```javascript
// Pulkit Agarwal 19323939
// Assignment-3 (Short Experiment with Serverless)
let functionCallTimeArray = [];
let counter = 0;
let sum = 0;

exports.handler = async (event) => {
    let ThisInvocation, TimeSinceLast;
    let TotalInvocationsOnThisContainer, AverageGapBetweenInvocations;
    let responseBody, response;
    let totalGap = 0;
    let functionCall = new Date().toISOString();

    let cmd;
    if (event.queryStringParameters && event.queryStringParameters.cmd) {
        cmd = event.queryStringParameters.cmd;
    }

    ThisInvocation = `${functionCall}`;
    functionCallTimeArray[counter] = functionCall; // Storing function calltime

    if (counter == 0) {
        TimeSinceLast = 0; // Function called the first Time
    } else {
        //Time gap between two consecutive function calls in milliseconds
        totalGap = (Date.parse(functionCallTimeArray[counter]) - Date.parse(functionCallTimeArray[counter - 1]));
        TimeSinceLast = `${(totalGap/1000).toFixed(2)}`;
        sum = sum + totalGap; // Sum of all Total Gaps
    }
    // Function call Count, counter + 1 because initialized from 0
    TotalInvocationsOnThisContainer = `${counter + 1}`;

    if (counter == 0){
        // Function called the first Time. Therefore Average is ZERO
        AverageGapBetweenInvocations = 0;
    } else {
        AverageGapBetweenInvocations = `${((sum/(counter+1))/1000).toFixed(2)}`;
    }

    if (cmd == 'RESET') { // All global variables reset to ZERO
        functionCallTimeArray = [];
        counter = 0;
        sum = 0;
        responseBody = { ThisInvocation };

    } else {
        counter = counter + 1;
        responseBody = { ThisInvocation, TimeSinceLast,
            TotalInvocationsOnThisContainer, AverageGapBetweenInvocations };
    }

    response = { body: JSON.stringify(responseBody) };
    return response;
};
```