

Program 4 report

By Youssef Beltagy

Here is the webpage url: <https://nosqldatabase-website.azurewebsites.net>

Here is the s3 object url: <https://chicken-in-the-cloud.s3-us-west-2.amazonaws.com/text.txt>

Here is the API url: <https://svj981o6h4.execute-api.us-west-2.amazonaws.com/prod>

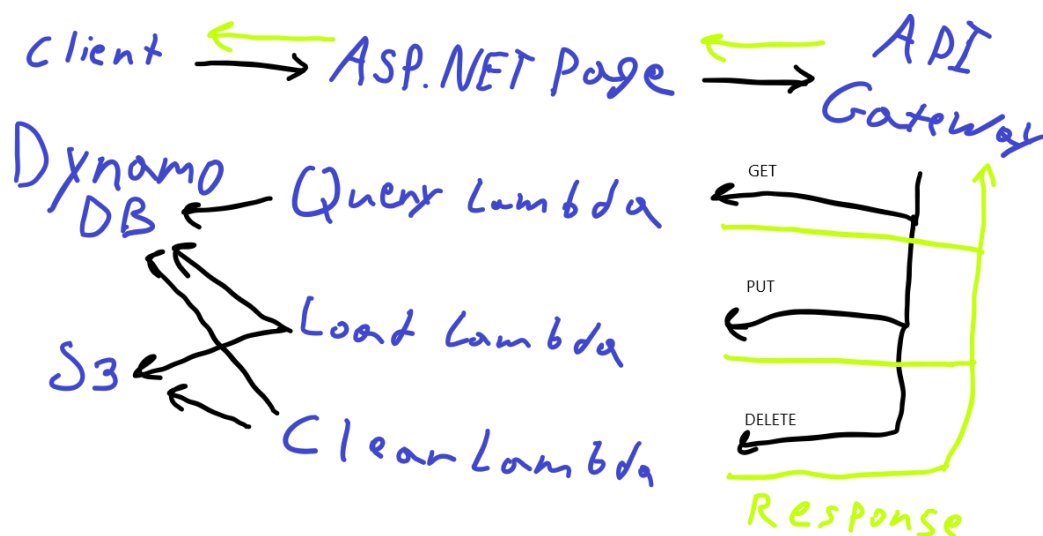
Design

I didn't make a website. I made a restful API using AWS's lambdas.

Initially, I intended to write JavaScript code that will execute in the user's browser, make the API calls, and display the results. However, because of CORS issues, I couldn't do that. So I made an asp.net webpage. If I was more experienced with JavaScript, I could have figured it out.

I wrote Lambda functions to handle all the logic. I provide access to these lambda functions through AWS's API-Gateway. I use GET to query the database, PUT to load data into the database, and DELETE to clear the database.

In this design diagram, the request is black and the response is green.



Scale

The website itself is unimportant and can be scaled horizontally without issues. The API is the bottle neck.

API Gateway and lambda can scale horizontally without issues. They don't even need distributed algorithms.

Because the logic can be scaled horizontally without issues, I expect the bottleneck here to be the storage. The storage must use distributed algorithms to maintain consistency. But since I'm using S3 and DynamoDB, AWS ensures high-scalability for my storage.

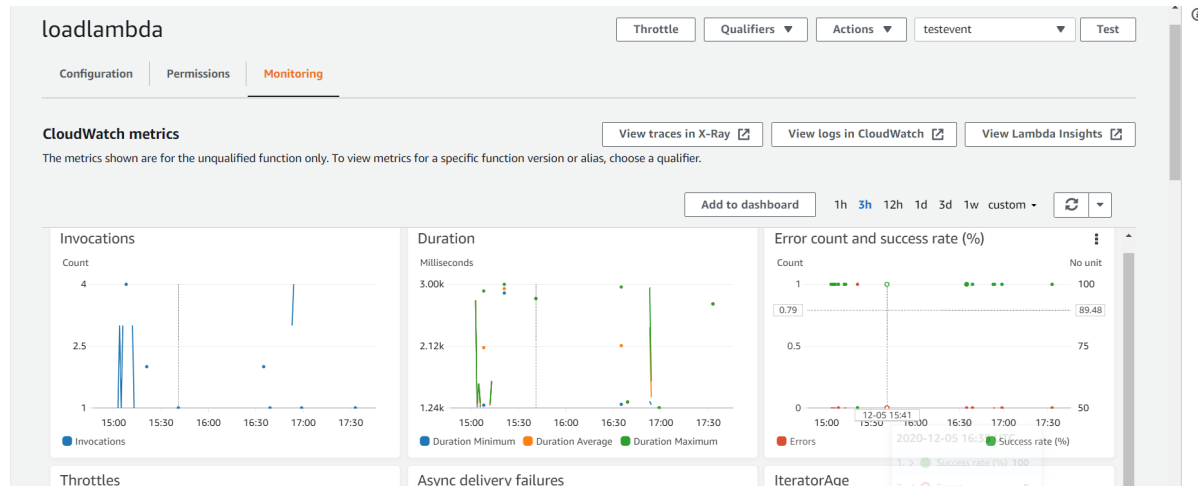
With the current code, this API/Site can be scaled to enterprise sizes without issues. So, I can automate the scaling of this application. I just need the money and incentive to scale it.

Monitoring

I monitor my API through Cloud Watch. However, because I made three separate lambdas, getting a full view of the website is cumbersome. However, I can find out exactly how many times each lambda was called.

In fact, I can monitor S3 and DynamoDB through Cloud Watch as well. I get a lot of information, including the PUT latency of the table.

But my opinion is that all of this information is useless right now because nobody uses my awesome API.



I monitor my APS.NET site through Azure's monitoring. However, I don't care about it that much. It is simply a way for you to access the API.

Function logic

All of my lambdas are in python.

Loadlambda copies Dr. Dimpsey's object to my bucket (with public access) and then uses my object to fill up the DynamoDB table. Make a PUT request with the API link to use Loadlambda.

Clearlambda removes the s3 object and clears the database using scan. Make a delete request to the API link to use ClearLambda.

Querylambda requires at least one input. Querylambda searches for a single entry if it is given both a last name and a first name. If only the last name was given, Querylambda uses the query DynamoDB API method to get all entries with the give last name. If only the last name was give, QueryLambda uses the scan DynamoDB API method to get all entries with the given first name.

SLA

I can multiply the SLA of the services to get an estimation of my service's SLA.

However, without agreeing on a more granular definition, it is hard to get a more precise SLA. I considered getting an SLA per Lambda. Then, for example, I would multiply the s3 SLA twice for Loadlambda because Loadlambda makes two requests to s3. However, that isn't meaningful unless I get a per-transaction SLA from Azure and AWS.

The API's SLA is $0.999 * 0.9999 * 0.9995 * 0.9995$ (s3 * DynamoDB * lambda * API-Gateway) = 0.99790144962. The SLA for my API is 99.79%.

The Website's SLA is $0.99790144962 * 0.9995$ (my-API * App-services) = 0.99740249889. The SLA for my website is 99.74%.

S3

<https://aws.amazon.com/s3/sla/>

99.9%

DynamoDB

<https://aws.amazon.com/dynamodb/sla/>

99.99%

Lambda

<https://aws.amazon.com/lambda/sla/>

99.95%

API Gateway

<https://aws.amazon.com/api-gateway/sla/>

99.95%

Since API Gateway is simply a forwarding service, I'm surprised it isn't close to 100%. Maybe because it ultimately points to a Lambda or a VM which has a 99.95% SLA? I think that makes sense, however, I will still take AWS's SLA at face value and use it.

Azure SLA

https://azure.microsoft.com/en-us/support/legal/sla/app-service/v1_4/

99.95%

Notes

I considered submitting a non-functional url and just put the site up later, but that felt like cheating.

However, I intend to improve the interface of my website. I want it to print the results in a more beautiful way. Since that has nothing to do with the logic.

Here is the current interface:

By Youssef Beltagy

A database for the people and by the people.
The data is loaded from Dr. Dimpsey's S3 object and loaded into the people dynamodb table.
Load and clear are self explanatory. Query must take a first name, a last name or both.

Load

Clear

Query

First Name

Last Name

Output:

© 2020 - People Database

