Serverless AWS Project:

Building a CRUD Serverless API with AWS Lambda, API Gateway, and a DynamoDB with cloudwatch and SNS notifications.

Problem Statement:

-A client has built a user registration web application for a company. The front-end is written in JavaScript and has to function with web API; while the backend is written in Python. The client wants this application to be deployed on AWS using serverless service options for the backend. Also, the client wants to monitor the application and get notified in case they are errors or too much traffic.

Solution Statement:

I am going to create an API with endpoints so that the frontend can use it, and also, create a lambda to manage the request from the API gateway and a dynamodb to store data. The lambda function will also communicate with the dynamodb. To store the frontend JavaScript code, I am going to create an Amazon S3 bucket to hold the code and an Amazon CloudFront distribution to serve the website globally; while Amazon Route 53 will manage the domain name, and AWS Certificate Manager will provide a valid SSL/TLS certificate. Lastly, AWS Cloudwatch is going to manage the logs.

Some Endpoints for the project:

-Health check: This endpoint will be used to study the applications' health. It has a GET method to return a 200 response when called.

-User: This endpoint will support GET, POST, DELETE, and PUT methods. It will be used to create, modify, delete, or display users' information.

-Users: This endpoint will be used to list all the users in the database.

Services to be used:

-AWS Lambda

-AWS API Gateway

-DynamoDB

-Amazon S3

-Amazon CloudFront

-Amazon Route 53

-AWS Certificate Manager

- AWS SNS

- AWS Cloudwatch

Lambda function description:

-The Lambda function is going to create, reads, updates, and deletes from DynamoDB. Also, the function uses events from API Gateway to determine how to interact with DynamoDB.

Creating a DynamoDB database

Creating an IAM role so that the lambda function will use to access the database

Role permissions

Step 2: Add permissions

[Edit]

Permissions policy summary

Policy name ↗		Type		Attached as	
AmazonDynamoDBFullAccess		AWS managed		Permissions policy	
CloudWatchFullAccess		AWS managed		Permissions policy	

Creating a Lambda function

Create function   Info

AWS Serverless Application Repository applications have moved to Create application.

Author from scratch ⦿	**Use a blueprint** ○	**Container imag**
Start with a simple Hello World example.	Build a Lambda application from sample code and configuration presets for common use cases.	Select a container i

Basic information

Function name
Enter a name that describes the purpose of your function.

    serverless-api-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 3.9   ▼

Architecture   Info
Choose the instruction set architecture you want for your function code.

Files for the Lambda process

Code source   Info

File   Edit   Find   View   Go   Tools   Window   [Test ▾]   [Deploy]   [Changes not deployed]

Go to Anything (Ctrl-P)

lambda_function ×   custom_encoder. ×   ⊕

▼ 📁 serverless-api-lambi ⚙▾
   📄 custom_encoder.py
   📄 lambda_function.py

```
122              'UpdateAttributes': response
123         }
124         return buildResponse(200, body)
125    except:
126         logger.exception('Do your custom error handling here. I am just
127
128 def deleteProduct(productId):
```

Create and Configure the API gateway

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 endp[...]

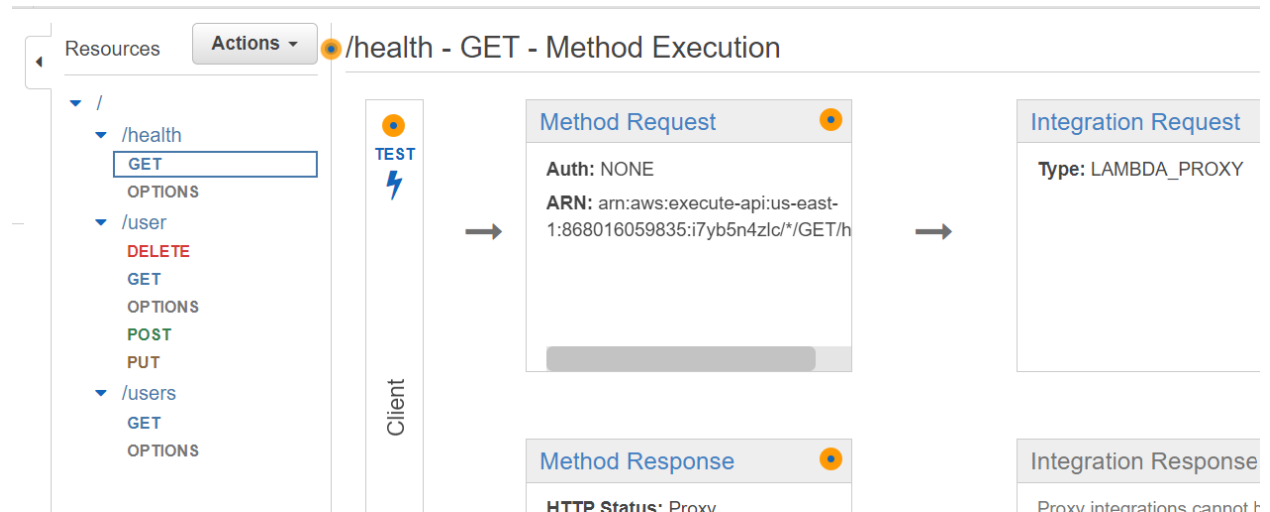⦿ **New API**   ○ **Import from Swagger or Open API 3**   ○ **Example API**

Settings

Choose a friendly name and description for your API.

API name*	serverless-rest-api

Create various endpoints for the API

After Deploying the API, now it's time to host the JavaScript application on S3

Create bucket Info

Buckets are containers for data stored in S3. Learn more ☑

General configuration

Bucket name

javascript-app

Bucket name must be globally unique and must not contain spaces or uppercase letters. See rules for bucket naming ☑

AWS Region

US East (N. Virginia) us-east-1 ▼

Copy settings from existing bucket - *optional*
Only the bucket settings in the following configuration are copied.

Configuring the S3 bucket to host the static website

Amazon S3 > Buckets > javascript-app > Edit static website hosting

Edit static website hosting Info

Static website hosting

Use this bucket to host a website or redirect requests. Learn more ☑

Static website hosting

○ Disable

● Enable

Hosting type

● Host a static website
  Use the bucket endpoint as the web address. Learn more ☑

○ Redirect requests for an object
  Redirect requests to another bucket or domain. Learn more ☑

S3 static website URL

Static website hosting
Use this bucket to host a website or redirect requests. Learn more ⬈

Static website hosting

Enabled

Hosting type

Bucket hosting

Bucket website endpoint

When you configure your bucket as a static website, the website is available at the AWS Region-specific website endpoint of the bucket. Learn more

📋 http://javascript-app.s3-website-us-east-1.amazonaws.com ⬈

Accessing the website with the S3 URL

↻ ⚠ Not secure | javascript-app.s3-website-us-east-1.amazonaws.com   ⬆ ☆ ⓖ

gister

Users

#	Avatar	First	Last	Username	Action
9982		Jon	Olv	Jo98	Edit Del
4524		Andrew	Bien	ad95	Edit Del

Creating a CloudFront distribution for the site

Create distribution

Origin

Origin domain
Choose an AWS origin, or enter your origin's domain name.

🔍 javascript-app.s3-website-us-east-1.amazonaws.com   ✕

Origin path - *optional*   Info
Enter a URL path to append to the origin domain name for origin requests.

Enter the origin path

Name
Enter a name for this origin.

javascript-app.s3-website-us-east-1.amazonaws.com

Add custom header - *optional*

Selecting a previous CNAME record that I created for the website

Alternate domain name (CNAME) - *optional*
Add the custom domain names that you use in URLs for the files served by this distribution.

niandreprojectforaws.online   **Remove**

Add item

ⓘ To add a list of alternative domain names, use the bulk editor.

Custom SSL certificate - *optional*
Associate a certificate from AWS Certificate Manager. The certificate must be in the US East (N. Virginia) Region (us-east-1).

niandreprojectforaws.online (7c99e4ee-9387-4234-a2f9-77470859a525)   ▼   ⟳

⊘ niandreprojectforaws.online ↗   Request certificate ↗

— Legacy clients support - $600/month prorated charge applies. Most customers do not need this.
  CloudFront allocates dedicated IP addresses at each CloudFront edge location to serve your content over HTTPS.
  ☐ Enabled

— Security policy
  The security policy determines the SSL or TLS protocol and the specific ciphers that CloudFront uses for HTTPS connections with
  viewers (clients).

CloudFront distribution with the alternative name

E38U7H18YP42TQ

General   Origins   Behaviors   Error pages   Geographic restrictions   Invalidations   Tags

Details

Distribution domain name
⧉ d3r3zq2k51l7pt.cloudfront.net

ARN
⧉ arn:aws:cloudfront::868016059835:distribution/E38U7
H18YP42TQ

Last modified
January 24, 2023 at 1:55:5

Settings

Description
-

Alternate domain names
niandreprojectforaws.online

Standard logging
Off

Price class

Custom SSL certificate

Cookie logging

Accessing the website with the alternative name

Now, we need to integrate Amazon CloudWatch

Creating an IAM role for logging into CloudWatch

Setting up logs for the API gateway

Creating a log destination group in CloudWatch for the logs

Create log group

Log group details

Log group name

cloudwatchloggroupapi

Retention setting

Never expire

KMS key ARN - optional

Tags

A tag is a label that you assign to an Amazon Web Services resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your Amazon Web Services costs.

I created a new user, and I was able to generate these logs

Log streams	Metric filters	Subscription filters	Contributor Insights	Tags	Data protection - *new*

Log streams (3)   C   Delete   Create log stream

Filter log streams or try prefix search   ☐ Exact match

	Log stream		Last event time
☐	cd978e7221e8b06ca709cd40e62a05b3	▽	2023-01-24 16:49:09 (UTC-06:00)
☐	5d43473d64b2731aa0bbf5fcf16bbf12		2023-01-24 16:49:09 (UTC-06:00)
☐	4ba5c501dcec1d3295eb9da0fb9e512e		2023-01-24 16:49:09 (UTC-06:00)

Looking through some of the log streams

CloudWatch > Log groups > API-Gateway-Execution-Logs_i7yb5n4zlc/register > cd978e7221e8b06ca709cd40e62a05b3

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 ⧉

▶	Timestamp	Message
		No older events at this moment. *Retry*
▶	2023-01-24T16:49:09.668-06:00	(3dd1d43c-97a2-4ed4-9317-ef30a3c9d683) Extended Request Id: fRP_6Ex7IAMF2lg=
▶	2023-01-24T16:49:09.668-06:00	(3dd1d43c-97a2-4ed4-9317-ef30a3c9d68… Verifying Usage Plan for request: 3dd1d43c-97a2-4ed4-9317-ef30a3c9d68…
▶	2023-01-24T16:49:09.669-06:00	(3dd1d43c-97a2-4ed4-9317-ef30a3c9d683) API Key authorized because method 'GET /users' does not require API K…
▶	2023-01-24T16:49:09.669-06:00	(3dd1d43c-97a2-4ed4-9317-ef30a3c9d683) Usage Plan check succeeded for API Key and API Stage i7yb5n4zlc/regis…
▶	2023-01-24T16:49:09.670-06:00	(3dd1d43c-97a2-4ed4-9317-ef30a3c9d683) Starting execution for request: 3dd1d43c-97a2-4ed4-9317-ef30a3c9d683
▶	2023-01-24T16:49:09.670-06:00	(3dd1d43c-97a2-4ed4-9317-ef30a3c9d683) HTTP Method: GET, Resource Path: /users
▶	2023-01-24T16:49:09.670-06:00	(3dd1d43c-97a2-4ed4-9317-ef30a3c9d683) Method request path: {}
▶	2023-01-24T16:49:09.670-06:00	(3dd1d43c-97a2-4ed4-9317-ef30a3c9d683) Method request query string: {}
▶	2023-01-24T16:49:09.670-06:00	(3dd1d43c-97a2-4ed4-9317-ef30a3c9d683) Method request headers: {sec-fetch-mode=cors, referer=https://niandre…
▶	2023-01-24T16:49:09.670-06:00	(3dd1d43c-97a2-4ed4-9317-ef30a3c9d683) Method request body before transformations:
▶	2023-01-24T16:49:09.670-06:00	(3dd1d43c-97a2-4ed4-9317-ef30a3c9d683) Endpoint request URI: https://lambda.us-east-1.amazonaws.com/2015-03-…

Configure CloudWatch logging for the AWS Lambda

Verifying to make sure that the role of the Lambda function has a policy

serverless-api-role

Delete

Allows Lambda functions to call AWS services on your behalf.

Summary

Edit

Creation date
January 23, 2023, 17:43 (UTC-06:00)

ARN
⧉ arn:aws:iam::868016059835:role/serverless-api-role

Last activity
✅ 21 minutes ago

Maximum session duration
1 hour

Permissions	Trust relationships	Tags	Access Advisor	Revoke sessions

Permissions policies (2) Info
You can attach up to 10 managed policies.

[🔄] [Simulate] [Remove] [Add permissions ▼]

🔍 Filter policies by property or policy name and press enter.

< 1 > ⚙

☐	Policy name ⧉	▽	Type	▽	Description
☐	⊕ 📦 CloudWatchFullAccess		AWS managed		Provides full access to CloudWatch.

Create a new user and see if the log was sent to the log group

Create user

Brittany

Alexis

brit98

brital@gmail.com

OK

		Action
		Edit Del
		Edit Del
		Edit Del
		Edit Del

Users

#	Avatar	First	Last	Username	Action
2710		kevin	Houst	kv123	Edit Del
9982		Jon	Olv	Jo98	Edit Del
1155		Sarah	moore	sara43	Edit Del
4524		Andrew	Bien	ad95	Edit Del
1854		Brittany	Alexis	brit98	Edit Del

Viewing the Log group for the Lambda function in CloudWatch

CloudWatch > Log groups > /aws/lambda/serverless-api-lambda

/aws/lambda/serverless-api-lambda

[Actions ▼] [View in Logs Insights] [Search log group]

▶ **Log group details**

Log streams | Metric filters | Subscription filters | Contributor Insights | Tags | Data protection - *new*

Log streams (14)   [⟳] [Delete] [Create log stream] [Search all log streams]

[🔍 Filter log streams or try prefix search]   ☐ Exact match   ‹ 1 › ⚙

☐	Log stream ▽	Last event time ▼
☐	2023/01/24/[$LATEST]1403bf753ec84f17b3db4c1d8776a0f4	2023-01-24 17:53:24 (UTC-06:00)
☐	2023/01/24/[$LATEST]03dd7793386d4572bddd9f77036231c5	2023-01-24 17:45:56 (UTC-06:00)
☐	2023/01/24/[$LATEST]261a1723eed74eb2848bc2a20ffb17d7	2023-01-24 17:29:22 (UTC-06:00)
☐	2023/01/24/[$LATEST]59d0f8d3a957404b8456a44692ef6e62	2023-01-24 17:26:49 (UTC-06:00)

Viewing some of the logs from the lambda function: After viewing this log, I realize that I had an error in the Lambda function. I used the keyword "sys" in the code to print out a message, but I forgot to import the sys module in the function.

CloudWatch > Log groups > /aws/lambda/serverless-api-lambda > 2023/01/24/[$LATEST]261a1723eed74eb2848bc2a20ffb17d7

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** 🗗

[⟳] [Actions ▼] [Create metric filter]

[🔍 Filter events]   [Clear] [1m] [30m] [1h] [12h] [Custom ▦]   [Display ▼] ⚙

▶	Timestamp	Message
		No older events at this moment. *Retry*
▶	2023-01-24T17:29:20.808-06:00	INIT_START Runtime Version: python:3.9.v16 Runtime Version ARN: arn:aws:lambda:us-east-1::runtime:07a48df201…
▶	2023-01-24T17:29:21.115-06:00	[INFO] 2023-01-24T23:29:21.115Z Found credentials in environment variables.
▶	2023-01-24T17:29:21.157-06:00	START RequestId: 1c47fdff-d156-4b0c-818a-2aedd395e923 Version: $LATEST
▶	2023-01-24T17:29:21.157-06:00	[INFO] 2023-01-24T23:29:21.157Z 1c47fdff-d156-4b0c-818a-2aedd395e923 {'resource': '/users', 'path': '/users'…
▶	2023-01-24T17:29:21.216-06:00	[ERROR] 2023-01-24T23:29:21.216Z 1c47fdff-d156-4b0c-818a-2aedd395e923 Do your custom error handling here. I …
▶	2023-01-24T17:29:21.217-06:00	[ERROR] NameError: name 'sys' is not defined Traceback (most recent call last): File "/var/task/lambda_fun…
▶	2023-01-24T17:29:21.220-06:00	END RequestId: 1c47fdff-d156-4b0c-818a-2aedd395e923
▶	2023-01-24T17:29:21.220-06:00	REPORT RequestId: 1c47fdff-d156-4b0c-818a-2aedd395e923 Duration: 62.94 ms Billed Duration: 63 ms Memory Size…
▶	2023-01-24T17:29:22.013-06:00	START RequestId: 1bae8965-eebf-4740-82df-0febc275ee1b Version: $LATEST
▶	2023-01-24T17:29:22.013-06:00	[INFO] 2023-01-24T23:29:22.013Z 1bae8965-eebf-4740-82df-0febc275ee1b {'resource': '/users', 'path': '/users'…
▶	2023-01-24T17:29:22.018-06:00	[ERROR] 2023-01-24T23:29:22.018Z 1bae8965-eebf-4740-82df-0febc275ee1b Do your custom error handling here. I …
▶	2023-01-24T17:29:22.018-06:00	[ERROR] NameError: name 'sys' is not defined Traceback (most recent call last): File "/var/task/lambda_fun…

-After fixing the above error, I rerun the function and the problem was fixed. Below is the output of a new log.

Configure CloudWatch logs for DynamoDB

When I created the DynamoDB, CloudWatch come with the newer create database because CloudWatch monitors take the data from DynamoDB and process it into readable metrics. Now, I just need to retrieve CloudWatch data from a table that was created in DynamoDB.

In the CloudWatch Metrics section, we can see our DynamoDB

DynamoDB has different metrics. I need to select the Table Metric

Browse | Query | Graphed metrics | Options | Source
Add math ▾ | Add query ▾

Metrics (22) Info

Graph with SQL | Graph search

N. Virginia ▾ | All > DynamoDB

🔍 Search for any metric, dimension, resource id or account id

| Table Operation Metrics | 5 | Table Metrics | 8 | Account Metrics | 9 |

These are some of the metrics

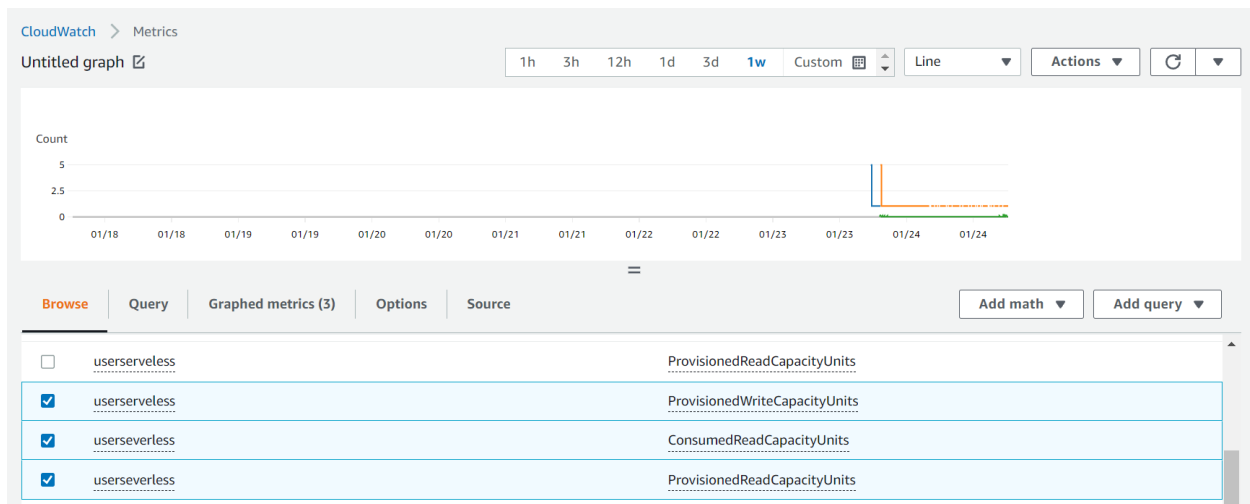**Metrics** (8) Info

Graph with SQL | Gra

N. Virginia ▾ | All > DynamoDB > Table Metrics

🔍 Search for any metric, dimension, resource id or account id

☐	TableName 8/8	▲	Metric name
☐	userserveless		ConsumedReadCapacityUnits
☐	userserveless		ConsumedWriteCapacityUnits
☐	userserveless		ProvisionedReadCapacityUnits

Graph output

CloudWatch > Metrics

Untitled graph ✎

1h | 3h | 12h | 1d | 3d | 1w | Custom 📅 ⬍ | Line ▾ | Actions ▾ | ↻ | ▾

Count
5
2.5
0

01/18 01/18 01/19 01/19 01/20 01/20 01/21 01/21 01/22 01/22 01/23 01/23 01/24 01/24

=

Browse | Query | Graphed metrics (3) | Options | Source
Add math ▾ | Add query ▾

☐	userserveless	ProvisionedReadCapacityUnits
☑	userserveless	ProvisionedWriteCapacityUnits
☑	userseverless	ConsumedReadCapacityUnits
☑	userseverless	ProvisionedReadCapacityUnits

At this moment, the client is going to use the basic analytics for bucket size and several objects, which are free and are calculated automatically to save money. However, if the client wants to view more detailed analytics for S3, the client will need to turn on the request metrics and the cost of the service can add up.