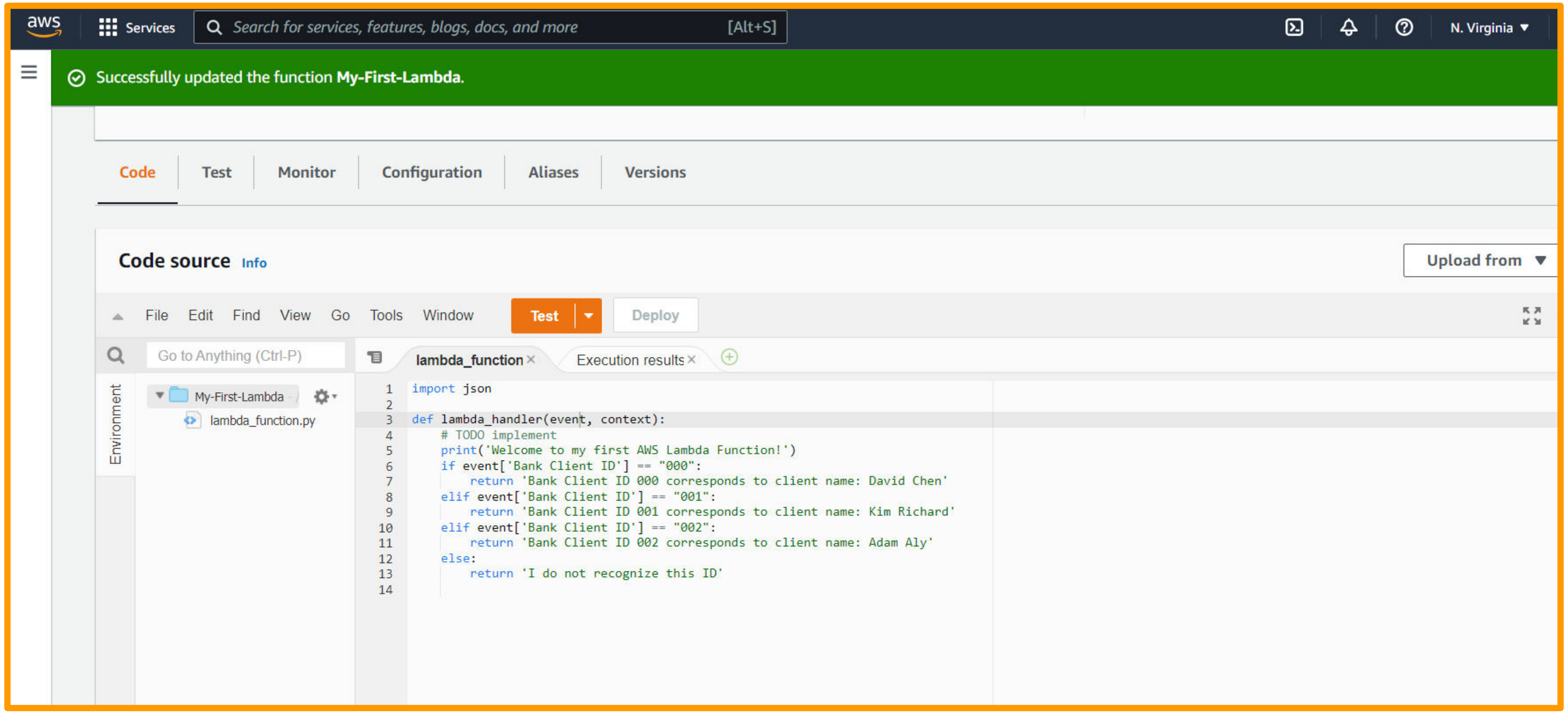
Last modified
33 seconds ago

Function ARN
arn:aws:lambda:us-east-1:422132866096:function:My-First-Lambda

```python
import json

def lambda_handler(event, context):
    # TODO implement
    return {
        'statusCode': 200,
        'body': json.dumps('Hello from Lambda!')
    }

```

DEMO: CREATE AN AWS LAMBDA FUNCTION USING CONSOLE

WRITE THIS FUNCTION BELOW AND CLICK DEPLOY

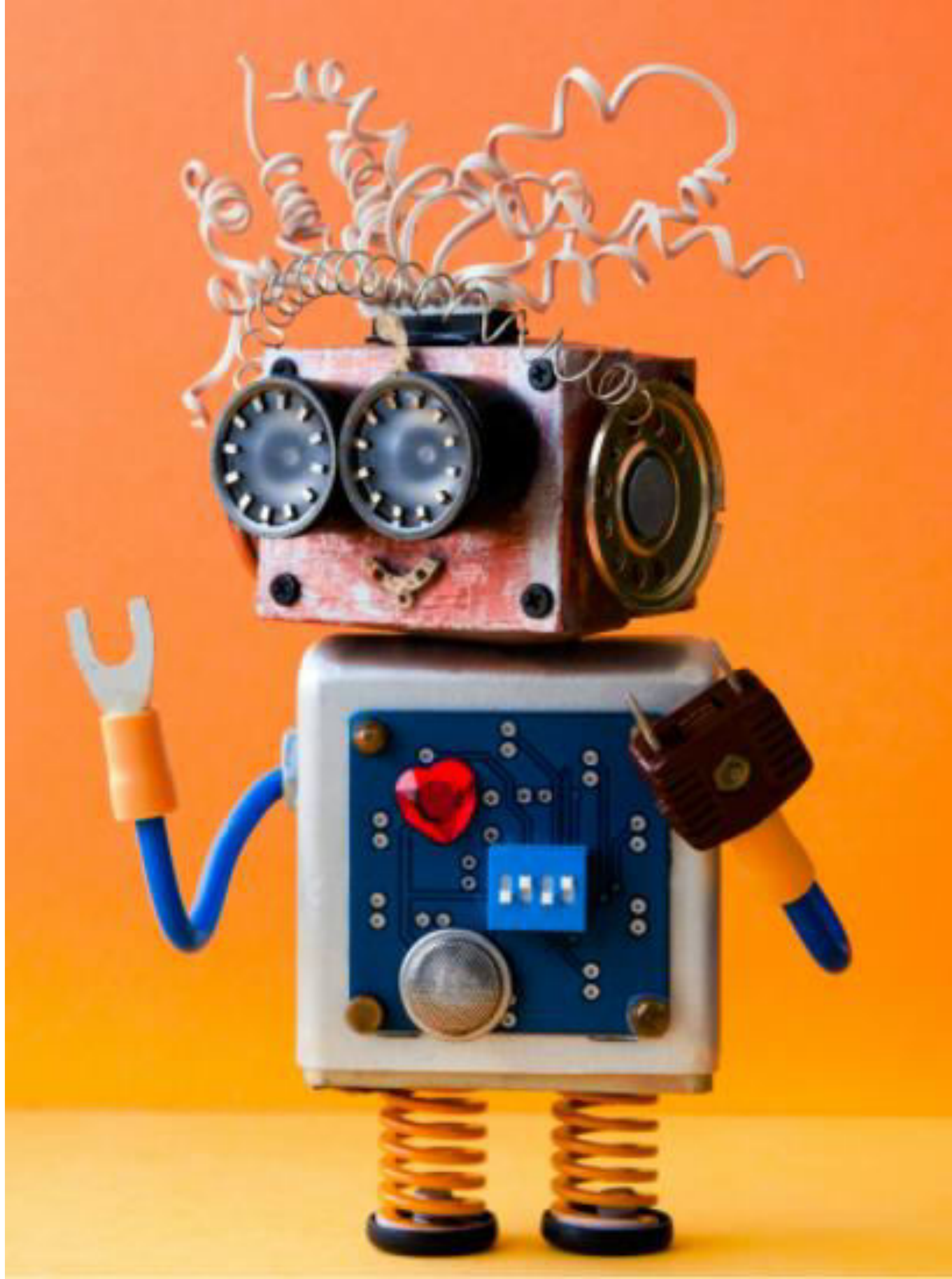


Successfully updated the function **My-First-Lambda**.

```
import json

def lambda_handler(event, context):
    # TODO implement
    print('Welcome to my first AWS Lambda Function!')
    if event['Bank Client ID'] == "000":
        return 'Bank Client ID 000 corresponds to client name: David Chen'
    elif event['Bank Client ID'] == "001":
        return 'Bank Client ID 001 corresponds to client name: Kim Richard'
    elif event['Bank Client ID'] == "002":
        return 'Bank Client ID 002 corresponds to client name: Adam Aly'
    else:
        return 'I do not recognize this ID'
```

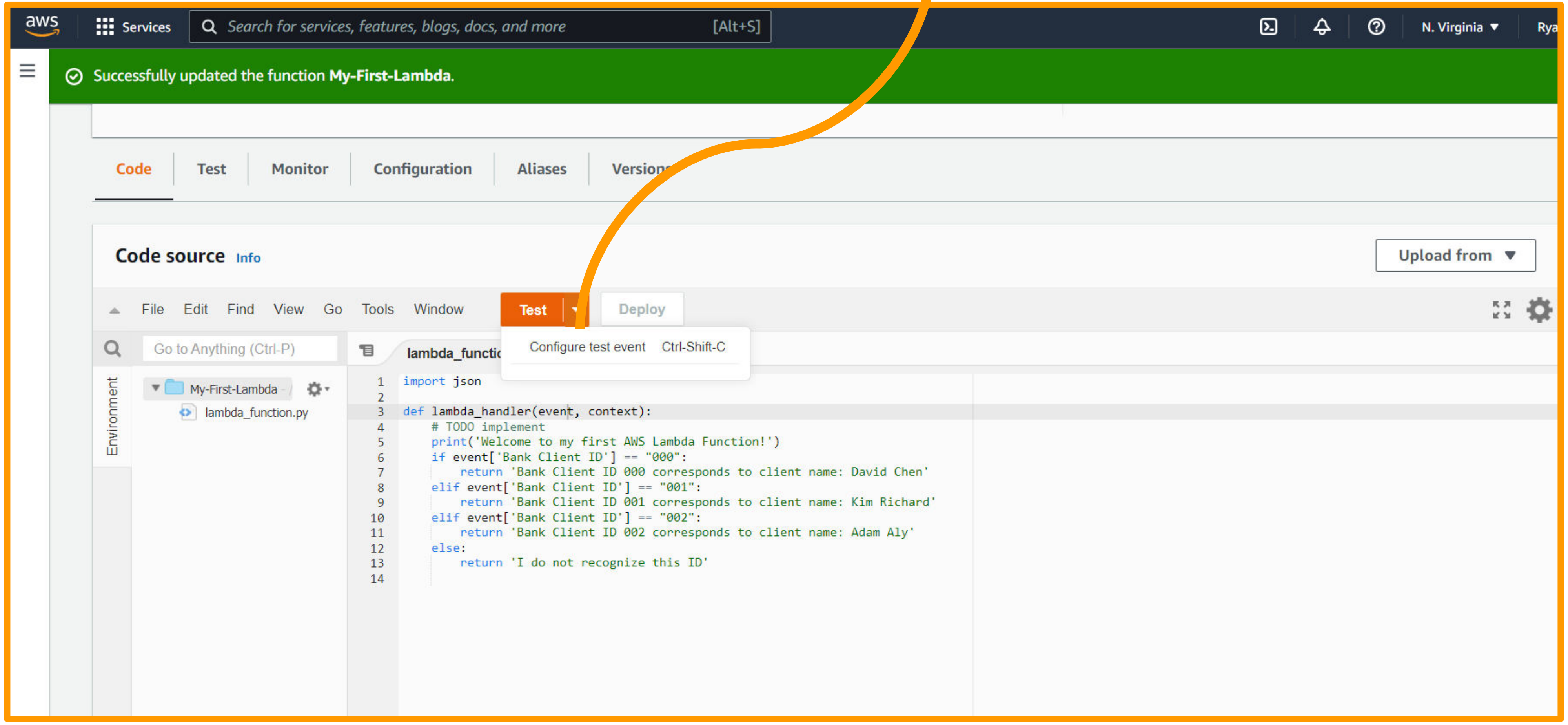
DEMO EXAMPLE 2: TEST AN AWS LAMBDA FUNCTION USING CONSOLE

EASY ADVANCED



DEMO: CREATE AN AWS LAMBDA FUNCTION USING CONSOLE

LET'S TEST THIS FUNCTION, CLICK ON "CONFIGURE TEST EVENT"

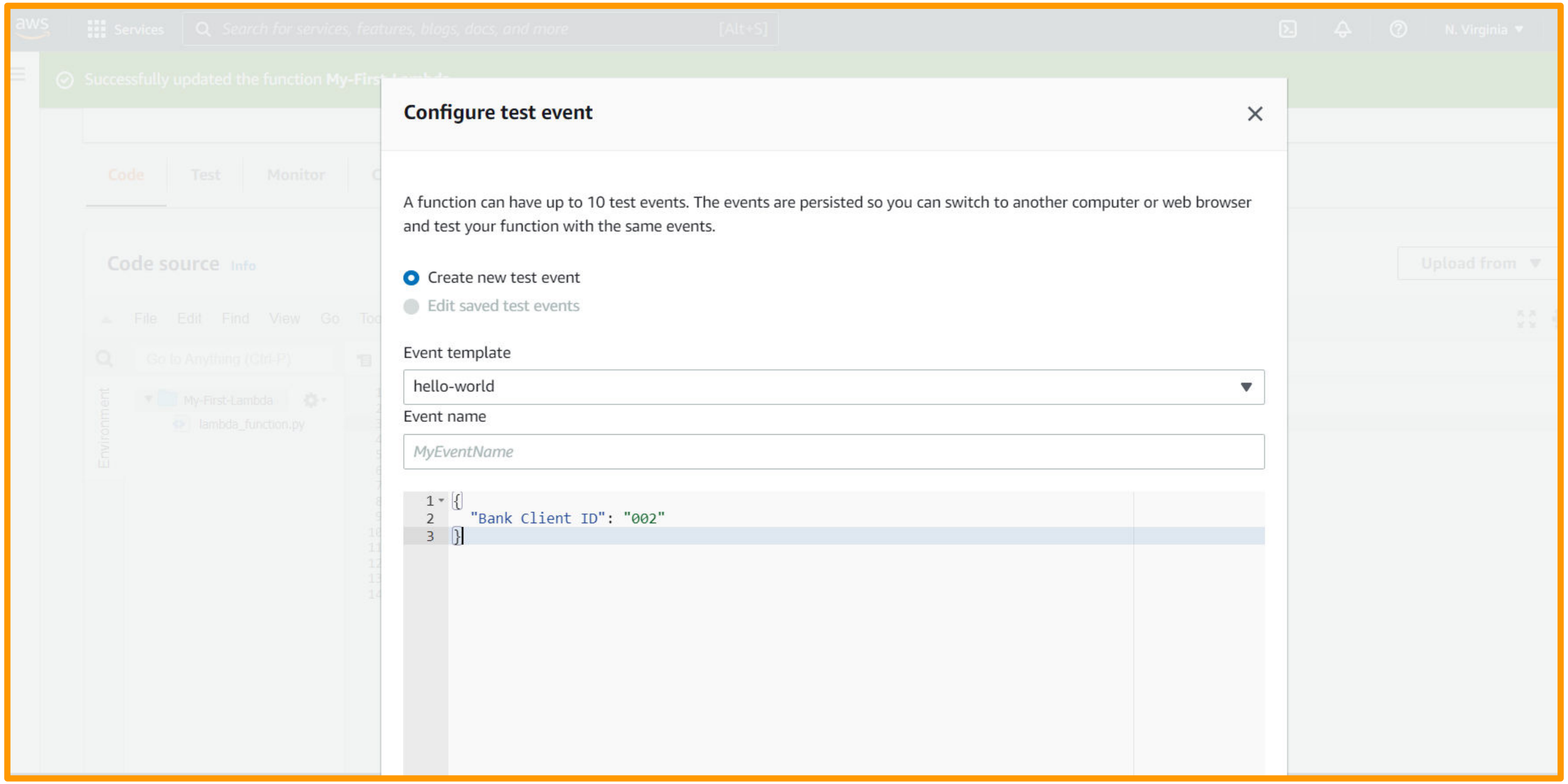


Successfully updated the function **My-First-Lambda**.

```
import json

def lambda_handler(event, context):
    # TODO implement
    print('Welcome to my first AWS Lambda Function!')
    if event['Bank Client ID'] == "000":
        return 'Bank Client ID 000 corresponds to client name: David Chen'
    elif event['Bank Client ID'] == "001":
        return 'Bank Client ID 001 corresponds to client name: Kim Richard'
    elif event['Bank Client ID'] == "002":
        return 'Bank Client ID 002 corresponds to client name: Adam Aly'
    else:
        return 'I do not recognize this ID'
```

DEMO: CREATE AN AWS LAMBDA FUNCTION USING CONSOLE

PROVIDE A NAME TO THE EVENT AND CLICK “CREATE”



Successfully updated the function My-First-Lambda

Configure test event

A function can have up to 10 test events. The events are persisted so you can switch to another computer or web browser and test your function with the same events.

(● Create new test event) Edit saved test events

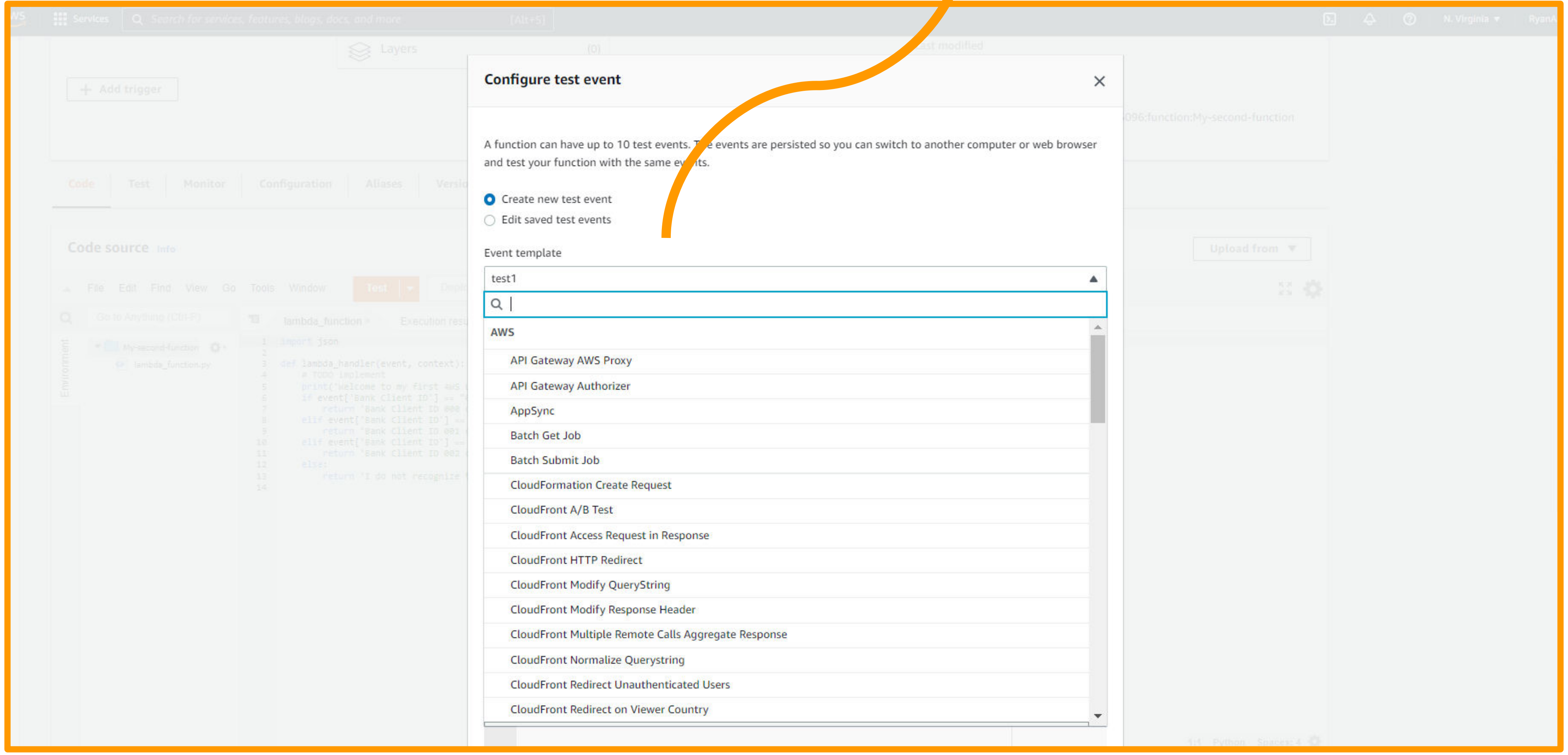
Event template: hello-world

Event name: MyEventName

```
{
  "Bank Client ID": "002"
}
```

DEMO: CREATE AN AWS LAMBDA FUNCTION USING CONSOLE

NOTE THAT YOU CAN SELECT SEVERAL TEST EVENT FROM THIS MENU



Configure test event

A function can have up to 10 test events. The events are persisted so you can switch to another computer or web browser and test your function with the same events.

(● Create new test event) ○ Edit saved test events

Event template

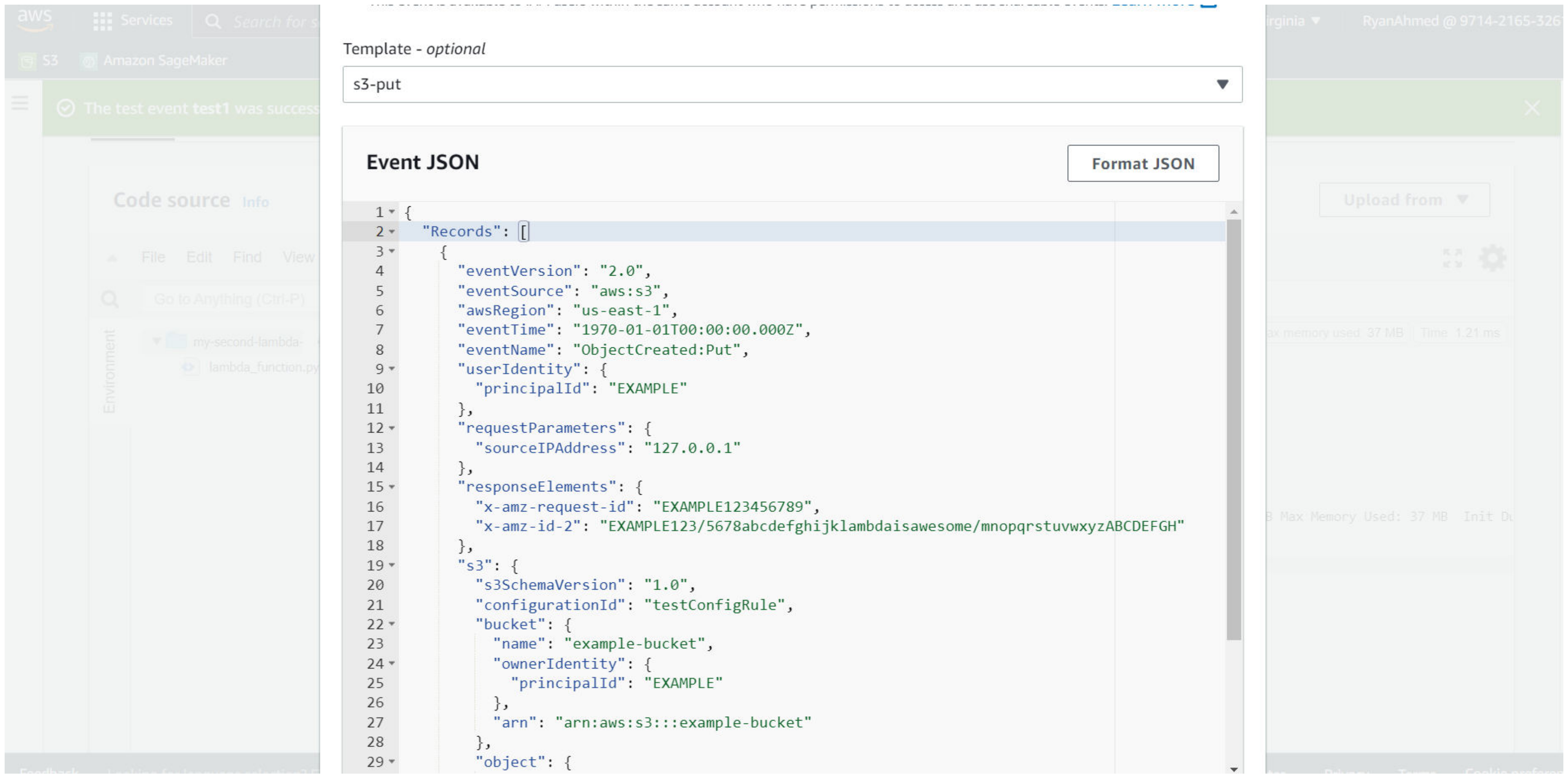
test1

AWS

- API Gateway AWS Proxy
- API Gateway Authorizer
- AppSync
- Batch Get Job
- Batch Submit Job
- CloudFormation Create Request
- CloudFront A/B Test
- CloudFront Access Request in Response
- CloudFront HTTP Redirect
- CloudFront Modify QueryString
- CloudFront Modify Response Header
- CloudFront Multiple Remote Calls Aggregate Response
- CloudFront Normalize Querystring
- CloudFront Redirect Unauthenticated Users
- CloudFront Redirect on Viewer Country

DEMO: CREATE AN AWS LAMBDA FUNCTION USING CONSOLE

EXPLORE THE S3-PUT EVENT FOR EXAMPLE



Template - *optional*

s3-put

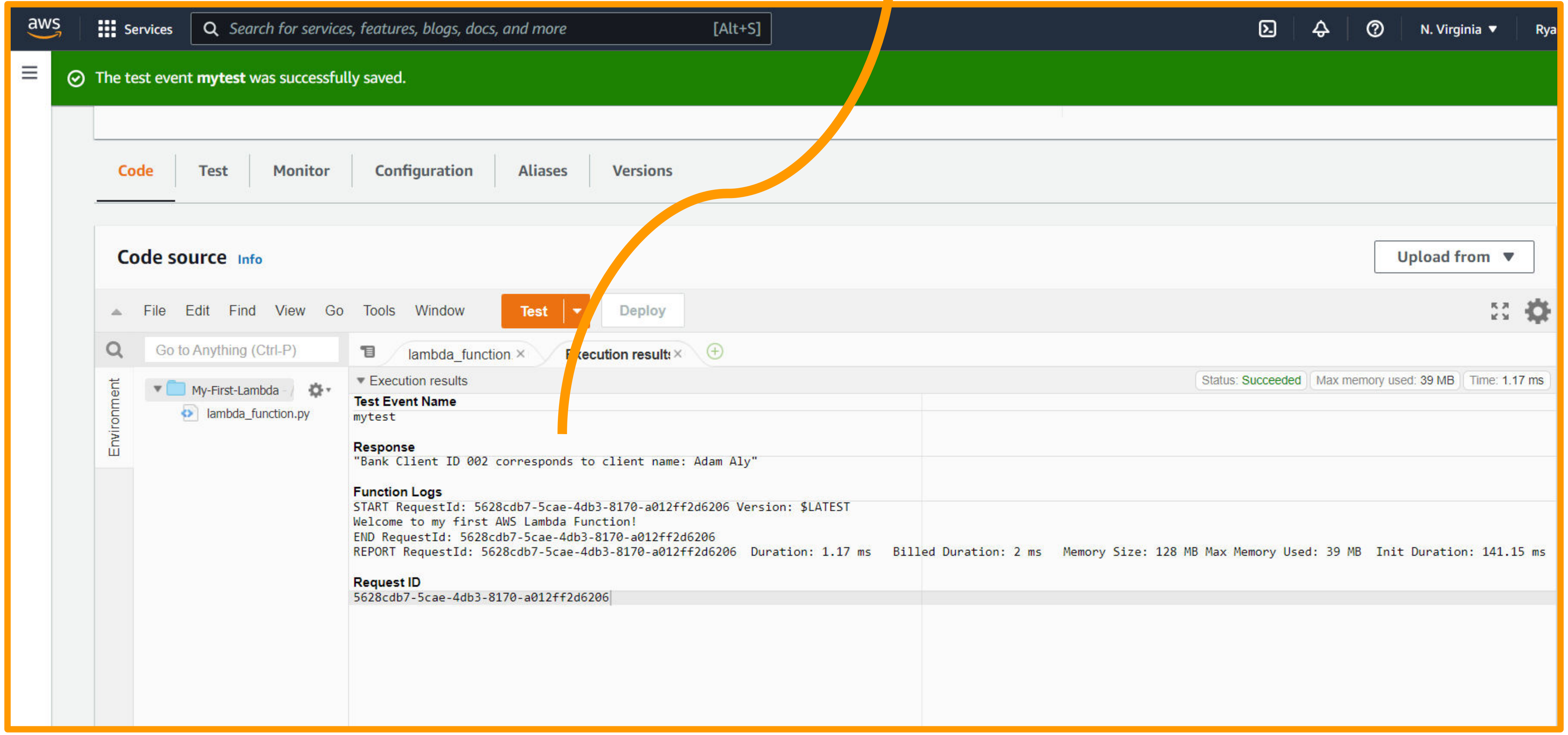
Event JSON

Format JSON

```
{
  "Records": [
    {
      "eventVersion": "2.0",
      "eventSource": "aws:s3",
      "awsRegion": "us-east-1",
      "eventTime": "1970-01-01T00:00:00.000Z",
      "eventName": "ObjectCreated:Put",
      "userIdentity": {
        "principalId": "EXAMPLE"
      },
      "requestParameters": {
        "sourceIPAddress": "127.0.0.1"
      },
      "responseElements": {
        "x-amz-request-id": "EXAMPLE123456789",
        "x-amz-id-2": "EXAMPLE123/5678abcdefghijklambdaisawesome/mnopqrstuvwxyzABCDEFGH"
      },
      "s3": {
        "s3SchemaVersion": "1.0",
        "configurationId": "testConfigRule",
        "bucket": {
          "name": "example-bucket",
          "ownerIdentity": {
            "principalId": "EXAMPLE"
          },
          "arn": "arn:aws:s3:::example-bucket"
        },
        "object": {
```

DEMO: CREATE AN AWS LAMBDA FUNCTION USING CONSOLE

CLICK TEST AND OBSERVE THE FUNCTION RESPONSE



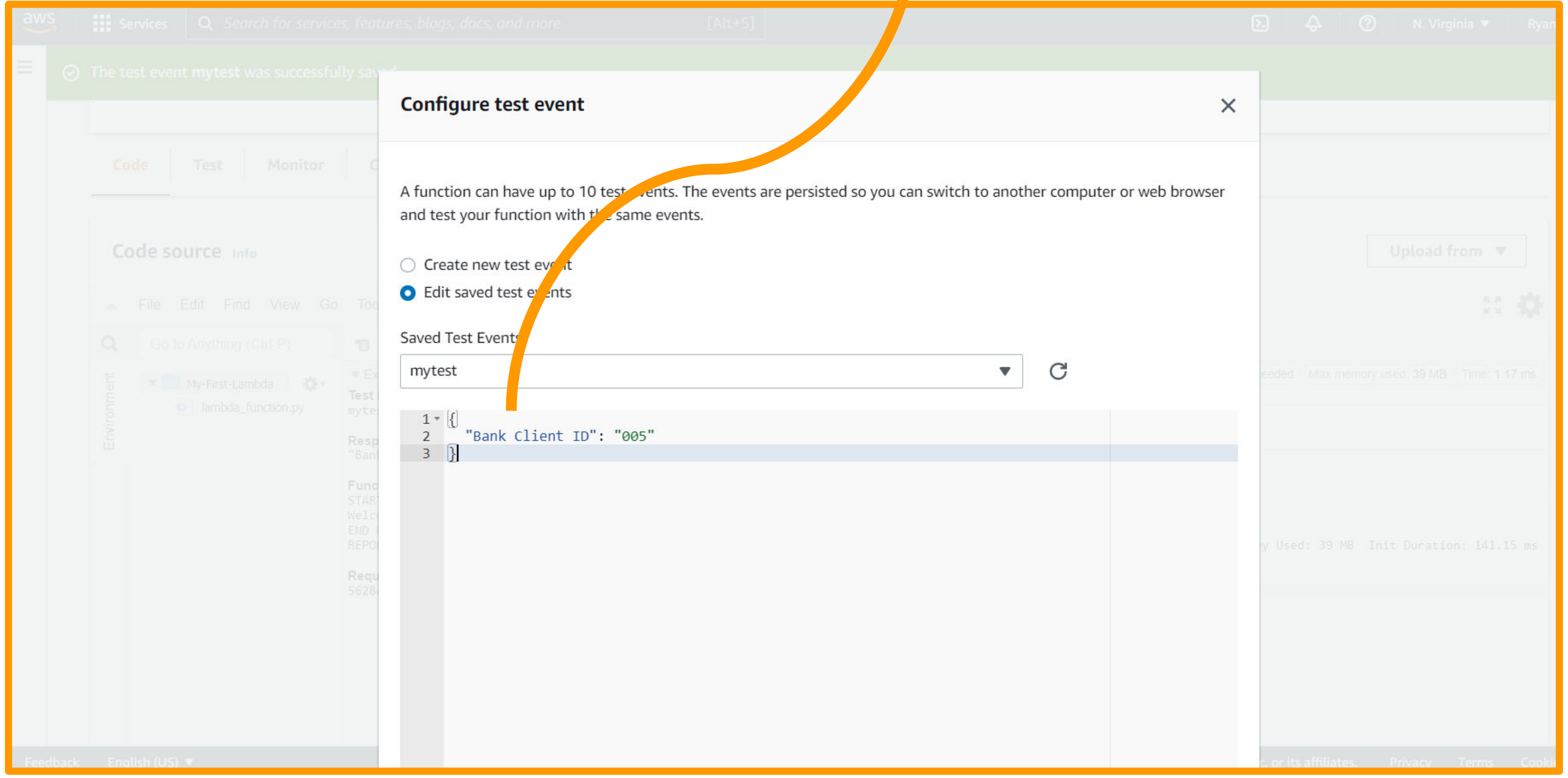
The test event **mytest** was successfully saved.

Execution results — Status: Succeeded | Max memory used: 39 MB | Time: 1.17 ms

```
Test Event Name
mytest

Response
"Bank Client ID 002 corresponds to client name: Adam Aly"

Function Logs
START RequestId: 5628cdb7-5cae-4db3-8170-a012ff2d6206 Version: $LATEST
Welcome to my first AWS Lambda Function!
END RequestId: 5628cdb7-5cae-4db3-8170-a012ff2d6206
REPORT RequestId: 5628cdb7-5cae-4db3-8170-a012ff2d6206  Duration: 1.17 ms   Billed Duration: 2 ms   Memory Size: 128 MB Max Memory Used: 39 MB  Init Duration: 141.15 ms

Request ID
5628cdb7-5cae-4db3-8170-a012ff2d6206
```

DEMO: CREATE AN AWS LAMBDA FUNCTION USING CONSOLE

LET'S TEST THE FUNCTION AGAIN BY SETTING "BANK CLIENT ID": 005



Configure test event

A function can have up to 10 test events. The events are persisted so you can switch to another computer or web browser and test your function with the same events.

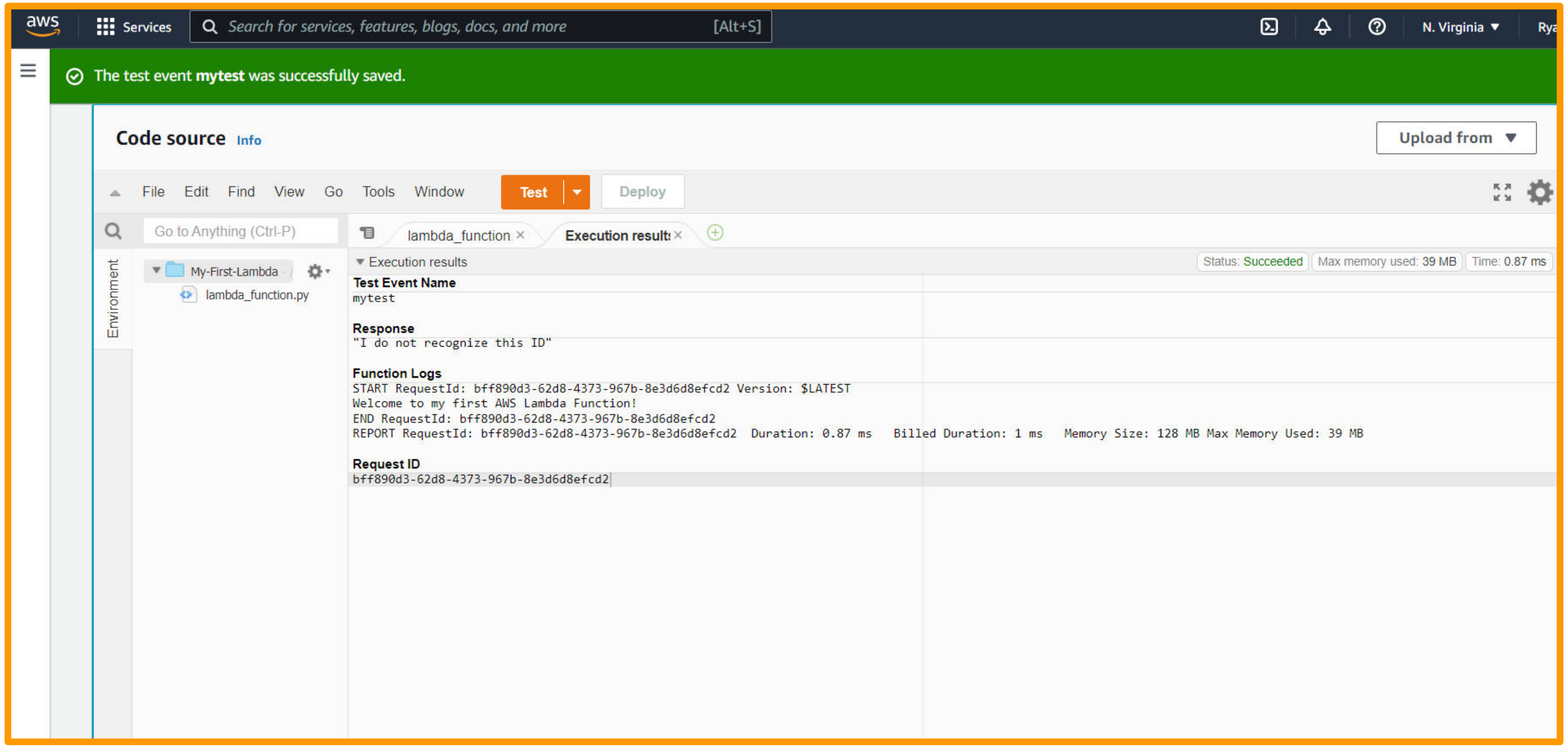
Create new test event [✓ Edit saved test events]

Saved Test Events: mytest

```
{
  "Bank Client ID": "005"
}
```

DEMO: CREATE AN AWS LAMBDA FUNCTION USING CONSOLE

NOTE THAT THE ELSE BRANCH HAS BEEN EXECUTED



Test Event Name
```
mytest
```

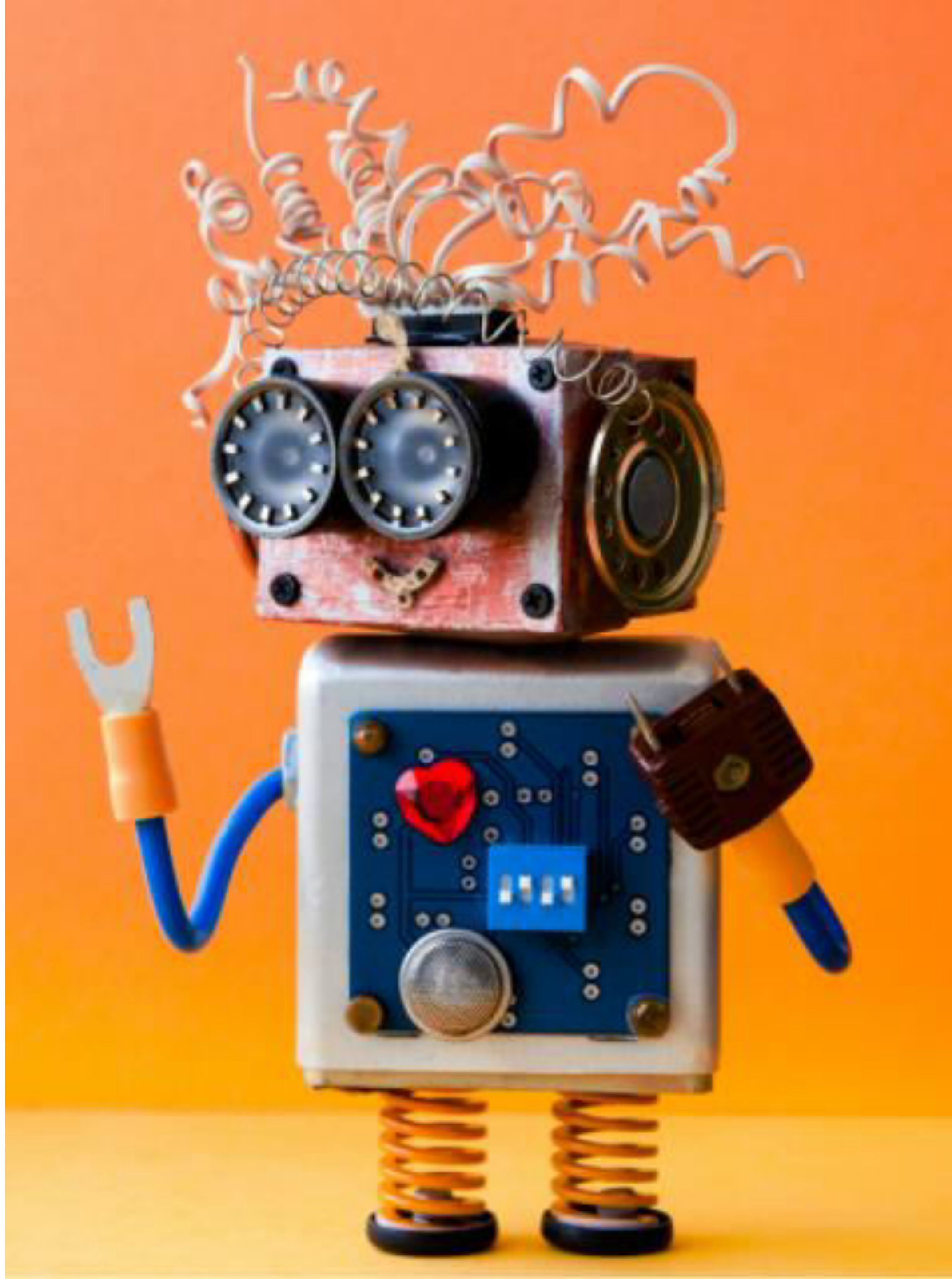
Response
```
"I do not recognize this ID"
```

Function Logs
```
START RequestId: bff890d3-62d8-4373-967b-8e3d6d8efcd2 Version: $LATEST
Welcome to my first AWS Lambda Function!
END RequestId: bff890d3-62d8-4373-967b-8e3d6d8efcd2
REPORT RequestId: bff890d3-62d8-4373-967b-8e3d6d8efcd2  Duration: 0.87 ms   Billed Duration: 1 ms   Memory Size: 128 MB Max Memory Used: 39 MB
```

Request ID
```
bff890d3-62d8-4373-967b-8e3d6d8efcd2
```

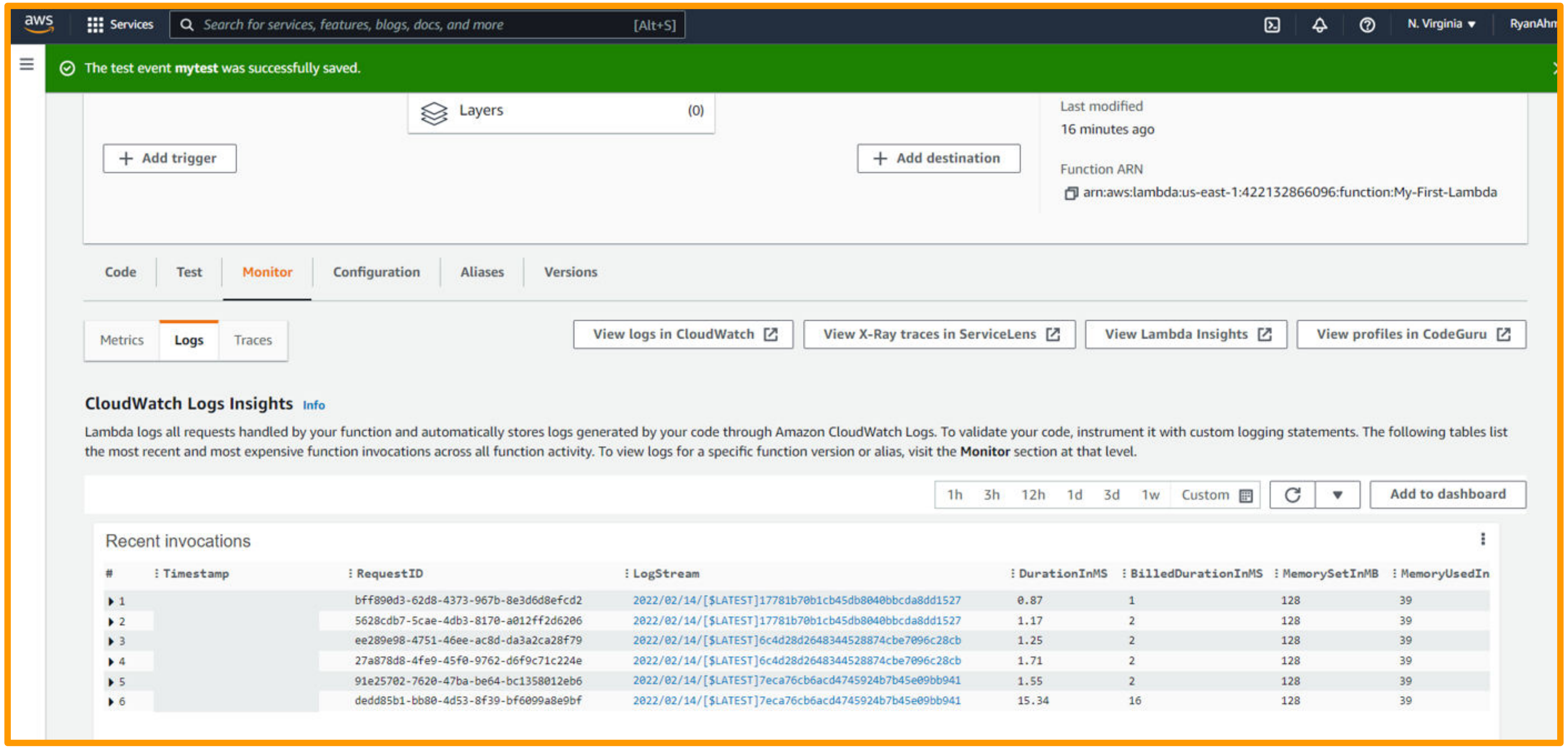
DEMO EXAMPLE 2: MONITOR AN AWS LAMBDA FUNCTION USING CONSOLE

EASY ▲ ADVANCED



DEMO: CREATE AN AWS LAMBDA FUNCTION USING CONSOLE

GO TO MONITOR AND LOGS TO TRACK THE LAMBDA FUNCTION REQUESTS, DURATION, BILLING..ETC



The test event **mytest** was successfully saved.

Last modified
16 minutes ago

Function ARN
arn:aws:lambda:us-east-1:422132866096:function:My-First-Lambda

CloudWatch Logs Insights Info

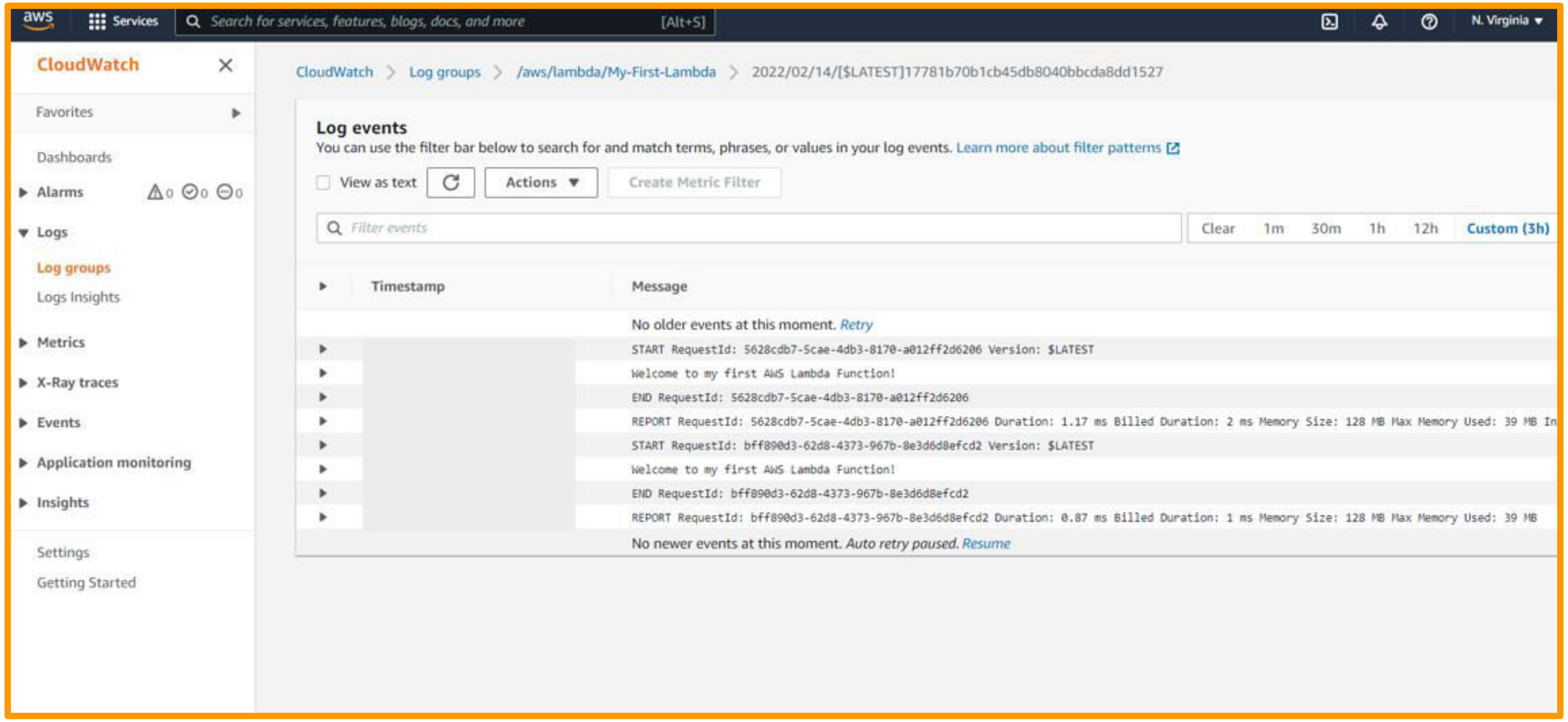
Lambda logs all requests handled by your function and automatically stores logs generated by your code through Amazon CloudWatch Logs. To validate your code, instrument it with custom logging statements. The following tables list the most recent and most expensive function invocations across all function activity. To view logs for a specific function version or alias, visit the **Monitor** section at that level.

Recent invocations

#	Timestamp	RequestID	LogStream	DurationInMS	BilledDurationInMS	MemorySetInMB	MemoryUsedIn
1		bff890d3-62d8-4373-967b-8e3d6d8efcd2	2022/02/14/[$LATEST]17781b70b1cb45db8040bbcda8dd1527	0.87	1	128	39
2		5628cdb7-5cae-4db3-8170-a012ff2d6206	2022/02/14/[$LATEST]17781b70b1cb45db8040bbcda8dd1527	1.17	2	128	39
3		ee289e98-4751-46ee-ac8d-da3a2ca28f79	2022/02/14/[$LATEST]6c4d28d2648344528874cbe7096c28cb	1.25	2	128	39
4		27a878d8-4fe9-45f0-9762-d6f9c71c224e	2022/02/14/[$LATEST]6c4d28d2648344528874cbe7096c28cb	1.71	2	128	39
5		91e25702-7620-47ba-be64-bc1358012eb6	2022/02/14/[$LATEST]7eca76cb6acd4745924b7b45e09bb941	1.55	2	128	39
6		dedd85b1-bb80-4d53-8f39-bf6099a8e9bf	2022/02/14/[$LATEST]7eca76cb6acd4745924b7b45e09bb941	15.34	16	128	39

DEMO: CREATE AN AWS LAMBDA FUNCTION USING CONSOLE

CLICK ON A SAMPLE LOG EVENT TO VIEW DETAILS ON CLOUDWATCH



CloudWatch > Log groups > /aws/lambda/My-First-Lambda > 2022/02/14/[$LATEST]17781b70b1cb45db8040bbcda8dd1527

Log events

You can use the filter bar below to search for and match terms, phrases, or values in your log events. Learn more about filter patterns

View as text | Actions | Create Metric Filter

Filter events | Clear | 1m | 30m | 1h | 12h | Custom (3h)

Timestamp	Message

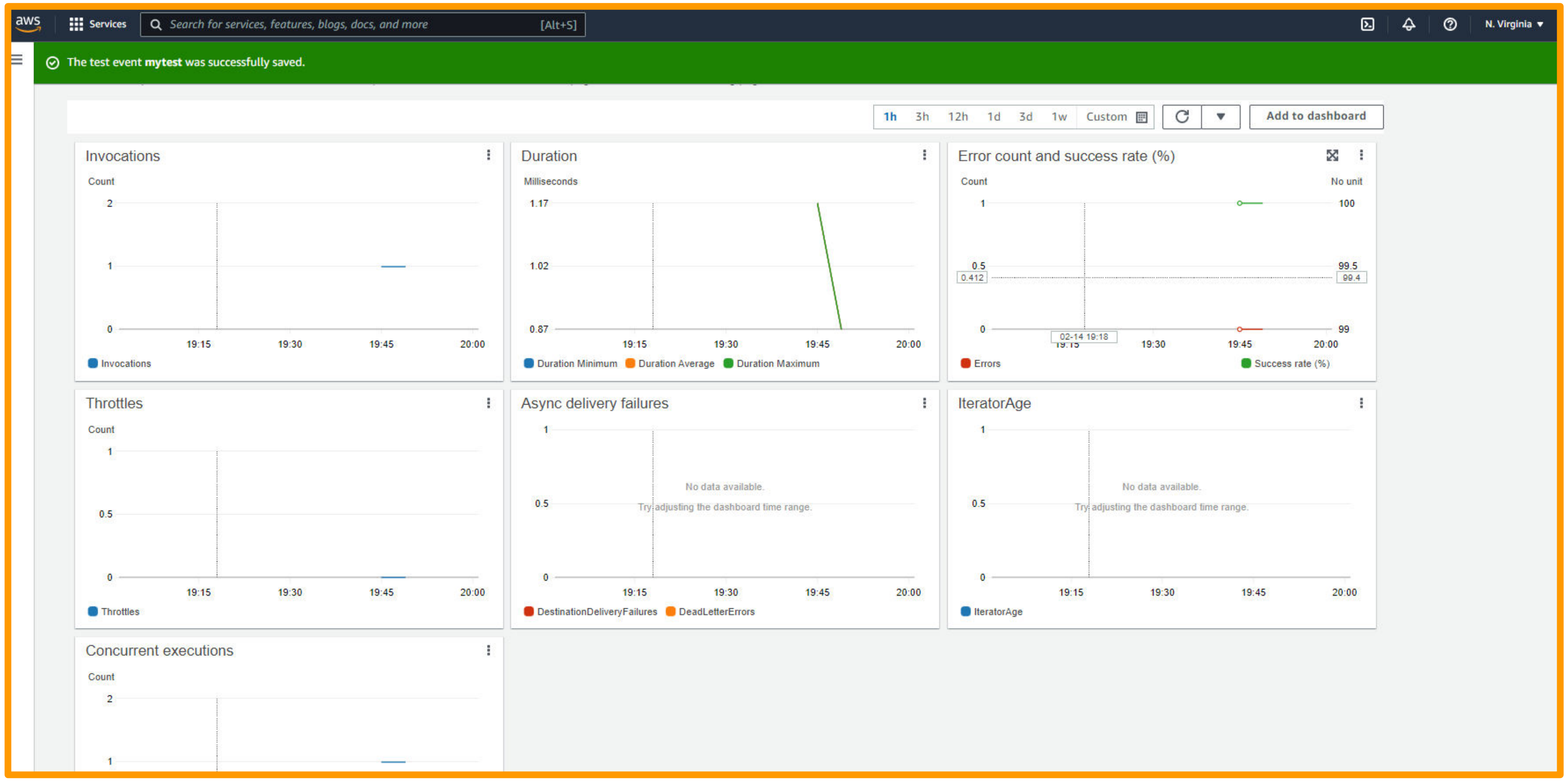
No older events at this moment. *Retry*

```
START RequestId: 5628cdb7-5cae-4db3-8170-a012ff2d6206 Version: $LATEST
Welcome to my first AWS Lambda Function!
END RequestId: 5628cdb7-5cae-4db3-8170-a012ff2d6206
REPORT RequestId: 5628cdb7-5cae-4db3-8170-a012ff2d6206 Duration: 1.17 ms Billed Duration: 2 ms Memory Size: 128 MB Max Memory Used: 39 MB In
START RequestId: bff890d3-62d8-4373-967b-8e3d6d8efcd2 Version: $LATEST
Welcome to my first AWS Lambda Function!
END RequestId: bff890d3-62d8-4373-967b-8e3d6d8efcd2
REPORT RequestId: bff890d3-62d8-4373-967b-8e3d6d8efcd2 Duration: 0.87 ms Billed Duration: 1 ms Memory Size: 128 MB Max Memory Used: 39 MB
```

No newer events at this moment. *Auto retry paused.* Resume

DEMO: CREATE AN AWS LAMBDA FUNCTION USING CONSOLE

GO TO MONITOR AND METRICS TO VIEW DASHBOARD WITH ALL METRICS INCLUDING INVOCATIONS

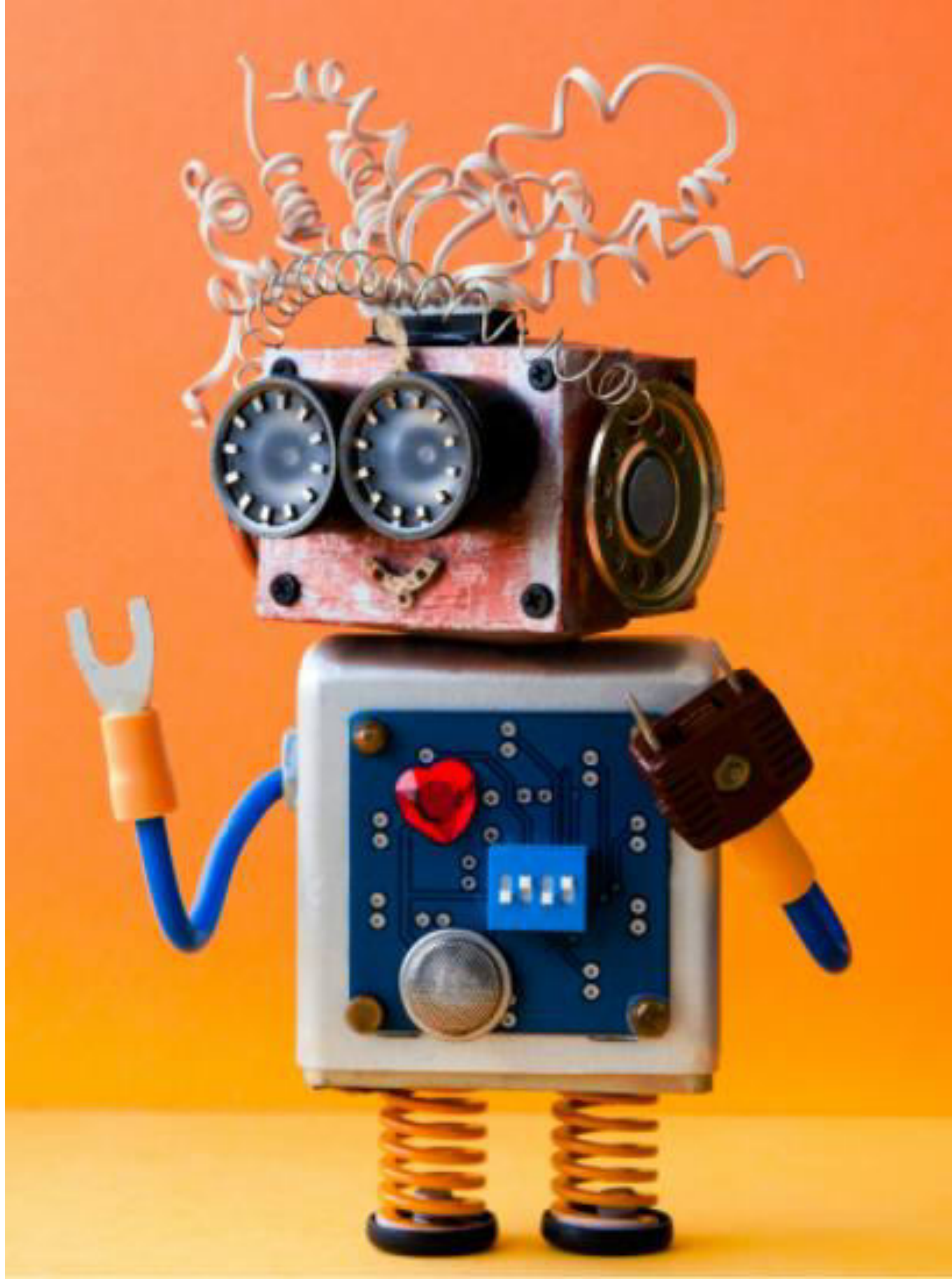


The test event **mytest** was successfully saved.

No data available.
Try adjusting the dashboard time range.

FINAL END-OF-DAY CAPSTONE PROJECT

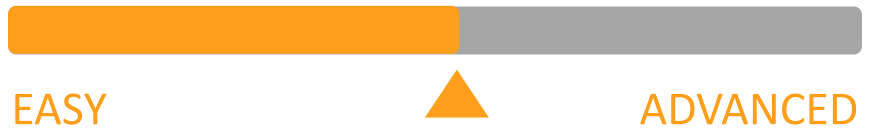
EASY ▲ ADVANCED

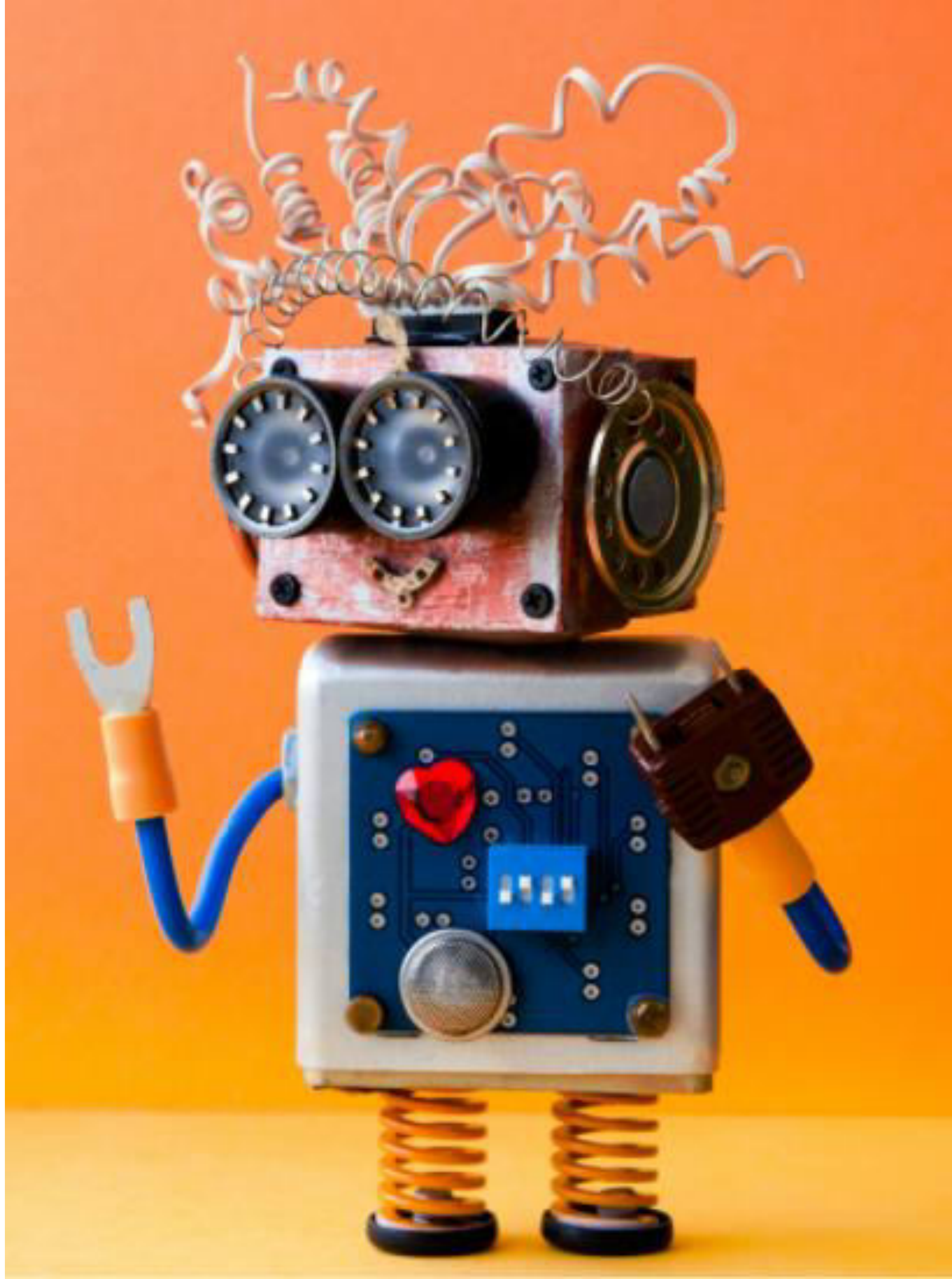


FINAL CAPSTONE PROJECT

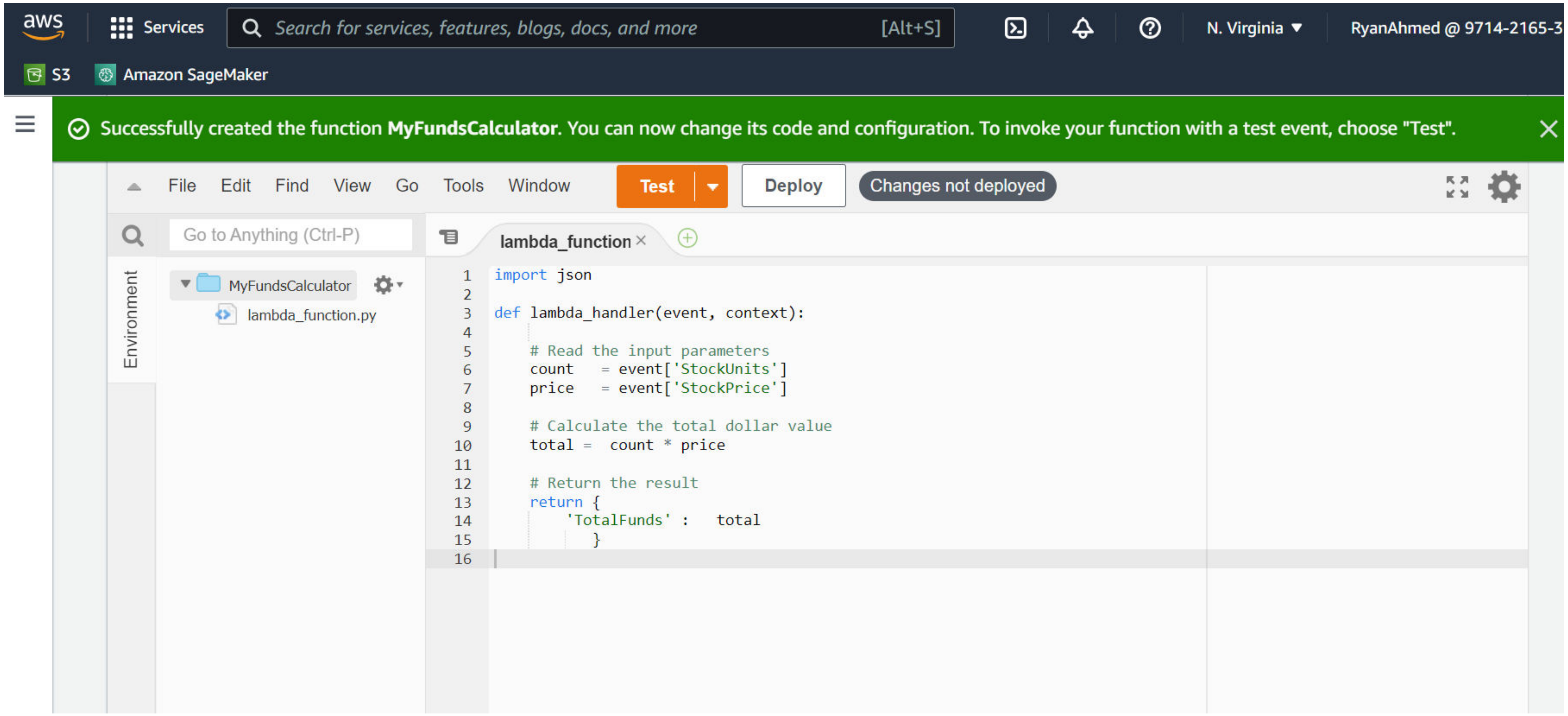
- Define a Lambda Function that takes in the number of stock units and stock price and calculates the total value of the portfolio.
- Return the total value of the portfolio.
- Configure the following test events:
    - Stock Units = 20, stock value = $1000
    - Stock Units = 5, stock value = $2000
- Assume that the default currency in the previous lambda function is in USD, we would like to enhance this function to either return USD or CAD currencies (assume exchange rate 1 USD = 1.3 CAD).
- Define and test the new Lambda Function.

FINAL END-OF-DAY CAPSTONE PROJECT SOLUTION





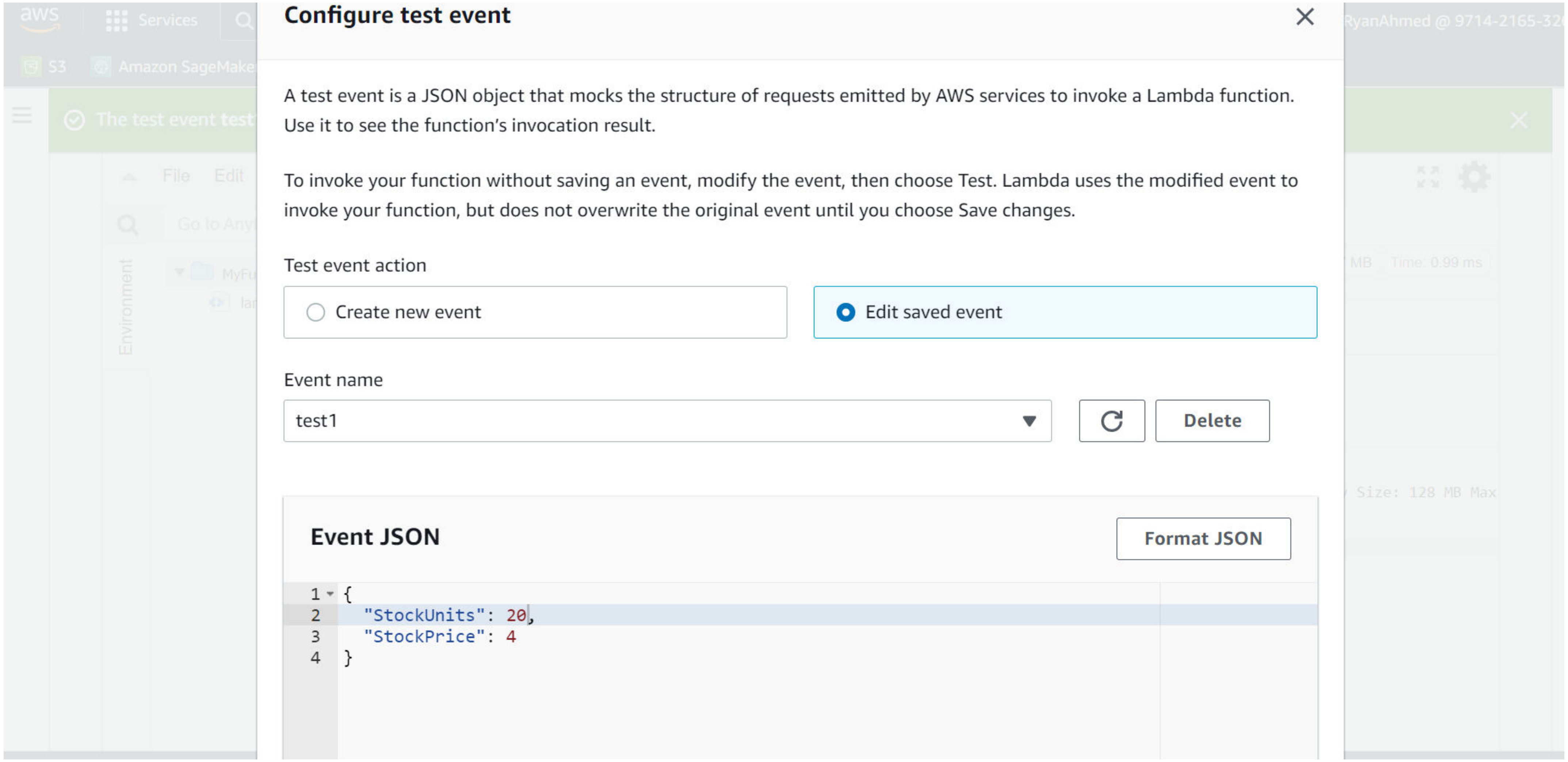
FINAL CAPSTONE PROJECT SOLUTION



Successfully created the function **MyFundsCalculator**. You can now change its code and configuration. To invoke your function with a test event, choose "Test".

```python
import json

def lambda_handler(event, context):

    # Read the input parameters
    count   = event['StockUnits']
    price   = event['StockPrice']

    # Calculate the total dollar value
    total =  count * price

    # Return the result
    return {
        'TotalFunds' :   total
        }

```

FINAL CAPSTONE PROJECT SOLUTION



Configure test event

A test event is a JSON object that mocks the structure of requests emitted by AWS services to invoke a Lambda function. Use it to see the function's invocation result.

To invoke your function without saving an event, modify the event, then choose Test. Lambda uses the modified event to invoke your function, but does not overwrite the original event until you choose Save changes.

Test event action

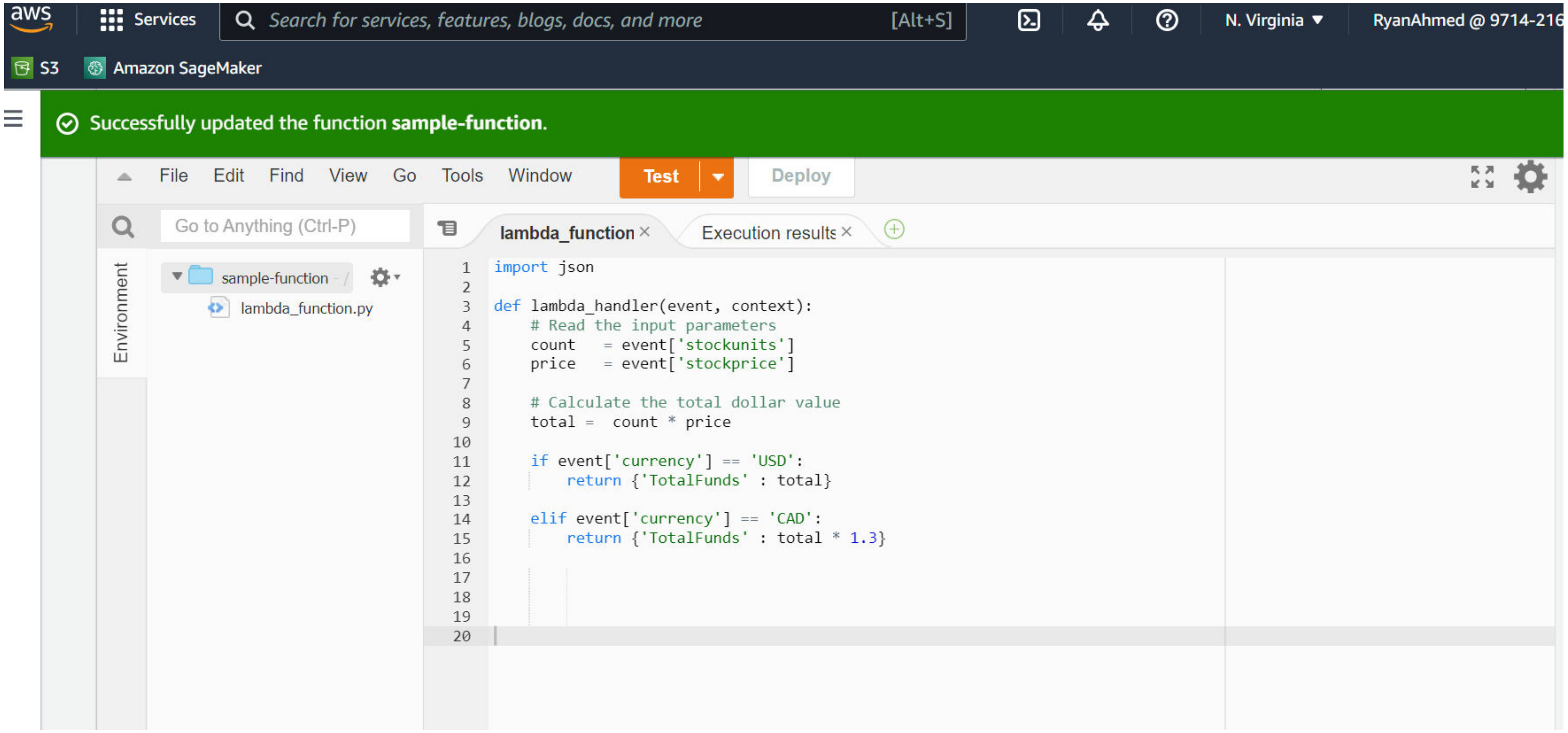
Create new event [✓ Edit saved event]

Event name: test1

Event JSON

```
{
  "StockUnits": 20,
  "StockPrice": 4
}
```

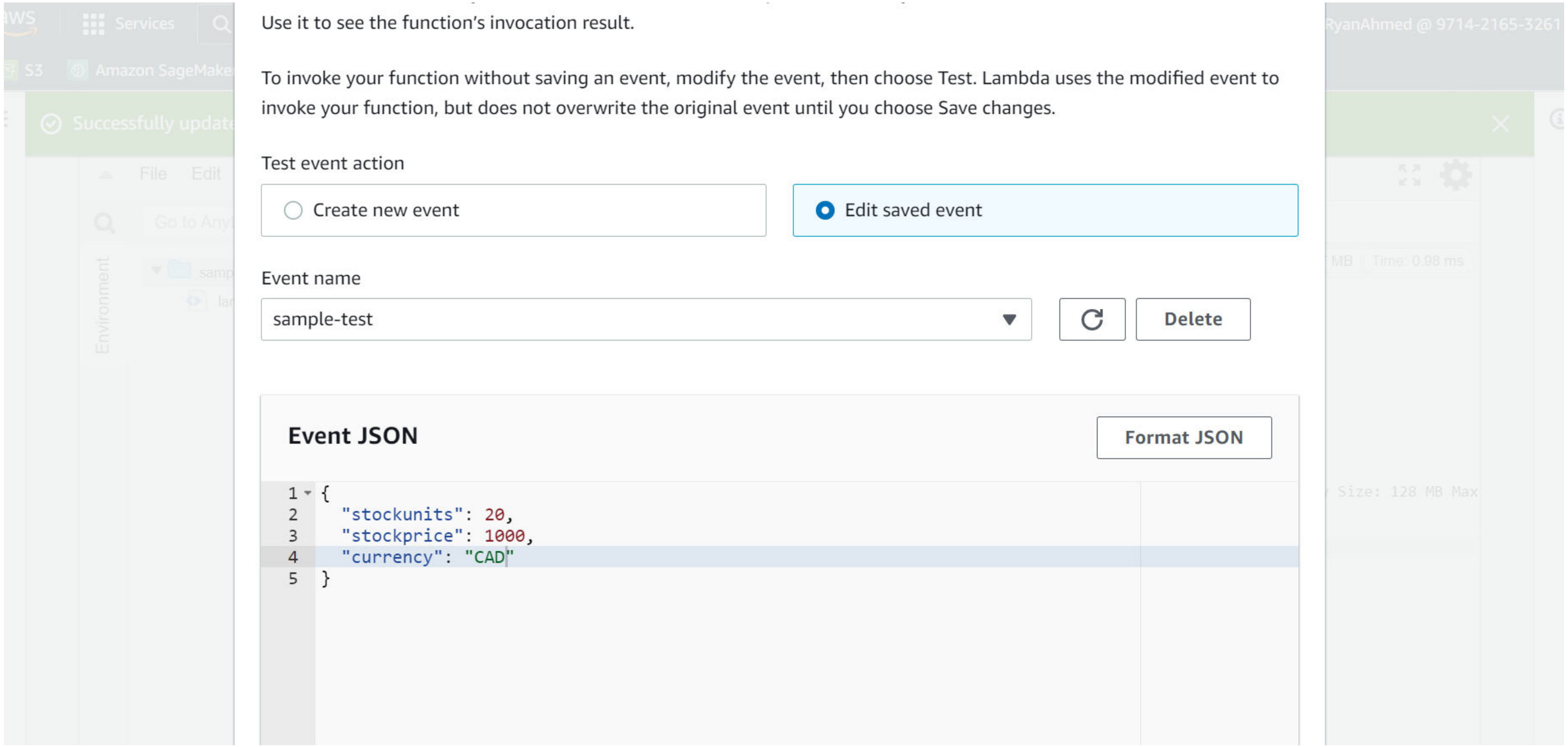
FINAL CAPSTONE PROJECT SOLUTION



Successfully updated the function **sample-function.**

```python
import json

def lambda_handler(event, context):
    # Read the input parameters
    count   = event['stockunits']
    price   = event['stockprice']

    # Calculate the total dollar value
    total =  count * price

    if event['currency'] == 'USD':
        return {'TotalFunds' : total}

    elif event['currency'] == 'CAD':
        return {'TotalFunds' : total * 1.3}
```

FINAL CAPSTONE PROJECT SOLUTION



Use it to see the function's invocation result.

To invoke your function without saving an event, modify the event, then choose Test. Lambda uses the modified event to invoke your function, but does not overwrite the original event until you choose Save changes.

Test event action

○ Create new event [✓ Edit saved event]

Event name

sample-test

Event JSON

```
{
  "stockunits": 20,
  "stockprice": 1000,
  "currency": "CAD"
}
```