



Lab 4 Azure Functions

By: Kevin Saye

IoT Solution Architect

December, 2017

This lab assumes you have completed Lab 2.

If you have any issues or concerns, please email: virtualbootcamphelp@microsoft.com.

Execution Time: 30 minutes.

Required Hardware:

- Windows 10 PC
- IoT Hardware kit: <https://www.adafruit.com/product/3605> or similar hardware.
- Access to a WiFi network (without a captive portal aka web page login)

Required Operating System:

- Windows 10

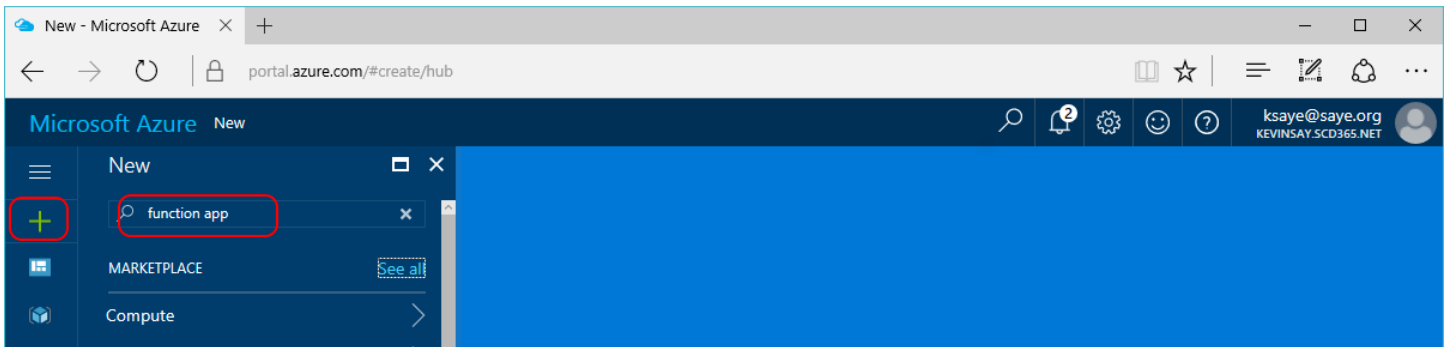
Other Requirements:

- Azure Subscription

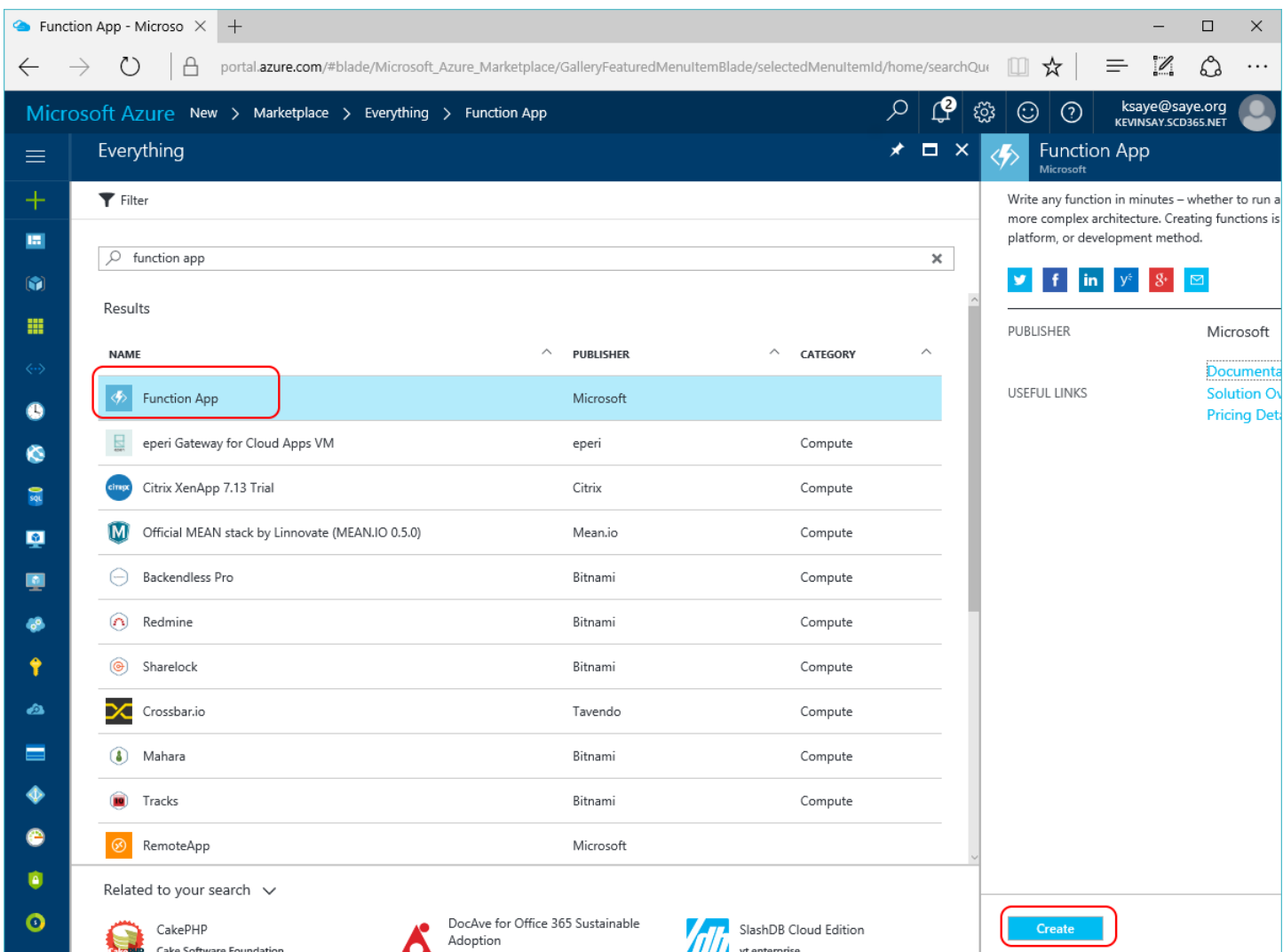
Required Software:

- None

- Step 1. Go to <http://portal.azure.com> and sign in.
- Step 2. If not already stopped, stop your Stream Analytics Job from Lab 3.
- Step 3. Click the Plus sign on the left and search for **Function App**.



- Step 4. Select the Function App by Microsoft and click Create.



Step 5. Give your function a unique name, select the correct Azure Subscription, select the Resource Group created in the prior lab, make sure it is in the desired location and click create.

Function App - Microso × +

portal.azure.com/#create/Microsoft.FunctionApp

Microsoft Azure New > Marketplace > Everything > Function App > Function App

Function App Create

* App name
kevinsayhol ✓
.azurewebsites.net

* Subscription
MSDN Personal

* Resource Group
☐ Create new ☒ Use existing
kevinsayhol

* Hosting Plan
Consumption Plan

* Location
East US

* Storage Account
function219b628086d2

☐ Pin to dashboard

Create Automation options

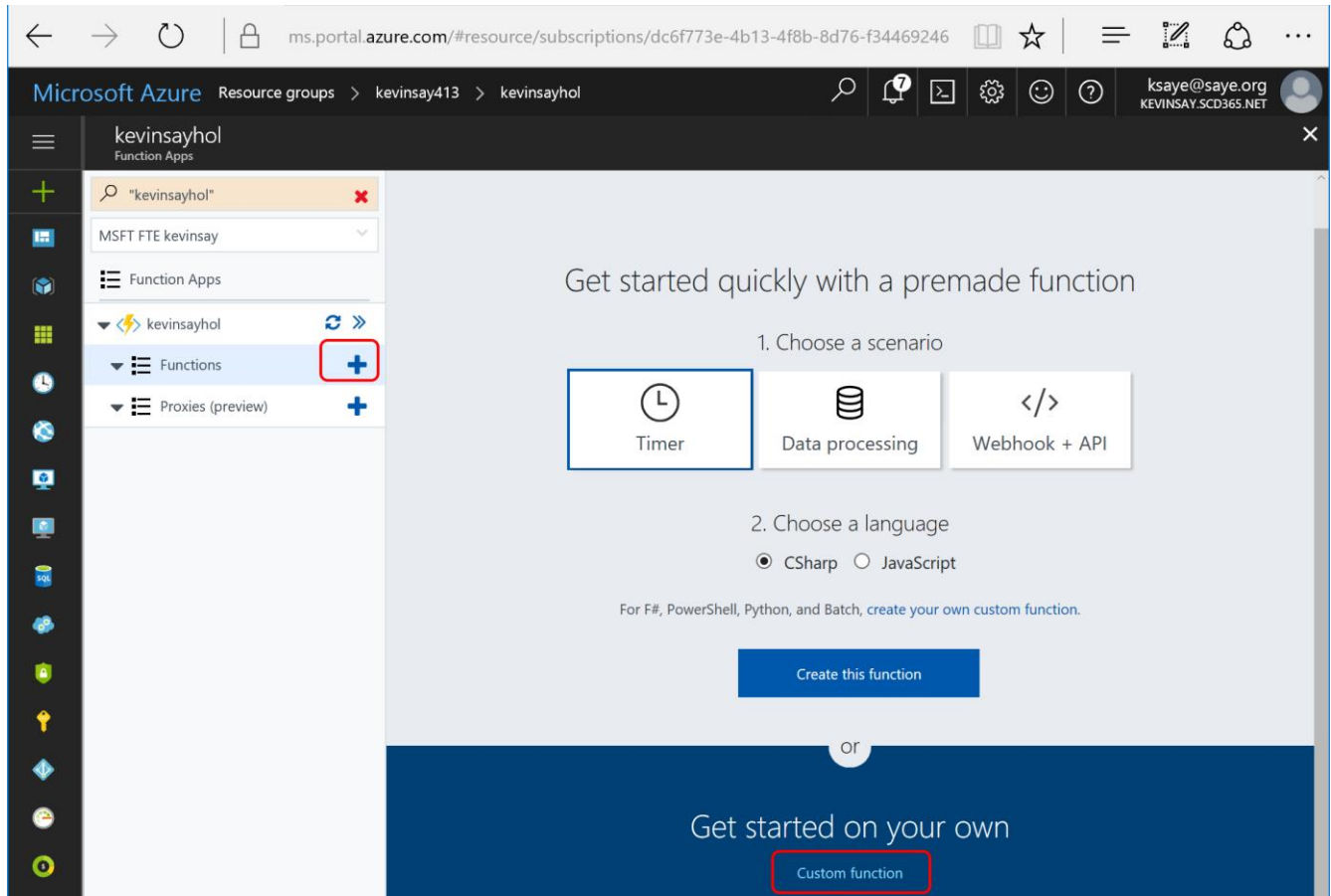
Step 6. After about a minute, click the Resource Group Icon on the left, select the resource group you created and click on the Function just created.

The screenshot shows the Microsoft Azure portal interface. The left sidebar contains the 'Resource groups' section with a search bar and a list of resource groups. The 'kevinssayhol' resource group is selected and highlighted. The main pane displays the 'kevinssayhol' resource group overview, including subscription details and a list of resources. The 'kevinssayhol' resource is highlighted in the list.

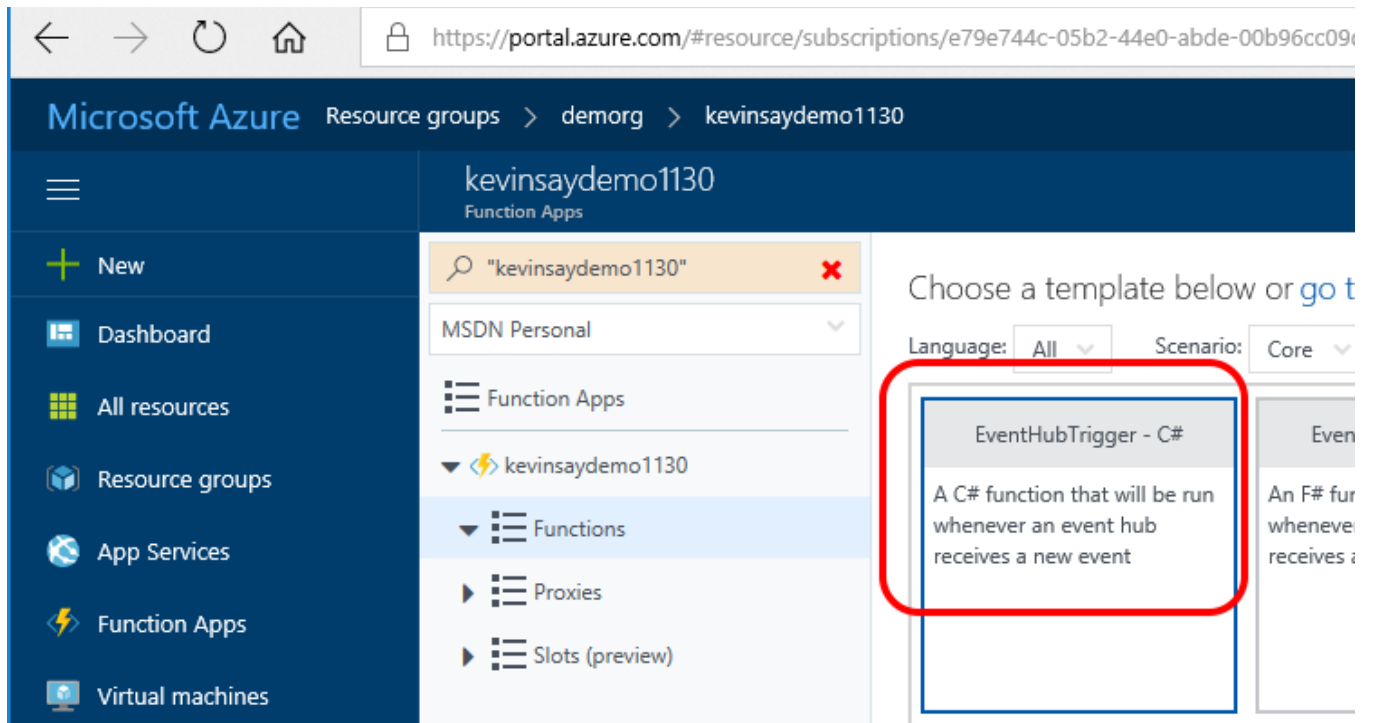
Subscription name (change) [MSDN Personal](#) Subscription ID: e79e744c-05b2-44e0-abde-00b96cc09de9
Deployments: 1 Deploying, 2 Succeeded Location: East US

NAME	TYPE	LOCATION
function219b628086d2	Storage account	East US
hol	IoT Hub	East US
iotStreamJob	Stream Analytics job	East US
kevinssayhol	App Service	East US

Step 7. Under your function name, click the + sign and click on Custom function.



Step 8. Click the "EventHubTrigger - C#" template.



Step 9. Name your function and type in the name of the IoT Hub in the event Hub Name area. Next click new on the Event Hub Connection.

The screenshot shows the Microsoft Azure portal interface for configuring a new function app. The breadcrumb navigation at the top indicates the path: Resource groups > iotbootcamp > kevinsayfunction1. The left sidebar shows the 'Functions' section under 'Function Apps' for 'kevinsayfunction1'. The main area displays a grid of trigger options. The 'EventHubTrigger - C#' trigger is highlighted with a red box. Below the grid, the 'Name your function' field contains 'EventHubTriggerCSharp1'. Under the 'Azure Event Hubs trigger' section, the 'Event Hub connection' field is empty, with a red box highlighting a 'new' button next to it. The 'Event Hub name' field contains 'samples-workitems' and the 'Event Hub consumer group' field contains '\$Default'.

Microsoft Azure Resource groups > iotbootcamp > kevinsayfunction1

kevinsayfunction1
Function Apps

"kevinsayfunction1" MSFT FTE kevinsay

Function Apps

kevinsayfunction1

Functions

Proxies

Slots (preview)

BlobTrigger - F#
An F# function that will be run whenever a blob is added to a specified container

BlobTrigger - JavaScript
A JavaScript function that will be run whenever a blob is added to a specified container

EventHubTrigger - C#
A C# function that will be run whenever an event hub receives a new event

ServiceBusQueueTrigger - C#
A C# function that will be run whenever a message is added to a specified Service Bus queue

ServiceBusQueueTrigger - F#
An F# function that will be run whenever a message is added to a specified Service Bus queue

ServiceBusQueueTrigger - JavaScript
A JavaScript function that will be run whenever a message is added to a specified Service Bus queue

Name your function
EventHubTriggerCSharp1

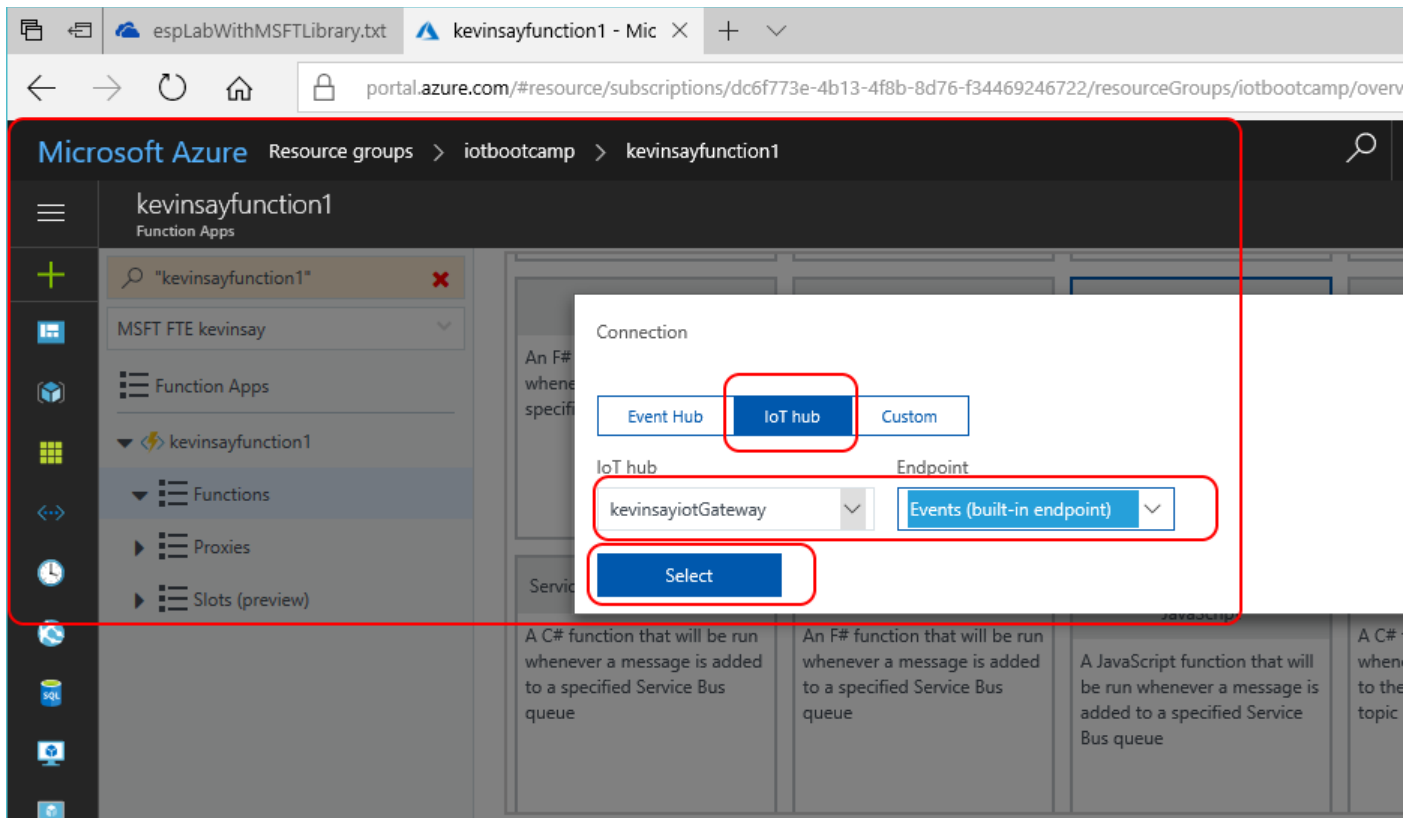
Azure Event Hubs trigger

Event Hub connection show value
new

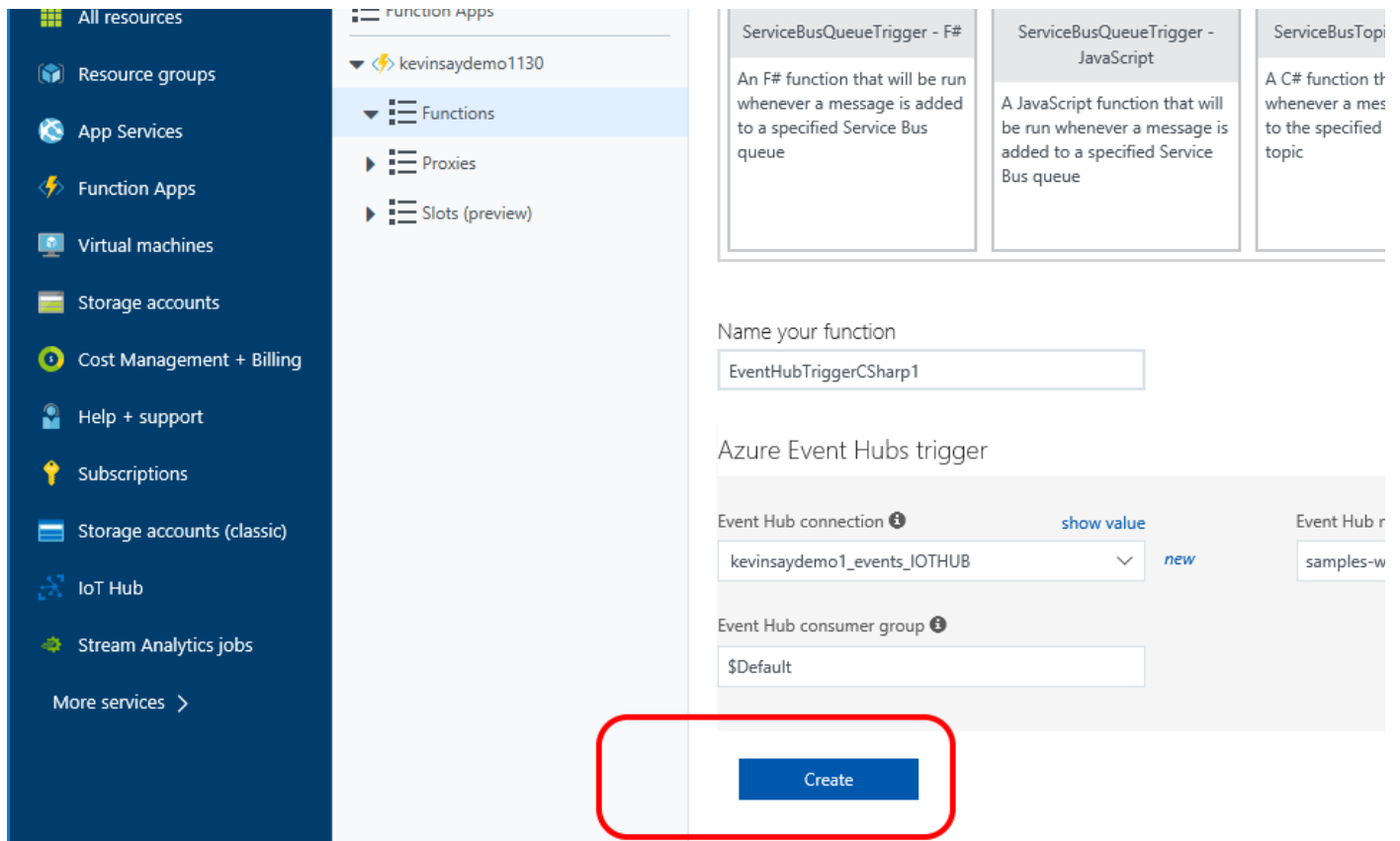
Event Hub name
samples-workitems

Event Hub consumer group
\$Default

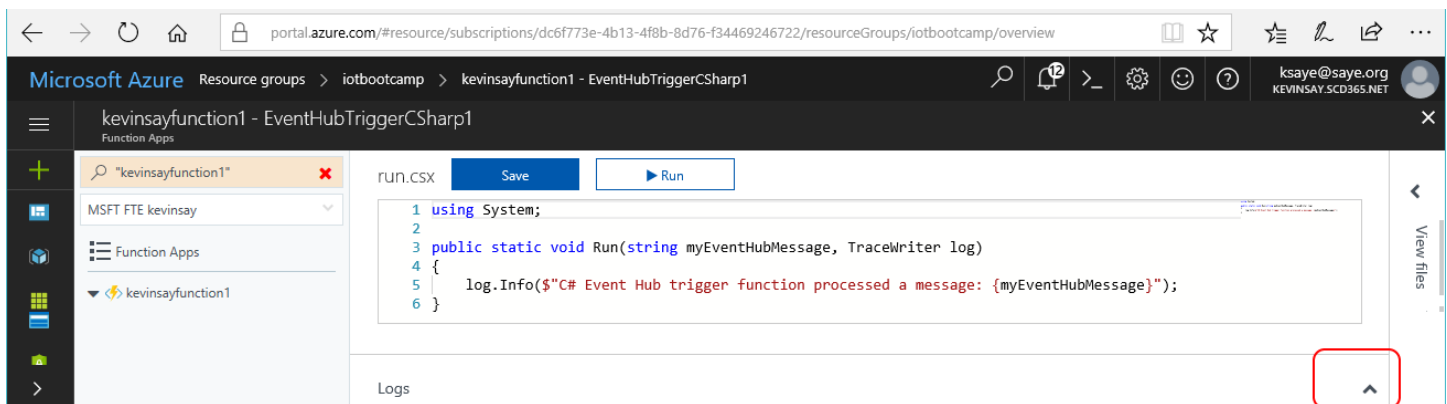
Step 10. At the "Connection" screen, select IoT hub, the name of your hub and then click Select.



Step 11. At the template screen, click create to finally create the Function:

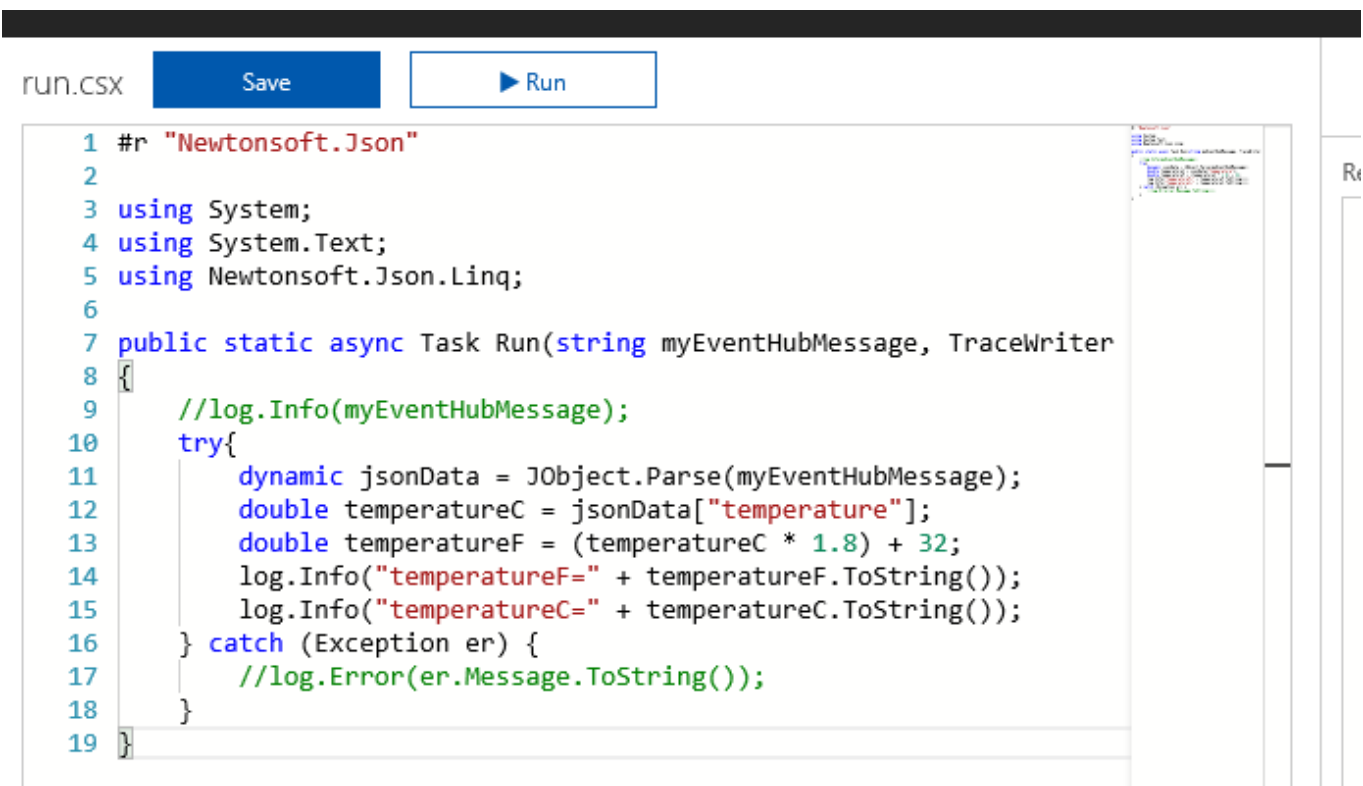


Step 12. When created you will see the default C# code just logs what it finds in the IoT Hub. Click the up arrow in the logs area to see data, as shown below:

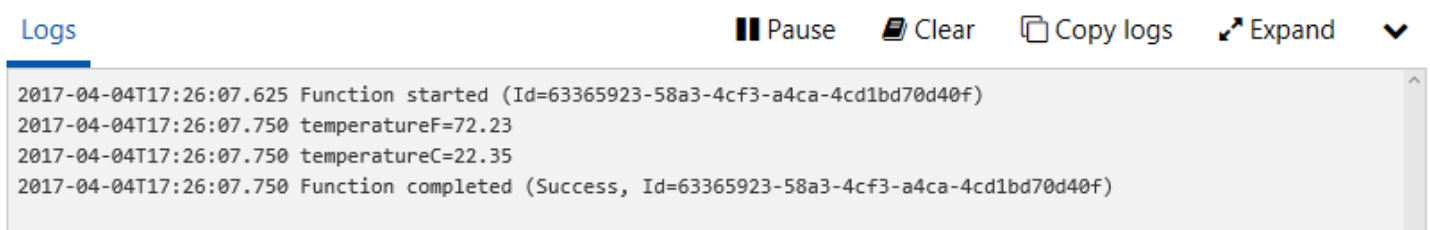


If you have data coming in, you will see it logged. If you do not see data, verify that your device is connected and sending data.

Step 13. Last, simply modify the code as shown below and click Run. You can just copy the content from: <https://tinyurl.com/FunctionIoTBC>

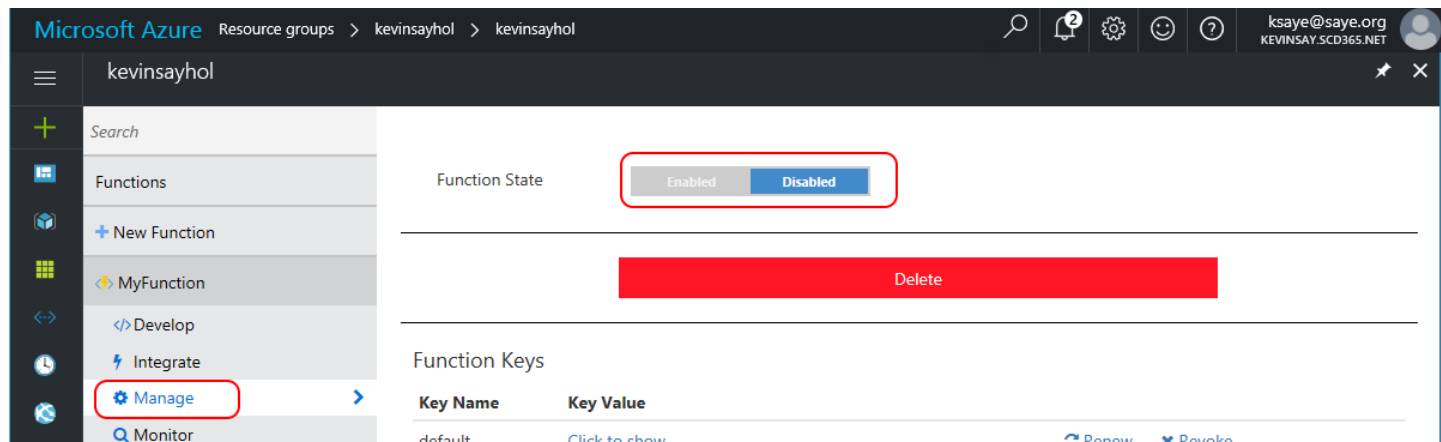


You should see the output as shown below:



We could also modify the code to call native libraries, save the data to storage, resubmit the data back to IoT Hub, the possibilities are endless. One example (<https://github.com/ksaye/IoTDemonstrations/blob/master/DogFoodMonitor/AzureFunction.cs>) sends a message to another device, which we will do tomorrow.

Note, because we use the same consumer group as Device Manager and Stream Analytics, we will have a conflict. Either: Disable Azure Function (shown below) or use a different consumer group.



This concludes the lab.