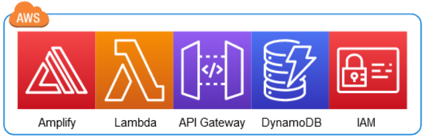
End-to-End AWS Web App by Daniel Lee

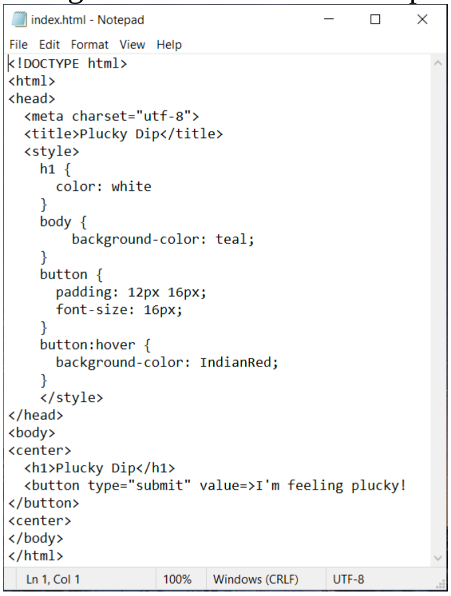
Objective: create a web app to generate numbers for the Euromillions lottery and store it to a database.

Services I'll be using



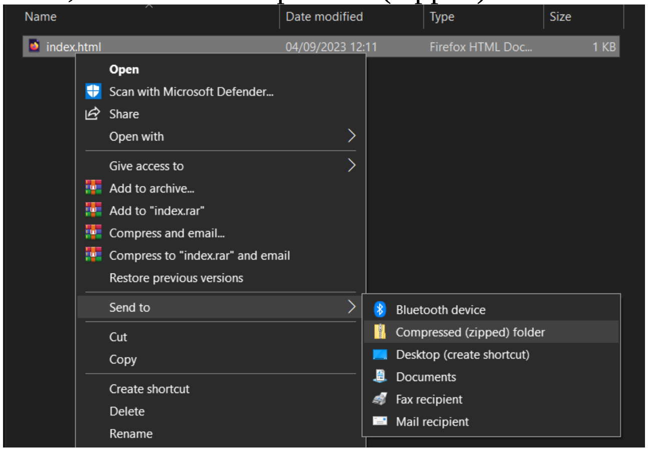
- Deploy and host a webpage with **Amplify**
- Create a Python **Lambda** function to generate the random numbers
- Create a REST API for the Lambda function using **API Gateway**
- Create a DynamDB table to store the results
- Using **IAM** to give Lambda permissions to write to DynamoDB

1. Create a simple index.html document using a text editor such as notepad

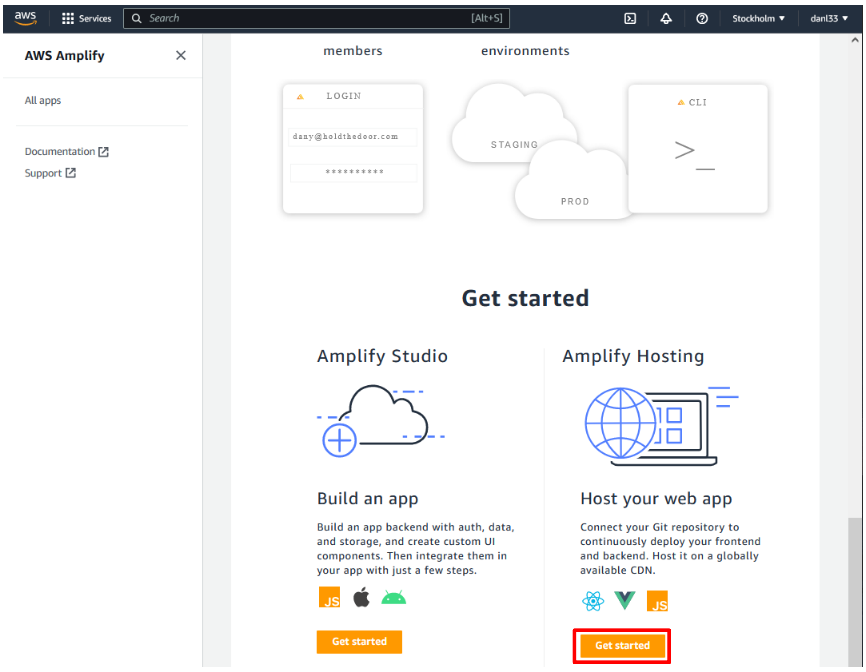


```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>Plucky Dip</title>
  <style>
    h1 {
      color: white
    }
    body {
        background-color: teal;
    }
    button {
      padding: 12px 16px;
      font-size: 16px;
    }
    button:hover {
      background-color: IndianRed;
    }
    </style>
</head>
<body>
<center>
  <h1>Plucky Dip</h1>
  <button type="submit" value=>I'm feeling plucky!
</button>
<center>
</body>
</html>
```

2. Once saved and closed, right-click, Send to > Compressed (zipped) folder



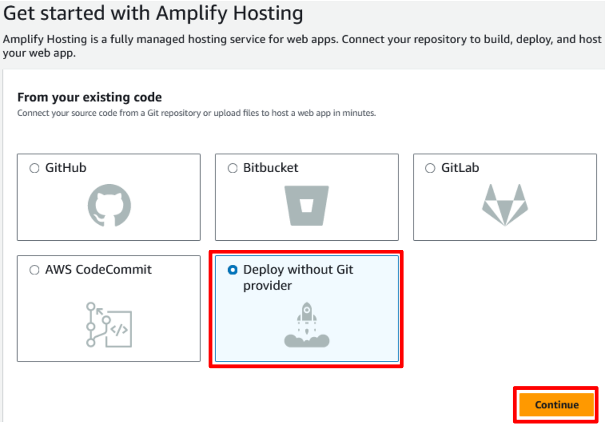
3. In your browser navigate to Amplify and click Get Started under Amplify Hosting



Build an app backend with auth, data, and storage, and create custom UI components. Then integrate them in your app with just a few steps.

Connect your Git repository to continuously deploy your frontend and backend. Host it on a globally available CDN.

4. Configure as follows



Get started with Amplify Hosting

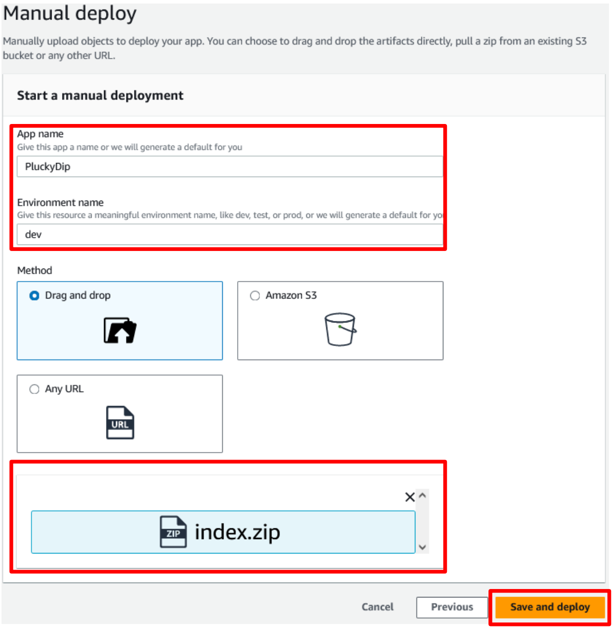
Amplify Hosting is a fully managed hosting service for web apps. Connect your repository to build, deploy, and host your web app.

From your existing code

Connect your source code from a Git repository or upload files to host a web app in minutes.

○ GitHub
○ Bitbucket
○ GitLab
○ AWS CodeCommit
● Deploy without Git provider

5. Provide an app and environment name, then drag and drop in the previously created zip file



Manual deploy

Manually upload objects to deploy your app. You can choose to drag and drop the artifacts directly, pull a zip from an existing S3 bucket or any other URL.

Start a manual deployment

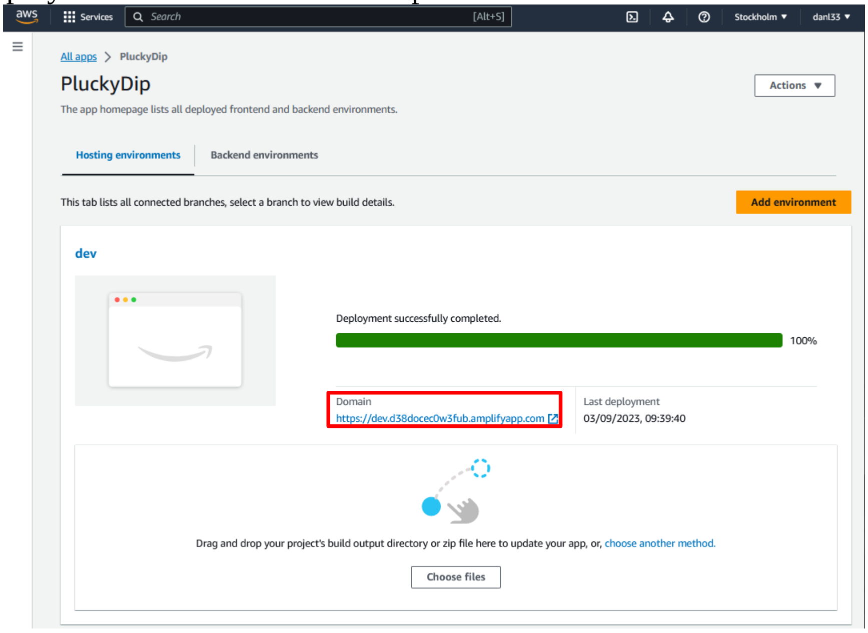
App name
Give this app a name or we will generate a default for you
PluckyDip

Environment name
Give this resource a meaningful environment name, like dev, test, or prod, or we will generate a default for yo
dev

Method
● Drag and drop
○ Amazon S3
○ Any URL

index.zip

6. Once successfully deployed click on the domain to open it in a new tab



All apps > PluckyDip

PluckyDip

The app homepage lists all deployed frontend and backend environments.

This tab lists all connected branches, select a branch to view build details.

dev

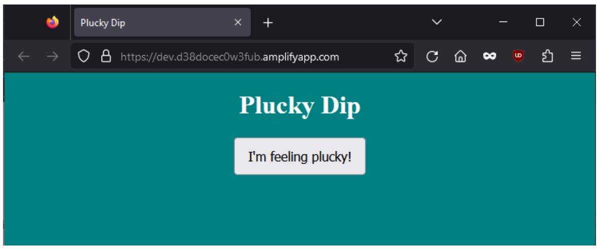
Deployment successfully completed.

100%

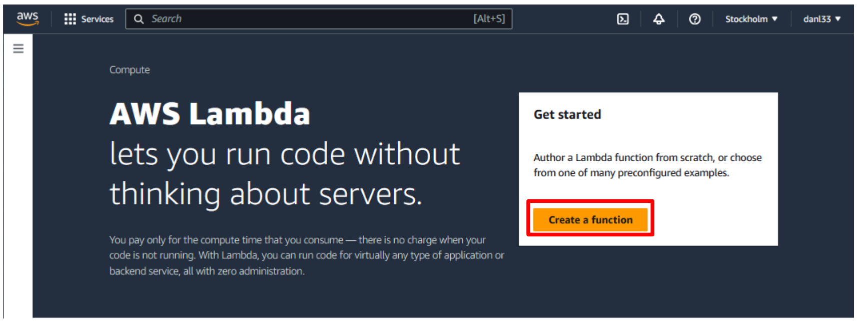
Domain
https://dev.d38docec0w3fub.amplifyapp.com

Last deployment
03/09/2023, 09:39:40

Drag and drop your project's build output directory or zip file here to update your app, or, choose another method.



7. In a new tab navigate to Lambda and click Create a function



Compute

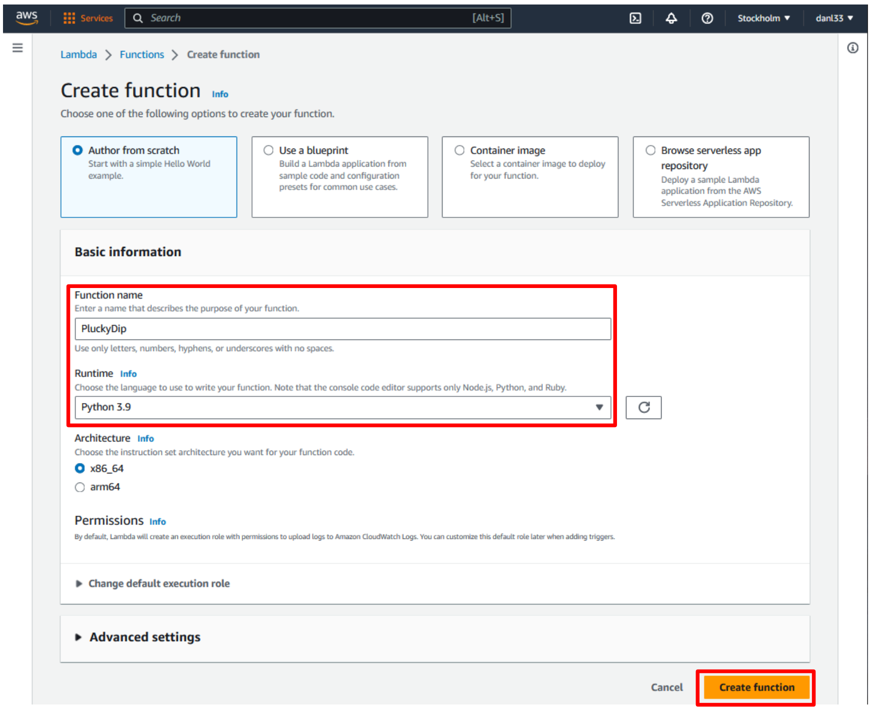
AWS Lambda lets you run code without thinking about servers.

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.

Get started

Author a Lambda function from scratch, or choose from one of many preconfigured examples.

8. Configure as follows



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.
PluckyDip
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 3.9

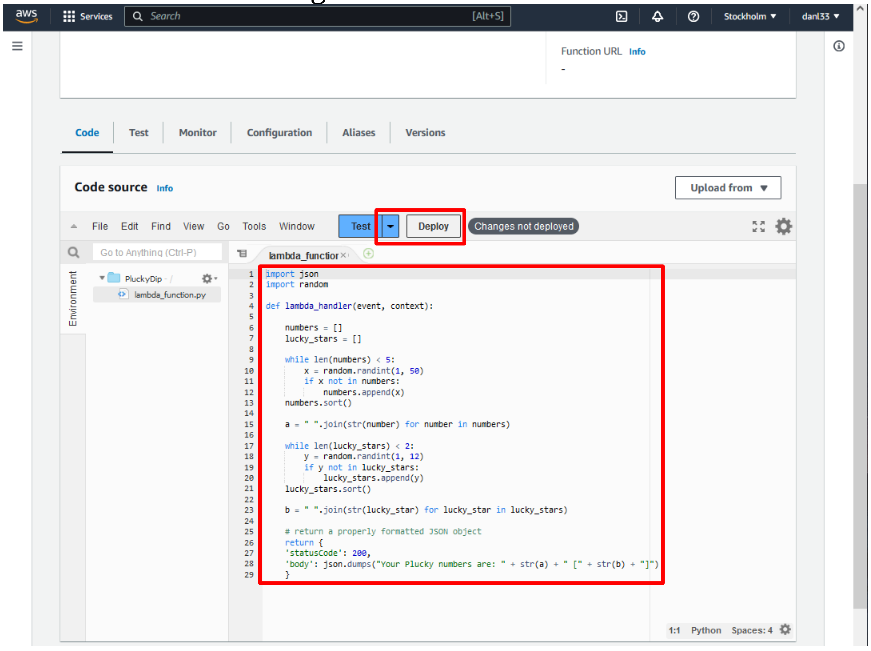
Architecture Info
Choose the instruction set architecture you want for your function code.
● x86_64
○ arm64

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

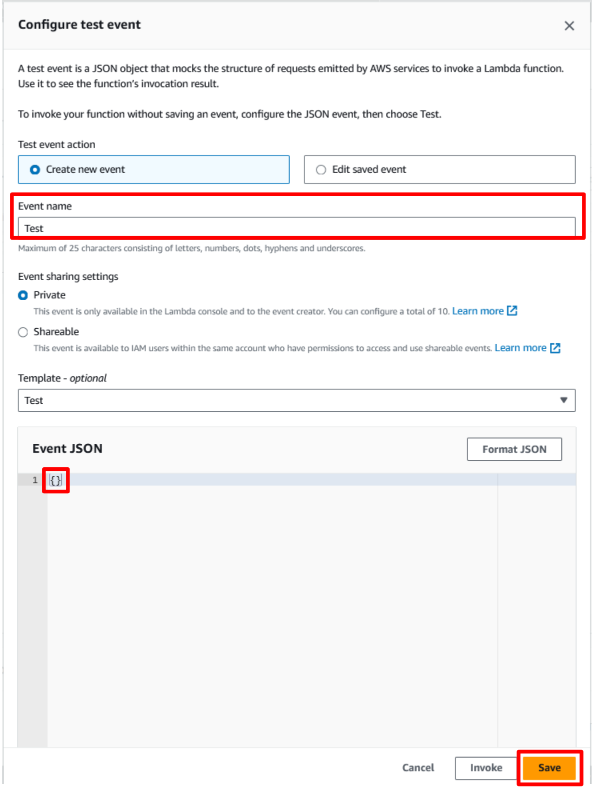
▸ Advanced settings

9. On the next scroll down to Code source and add in your python code. Click deploy to save changes. Then click on the down arrow next to Test and select Configure test event



```python
import json
import random

def lambda_handler(event, context):

    numbers = []
    lucky_stars = []

    while len(numbers) < 5:
        x = random.randint(1, 50)
        if x not in numbers:
            numbers.append(x)
    numbers.sort()

    a = " ".join(str(number) for number in numbers)

    while len(lucky_stars) < 2:
        y = random.randint(1, 12)
        if y not in lucky_stars:
            lucky_stars.append(y)
    lucky_stars.sort()

    b = " ".join(str(lucky_star) for lucky_star in lucky_stars)

    # return a properly formatted JSON object
    return {
    'statusCode': 200,
    'body': json.dumps("Your Plucky numbers are: " + str(a) + " [" + str(b) + "]")
    }
```

10. Configure as follows. *Note: As the python script doesn't require any input values we can simply put "{}"*



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, configure the JSON event, then choose Test.

Test event action

● Create new event ○ Edit saved event

Event name

Test

Maximum of 25 characters consisting of letters, numbers, dots, hyphens and underscores.

Event sharing settings

● Private

This event is only available in the Lambda console and to the event creator. You can configure a total of 10. Learn more

○ Shareable

This event is available to IAM users within the same account who have permissions to access and use shareable events. Learn more

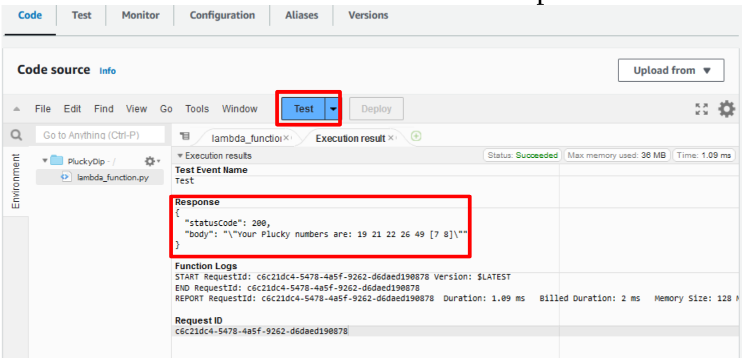
Template - *optional*

Test

Event JSON

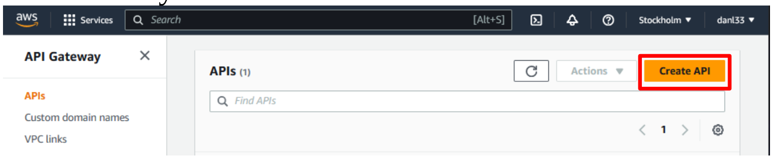
```
{}
```

11. Now click the Test button and the results will show a successful response

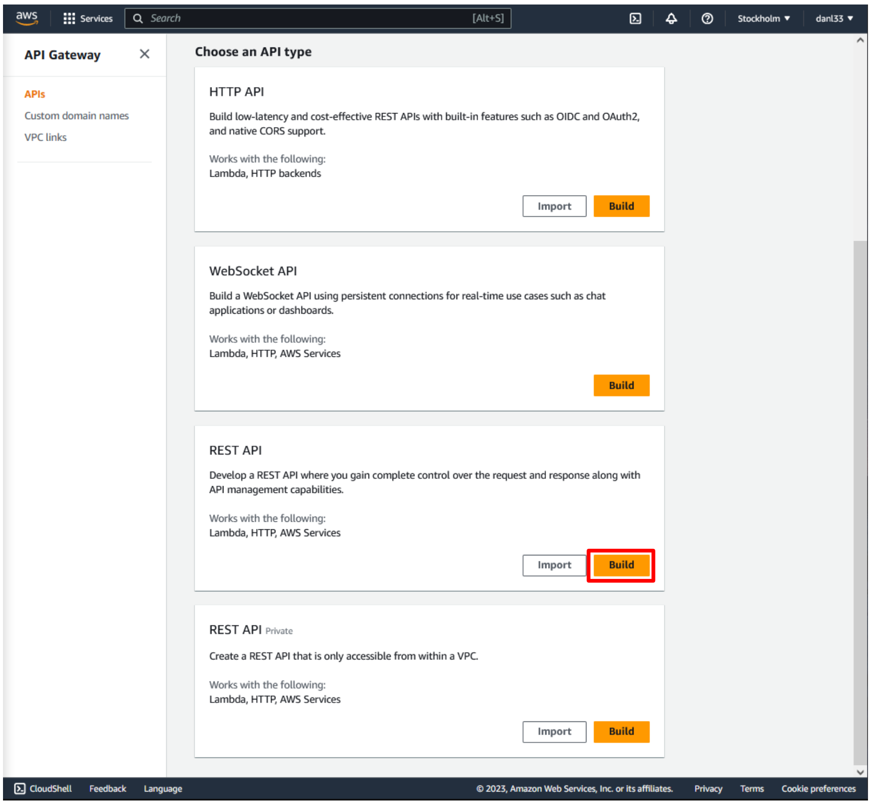


```
Execution results                    Status: Succeeded  Max memory used: 36 MB  Time: 1.09 ms
Test Event Name
Test

Response
{
  "statusCode": 200,
  "body": "\"Your Plucky numbers are: 19 21 22 26 49 [7 8]\""
}

Function Logs
START RequestId: c6c21dc4-5478-4a5f-9262-d6daed190878 Version: $LATEST
END RequestId: c6c21dc4-5478-4a5f-9262-d6daed190878
REPORT RequestId: c6c21dc4-5478-4a5f-9262-d6daed190878  Duration: 1.09 ms   Billed Duration: 2 ms   Memory Size: 128 M

Request ID
c6c21dc4-5478-4a5f-9262-d6daed190878
```

12. In a new tab navigate to API Gateway and click Create API



13. Click Build under Rest API



Choose an API type

HTTP API

Build low-latency and cost-effective REST APIs with built-in features such as OIDC and OAuth2, and native CORS support.

Works with the following:
Lambda, HTTP backends

WebSocket API

Build a WebSocket API using persistent connections for real-time use cases such as chat applications or dashboards.

Works with the following:
Lambda, HTTP, AWS Services

REST API

Develop a REST API where you gain complete control over the request and response along with API management capabilities.

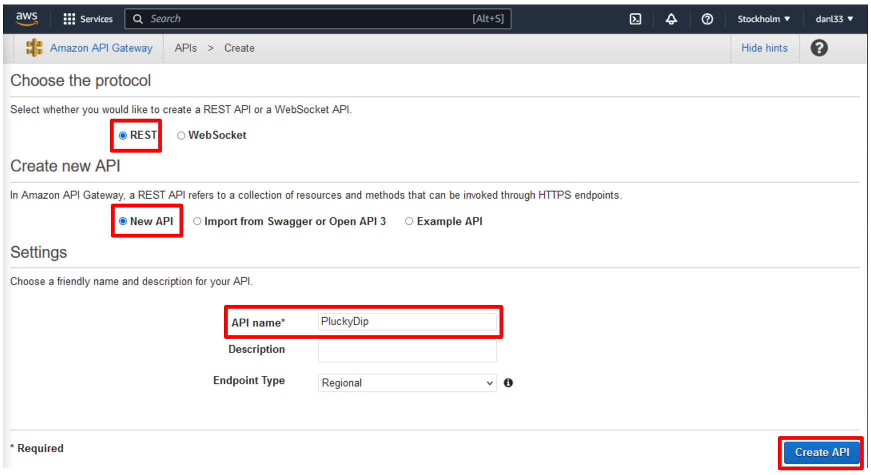
Works with the following:
Lambda, HTTP, AWS Services

REST API Private

Create a REST API that is only accessible from within a VPC.

Works with the following:
Lambda, HTTP, AWS Services

14. Configure as follows



Choose the protocol

Select whether you would like to create a REST API or a WebSocket API.

● REST ○ WebSocket

Create new API

In Amazon API Gateway, a REST API refers to a collection of resources and methods that can be invoked through HTTPS endpoints.

● New API ○ Import from Swagger or Open API 3 ○ Example API

Settings

Choose a friendly name and description for your API.

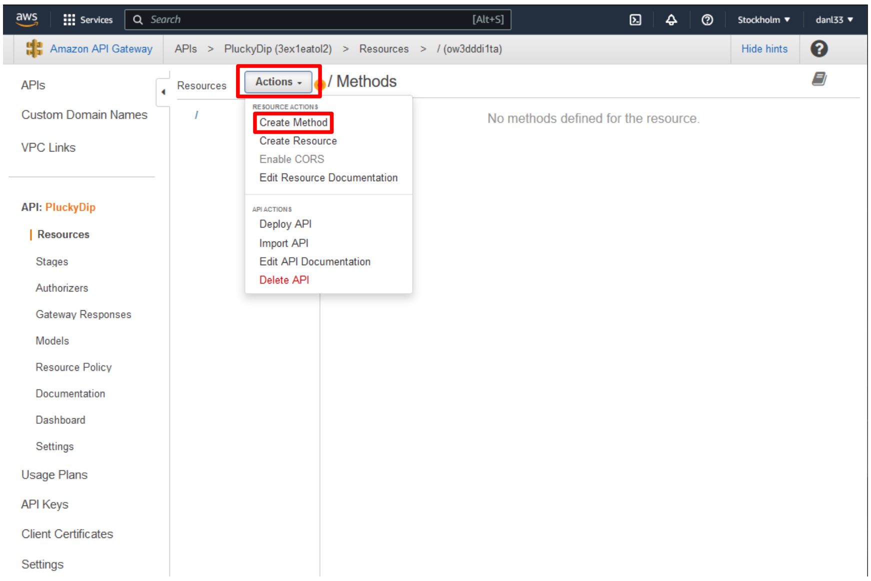
API name* PluckyDip

Description

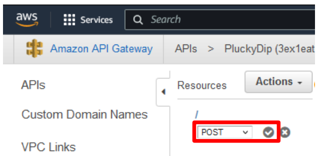
Endpoint Type Regional

* Required

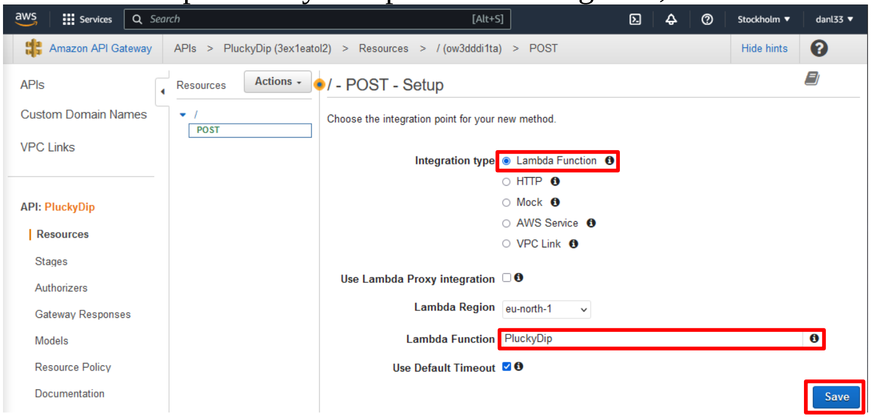
15. On the next screen click Action > Create Method



16. Change the method type to POST then click on the tick



17. Configure to the Lambda function previously set up. After clicking save, hit ok on the confirmation pop-up.



/ - POST - Setup

Choose the integration point for your new method.

Integration type ● Lambda Function ○ HTTP ○ Mock ○ AWS Service ○ VPC Link

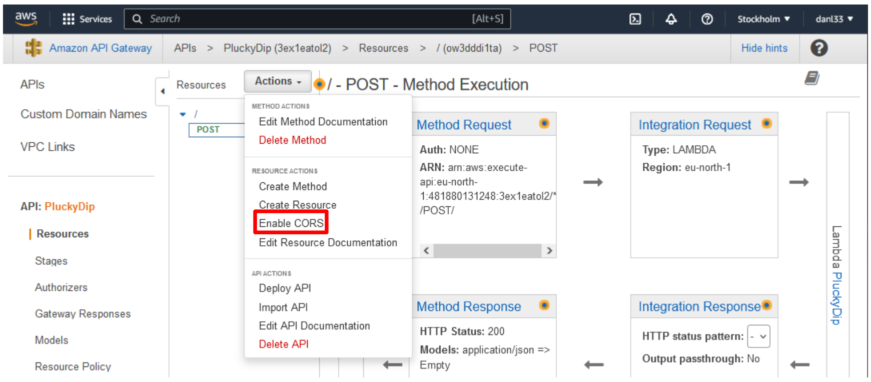
Use Lambda Proxy integration □

Lambda Region eu-north-1

Lambda Function PluckyDip

Use Default Timeout ☑

18. Next click Action > Enable CORS



Auth: NONE

ARN: arn:aws:execute-api:eu-north-1:481880131248:3ex1eatol2/*/POST/

Type: LAMBDA

Region: eu-north-1

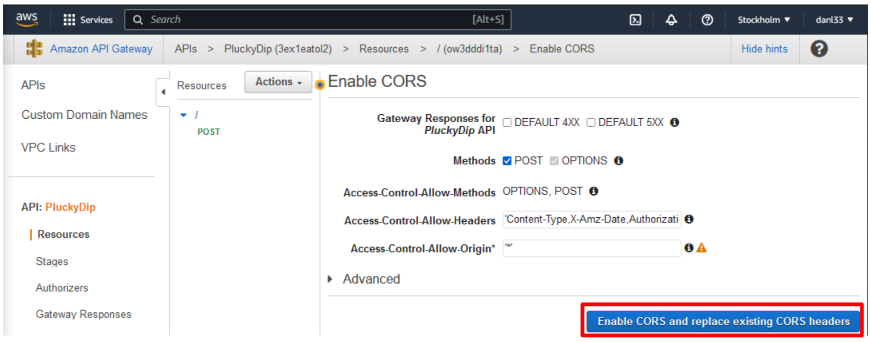
HTTP Status: 200

Models: application/json => Empty

HTTP status pattern: -

Output passthrough: No

19. Click Enable CORS and replace existing CORS headers. Hit Yes, replace existing values on the confirmation pop-up.



Enable CORS

Gateway Responses for PluckyDip API □ DEFAULT 4XX □ DEFAULT 5XX

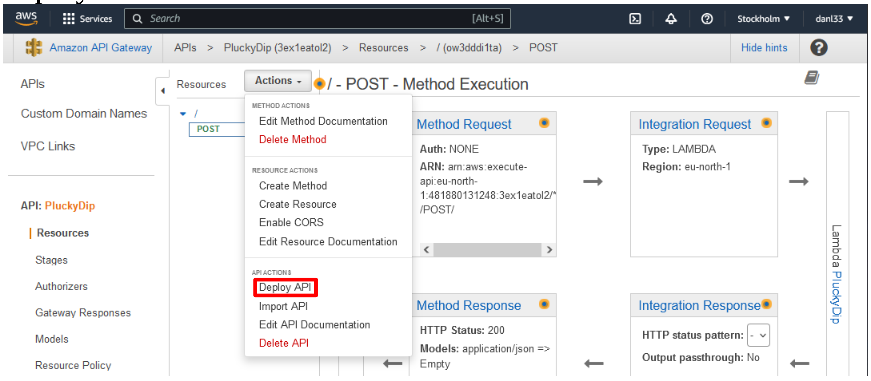
Methods ☑ POST ☑ OPTIONS

Access-Control-Allow-Methods OPTIONS, POST

Access-Control-Allow-Headers 'Content-Type,X-Amz-Date,Authorizati

Access-Control-Allow-Origin* '*'

20. Next click Action > Deploy API



Auth: NONE

ARN: arn:aws:execute-api:eu-north-1:481880131248:3ex1eatol2/*/POST/

Type: LAMBDA

Region: eu-north-1

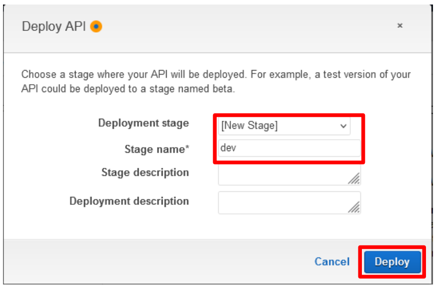
HTTP Status: 200

Models: application/json => Empty

HTTP status pattern: -

Output passthrough: No

21. Configure and deploy a new stage as follows



Choose a stage where your API will be deployed. For example, a test version of your API could be deployed to a stage named beta.

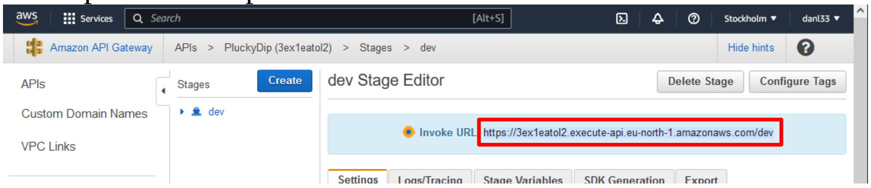
Deployment stage [New Stage]

Stage name* dev

Stage description

Deployment description

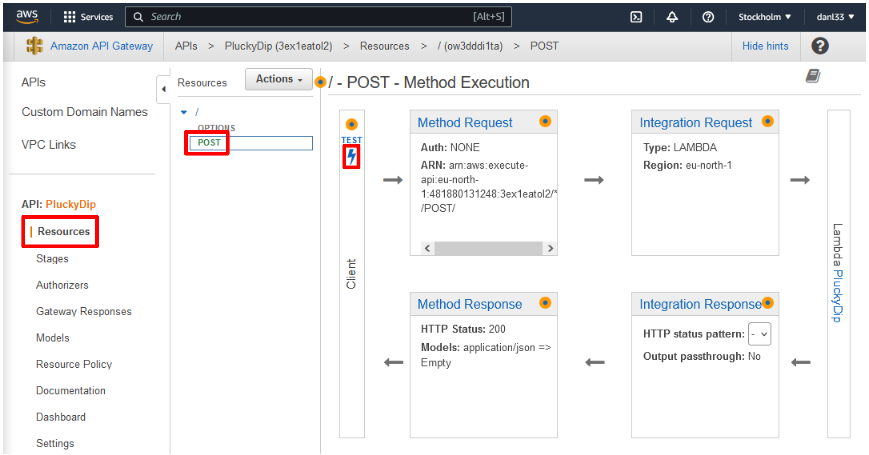
22. Copy the Invoke URL and paste to notepad for future reference



dev Stage Editor

Invoke URL: https://3ex1eatol2.execute-api.eu-north-1.amazonaws.com/dev

23. Click Resources > POST > ⚡



/ - POST - Method Execution

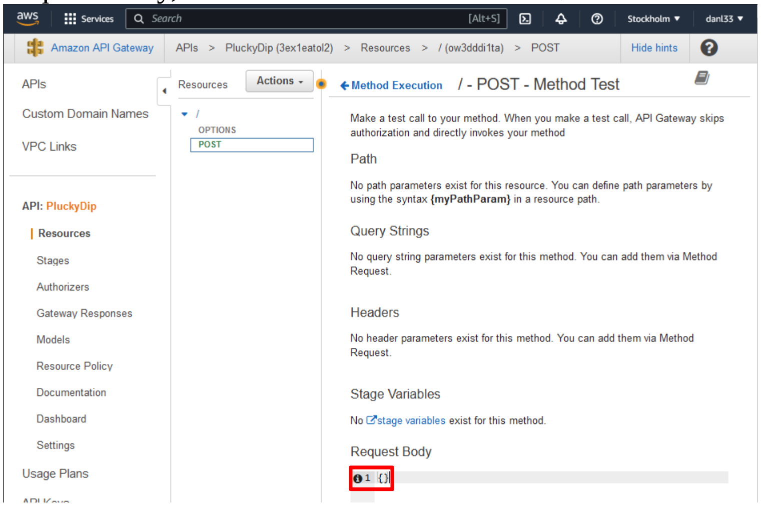
Method Request
Auth: NONE
ARN: arn:aws:execute-api:eu-north-1:481880131248:3ex1eatol2/*/POST/

Integration Request
Type: LAMBDA
Region: eu-north-1

Method Response
HTTP Status: 200
Models: application/json => Empty

Integration Response
HTTP status pattern: -
Output passthrough: No

24. Again, put “{}” into the Request body, then scroll down to click the ⚡Test button



/ - POST - Method Test

Make a test call to your method. When you make a test call, API Gateway skips authorization and directly invokes your method

Path

No path parameters exist for this resource. You can define path parameters by using the syntax {myPathParam} in a resource path.

Query Strings

No query string parameters exist for this method. You can add them via Method Request.

Headers

No header parameters exist for this method. You can add them via Method Request.

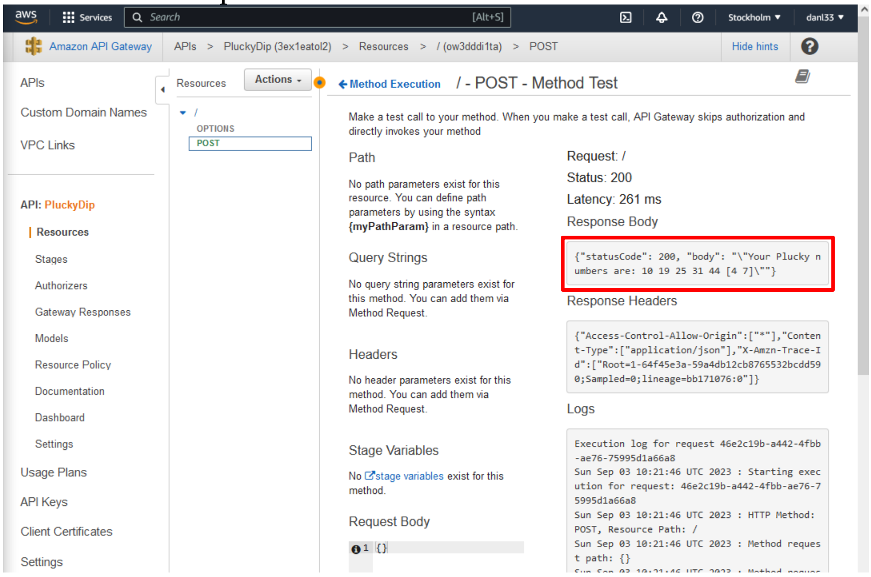
Stage Variables

No stage variables exist for this method.

Request Body

```
1 {}
```

25. The results will show a successful response



/ - POST - Method Test

Make a test call to your method. When you make a test call, API Gateway skips authorization and directly invokes your method

Path

No path parameters exist for this resource. You can define path parameters by using the syntax {myPathParam} in a resource path.

Query Strings

No query string parameters exist for this method. You can add them via Method Request.

Headers

No header parameters exist for this method. You can add them via Method Request.

Stage Variables

No stage variables exist for this method.

Request Body

```
1 {}
```

Request: /

Status: 200

Latency: 261 ms

Response Body

```
{"statusCode": 200, "body": "\"Your Plucky n
umbers are: 10 19 25 31 44 [4 7]\""}
```

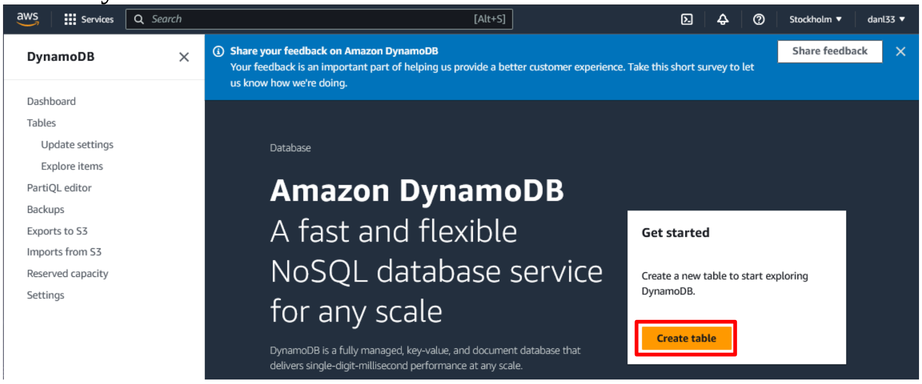
Response Headers

```
{"Access-Control-Allow-Origin":["*"],"Conten
t-Type":["application/json"],"X-Amzn-Trace-I
d":["Root=1-64f45e3a-59a4db12cb8765532bcdd59
0;Sampled=0;lineage=bb171076:0"]}
```

Logs

```
Execution log for request 46e2c19b-a442-4fbb
-ae76-75995d1a66a8
Sun Sep 03 10:21:46 UTC 2023 : Starting exec
ution for request: 46e2c19b-a442-4fbb-ae76-7
5995d1a66a8
Sun Sep 03 10:21:46 UTC 2023 : HTTP Method:
POST, Resource Path: /
Sun Sep 03 10:21:46 UTC 2023 : Method reques
t path: {}
```

26. In a new tab navigate to DynamoDB and click Create table



Share your feedback on Amazon DynamoDB

Your feedback is an important part of helping us provide a better customer experience. Take this short survey to let us know how we're doing.

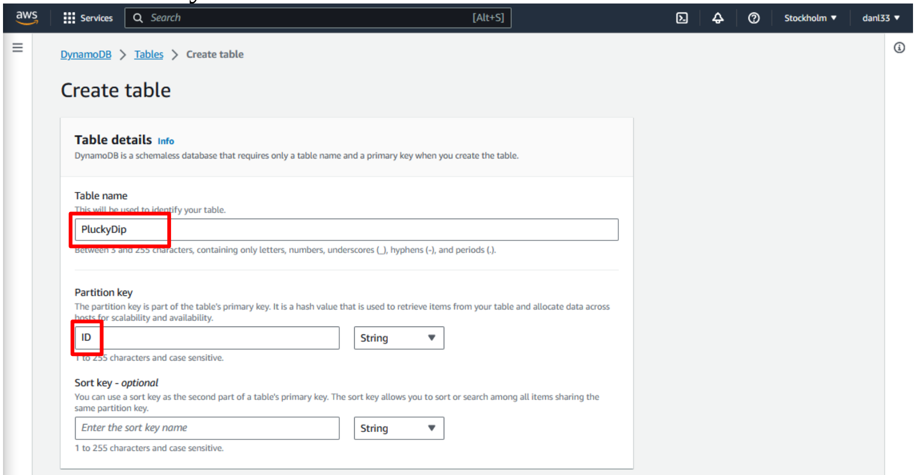
Amazon DynamoDB

A fast and flexible NoSQL database service for any scale

DynamoDB is a fully managed, key-value, and document database that delivers single-digit-millisecond performance at any scale.

Create a new table to start exploring DynamoDB.

27. Set a Table name and Partition key then scroll to the bottom and click the Create table button



Create table

Table details

DynamoDB is a schemaless database that requires only a table name and a primary key when you create the table.

Table name

This will be used to identify your table.

PluckyDip

Between 3 and 255 characters, containing only letters, numbers, underscores (_), hyphens (-), and periods (.).

Partition key

The partition key is part of the table's primary key. It is a hash value that is used to retrieve items from your table and allocate data across hosts for scalability and availability.

ID | String ▾

1 to 255 characters and case sensitive.

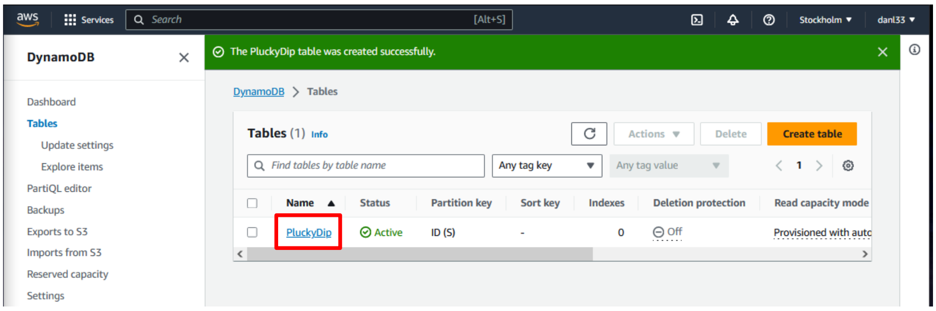
Sort key - *optional*

You can use a sort key as the second part of a table's primary key. The sort key allows you to sort or search among all items sharing the same partition key.

Enter the sort key name | String ▾

1 to 255 characters and case sensitive.

28. Click on on the table once created

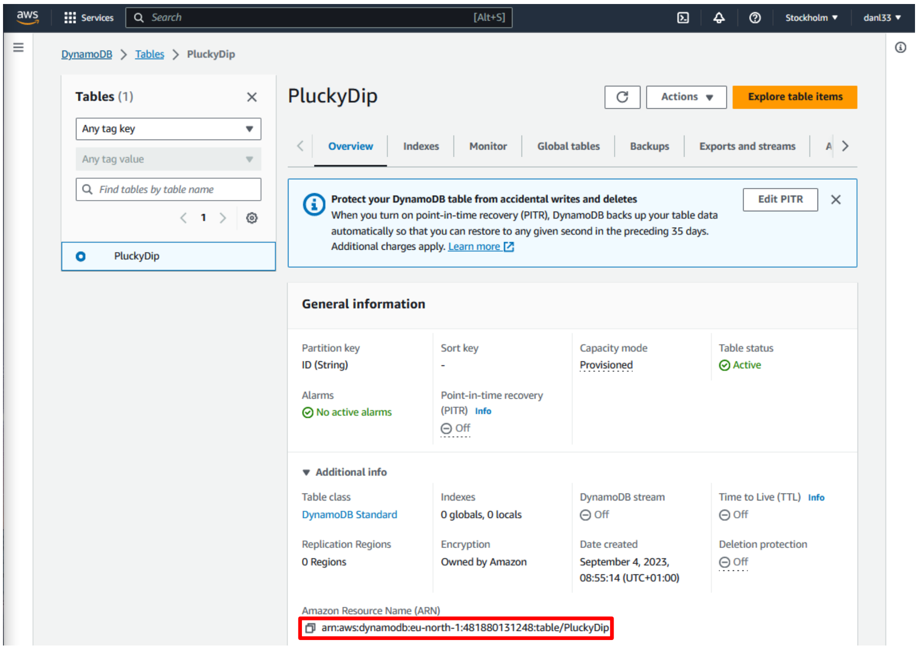


The PluckyDip table was created successfully.

Tables (1)

□	Name ▲	Status	Partition key	Sort key	Indexes	Deletion protection	Read capacity mode
□	PluckyDip	✓ Active	ID (S)	-	0	⊖ Off	Provisioned with autc

29. Under General Information, click on Additonal Info to expand and copy the ARN into notepad



PluckyDip

Protect your DynamoDB table from accidental writes and deletes

When you turn on point-in-time recovery (PITR), DynamoDB backs up your table data automatically so that you can restore to any given second in the preceding 35 days. Additional charges apply. Learn more

General information

Partition key	Sort key	Capacity mode	Table status
ID (String)	-	Provisioned	✓ Active
Alarms	Point-in-time recovery (PITR) Info		
✓ No active alarms	⊖ Off		

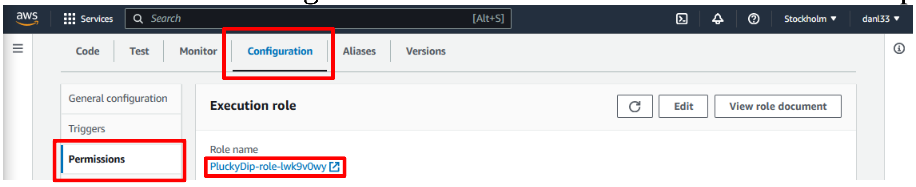
▾ Additional info

Table class	Indexes	DynamoDB stream	Time to Live (TTL) Info
DynamoDB Standard	0 globals, 0 locals	⊖ Off	⊖ Off
Replication Regions	Encryption	Date created	Deletion protection
0 Regions	Owned by Amazon	September 4, 2023, 08:55:14 (UTC+01:00)	⊖ Off

Amazon Resource Name (ARN)

arn:aws:dynamodb:eu-north-1:481880131248:table/PluckyDip

30. Return to the Lambda tab and select Configuration > Permissions and click on the role top open in a new tab



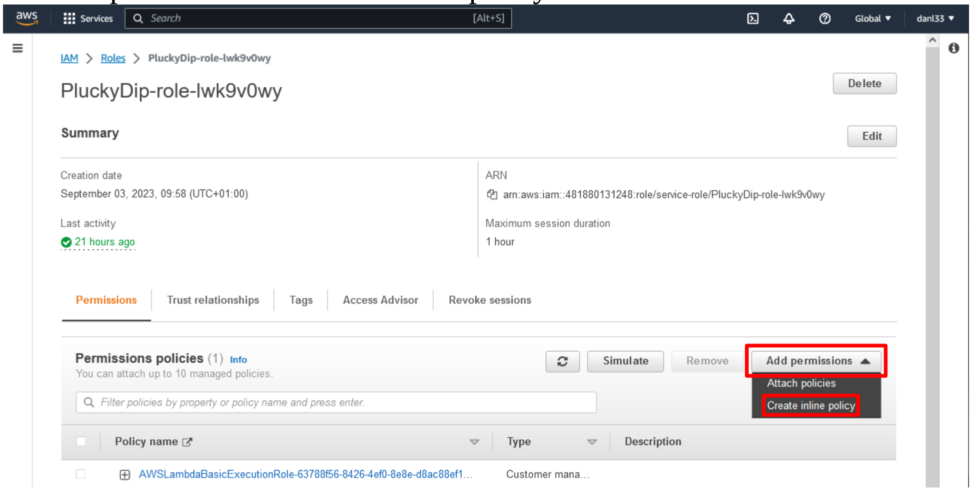
Code | Test | Monitor | Configuration | Aliases | Versions

General configuration
Triggers
Permissions

Execution role

Role name
PluckyDip-role-lwk9v0wy

31. In the new tab click Add permissions > Create inline policy



IAM > Roles > PluckyDip-role-lwk9v0wy

PluckyDip-role-lwk9v0wy

Summary

Creation date
September 03, 2023, 09:58 (UTC+01:00)

ARN
arn:aws:iam::481880131248:role/service-role/PluckyDip-role-lwk9v0wy

Last activity
21 hours ago

Maximum session duration
1 hour

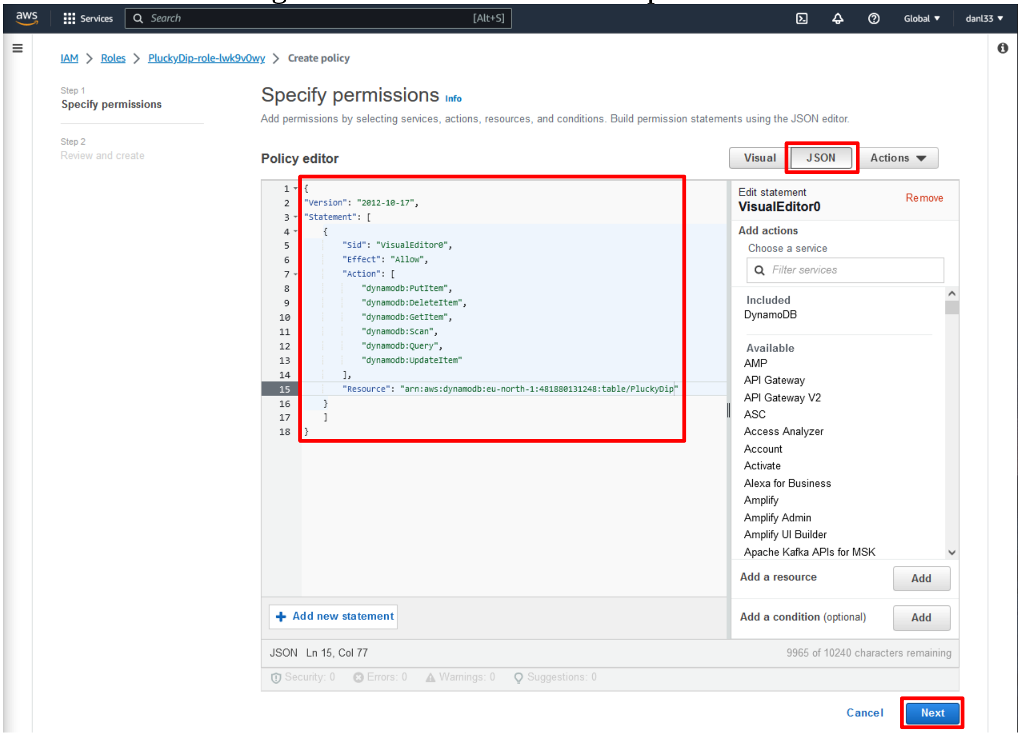
Permissions | Trust relationships | Tags | Access Advisor | Revoke sessions

Permissions policies (1) Info
You can attach up to 10 managed policies.

Add permissions
- Attach policies
- Create inline policy

Policy name	Type	Description
AWSLambdaBasicExecutionRole-63788f56-8426-4ef0-8e8e-d8ac88ef1...	Customer mana...	

32. Click JSON and add in the following which includes the ARN copied earlier



IAM > Roles > PluckyDip-role-lwk9v0wy > Create policy

Step 1
Specify permissions

Step 2
Review and create

Specify permissions Info
Add permissions by selecting services, actions, resources, and conditions. Build permission statements using the JSON editor.

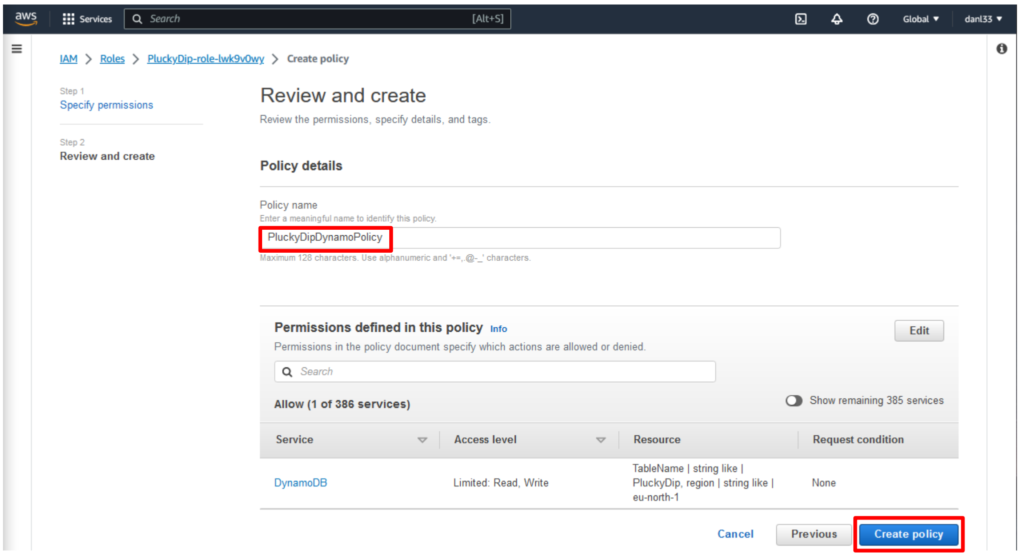
Policy editor

```
{
"Version": "2012-10-17",
"Statement": [
    {
        "Sid": "VisualEditor0",
        "Effect": "Allow",
        "Action": [
            "dynamodb:PutItem",
            "dynamodb:DeleteItem",
            "dynamodb:GetItem",
            "dynamodb:Scan",
            "dynamodb:Query",
            "dynamodb:UpdateItem"
        ],
        "Resource": "arn:aws:dynamodb:eu-north-1:481880131248:table/PluckyDip"
    }
    ]
}
```

JSON Ln 15, Col 77
9965 of 10240 characters remaining

Security: 0 Errors: 0 Warnings: 0 Suggestions: 0

33. Set a policy name



IAM > Roles > PluckyDip-role-lwk9v0wy > Create policy

Step 1
Specify permissions

Step 2
Review and create

Review and create
Review the permissions, specify details, and tags.

Policy details

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

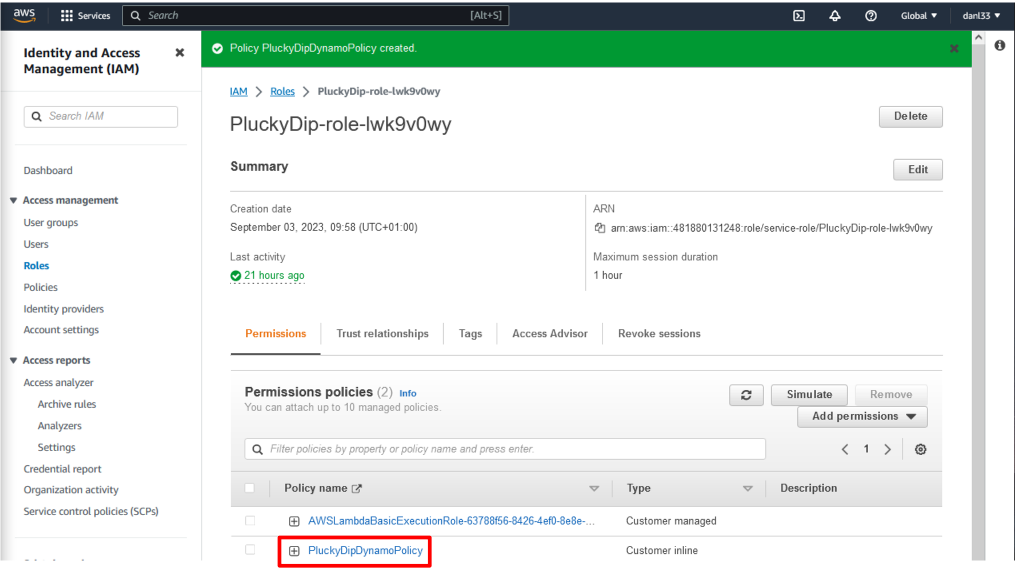
Permissions defined in this policy Info
Permissions in the policy document specify which actions are allowed or denied.

Allow (1 of 386 services)

Show remaining 385 services

Service	Access level	Resource	Request condition
DynamoDB	Limited: Read, Write	TableName | string like | PluckyDip, region | string like | eu-north-1	None

34. The new policy will now be listed as follows



Policy PluckyDipDynamoPolicy created.

IAM > Roles > PluckyDip-role-lwk9v0wy

PluckyDip-role-lwk9v0wy

Summary

Creation date: September 03, 2023, 09:58 (UTC+01:00)

ARN: arn:aws:iam::481880131248:role/service-role/PluckyDip-role-lwk9v0wy

Last activity: 21 hours ago

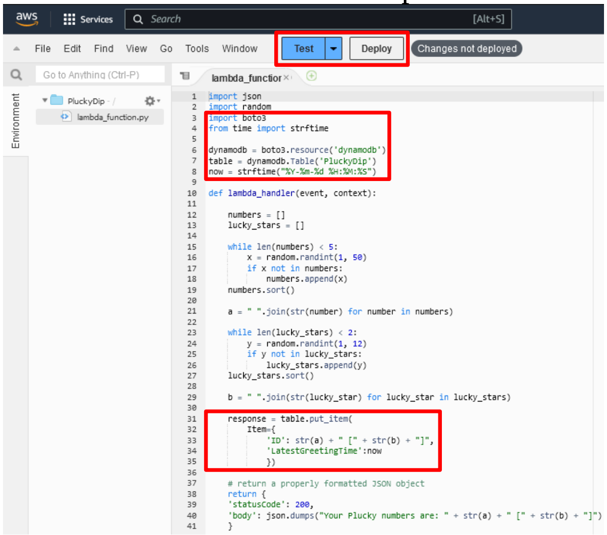
Maximum session duration: 1 hour

Permissions policies (2)

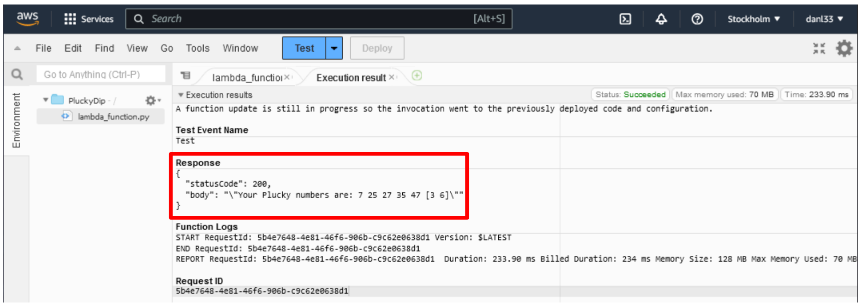
You can attach up to 10 managed policies.

Policy name	Type	Description
AWSLambdaBasicExecutionRole-63788f56-8426-4ef0-8e8e-...	Customer managed	
PluckyDipDynamoPolicy	Customer inline	

35. Return to the Lambda tab, and update the code to enable the results to be written to a DynamoDB table along wit a timestamp. Again, Deploy and Test to acheive a successful response.



```python
import json
import random
import boto3
from time import strftime

dynamodb = boto3.resource('dynamodb')
table = dynamodb.Table('PluckyDip')
now = strftime("%Y-%m-%d %H:%M:%S")

def lambda_handler(event, context):

    numbers = []
    lucky_stars = []

    while len(numbers) < 5:
        x = random.randint(1, 50)
        if x not in numbers:
            numbers.append(x)
    numbers.sort()

    a = " ".join(str(number) for number in numbers)

    while len(lucky_stars) < 2:
        y = random.randint(1, 12)
        if y not in lucky_stars:
            lucky_stars.append(y)
    lucky_stars.sort()

    b = " ".join(str(lucky_star) for lucky_star in lucky_stars)

    response = table.put_item(
        Item={
            'ID': str(a) + " [" + str(b) + "]",
            'LatestGreetingTime':now
            })

    # return a properly formatted JSON object
    return {
    'statusCode': 200,
    'body': json.dumps("Your Plucky numbers are: " + str(a) + " [" + str(b) + "]")
    }
```

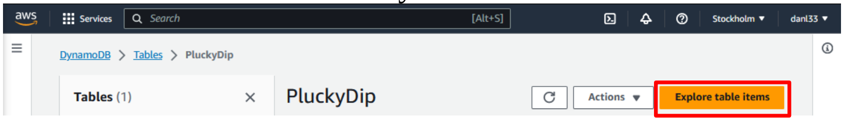


Execution results

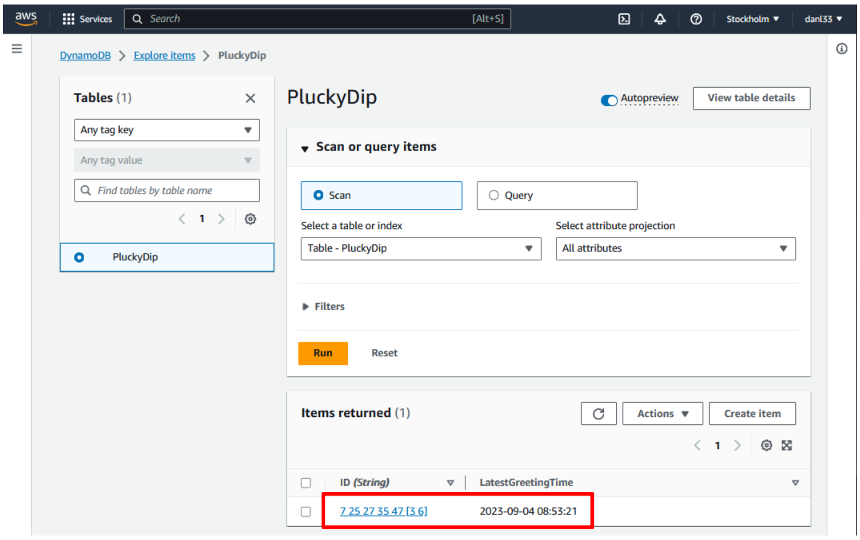
Status: Succeeded | Max memory used: 70 MB | Time: 233.90 ms

```
A function update is still in progress so the invocation went to the previously deployed code and configuration.

Test Event Name
Test

Response
{
  "statusCode": 200,
  "body": "\"Your Plucky numbers are: 7 25 27 35 47 [3 6]\""
}

Function Logs
START RequestId: 5b4e7648-4e81-46f6-906b-c9c62e0638d1 Version: $LATEST
END RequestId: 5b4e7648-4e81-46f6-906b-c9c62e0638d1
REPORT RequestId: 5b4e7648-4e81-46f6-906b-c9c62e0638d1  Duration: 233.90 ms Billed Duration: 234 ms Memory Size: 128 MB Max Memory Used: 70 MB

Request ID
5b4e7648-4e81-46f6-906b-c9c62e0638d1
```

36. Return to the DynamoDB tab and check that the entry has been added



DynamoDB > Tables > PluckyDip



DynamoDB > Explore items > PluckyDip

Tables (1)

PluckyDip

Scan or query items

● Scan ○ Query

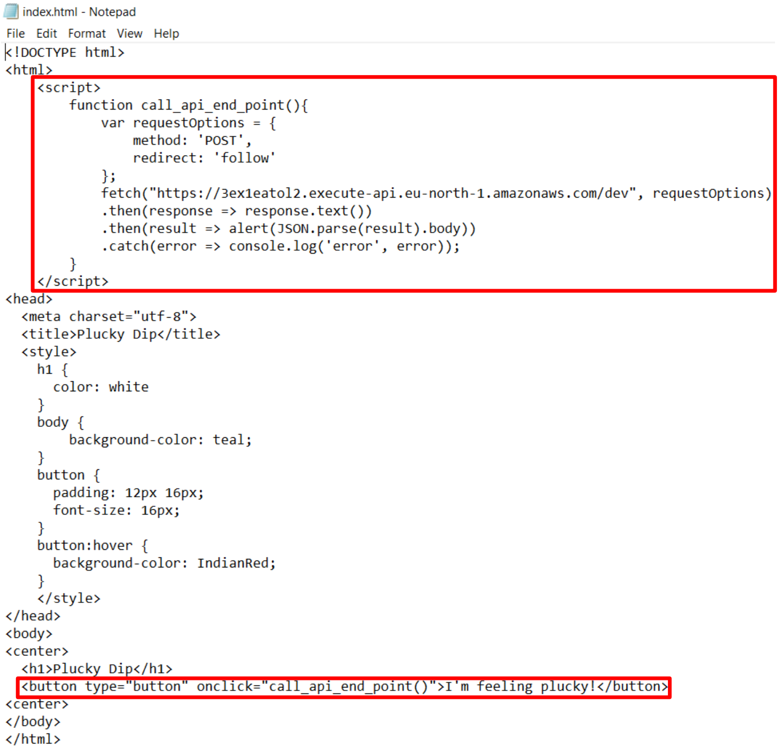
Select a table or index: Table - PluckyDip

Select attribute projection: All attributes

Items returned (1)

ID (String)	LatestGreetingTime
7 25 27 35 47 [3 6]	2023-09-04 08:53:21

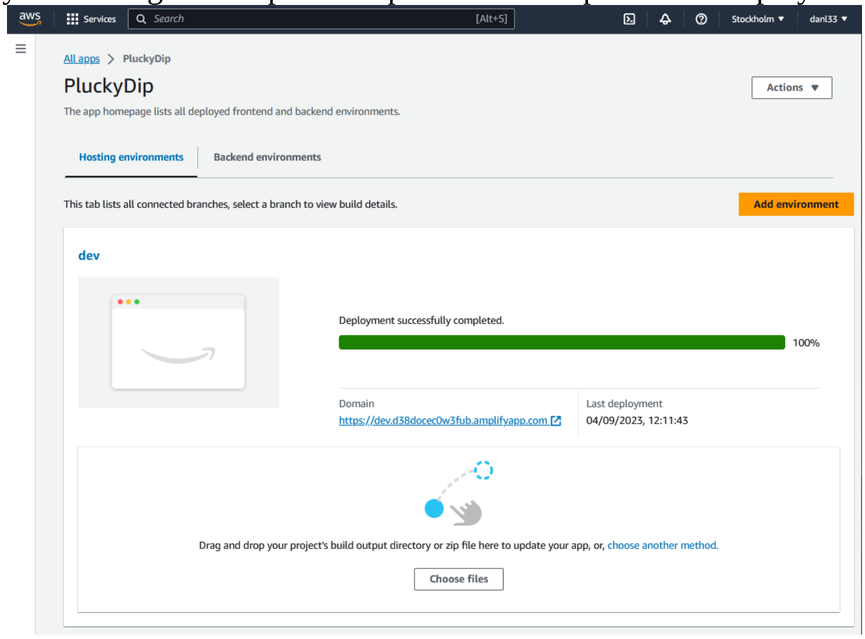
37. Re-open the index.html file in notepad and update the code to call the API on the button click. Save, close and zip



index.html - Notepad

```
<!DOCTYPE html>
<html>
    <script>
        function call_api_end_point(){
            var requestOptions = {
                method: 'POST',
                redirect: 'follow'
            };
            fetch("https://3ex1eatol2.execute-api.eu-north-1.amazonaws.com/dev", requestOptions)
            .then(response => response.text())
            .then(result => alert(JSON.parse(result).body))
            .catch(error => console.log('error', error));
        }
    </script>
<head>
  <meta charset="utf-8">
  <title>Plucky Dip</title>
  <style>
    h1 {
      color: white
    }
    body {
        background-color: teal;
    }
    button {
      padding: 12px 16px;
      font-size: 16px;
    }
    button:hover {
      background-color: IndianRed;
    }
    </style>
</head>
<body>
<center>
  <h1>Plucky Dip</h1>
  <button type="button" onclick="call_api_end_point()">I'm feeling plucky!</button>
<center>
</body>
</html>
```

38. Return to the Amplify tab to drag and drop in the updated index.zip file to re-deploy Plucky Dip



All apps > PluckyDip

PluckyDip

The app homepage lists all deployed frontend and backend environments.

Hosting environments | Backend environments

This tab lists all connected branches, select a branch to view build details.

dev

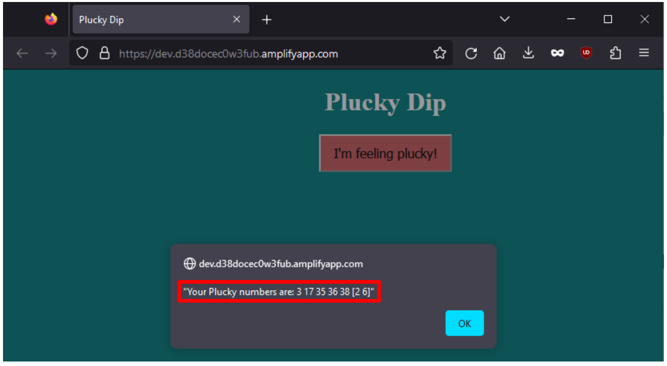
Deployment successfully completed. 100%

Domain: https://dev.d38docec0w3fub.amplifyapp.com

Last deployment: 04/09/2023, 12:11:43

Drag and drop your project's build output directory or zip file here to update your app, or, choose another method.

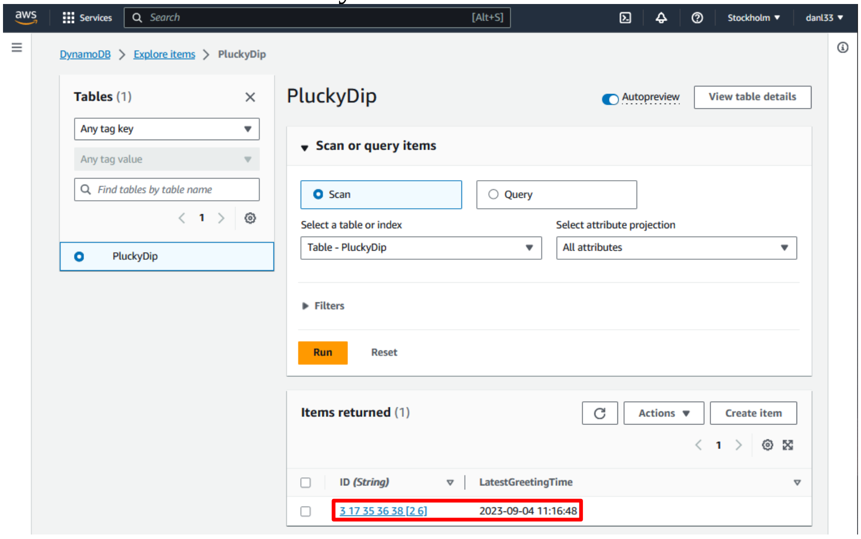
39. Return to the Plucky Dip tab, refresh and click the button to call the API



dev.d38docec0w3fub.amplifyapp.com

"Your Plucky numbers are: 3 17 35 36 38 [2 6]"

40. Return to the DynamoDB tab to ensure the entry was written to the table



● Scan ○ Query

Select a table or index: Table - PluckyDip

Select attribute projection: All attributes

Items returned (1)

□	ID (String)	LatestGreetingTime
□	3 17 35 36 38 [2 6]	2023-09-04 11:16:48