1/18/22, 7:39 PM Microsoft OpenHack

OVERVIEW OPEN HACK GUIDE OPEN HACK ENVIRONMENT PROVIDE FEEDBACK MESSAGES

1 10 2 3 4 5 6 7 8 9

Process POS sales events

Background

Best For You Organics Company (BFYOC) has started a pilot program to allow their point of sale (POS) systems to send sales information directly to the cloud. The hope is it will allow BFYOC to analyze and respond to sales data from their stores much faster.

Challenge

First, create an Event Hubs namespace with default values in your Azure Subscription, then create a new Event Hub in it **with 32 partitions**. Then identify and copy the following information:

- A shared access key connection string with Send permissions from the shared access policies of your Event Hubs **namespace** (not the Event Hub in the namespace)
- The Event Hub name that you created inside your Event Hubs namespace

With that information make a post call to the /team/registerEventHub method of the <u>OpenHack API System (https://petstore.swagger.io/?url=https://serverlessohmanagementapi.trafficmanager.net/api/definition)</u>

You will be required to post your team table number, the Event Hubs shared access key connection string, and the Event Hub name.

This will trigger the POS system to start sending batches of UTF-8 encoded events with sales details into your Event Hub. Each batch will contain multiple sales events in JSON format from the POS systems.

Here is an example sale event:

```
"header": {
    "salesNumber": "0c423398-3c7c-0682-7519-4701c445ed7a",
    "dateTime": "2019-09-11T06:04:43.6819622-07:00",
    "locationId": "00d8ea6f-935c-2cca-9bbc-f56b5a091621",
    "locationName": "Lorena's Ice Cream Parlor",
    "locationAddress": "865 Olson Cape",
    "locationPostcode": "03030",
    "totalCost": "123.40",
    "totalTax": "12.34",
    "receiptUrl": "https://serverlessohsales.blob.core.windows.net/TheReceipt.pdf"
},
"details":
[
    {
        "productId": "65ab124a-9b2c-4294-a52d-18839364ef15",
        "quantity": "10",
        "unitCost": "10.40",
        "totalCost": "104.00",
        "totalTax": "10.40",
        "productName": "Durian Durian",
        "productDescription": "Smells suspect but tastes... also suspect."
    },
    {
        "productId": "75542e38-563f-436f-adeb-f426f1dabb5c",
        "quantity": "1",
        "unitCost": "3.40",
        "totalCost": "3.40",
        "totalTax": "0.34",
        "productName": "Starfruit Explosion",
        "productDescription": "This starfruit ice cream is out of this world!"
    },
        "productId": "80bab959-ef8b-4ae3-8bf2-e876d77277b6",
        "quantity": "2",
        "unitCost": "8.00",
        "totalCost": "16.00",
        "totalTax": "1.60",
        "productName": "French Vanilla",
        "productDescription": "It's vanilla ice cream."
    }
]
```

Not every record will have a URL for the receiptUrl property. Some records may have a null value.

Your first challenge is to work as a team to implement a solution with an Azure Function using either the Consumption or Premium plan (Serverless) that receives these events from Event Hubs **in batches**, loops through each of them (debatching), and then saves **one entry per sale event** into the same database from the previous challenge. Note: if using CosmosDB this can be in a separate collection.

Your second challenge is to configure your Azure Functions solution to have a maximum event count received per receive loop (maximum batch size) of 64 and a prefetch count value of 256 when retrieving batches from the Event Hub.

Once you have your solution working, make a post call to the /team/boost/:teamTableNumber method of the <u>OpenHack API System</u> (https://petstore.swagger.io/?url=https://serverlessohmanagementapi.trafficmanager.net/api/definition) You will be required to post your team table number. This will enable the system to start sending a larger number of events into your Event Hub.

Your third and final challenge is to monitor your Azure Functions solution and identify the maximum number of instances that your Azure Functions scaled up to 'behind the scenes' after turning boost mode on.

Note: The Serverless Open Hack will automatically reset the rate of events to the initial value 4 hours after your team turns on boost mode. Simply make an additional post call to turn it back on if need be.

Success Criteria

- Demonstrate to your coach your implementation that debatches and inserts those events from Event Hub into your database
- Demonstrate to your coach that your processor is retrieving messages with a maximum batch size of 64 and prefetch count of 256
- Demonstrate how you tracked the number of instances that your Azure Functions scaled up to

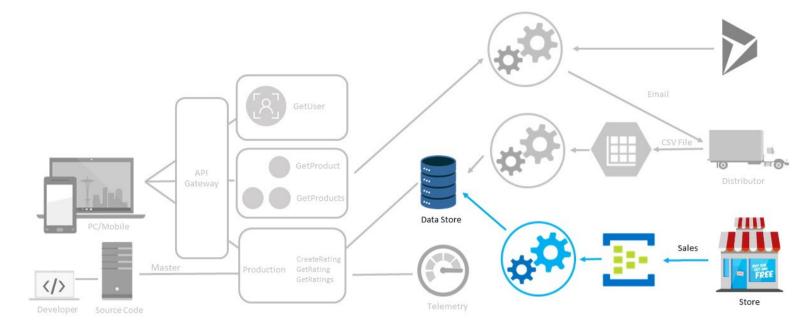
OpenHack © Microsoft 2021. All Rights Reserved - Powered By Opsgility Privacy.

1/18/22, 7:39 PM Microsoft OpenHack

References

- What is Event Hubs? (https://docs.microsoft.com/azure/event-hubs/event-hubs-what-is-event-hubs)
- Scaling with Event Hubs (https://docs.microsoft.com/azure/event-hubs/event-hubs-scalability)
- Azure Event Hubs bindings for Azure Functions (https://docs.microsoft.com/azure/azure-functions/functions-bindings-event-hubs)
- <u>Functions Event Hubs Trigger Scaling (https://docs.microsoft.com/azure/azure-functions/functions-bindings-event-hubs#trigger---scaling)</u>
- Azure Event Hubs Event Processor Host overview (https://docs.microsoft.com/azure/event-hubs/event-hubs-event-processor-host)
- <u>Processing 100,000 Events Per Second on Azure Functions (https://blogs.msdn.microsoft.com/appserviceteam/2017/09/19/processing-100000-events-per-second-on-azure-functions/)</u>
- A tour of Analytics in Application Insights (https://docs.microsoft.com/azure/application-insights/app-insights-analytics-tour)
- Azure Functions scale and hosting (https://docs.microsoft.com/azure/azure-functions/functions-scale)

Progress Diagram



Process POS sales event progress diagram