

```
1 let invocations = [];
2
3 exports.handler = async (event) => {
4     let timestamp = new Date().valueOf();
5     let isoString = new Date().toISOString();
6     invocations.push(timestamp);
7
8     let params = event.queryStringParameters;
9     if (params && params.hasOwnProperty('cmd') && params.cmd == 'RESET') {
10         invocations = [];
11         return {
12             statusCode: 200,
13             body: JSON.stringify({ ThisInvocation: isoString })
14         };
15     }
16
17     let timeSinceLast = 0;
18     let averageGap = 0;
19     if (invocations.length > 1) {
20         timeSinceLast = (timestamp - invocations[invocations.length - 2]) / 1000;
21         for (var i = 1; i < invocations.length; i++) {
22             averageGap += invocations[i] - invocations[i - 1];
23         }
24         averageGap /= (invocations.length - 1) * 1000;
25     }
26
27     let response = {
28         statusCode: 200,
29         body: JSON.stringify({
30             ThisInvocation: isoString,
31             TimeSinceLast: timeSinceLast,
32             TotalInvocationsOnThisContainer: invocations.length,
33             AverageGapBetweenInvocations: averageGap
34         })
35     };
36
37     return response;
38 };
```