

Microsoft Word - Assignment3.docx

myfirstfunc - Lambda

← → ↺ 🏠

https://console.aws.amazon.com/lambda/home?region=us-east-1#/functions/myfirstfunc?tab=configuration

☆

aws

Services ▾

🔔

N. Virginia ▾

Support ▾

▲

File Edit Find View Go Tools Window

Test ▾ Deploy

⌵ ⌵ ⌵ ⚙

Environment

myfirstfunc /

index.js

index.js

```
1 var requestTimeArray = [];
2 var timeBetweenInvocationsArray = [];
3
4 exports.handler = async (event) => {
5
6     // STEP 0: RESET if specified by the event parameters
7     if(event.queryStringParameters && event.queryStringParameters.cmd) {
8         if(event.queryStringParameters.cmd == "RESET") {
9             requestTimeArray = [];
10            timeBetweenInvocationsArray = [];
11        }
12    }
13
14    // STEP 1: log the current time and the number of invocations
15    requestTimeArray.push(new Date());
16    var invocationsCount = requestTimeArray.length;
17
18    // STEP 2: If there was a previous invocation, log the amount of time passed since (to the nearest second)
19    var timeSinceLast = requestTimeArray[invocationsCount-1] - requestTimeArray[invocationsCount-2] || 0;
20    if(timeSinceLast != 0){
21        timeSinceLast = Math.round(timeSinceLast / 1000);
22        timeBetweenInvocationsArray.push(timeSinceLast)
23    }
24
25    // STEP 3: compute the average of the logged invocation gaps (to the nearest second)
26    var invocationsGapSum = timeBetweenInvocationsArray.reduce((a,b) => (a+b), 0); // short lambda function to compute the sum of the array
27    var invocationsGapAverage = (invocationsGapSum / timeBetweenInvocationsArray.length);
28    invocationsGapAverage = Math.round(invocationsGapAverage)
29
30    if(invocationsCount == 1){
31        // first invocation or a reset, so only return the ThisInvocation field
32        return {
33            statusCode: 200,
34            body: JSON.stringify({
35                ThisInvocation: requestTimeArray[invocationsCount-1].toISOString()
36            })
37        };
38    }
39    else{
40        return {
41            statusCode: 200,
42            body: JSON.stringify({
43                ThisInvocation: requestTimeArray[invocationsCount-1].toISOString(),
44                TimeSinceLast : timeSinceLast,
45                TotalInvocationsOnThisContainer : invocationsCount,
46                AverageGapBetweenInvocations : invocationsGapAverage
47            })
48        };
49    }
50 }
51 };
52
```

Execution Result ×

Execution results

No execution results yet

52:1

JavaScript

Spaces: 4

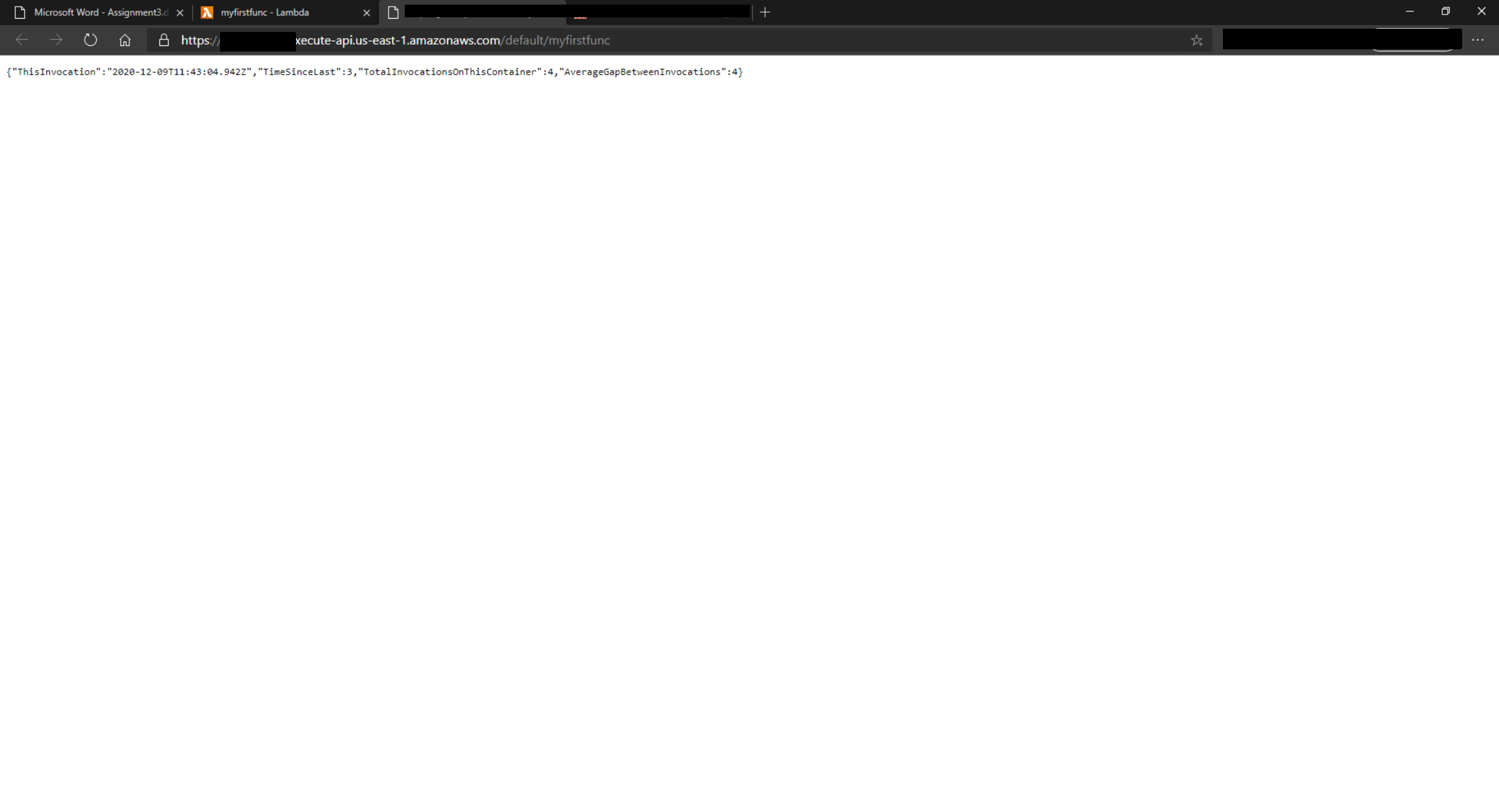
Feedback

English (US) ▾

© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Privacy Policy

Terms of Use



{"ThisInvocation": "2020-12-09T11:43:24.280Z"}