Icon

Description automatically generated

* [**Attendee Guide**](https://openhackguides.blob.core.windows.net/userguides/OpenHack%20Portal%20User%20Guide%20-%20Attendee.pdf)

[**Logout**](https://openhack.skillmeup.com/account/signout)

* [**OVERVIEW**](https://openhack.skillmeup.com/labengine/modules/microsoft-open-hack-serverless_637802163179013632#labOverview)
* [**OPEN HACK GUIDE**](https://openhack.skillmeup.com/labengine/modules/microsoft-open-hack-serverless_637802163179013632#labGuide)
* [**OPEN HACK ENVIRONMENT**](https://openhack.skillmeup.com/labengine/modules/microsoft-open-hack-serverless_637802163179013632#labEnvironmentTab)
* [**PROVIDE FEEDBACK**](https://openhack.skillmeup.com/labengine/modules/microsoft-open-hack-serverless_637802163179013632#NotesTab)
* [**MESSAGES**](https://openhack.skillmeup.com/labengine/modules/microsoft-open-hack-serverless_637802163179013632#MessagesTab)
* **1** **2** **3** **4** **5** **6** **7** **8** **9** **10**

Development Environment Configuration

Please ensure your machine conforms to these prerequisites.

* A modern laptop running Windows 10, Mac OSX 10.12 or higher, or one of [these Ubuntu versions](https://github.com/Azure/azure-functions-core-tools#linux)
* Your preferred IDE (integrated development environments), like Visual Studio Code or Visual Studio:
  + If using Visual Studio for Windows, make sure you have the [latest Visual Studio version](https://visualstudio.microsoft.com/) with the Azure development option selected.
  + If using Visual Studio Code in Windows, OSX, or Linux make sure you have the latest Visual Studio Code version for your OS, and
    - [VSCode Azure Functions extension](https://marketplace.visualstudio.com/items?itemName=ms-azuretools.vscode-azurefunctions)
    - [Azure Functions Core Tools](https://docs.microsoft.com/azure/azure-functions/functions-run-local#v3) version 3x
  + Optionally, you may just desire to get the Azure Tools extension, which includes functions and many other Azure resources
    - [Azure Tools VSCode Extensions](https://marketplace.visualstudio.com/items?itemName=ms-vscode.vscode-node-azure-pack)

Language specific prerequisites

Depending on the language your team uses, there are additional prerequisites that your machine will need to have. Please review these language specific prerequisites to ensure your machine is ready for the open hack.

C# .NET Core

* Either [Visual Studio](https://visualstudio.microsoft.com/)

or

* [Visual Studio Code](https://code.visualstudio.com/) on one of the [supported platforms](https://code.visualstudio.com/docs/supporting/requirements#_platforms).
  + Use of the Visual Studio Code [C# Extension](https://marketplace.visualstudio.com/items?itemName=ms-vscode.csharp) is recommended.
  + [Azure Functions Core Tools](https://docs.microsoft.com/azure/azure-functions/functions-run-local#v3) version 3x

.Net 6.0 with Azure Functions v4 has not been validated for these OpenHack challenges (and are currently not available for Azure functions v3). Please use Azure Functions v3 and .NET 3.1. Refer to [supported languages](https://docs.microsoft.com/azure/azure-functions/supported-languages) to determine when .Net 6.0 and Azure functions v4 are no longer in preview state. At that time, you may choose to attempt the exercises in Functions v4 and .Net 6.

Java/Maven

* [Java Developer Kit](https://aka.ms/azure-jdks) version 8
* [Apache Maven](https://maven.apache.org/) version 3.0 or later
* [Azure CLI](https://docs.microsoft.com/cli/azure)
* [Azure Functions Core Tools](https://docs.microsoft.com/azure/azure-functions/functions-run-local#v3) version 3.x

Refer to [Azure Functions Java Developer guide](https://docs.microsoft.com/azure/azure-functions/functions-reference-java) for latest version support information

JavaScript

* [Visual Studio Code](https://code.visualstudio.com/) on one of the [supported platforms](https://code.visualstudio.com/docs/supporting/requirements#_platforms)
* [Azure Functions Core Tools](https://docs.microsoft.com/azure/azure-functions/functions-run-local#v3) version 3x
* [Node.js](https://nodejs.org/) Active LTS and Maintenance LTS versions (8.11.1 and 10.14.1 recommended)

Refer to [Azure Function Supported Languages](https://docs.microsoft.com/azure/azure-functions/functions-reference-java) for further Node supported version information

Python

**Previous Versions**

If you have 3.6.x or 3.7.x you **should** have no issues with those versions, just be certain to select the correct version when creating your function apps at Azure.

**Use a VM instead of multiple versions**

It is recommended you do not do a side-by-side installation of a new python version for this challenge. To be clean and avoid potential issues, consider using a low-cost VM running Linux at Azure and install the latest python tools there.

**Use VSCode with the Azure Functions Extension**

It is very easy to work with VSCode with python functions at Azure. Although you may be new to VSCode, it is recommended due to the high availability of extensions that can easily run (locally) and publish Azure functions to your integrated Azure subscription.

* [Python 3.8.x](https://www.python.org/downloads/)
  + Using VSCode with Azure Functions extension makes it easy to select the compiler. language, and project and integrate directly to your function app at Azure, including deployment.
  + VS Code users on Windows, in the past, python interpreter selection can be troublesome. Review [Python in VS Code environments](https://code.visualstudio.com/docs/languages/python#_environments) for assistance with this if you run into issues.
* [VSCode](https://code.visualstudio.com/download)
* [VSCode Azure Functions extension](https://marketplace.visualstudio.com/items?itemName=ms-azuretools.vscode-azurefunctions)
* [Azure Functions Core Tools](https://docs.microsoft.com/azure/azure-functions/functions-run-local#v3) version 3x. You need this to run/test locally.
* [Azure CLI](https://docs.microsoft.com/cli/azure)
* Recommended for Mac OSX and Linux users to install [Azurite v2](https://www.nuget.org/packages/Azurite/)
* [VSCode Azure Storage extension](https://marketplace.visualstudio.com/items?itemName=ms-azuretools.vscode-azurestorage)

Optional - Infrastructure as Code

Team members who are participating to learn how to support and manage “serverless” services in Azure should prepare their machines with:

* IDE of Choice
* Azure CLI
* Extensions or tooling required to support Azure deployments via ARM, Bicep, Terraform, etc.

OpenHack © Microsoft 2021. All Rights Reserved - Powered By [Opsgility](https://opsgility.com/" \t "_blank) [Privacy.](https://openhack.skillmeup.com/Home/Privacy)

Icon

Description automatically generated

* [**Attendee Guide**](https://openhackguides.blob.core.windows.net/userguides/OpenHack%20Portal%20User%20Guide%20-%20Attendee.pdf)

[**Logout**](https://openhack.skillmeup.com/account/signout)

* [**OVERVIEW**](https://openhack.skillmeup.com/labengine/modules/microsoft-open-hack-serverless_637802163179013632#labOverview)
* [**OPEN HACK GUIDE**](https://openhack.skillmeup.com/labengine/modules/microsoft-open-hack-serverless_637802163179013632#labGuide)
* [**OPEN HACK ENVIRONMENT**](https://openhack.skillmeup.com/labengine/modules/microsoft-open-hack-serverless_637802163179013632#labEnvironmentTab)
* [**PROVIDE FEEDBACK**](https://openhack.skillmeup.com/labengine/modules/microsoft-open-hack-serverless_637802163179013632#NotesTab)
* [**MESSAGES**](https://openhack.skillmeup.com/labengine/modules/microsoft-open-hack-serverless_637802163179013632#MessagesTab)
* **1** **2** **3** **4** **5** **6** **7** **8** **9** **10**

Create your first serverless function & workflow

Background

In this challenge you will start creating your first serverless functions and workflows, which are key skills for the upcoming challenges which make up this OpenHack.

Challenge

Your first goal is for each team member to create and debug an HTTP triggered Azure Function locally on your own development machine. This function has the following requirements:

* **Verb**: GET
* **Input parameters**: A productId value is passed to it as a query parameter or an HTTP route
* **Result**: “The product name for your product id {the id passed in to the function} is Starfruit Explosion”

The second goal is for each team member to create their own Azure Resource Group in your team’s Azure Subscription, and then deploy the function from the first goal to it. **Make sure to use different, unique Function App names!** Then test the function after deployed.

Your final goal for this challenge is for each team member to create an HTTP triggered Logic App in their Resource Group. The Logic App has the following trigger requirements:

* **Verb**: POST
* **Input payload example**:
* {
* "productId": "75542e38-563f-436f-adeb-f426f1dabb5c"

}

Next, you should configure the Logic App to call the Azure Function created earlier, passing the productId into it, then reply successfully with the response text content of **“{response from function} and the description is This starfruit ice cream is out of this world!”**.

So, for the example inputs above the Logic App would reply with **“The product name for your product id 75542e38-563f-436f-adeb-f426f1dabb5c is Starfruit Explosion and the description is This starfruit ice cream is out of this world!”**.

**Hint**: To pass query parameters from Logic Apps to a function in the Azure Functions action, you can write the name and values as JSON in the Queries field. Example:

{

"queryParamName": "queryparamValue"

}

Success Criteria

*Given the complexity and variety of potential solutions, the goals for all challenge are verified by your team’s coach.*

* Everyone on your team must demonstrate that you created an HTTP triggered Azure Function locally that accepts a GET verb. Show to your coach that you can call that function and receive a successful response with the right text provided and the value of productId properly filled in.
* Everyone on your team must demonstrate that you deployed that HTTP triggered Azure Function to Azure. Show to your coach that you can call that deployed function and receive a successful response.
* Everyone on your team must demonstrate that you can call the Logic App via HTTP POST which then calls the Function, and that it replies successfully with the correct extended text.
* Infrastructure Bonus: If your team has a member working on IaC, demonstrate the ability to deploy an Azure Function and Logic App services with code, instead of performing the tasks related to the other success critera.

References

* [An introduction to Azure Functions](https://docs.microsoft.com/azure/azure-functions/functions-overview)
* [Supported languages in Azure Functions](https://docs.microsoft.com/azure/azure-functions/supported-languages)
* [Write Azure functions in VSCode](https://docs.microsoft.com/azure/azure-functions/functions-develop-vs-code?tabs=csharp)
* [Test your project locally](https://docs.microsoft.com/azure/developer/javascript/tutorial-vscode-serverless-node-03)
* [Strategies for testing your code in Azure Functions](https://docs.microsoft.com/azure/azure-functions/functions-test-a-function)
* [Code and test Azure Functions locally](https://docs.microsoft.com/azure/azure-functions/functions-develop-local)
* [Create a function on Linux using a custom container](https://docs.microsoft.com/azure/azure-functions/functions-create-function-linux-custom-image)
* [Logic Apps Overview](https://docs.microsoft.com/azure/logic-apps/logic-apps-overview)
* [Create a Logic App](https://docs.microsoft.com/azure/logic-apps/quickstart-create-first-logic-app-workflow)
* [Logic Apps Call, trigger, or nest workflows with HTTP endpoints in Logic Apps](https://docs.microsoft.com/azure/logic-apps/logic-apps-http-endpoint)
* [First Azure Function Visual Studio](https://docs.microsoft.com/azure/azure-functions/functions-create-your-first-function-visual-studio)
* [Azure Functions Python developer guide](https://docs.microsoft.com/azure/azure-functions/functions-reference-python)
* [Azure Functions create first Python Function](https://docs.microsoft.com/azure/azure-functions/functions-create-first-function-python) for latest version/environment configuration information.

Bonus - Infrastructure as Code

It’s important for all teams to be able to deploy and experiment with new technologies to help determine how they can be used, but at some point they must be operationalized. While the application developers are learning to work with Azure Functions and Logic Apps, develop code that can deploy those services in a repeatable manner using a consistant naming pattern. Consider common domain specific languages for infrastructure deployments and work with your team so they can be prepared to use these components for challenges going forward.

* [What is Infrastructure as Code?](https://docs.microsoft.com/en-us/devops/deliver/what-is-infrastructure-as-code)
* [Cloud-Native IaC](https://docs.microsoft.com/en-us/dotnet/architecture/cloud-native/infrastructure-as-code)
* [What is Azure Resource Manager](https://docs.microsoft.com/en-us/azure/azure-resource-manager/management/overview)
* [Azure Resource Manager Templates](https://azure.microsoft.com/en-in/services/arm-templates/)
* [Terraform AzureRM](https://registry.terraform.io/providers/hashicorp/azurerm/latest/docs)
* [Azure Bicep](https://github.com/Azure/bicep)

OpenHack © Microsoft 2021. All Rights Reserved - Powered By [Opsgility](https://opsgility.com/" \t "_blank) [Privacy.](https://openhack.skillmeup.com/Home/Privacy)

Icon

Description automatically generated

* [**Attendee Guide**](https://openhackguides.blob.core.windows.net/userguides/OpenHack%20Portal%20User%20Guide%20-%20Attendee.pdf)

[**Logout**](https://openhack.skillmeup.com/account/signout)

* [**OVERVIEW**](https://openhack.skillmeup.com/labengine/modules/microsoft-open-hack-serverless_637802163179013632#labOverview)
* [**OPEN HACK GUIDE**](https://openhack.skillmeup.com/labengine/modules/microsoft-open-hack-serverless_637802163179013632#labGuide)
* [**OPEN HACK ENVIRONMENT**](https://openhack.skillmeup.com/labengine/modules/microsoft-open-hack-serverless_637802163179013632#labEnvironmentTab)
* [**PROVIDE FEEDBACK**](https://openhack.skillmeup.com/labengine/modules/microsoft-open-hack-serverless_637802163179013632#NotesTab)
* [**MESSAGES**](https://openhack.skillmeup.com/labengine/modules/microsoft-open-hack-serverless_637802163179013632#MessagesTab)
* **1** **2** **3** **4** **5** **6** **7** **8** **9** **10**

Finish the Ice Cream Ratings API

Background

Best For You Organics Company (BFYOC) has begun creation of a mobile application and public facing website, both of which will be used by customers to submit ratings and feedback of their ice cream. Both the website and application will be calling a set of APIs. BFYOC would like your team to create APIs to allow customers to submit ratings, and for business users to retrieve the ratings for reporting purposes.

It’s difficult to predict the load generated by user reviews which can be impacted by several factors. Introduction or cancellation of a flavor will cause a spike in reviews, as will local events, like concerts or an unexpected heatwave. As a result, the solution you implement must be able to scale dynamically, and minimize costs whenever possible.

BFYOC would like to ensure that the proper engineering fundamentals are implemented from the beginning of the application development and want to ensure the team establishes a pipeline where updates are [integrated](https://docs.microsoft.com/azure/devops/learn/what-is-continuous-integration/) and [deployed](https://docs.microsoft.com/azure/devops/learn/what-is-continuous-delivery/) continuously. For compliance reasons, they also need to be able to track changes, rollback any deployments as necessary. The BFYOC security team will be auditing CI/CD pipelines to ensure that sensitive secrets are properly stored and accessed securely using RBAC.

Challenge

Setup

Your team has been asked to implement a solution that will automatically deploy the Ratings API. The solution needs to include a process to revert to a previous (code) commit and redeploy, as well as tracking all changes made to the code.

Your solution should handle automatically building the code every time it is checked/merged in to the master branch and generate an artifact that can be deployed. Assuming the build succeeds and applicable unit tests pass, the artifact should then be automatically deployed to an instance of your Ratings API.

Take this opportunity to discuss as a team how you will be working as team. Implement the appropriate branching strategies that will enable collaboration.

If someone on your team will be developing code for deploying the infrastructure, consider ways to integrate this code deployment as well.

API Development

This challenge will require information about Users and Products. Three APIs have already been created for getting info about Users and Products:

* [Get Products](https://serverlessohapi.azurewebsites.net/api/GetProducts)
* [Get Product](https://serverlessohapi.azurewebsites.net/api/GetProduct?productId=75542e38-563f-436f-adeb-f426f1dabb5c) (expects a productId query parameter in the URL)
* [Get Users](https://serverlessohapi.azurewebsites.net/api/GetUsers)
* [Get User](https://serverlessohapi.azurewebsites.net/api/GetUser?userId=cc20a6fb-a91f-4192-874d-132493685376) (expects a userId query parameter in the URL)

You now should work together as a team to create an Azure Function App in your Azure subscription that will implement the Ice Cream Ratings part of the API.

Your challenge is to define, create and deploy three functions in this function app:

**CreateRating**

* **Verb**: POST
* **Input payload example**:

{

"userId": "cc20a6fb-a91f-4192-874d-132493685376",

"productId": "4c25613a-a3c2-4ef3-8e02-9c335eb23204",

"locationName": "Sample ice cream shop",

"rating": 5,

"userNotes": "I love the subtle notes of orange in this ice cream!"

}

* **Requirements**
  + Validate both userId and productId by calling the existing API endpoints. You can find a user id to test with from the sample payload above
  + Add a property called id with a GUID value
  + Add a property called timestamp with the current UTC date time
  + Validate that the rating field is an integer from 0 to 5
  + Use a data service to store the ratings information to the backend
  + Return the entire review JSON payload with the newly created id and timestamp
* **Output payload example**:

{

"id": "79c2779e-dd2e-43e8-803d-ecbebed8972c",

"userId": "cc20a6fb-a91f-4192-874d-132493685376",

"productId": "4c25613a-a3c2-4ef3-8e02-9c335eb23204",

"timestamp": "2018-05-21 21:27:47Z",

"locationName": "Sample ice cream shop",

"rating": 5,

"userNotes": "I love the subtle notes of orange in this ice cream!"

}

**GetRating**

* **Verb**: GET
* **Query string or route parameter**: ratingId
* **Requirements**
  + Get the rating from your database and return the entire JSON payload for the review identified by the id
  + Additional route parameters or query string values may be used if necessary.
* **Output payload example**:

{

"id": "79c2779e-dd2e-43e8-803d-ecbebed8972c",

"userId": "cc20a6fb-a91f-4192-874d-132493685376",

"productId": "4c25613a-a3c2-4ef3-8e02-9c335eb23204",

"timestamp": "2018-05-21 21:27:47Z",

"locationName": "Sample ice cream shop",

"rating": 5,

"userNotes": "I love the subtle notes of orange in this ice cream!"

}

**GetRatings**

* **Verb**: GET
* **Query string or route parameter**: userId
* **Requirements**
  + Get the ratings for the user from your database and return the entire JSON payload for the reviews for the user identified by the id.
  + Additional route parameters or query string values may be used if necessary.
* **Output payload example**:

[

{

"id": "79c2779e-dd2e-43e8-803d-ecbebed8972c",

"userId": "cc20a6fb-a91f-4192-874d-132493685376",

"productId": "4c25613a-a3c2-4ef3-8e02-9c335eb23204",

"timestamp": "2018-05-21 21:27:47Z",

"locationName": "Sample ice cream shop",

"rating": 5,

"userNotes": "I love the subtle notes of orange in this ice cream!"

},

{

"id": "8947f7cc-6f4c-49ed-a7aa-62892eac8f31",

"userId": "cc20a6fb-a91f-4192-874d-132493685376",

"productId": "e4e7068e-500e-4a00-8be4-630d4594735b",

"timestamp": "2018-05-20 09:02:30Z",

"locationName": "Another Sample Shop",

"rating": 4,

"userNotes": "I really enjoy this grape ice cream!"

}

]

Besides tools like Curl and Postman, you can use [the Rating Test Website](https://softserverless-rating.trafficmanager.net/) to test your CreateRating method.

**Hint**: If using the Rating Test Website make sure to add the <https://softserverless-rating.trafficmanager.net/> website url to your function’s CORS rules!

Success Criteria

* Have the rating API functions’ code in source control
* Demonstrate a working CI/CD pipeline for your team’s coaches. If there is an infrastructure focused member of the team, include support for the deployment of the infrastructure as well.
* Demonstrate sensitive environment data is stored securely, processed securely and under RBAC for your team’s coaches
* Provide your coach the three endpoint URLs. The coach should be able to make a call to each of the endpoints and get successful results
* The coach should be able to use GetRating and GetRatings to query for ratings that the coach creates with CreateRating
* The endpoints should return standard HTTP status codes. For example, 404 when items are not found.

References

* [What is Continuous Integration?](https://docs.microsoft.com/devops/develop/what-is-continuous-integration)
* [What is Continuous Delivery?](https://docs.microsoft.com/devops/deliver/what-is-continuous-delivery)
* [Continuous deployment for Azure Functions](https://docs.microsoft.com/azure/azure-functions/functions-continuous-deployment)
* [Azure DevOps Pipelines](https://docs.microsoft.com/azure/devops/pipelines/get-started/?view=azure-devops)
* [Git Actions Azure Functions Action](https://github.com/marketplace/actions/azure-functions-action)
* [Deploy Azure Functions with Jenkins](https://docs.microsoft.com/azure/jenkins/jenkins-azure-functions-deploy)
* [Deploy Azure Functions with Octopus Deploy](https://octopus.com/blog/azure-functions)
* Review the [Supported Languages in Azure Functions](https://docs.microsoft.com/azure/azure-functions/supported-languages) guide and go into the specific guide for the language you are using
* [Azure Functions triggers and bindings concepts](https://docs.microsoft.com/azure/azure-functions/functions-triggers-bindings)
* [Introduction to Cosmos DB](https://docs.microsoft.com/azure/cosmos-db/introduction)
* [Azure Cosmos DB bindings for Azure Functions](https://docs.microsoft.com/azure/azure-functions/functions-bindings-cosmosdb-v2)
* [Azure Functions HTTP and webhook bindings](https://docs.microsoft.com/azure/azure-functions/functions-bindings-http-webhook)
* [Store unstructured data using Azure Functions and Azure Cosmos DB](https://docs.microsoft.com/azure/azure-functions/functions-integrate-store-unstructured-data-cosmosdb)
* [Getting started with JSON features in Azure SQL Database](https://docs.microsoft.com/azure/sql-database/sql-database-json-features)
* [Azure Table storage overview](https://docs.microsoft.com/azure/cosmos-db/table-storage-overview)
* [Use Key Vault references for App Service and Azure Functions](https://docs.microsoft.com/azure/app-service/app-service-key-vault-references)
* [Use Azure Key Vault secrets in Azure Pipelines](https://docs.microsoft.com/azure/devops/pipelines/release/azure-key-vault?view=azure-devops)
* [What are managed identities for Azure resources?](https://docs.microsoft.com/azure/active-directory/managed-identities-azure-resources/overview)
* [How to use managed identities for App Service and Azure Functions](https://docs.microsoft.com/azure/app-service/overview-managed-identity?context=%2Fazure%2Factive-directory%2Fmanaged-identities-azure-resources%2Fcontext%2Fmsi-context&tabs=dotnet)
* [Provide access to Key Vault keys, certificates, and secrets with an Azure role-based access control](https://docs.microsoft.com/azure/key-vault/general/rbac-guide)

Progress Diagram

Diagram

Description automatically generatedRatings API progress diagram

Bonus - Infrastructure as Code

The application developers working with the APIs will be considering a variety of Azure services and features. You should continue working in conjuction with the developers to ensure these services are also captured in code and can be used as the destination for the application deployments.

* [Repeatable Infrastructure](https://docs.microsoft.com/en-us/azure/architecture/framework/devops/automation-infrastructure)
* [Deploying ARM with GitHub Actions](https://docs.microsoft.com/en-us/azure/azure-resource-manager/templates/deploy-github-actions)
* [Automate Terraform with GitHub Actions](https://learn.hashicorp.com/tutorials/terraform/github-actions)

OpenHack © Microsoft 2021. All Rights Reserved - Powered By [Opsgility](https://opsgility.com/" \t "_blank) [Privacy.](https://openhack.skillmeup.com/Home/Privacy)

Icon

Description automatically generated

* [**Attendee Guide**](https://openhackguides.blob.core.windows.net/userguides/OpenHack%20Portal%20User%20Guide%20-%20Attendee.pdf)

[**Logout**](https://openhack.skillmeup.com/account/signout)

* [**OVERVIEW**](https://openhack.skillmeup.com/labengine/modules/microsoft-open-hack-serverless_637802163179013632#labOverview)
* [**OPEN HACK GUIDE**](https://openhack.skillmeup.com/labengine/modules/microsoft-open-hack-serverless_637802163179013632#labGuide)
* [**OPEN HACK ENVIRONMENT**](https://openhack.skillmeup.com/labengine/modules/microsoft-open-hack-serverless_637802163179013632#labEnvironmentTab)
* [**PROVIDE FEEDBACK**](https://openhack.skillmeup.com/labengine/modules/microsoft-open-hack-serverless_637802163179013632#NotesTab)
* [**MESSAGES**](https://openhack.skillmeup.com/labengine/modules/microsoft-open-hack-serverless_637802163179013632#MessagesTab)
* **1** **2** **3** **4** **5** **6** **7** **8** **9** **10**

System telemetry and API Management

Background

Now that the Best For You Organics (BFYOC) API is complete, they want to start tracking telemetry from their API calls to be able to monitor the solution. After all, if you can’t measure it, you can’t improve it!

BFYOC would also like to begin exposing their APIs. They want to ensure the the APIs use consistent URL. They also want them exposed as different suites of APIs. They would like to have a different suite of APIs exposed to the mobile application, internal business users running reports and external partners. They also want to ensure that their mobile users receive priority access to those APIs by throttling requests from both internal business users and external partners.

Telemetry

The first part of this challenge is to work as a team to enable telemetry for the production Ratings API functions. BFYOC has requested a table derived from telemetry that displays the following information about your Ratings functions:

* The number of times each function was run in the past hour
* The average duration of each function in that hour
* There should be one row for each of the three Ratings functions your team created earlier

**Hint**: Application Insights may take a few minutes before showing telemetry from your functions

API Exposure

Uniform URL

Next your team is to create a uniformly addressed version of the entire API. This includes the GetUser, GetProduct and GetProducts APIs that already existed, and all the production functions your team created: CreateRating, GetRating, and GetRatings. Your team should expose all of these operations under the same base URL, as they currently are under different base URLs.

Product suites

Your team will now need to create 3 suites of APIs. One for each of the use cases: Mobile Application, Internal business users and External Partners. The team will need to configure the products so that only the required operations are exposed to the different product suites. Those requirements are as follows:

**Mobile Applications**

The Mobile Application requires access to all of the APIs. Each of them is required for different areas of the application’s UX. So the operations that need to be configured for this product suite are as follows:

* GetUser
* GetProduct
* GetProducts
* CreateRating
* GetRating
* GetRatings

**Internal business users**

The Internal business users use the APIs for reporting purposes. They need access to the product and rating information but shouldn’t be using the user operation or be able to create ratings. So the operations that should be exposed to this product suite are as follows:

* GetProduct
* GetProducts
* GetRating
* GetRatings

**External Partners**

The External Partners use case is to be able to see products that BYFOC has to offer, so should only have the product operations exposed to them. So this product suite should only have the following operations exposed to them:

* GetProduct
* GetProducts

Throttling

Finally, we need to ensure that our Mobile Application users are given priority over use of these APIs. We need to create some throttling on the operations exposed to the Internal business users and the External Partners.

Your team will need to create policies that ensure that the Internal business users do not make more than 10 calls within a 60 second time period on the operations that are exposed through this product suite.

The External Partners should also be throttled, but shouldn’t be throttled as greatly. The operations exposed to our partners should be limited to 15 calls per 60 seconds.

No need to store the results in a data store, on screen display of the results is sufficient

The mobile applications users shouldn’t have any throttling configured on the operations exposed to them.

Success Criteria

* Demonstrate to your coach the implementation of how to check the total number of times and the average duration of each function for the past hour based on telemetry. The results should be displayed as a table, in the same result set, and have one row for each of the three production Ratings API functions you created in the previous challenge.

No need to store the results in a data store, on screen display of the results is sufficient

* Demonstrate to your coach the implementation of exposing a uniform API under the same base URL for all functions
* Demonstrate to your coach the configuration of the 3 different product suites and the operations exposed to each of them.
* Demonstrate to your coach the implementation of the rate limiting requirements. The results should have different policies configured on the Internal business users product suite then on the External Partners product suite. There should be no throttling configured on the Mobile Applications product suite.

References

* [Monitor Azure Functions](https://docs.microsoft.com/azure/azure-functions/functions-monitoring)
* [Azure Monitor Log Queries](https://docs.microsoft.com/azure/azure-monitor/log-query/log-query-overview)
* [Azure API Management Overview](https://docs.microsoft.com/azure/api-management/api-management-key-concepts)
* [Azure API Management Tiers](https://docs.microsoft.com/azure/api-management/api-management-features)
* [Azure API Management Policies](https://docs.microsoft.com/azure/api-management/set-edit-policies)
* [Azure API Management Access Restriction Policies](https://docs.microsoft.com/azure/api-management/api-management-access-restriction-policies)

Progress Diagram

Diagram

Description automatically generatedSystem telemetry and API Management progress diagram

Bonus - Infrastructure as Code

Participate in discussion with your team around the configuration and managment of the APIs. Think about what components of the API Management solution that should be reflected as Infrastructure.

* [Azure API Management Resource Kit](https://github.com/Azure/azure-api-management-devops-resource-kit)

OpenHack © Microsoft 2021. All Rights Reserved - Powered By [Opsgility](https://opsgility.com/" \t "_blank) [Privacy.](https://openhack.skillmeup.com/Home/Privacy)

Icon

Description automatically generated

* [**Attendee Guide**](https://openhackguides.blob.core.windows.net/userguides/OpenHack%20Portal%20User%20Guide%20-%20Attendee.pdf)

[**Logout**](https://openhack.skillmeup.com/account/signout)

* [**OVERVIEW**](https://openhack.skillmeup.com/labengine/modules/microsoft-open-hack-serverless_637802163179013632#labOverview)
* [**OPEN HACK GUIDE**](https://openhack.skillmeup.com/labengine/modules/microsoft-open-hack-serverless_637802163179013632#labGuide)
* [**OPEN HACK ENVIRONMENT**](https://openhack.skillmeup.com/labengine/modules/microsoft-open-hack-serverless_637802163179013632#labEnvironmentTab)
* [**PROVIDE FEEDBACK**](https://openhack.skillmeup.com/labengine/modules/microsoft-open-hack-serverless_637802163179013632#NotesTab)
* [**MESSAGES**](https://openhack.skillmeup.com/labengine/modules/microsoft-open-hack-serverless_637802163179013632#MessagesTab)
* **1** **2** **3** **4** **5** **6** **7** **8** **9** **10**

Notify all distributors of the new ice cream line

Background

The food development, marketing, production, and quality teams at Best For You Organics (BFYOC) have released the details about their new line of ice creams to the sales department. The sales department has now asked your team to allow for notification of the main contact at each distributor with the information about the new line so they can review and start sending in orders.

In this challenge you will create a workflow between Best For You Organics CRM systems (Dynamics 365) and the company’s distributors.

Challenge

Your team should implement an HTTP triggered workflow that will generate an HTML email with the list of products, and then send an email to each of the main contacts at the distributors.

BFYOC has provided HTML that you can use as a starting point for the body of the emails.

**Note that this HTML includes a section that you need to programmatically loop through each product to finish creating the table.**

<!DOCTYPE html>

<html>

<body style="background-color: whitesmoke; color: #454545; font-family:'Gill Sans',

'Gill Sans MT', Calibri, 'Trebuchet MS', sans-serif; padding-bottom: 3em;">

<table style="width:100%; color:#454545">

<tr>

<td style="width:11em;">

<img style="margin-left:1em;"

src="https://serverlessoh.azureedge.net/public/ice-cream-2202561\_320-circle.jpg"

height="160" width="160" alt="Fruit Ice Cream">

</td>

<td>

<p style="font-style: italic; font-size: 50px;

font-weight:600; margin-left: 1em;">Best For You Organics</p>

</td>

</tr>

</table>

<p style="text-align: center; font-style: italic; font-size:

80px;">New Ice Cream Line!</p>

<p style="margin:2em 0em; font-size: 20px; text-align: center;">

Best For You Organics have a new line of fruit flavored ice creams.

Below is the information so you can start the ordering process:

</p>

<table style="width:100%; border-top: 1px solid #454545;

border-bottom: 1px solid #454545; color:#454545; padding: 1em; font-size: 20px;">

<thead>

<tr>

<th style="padding-bottom: 1em;" align="left">Ice Cream</th>

<th style="padding-bottom: 1em;" align="left">Description</th>

<th style="padding-bottom: 1em;" align="left">Product ID</th>

</tr>

</thead>

<tbody style="font-size: 16px;">

<!-- LOOP THROUGH EACH PRODUCT HERE AND CREATE A TABLE ROW ENTRY FOR EACH -->

</tbody>

</table>

<p style="text-align: center; margin-top: 3em;font-size: 20px;">Please contact

your representative at Best For You Organics to get more information..</p>

</body>

</html>

You can access the list of products via the GetProducts method exposed via the API Managament endpoint you created in the last challenge.

You can retrieve the main contacts of the distributors from the distributor’s Dynamics 365 CRM systems with the following detail:

* User: duser@microsoftopenhack.onmicrosoft.com
* Password: pass@word1
* Environment: Microsoft OpenHack (msftopenhack)
* Entity: Contacts

Viewing records to select the entity from the distributor’s Dynamics 365 CRM system can take several seconds. Please be patient.

The Email field from each contact contains the email address to be used for each distributor contact.

The email should also contain your team number in the subject. Feel free to also add the email address of someone in your team in CC so you can see what the emails look like.

HINT: The same credentials for user and password as listed above can also be used to send emails from Office 365 Outlook Add your own email address to the list and/or review the workflow executions to see the emails sent to validate the emails are sent as expected to multiple distributors.

Success Criteria

* The email looks correct in an HTML format and with the list of products
* Your coach will be able to validate that the distributors have received the email with your HTML content
* Make sure your team number is in the email subject
* Once the coach has validated the email, show the implementation to them

References

* [Create and manage records in Dynamics 365 using Azure Logic Apps](https://docs.microsoft.com/azure/connectors/connectors-create-api-crmonline)
* [Microsoft Dataverse (legacy)](https://docs.microsoft.com/connectors/commondataservice/)
* [List Records using a Common Data Service (Microsoft Dataverse) connector](https://docs.microsoft.com/azure/connectors/connect-common-data-service#list-records-based-on-a-filter)
* [Office 365 Connector](https://docs.microsoft.com/connectors/office365/)
* [Loops in Logic Apps](https://docs.microsoft.com/azure/logic-apps/logic-apps-control-flow-loops)
* [Create variables for saving and managing values in Azure Logic Apps](https://docs.microsoft.com/azure/logic-apps/logic-apps-create-variables-store-values)
* [Perform data operations in Azure Logic Apps](https://docs.microsoft.com/azure/logic-apps/logic-apps-perform-data-operations)
* [Logic Apps workflow actions and triggers](https://docs.microsoft.com/azure/logic-apps/logic-apps-workflow-actions-triggers)

Progress Diagram

Diagram

Description automatically generatedDistributor notification progress diagram

OpenHack © Microsoft 2021. All Rights Reserved - Powered By [Opsgility](https://opsgility.com/" \t "_blank) [Privacy.](https://openhack.skillmeup.com/Home/Privacy)

Icon

Description automatically generated

* [**Attendee Guide**](https://openhackguides.blob.core.windows.net/userguides/OpenHack%20Portal%20User%20Guide%20-%20Attendee.pdf)

[**Logout**](https://openhack.skillmeup.com/account/signout)

* [**OVERVIEW**](https://openhack.skillmeup.com/labengine/modules/microsoft-open-hack-serverless_637802163179013632#labOverview)
* [**OPEN HACK GUIDE**](https://openhack.skillmeup.com/labengine/modules/microsoft-open-hack-serverless_637802163179013632#labGuide)
* [**OPEN HACK ENVIRONMENT**](https://openhack.skillmeup.com/labengine/modules/microsoft-open-hack-serverless_637802163179013632#labEnvironmentTab)
* [**PROVIDE FEEDBACK**](https://openhack.skillmeup.com/labengine/modules/microsoft-open-hack-serverless_637802163179013632#NotesTab)
* [**MESSAGES**](https://openhack.skillmeup.com/labengine/modules/microsoft-open-hack-serverless_637802163179013632#MessagesTab)
* **1** **2** **3** **4** **5** **6** **7** **8** **9** **10**

Process distributor order batch files

Background

Best For You Organics Company (BFYOC) receives orders from distributors via flat files, similar to that of many businesses. These files come in batches of three, however each file can come in at different times. BFYOC would like your team to create an implementation that can process the files as a batch, storing the data contained as a single JSON object in a database.

Challenge

The purchase order information from some of the distributors arrive in batches. Each batch is made up of exactly three different files, each with different information about orders, as described below:

* Type 1: order header details. For example: 20180518151300-OrderHeaderDetails.csv
* Type 2: order line items. For example: 20180518151300-OrderLineItems.csv
* Type 3: product information for the batch. For example: 20180518151300-ProductInformation.csv

For these example file names the batch that binds them together is the file name prefix 20180518151300. Be sure to review the contents of the files, as a single order header details file may contain multiple orders.

Before continuing to the challenge, create a Storage Account of kind StorageV2 (general purpose v2) in your Azure Subscription, and then create a new Blob container in it. (If your team has a member who is focusing on IaC, this Storage Account must be deployed using code.)

You will be required to post your team table number, the storage account connection string, and the blob container name. The distributor systems will start sending batches of flat files with order details into your blob storage container every 1 minute. The date based prefix of the file names will indicate what files belong to the same batch.

HINT: Your team table number will be assigned if one exists. If not, you will need to create a unique table name such as <yourcity>-table-<sometablenumber>. You will need this table number in future challenges, so you may wish to write it down or keep it somewhere you can easily retrieve for future registrations. Review the swagger documentation or discuss with your coach for more information.

Make an HTTP POST call to the /team/registerStorageAccount endpoint of the [Serverless OpenHack API System](https://petstore.swagger.io/?url=https://serverlessohmanagementapi.trafficmanager.net/api/definition) to register your team’s storage account.

**Take note of the teamTableNumber used in this registration. It will be required in later challenges.**

**Make sure to note that the API is case-sensitive, so all parameters and parameter values must match case. Review the swagger documentation to make sure you have the BODY composed correctly**[**here**](https://petstore.swagger.io/?url=https://serverlessohmanagementapi.trafficmanager.net/api/definition#/Register%20Storage%20Account/register)

Your challenge is to work as a team to create a solution that:

* Waits until all the files for the same batch arrive before processing them.
* Once all three files for a batch have been received, make an HTTP POST call to the /order/combineOrderContent endpoint of the [Serverless OpenHack API System](https://petstore.swagger.io/?url=https://serverlessohmanagementapi.trafficmanager.net/api/definition), which will combine the content and return a single JSON document per order.
* Insert the JSON document of each order into your database as a separate entry/record.

**Note: The Serverless Open Hack will periodically check team registrations and delete those with invalid storage accounts and services used in later challenges. To ensure your team’s registration isn’t deleted, be sure not to delete the Azure services used here and subsequent challenges.**

HINT: Make sure to register EventGrid as a resource provider on your Azure Subscription before trying to respond to storage events. Failure to register Event Grid will result in an inability to create subscriptions that respond to Azure storage events

Success Criteria

* Demonstrate to your coach how you implemented the workflow to read, combine, and insert the data from the batches of flat files into one JSON document per order into a database. All the details from the flat files need to be in the JSON documents generated.
* Infrastructure Bonus: If your team has a member working on IaC, include the required storage account as part of the IaC deployment.

References

* [Register Event Grid](https://docs.microsoft.com/en-us/azure/event-grid/custom-event-quickstart-portal)
* [An introduction to Azure Event Grid](https://docs.microsoft.com/azure/event-grid/overview)
* [Event Grid: Reacting to Blob Storage events](https://docs.microsoft.com/azure/storage/blobs/storage-blob-event-overview)
* [Durable Functions overview](https://docs.microsoft.com/azure/azure-functions/durable-functions-overview)
* [Send, receive, and batch process messages in Logic Apps](https://docs.microsoft.com/azure/logic-apps/logic-apps-batch-process-send-receive-messages)

Progress Diagram

Diagram

Description automatically generatedProcess distributor order batch files progress diagram

OpenHack © Microsoft 2021. All Rights Reserved - Powered By [Opsgility](https://opsgility.com/" \t "_blank) [Privacy.](https://openhack.skillmeup.com/Home/Privacy)

Icon

Description automatically generated

* [**Attendee Guide**](https://openhackguides.blob.core.windows.net/userguides/OpenHack%20Portal%20User%20Guide%20-%20Attendee.pdf)

[**Logout**](https://openhack.skillmeup.com/account/signout)

* [**OVERVIEW**](https://openhack.skillmeup.com/labengine/modules/microsoft-open-hack-serverless_637802163179013632#labOverview)
* [**OPEN HACK GUIDE**](https://openhack.skillmeup.com/labengine/modules/microsoft-open-hack-serverless_637802163179013632#labGuide)
* [**OPEN HACK ENVIRONMENT**](https://openhack.skillmeup.com/labengine/modules/microsoft-open-hack-serverless_637802163179013632#labEnvironmentTab)
* [**PROVIDE FEEDBACK**](https://openhack.skillmeup.com/labengine/modules/microsoft-open-hack-serverless_637802163179013632#NotesTab)
* [**MESSAGES**](https://openhack.skillmeup.com/labengine/modules/microsoft-open-hack-serverless_637802163179013632#MessagesTab)
* **1** **2** **3** **4** **5** **6** **7** **8** **9** **10**

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 [OpenHack API System](https://petstore.swagger.io/?url=https://serverlessohmanagementapi.trafficmanager.net/api/definition" \t "_blank)

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 [OpenHack API System](https://petstore.swagger.io/?url=https://serverlessohmanagementapi.trafficmanager.net/api/definition" \t "_blank) 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
* Infrastructure Bonus - If your team has an IaC focused member, the event hub and namespace must be created using the code, processes and pipeline in place.

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)
* [Functions Event Hubs Trigger Scaling](https://docs.microsoft.com/azure/azure-functions/functions-bindings-event-hubs#trigger---scaling)
* [Azure Event Hubs Event Processor Host overview](https://docs.microsoft.com/azure/event-hubs/event-hubs-event-processor-host)
* [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/)
* [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

Diagram

Description automatically generatedProcess POS sales event progress diagram

OpenHack © Microsoft 2021. All Rights Reserved - Powered By [Opsgility](https://opsgility.com/" \t "_blank) [Privacy.](https://openhack.skillmeup.com/Home/Privacy)

Icon

Description automatically generated

* [**Attendee Guide**](https://openhackguides.blob.core.windows.net/userguides/OpenHack%20Portal%20User%20Guide%20-%20Attendee.pdf)

[**Logout**](https://openhack.skillmeup.com/account/signout)

* [**OVERVIEW**](https://openhack.skillmeup.com/labengine/modules/microsoft-open-hack-serverless_637802163179013632#labOverview)
* [**OPEN HACK GUIDE**](https://openhack.skillmeup.com/labengine/modules/microsoft-open-hack-serverless_637802163179013632#labGuide)
* [**OPEN HACK ENVIRONMENT**](https://openhack.skillmeup.com/labengine/modules/microsoft-open-hack-serverless_637802163179013632#labEnvironmentTab)
* [**PROVIDE FEEDBACK**](https://openhack.skillmeup.com/labengine/modules/microsoft-open-hack-serverless_637802163179013632#NotesTab)
* [**MESSAGES**](https://openhack.skillmeup.com/labengine/modules/microsoft-open-hack-serverless_637802163179013632#MessagesTab)
* **1** **2** **3** **4** **5** **6** **7** **8** **9** **10**

Pub/Sub and Network Integration

Background

Executives at Best For You Organics Company (BFYOC) have taken notice of the success of the recently implemented solution to enable Point of Sale (POS) terminals to send sales data to the cloud, and they want to leverage this new data even more. They have started a pilot program to update BFYOC POS terminals to randomly capture a PDF version of the sales receipt in addition to the detailed transaction data. They would like to save all the transaction data to their private storage in the cloud for data archival and future analysis. Additionally, executives are concerned about the potential for fraudulent transactions, and would like to add additional rigor for transactions which exceed $100.

After learning of the executive’s high-level requirements, the cloud architecture team has decided to implement a publisher/subscriber (pub/sub) messaging technique to capture POS sales data which has a receipt. They feel they can use a message filtering pattern to identify sales data which has a total transaction cost greater than $100.

Additionally, because BFYOC is highly protective of customer and sales data, they want to ensure all the data is safely stored within resources which are accessible only from services within, or integrated with, BFYOC’s virtual network in Azure. The files stored in blob storage are able to be used by other systems within BFYOC’s virtual network, such as data analysis and fraud detection processes. Access to the pub/sub messaging system chosen must also be limited to only resources integrated into the BFYOC’s virtual network.

BFYOC has an existing virtual network containing at least two subnets. They also have a Azure storage account configured to use Azure Virtual Network service endpoints to restrict access from any source except the two subnets within the virtual network. An Azure Virtual Machine is available to be used as a jumpbox for viewing data from services within the virtual network.

Diagram

Description automatically generatedVirtual Network with subnets and blob storage service endpoints

Challenge

Your first challenge is to modify the Azure Function used in the previous challenge to send the specified receipt information to a pub/sub messaging system, but only if there is a receipt. Using the POS data sent via the Event Hub from the previous challenge, you can create a new JSON document with a structure similar to the example below that will be sent to your pub/sub messaging system.

{

"totalItems": 8,

"totalCost": 123.40,

"salesNumber": "0c423398-3c7c-0682-7519-4701c445ed7a",

"salesDate": "09/11/2019 06:04:43",

"storeLocation": "00d8ea6f-935c-2cca-9bbc-f56b5a091621",

"receiptUrl": "https://serverlessohsales.blob.core.windows.net/TheReceipt.pdf"

}

Next, you will need to create a process(es) which is able to filter based on the total cost in each message, and accomplish the following tasks:

1. If the total cost is greater than or equal to $100:
   * retrieve the PDF file of the receipt using the receiptUrl value
   * base64 encode the PDF
   * create a JSON object which includes the receipt data, along with the base64 encoded version of the receipt
   * save the JSON object with a unique name to the “receipts-high-value” container within the provided Azure Storage account (accessible only from the virtual network)

The format for the saved data should be as follows:

{

"Store": "00d8ea6f-935c-2cca-9bbc-f56b5a091621",

"SalesNumber": "0c423398-3c7c-0682-7519-4701c445ed7a",

"TotalCost": 123.40,

"Items": 8,

"SalesDate": "09/11/2019 06:04:43",

"ReceiptImage":"V2VsY29tZSB0byBTZXJ2ZXJsZXNzIE9wZW5IYWNrIQ=="

}

1. If the total cost is less than $100:
   * create a JSON object which includes the receipt data
   * save the JSON object with a unique name to the “receipts” container within the provided Azure Storage account (accessible only from the provided virtual network)

The format for the saved data should be as follows:

{

"Store": "00d8ea6f-935c-2cca-9bbc-f56b5a091621",

"SalesNumber": "0c423398-3c7c-0682-7519-4701c445ed7a",

"TotalCost": 6.58,

"Items": 1,

"SalesDate": "09/02/2019 10:36:17"

}

When determining how to perform the filtering of messages, it is important to remember that BFYOC has built a distributed system which may, over time, require additional processing streams. The cloud architecture team wants to ensure the solution avoids filtering “in code” by leveraging features of the selected messaging system. The selected messaging system needs to also be configured to ensure it is only accessible from resources integrated with the virtual network.

HINT: All receipts should be sent to the messaging system, regardless of overall total cost.

Success Criteria

* Demonstrate the Azure Function which publishes data to a pub/sub messaging service.
* Demonstrate the ability to filter messages to the appropriate process based on the total cost of purchased items.
* Demonstrate the configuration of the pub/sub messaging system to ensure it is only accessible from inside the BFYOC’s virtual network.
* Demonstrate the ability to retrieve messages from the pub/sub service and save JSON files with a unique name to the proper blob storage container in the provided Azure Storage account (which uses service endpoints).

References

* [Claim-Check Pattern](https://docs.microsoft.com/azure/architecture/patterns/claim-check)
* [Azure Virtual Network Service Endpoints](https://docs.microsoft.com/azure/virtual-network/virtual-network-service-endpoints-overview)
* [Azure Functions networking options](https://docs.microsoft.com/azure/azure-functions/functions-networking-options)
* [Azure Functions Premium plan (network connectivity)](https://docs.microsoft.com/azure/azure-functions/functions-premium-plan#private-network-connectivity)
* [Azure Functions binding expression patterns](https://docs.microsoft.com/azure/azure-functions/functions-bindings-expressions-patterns#create-guids)
* [Azure Functions triggers and bindings concepts](https://docs.microsoft.com/azure/azure-functions/functions-triggers-bindings)
* [Azure Functions Recipes - Triggers and bindings](https://docs.microsoft.com/sandbox/functions-recipes/triggers-bindings)
* [Azure Service Bus Queues, Topics and Subscriptions](https://docs.microsoft.com/azure/service-bus-messaging/service-bus-queues-topics-subscriptions)
* [Azure Service Bus bindings for Azure Functions](https://docs.microsoft.com/azure/azure-functions/functions-bindings-service-bus)
* [Azure Service Bus Topics and Filters](https://docs.microsoft.com/azure/service-bus-messaging/topic-filters)
* [Create a Service Bus namespace with topic, subscription, and rule using an Azure Resource Manager template](https://docs.microsoft.com/azure/service-bus-messaging/service-bus-resource-manager-namespace-topic-with-rule)
* [Azure CLI (Service Bus)](https://docs.microsoft.com/cli/azure/servicebus/topic/subscription/rule?view=azure-cli-latest)
* [Azure Service Bus Samples (GitHub)](https://github.com/Azure/azure-service-bus/tree/master/samples)
* [Service Bus Explorer](https://github.com/paolosalvatori/ServiceBusExplorer)
* [Allow access to Azure Service Bus namespaces via private endpoints](https://docs.microsoft.com/azure/service-bus-messaging/private-link-service#:~:text=If%20you%20already%20have%20an%20existing%20namespace%2C%20you,which%20you%20want%20to%20add%20a%20private%20endpoint.)
* [What is Azure Private Link](https://docs.microsoft.com/azure/private-link/private-link-overview)

Progress Diagram

Diagram

Description automatically generatedPub/Sub and VNet Integration Progress Diagram

Bonus - Infrastructure as Code

Continue to codify the various services used by the development team for this application pilot. However, the BFYOC team that manages the shared neworking and storage has their own process for managing and updating those resources. Ensure that your code does not make any changes to those resources directly, as they could be reverted, causing conflicts or failures.

OpenHack © Microsoft 2021. All Rights Reserved - Powered By [Opsgility](https://opsgility.com/" \t "_blank) [Privacy.](https://openhack.skillmeup.com/Home/Privacy)

Icon

Description automatically generated

* [**Attendee Guide**](https://openhackguides.blob.core.windows.net/userguides/OpenHack%20Portal%20User%20Guide%20-%20Attendee.pdf)

[**Logout**](https://openhack.skillmeup.com/account/signout)

* [**OVERVIEW**](https://openhack.skillmeup.com/labengine/modules/microsoft-open-hack-serverless_637802163179013632#labOverview)
* [**OPEN HACK GUIDE**](https://openhack.skillmeup.com/labengine/modules/microsoft-open-hack-serverless_637802163179013632#labGuide)
* [**OPEN HACK ENVIRONMENT**](https://openhack.skillmeup.com/labengine/modules/microsoft-open-hack-serverless_637802163179013632#labEnvironmentTab)
* [**PROVIDE FEEDBACK**](https://openhack.skillmeup.com/labengine/modules/microsoft-open-hack-serverless_637802163179013632#NotesTab)
* [**MESSAGES**](https://openhack.skillmeup.com/labengine/modules/microsoft-open-hack-serverless_637802163179013632#MessagesTab)
* **1** **2** **3** **4** **5** **6** **7** **8** **9** **10**

Alerting and notification

Background

Best For You Organics Company’s management would like to be alerted when there’s a high number of bad reviews for a specific ice cream. They are hoping they can take advantage of the system you created earlier to react to business related information that is raised from custom telemetry in your code.

Challenge

Your first challenge is to update the CreateRating function and put the userNotes field through a sentiment analysis service. You should then use the result of the sentiment analysis to add an extra field, sentimentScore, to the JSON before saving to the database, and use custom telemetry to record information about the negative review. Of particular interest for this challenge will be ratings that contain any sentences which receive a negative sentiment result having a score of 0.7 or greater.

To start receiving automated ratings, make a post call to the /team/registerRatingEndpoint method of the OpenHack API System.

* OpenHack API OpenUI Definition: [Openhack Swagger](https://petstore.swagger.io/?url=https://serverlessohmanagementapi.trafficmanager.net/api/definition" \t "_blank)

Your second challenge is to implement a solution to generate an email alert to your coach whenever at least five bad ice cream reviews with a negative sentiment result greater than or equal to 0.7 on any sentences have been received in the past five minutes. The email should contain the bad reviews information, or a link to the information about the reviews.

Hint: This is five individual reviews containing at least one sentence with a negative sentiment score, not five total sentences in one or more reviews. The threshold should be greater than or equal to five negative reviews in the previous five minutes.

Success Criteria

* GetRating and GetRatings should return the sentimentScore field with the sentiment analysis result from the userNotes for new ratings created
* Your coach should receive an email with an alert containing information about the bad reviews after five are created within five minutes by the automated system

References

* [How to detect sentiment in Text Analytics](https://docs.microsoft.com/azure/cognitive-services/text-analytics/how-tos/text-analytics-how-to-sentiment-analysis)
* [Monitor Azure Functions](https://docs.microsoft.com/azure/azure-functions/functions-monitoring)
* [Log alerts in Azure Monitor - Alerts for Application Insights](https://docs.microsoft.com/azure/monitoring-and-diagnostics/monitor-alerts-unified-log)

Progress Diagram

Diagram

Description automatically generatedAlerting and notification progress diagram

OpenHack © Microsoft 2021. All Rights Reserved - Powered By [Opsgility](https://opsgility.com/" \t "_blank) [Privacy.](https://openhack.skillmeup.com/Home/Privacy)

Icon

Description automatically generated

* [**Attendee Guide**](https://openhackguides.blob.core.windows.net/userguides/OpenHack%20Portal%20User%20Guide%20-%20Attendee.pdf)

[**Logout**](https://openhack.skillmeup.com/account/signout)

* [**OVERVIEW**](https://openhack.skillmeup.com/labengine/modules/microsoft-open-hack-serverless_637802163179013632#labOverview)
* [**OPEN HACK GUIDE**](https://openhack.skillmeup.com/labengine/modules/microsoft-open-hack-serverless_637802163179013632#labGuide)
* [**OPEN HACK ENVIRONMENT**](https://openhack.skillmeup.com/labengine/modules/microsoft-open-hack-serverless_637802163179013632#labEnvironmentTab)
* [**PROVIDE FEEDBACK**](https://openhack.skillmeup.com/labengine/modules/microsoft-open-hack-serverless_637802163179013632#NotesTab)
* [**MESSAGES**](https://openhack.skillmeup.com/labengine/modules/microsoft-open-hack-serverless_637802163179013632#MessagesTab)
* **1** **2** **3** **4** **5** **6** **7** **8** **9** **10**

Putting it All Together

Background

As Best For You Organics Company (BFYOC) now has data from customer reviews, distributor orders, and POS sales for the new fruit flavored ice cream line, the product managers would like to know how the new line of ice creams are doing.

Challenge

Your challenge is to work on a solution that monitors and forwards the reviews, distributor orders, and POS sales data to an event-processing engine that will then, every 5 minutes, summarize the following:

* A list of ice creams based on total distributor orders total sales, from highest to lowest
* A list of ice creams based on total POS sales, from highest to lowest
* The average sentiment analysis value for each ice cream

Aggregate the data for each 5 minute tumbling window and then use that data to generate a report for your manager (your coach) with these insights.

Success Criteria

* Demonstrate to your coach how you implemented the solution
* The coach should have access to a report that includes all the required info as per the Challenge section.
* Infrastructure Bonus - If your team has an IaC focused member, demonstrate how the new services can be deployed using the code, processes and pipeline in place.

References

* [What is Stream Analytics?](https://docs.microsoft.com/azure/stream-analytics/stream-analytics-introduction)
* [Stream data as input into Stream Analytics](https://docs.microsoft.com/azure/stream-analytics/stream-analytics-define-inputs)
* [Query examples for common Stream Analytics usage patterns](https://docs.microsoft.com/azure/stream-analytics/stream-analytics-stream-analytics-query-patterns)
* [Azure Stream Analytics output to Azure Cosmos DB](https://docs.microsoft.com/azure/stream-analytics/stream-analytics-documentdb-output)
* [Working with the change feed support in Azure Cosmos DB](https://docs.microsoft.com/azure/cosmos-db/change-feed)

Progress Diagram

Diagram

Description automatically generatedFinal progress diagram

OpenHack © Microsoft 2021. All Rights Reserved - Powered By [Opsgility](https://opsgility.com/" \t "_blank) [Privacy.](https://openhack.skillmeup.com/Home/Privacy)