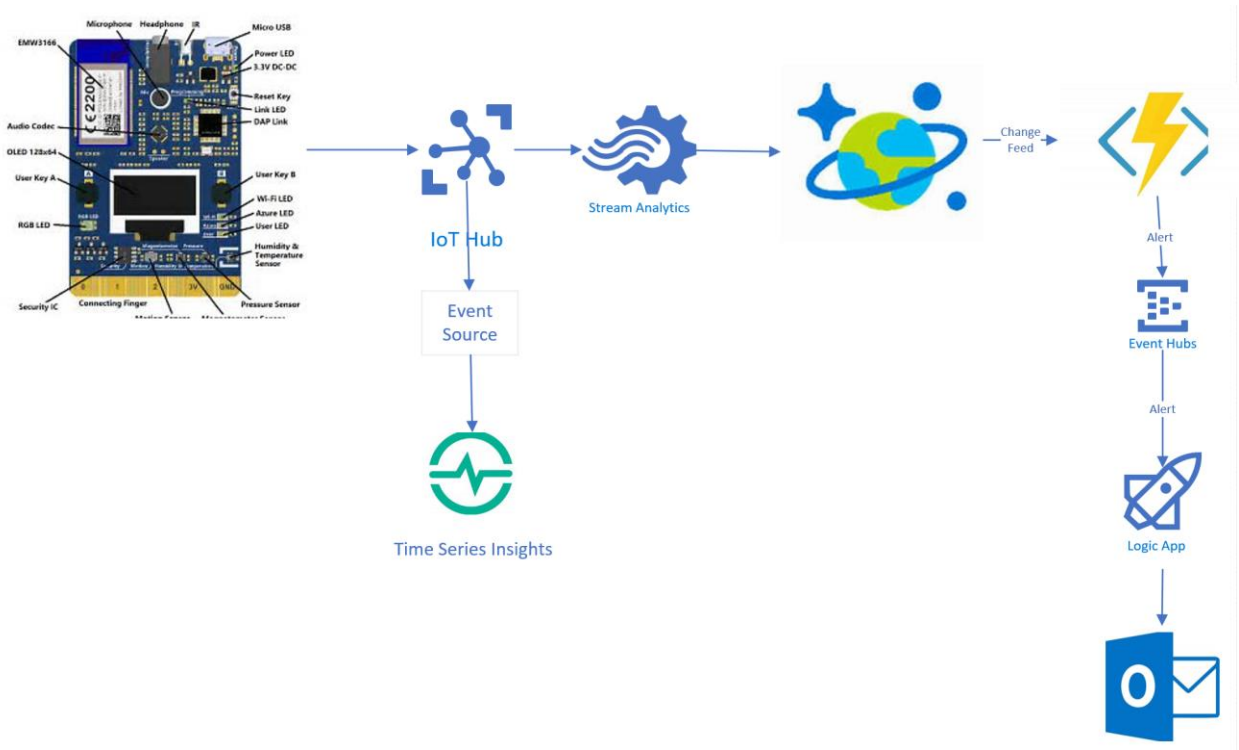
AZURE IOT & COSMOS DB WORKSHOP

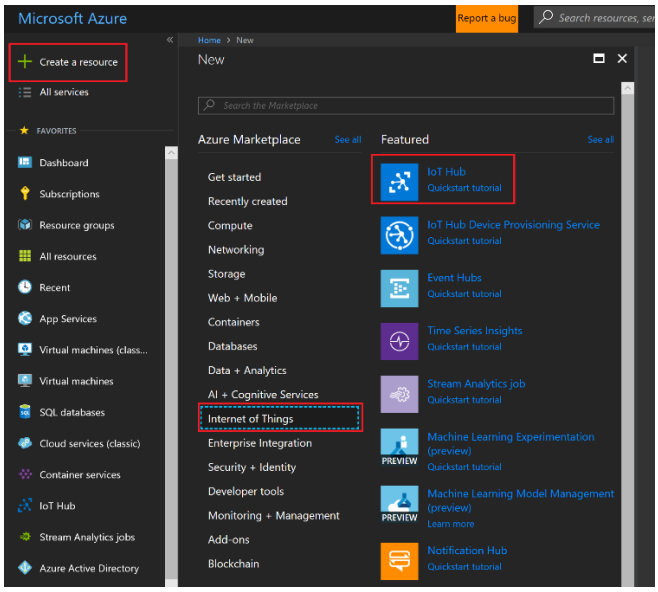
Rafat Sarosh / Ranga Vadlamudi

Contents

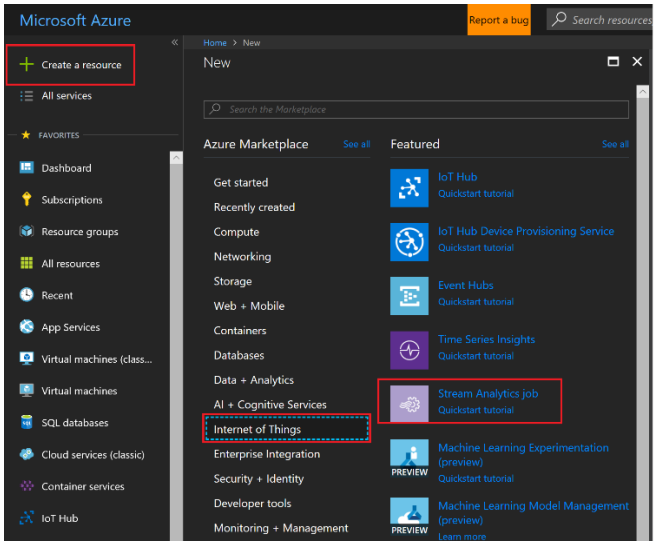

Microphone
Headphone
IR
Micro USB
EMW3166
Power LED
3.3V DC-DC
Reset Key
Link LED
DAP Link
Audio Codec
OLED 128x64
User Key A
User Key B
Wi-Fi LED
Azure LED
User LED
RGB LED
Humidity & Temperature Sensor
Security IC
Connecting Finger
Pressure Sensor
IoT Hub
Event Source
Time Series Insights
Stream Analytics
Change Feed
Alert
Event Hubs
Alert
Logic App

Steps to Azure IoT & CosmosDB Hands on Workshop

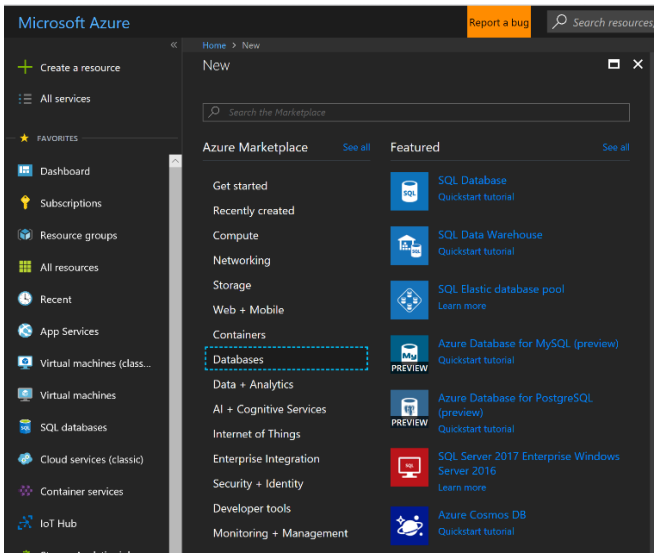
1. Setup Azure Services
   - Create IoTHub



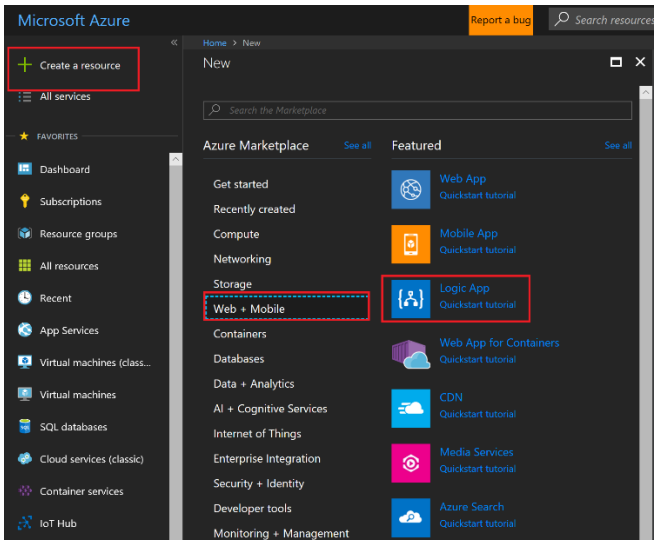
   - Create Stream Analytics Job



- Create Cosmos DB Account



- Create Logic App



Azure Stream Analytics

2. Setup & Start Azure Stream Analytics
   a) Configure Input and Output
   Input: Select IoTHub you created in the resource group

   Output: Select Cosmos DB you create in the resource group

Output goes to CosmosDB, name it CosmosDB

Define a stream function

Here is a screen cast https://youtu.be/418VvIt5HPc?t=18m21s

Stream Function

```
SELECT [deviceId] AS [deviceId],
                avg(temperature) AS avgtemp
                 INTO CosmosDB
                FROM
                   iothub
               GROUP BY
               deviceId,
               TumblingWindow(second, 15);
```

Create Event Hub

Create an EventHub which will receive data from Azure Function. Take a note of keys and name etc for Azure Function define below.

Create an Azure Function

Using Visual Studio, it is straight forward. Please see the screen cast to learn how to create an Azure Function in visual studio https://youtu.be/Mnq0O91i-0s?t=1m42s . Writing AzureFunction is simple on Portal too, please see the details below.

Using Visual Studio

Code

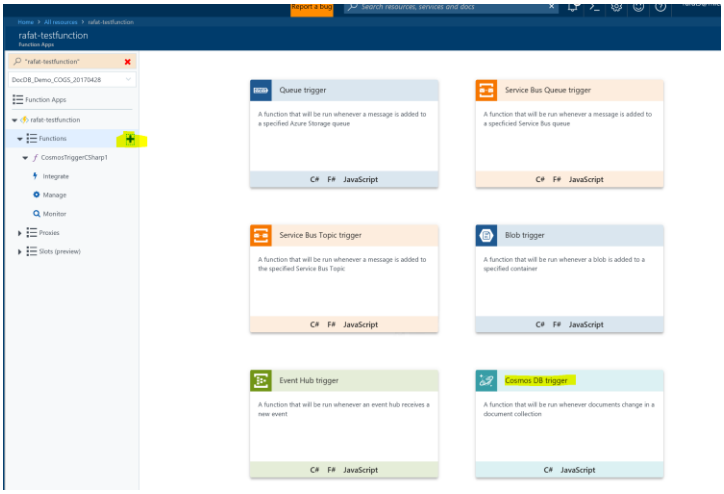
```
using System.Collections.Generic;
using System.Configuration;
using System.Text;
using Microsoft.Azure.Documents;
using Microsoft.Azure.EventHubs;
using Microsoft.Azure.WebJobs;
using Microsoft.Azure.WebJobs.Host;

namespace Lambda
{
    public static class StreamProcessor
    {
        [FunctionName("StreamProcessor")]
        public static void Run([CosmosDBTrigger(
            databaseName: "IoT",
            collectionName: "IoT",
            ConnectionStringSetting = "DBConnection",
            LeaseCollectionName = "leases")]IReadOnlyList<Document> input, TraceWriter log)
        {

            string connectionString = ConfigurationSettings.AppSettings["EventHubConnection"];
            var connectionStringBuilder = new EventHubsConnectionStringBuilder(connectionString)
            {
                EntityPath = "rafat-eventhub1" //Change as per your eventhub name
            };
            var client =
Microsoft.Azure.EventHubs.EventHubClient.CreateFromConnectionString(connectionStringBuilder.ToString());

            foreach (var doc in input)
            {
                string json = string.Format("{{\"iotid\":\"{0}\",\"temp\":{1}}}",
doc.GetPropertyValue<string>("iotid"), doc.GetPropertyValue<string>("temp"));
                EventData data = new Microsoft.Azure.EventHubs.EventData(Encoding.UTF8.GetBytes(json));
                client.SendAsync(data);
            }
        }
    }
}
```

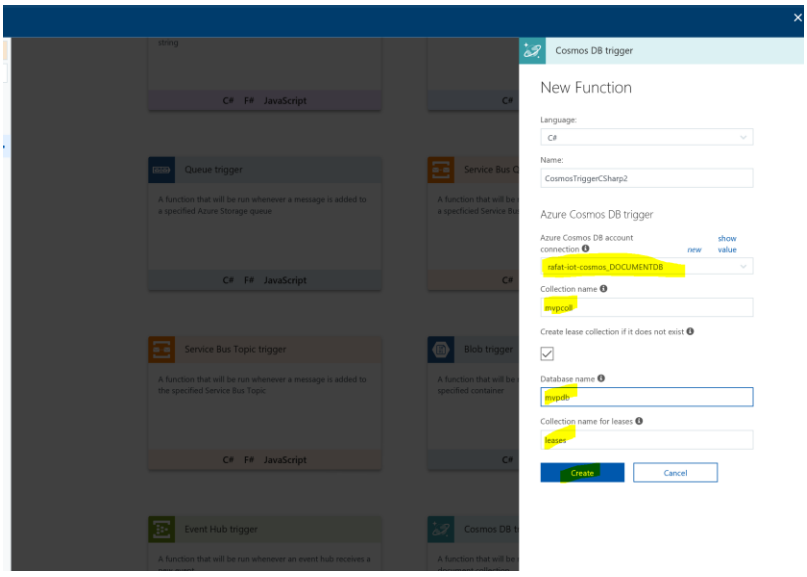
LocalSetting

```
{
    "IsEncrypted": false,
  "Values": {
    "AzureWebJobsStorage": "DefaultEndpointsProtocol=
    "AzureWebJobsDashboard": "DefaultEndpointsProtocol=https; ",
    "DBConnection": "AccountEndpoint=”
    "EventHubConnection": "Endpoint=sb ="

  }
}
```

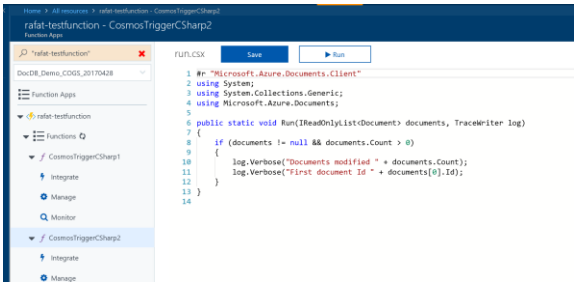
How to write the function directly on the portal

Create a Function app on the azure portal





You will see a template of function as follows:



Replace the above code with the following:

Code

```
#r "D:\home\site\wwwroot\CosmosTriggerCSharp1\bin\Microsoft.Azure.Documents.Client.dll"
#r "D:\home\site\wwwroot\CosmosTriggerCSharp1\bin\Microsoft.Azure.EventHubs.dll"


using System.Collections.Generic;
using System.Configuration;
using System.Text;
using Microsoft.Azure.Documents;
using Microsoft.Azure.EventHubs;

public static void Run(IReadOnlyList<Document> documents, TraceWriter log){

   if (documents != null && documents.Count > 0)    {

      string connectionString = ConfigurationManager.ConnectionStrings["EventHubConnection"].ConnectionString;
log.Verbose(connectionString);
      var connectionStringBuilder = new EventHubsConnectionStringBuilder(connectionString)  {

            EntityPath = "mvp-eventhub"
     };

      var client = Microsoft.Azure.EventHubs.EventHubClient.CreateFromConnectionString(connectionStringBuilder.ToString());
      foreach (var doc in documents) {
            string json = string.Format("{{\"iotid\":\"{0}\",\"temp\":{1}}}", doc.GetPropertyValue<string>("iotid"),
doc.GetPropertyValue<string>("temp"));
            EventData data = new Microsoft.Azure.EventHubs.EventData(Encoding.UTF8.GetBytes(json));
            client.SendAsync(data);
      }
   }
}
```

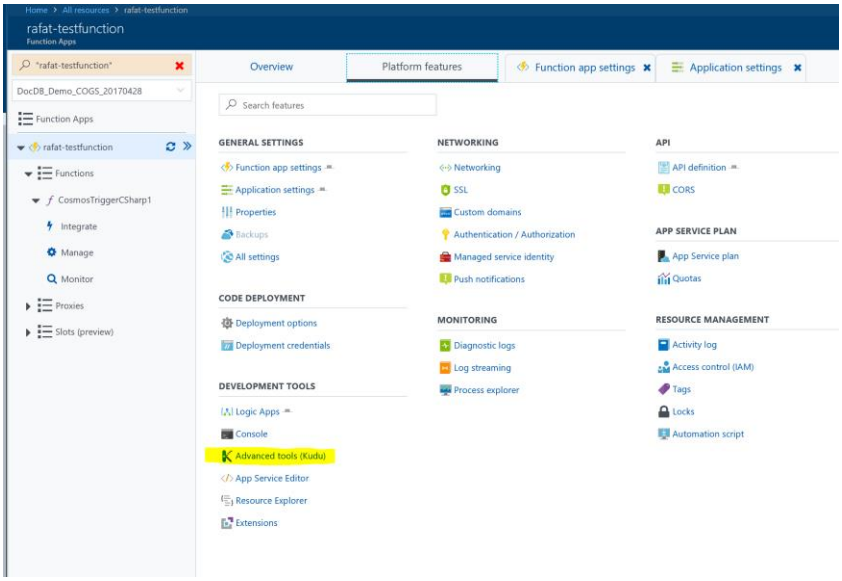
Function.json

No change needed in function.json

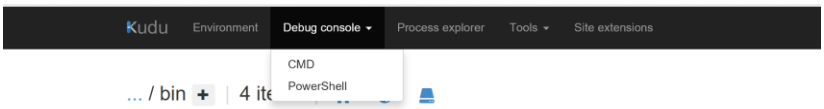
```
{

 "bindings": [
   {
     "type": "cosmosDBTrigger",
     "name": "documents",
     "direction": "in",
     "leaseCollectionName": "mvpleases",
     "connectionStringSetting": "rafat-iot-cosmos_DOCUMENTDB",
     "databaseName": "mvpdb",
     "collectionName": "mvpcoll",
     "createLeaseCollectionIfNotExists": true
   } ],

 "disabled": false

}
```

Upload the DLL to Azure Function

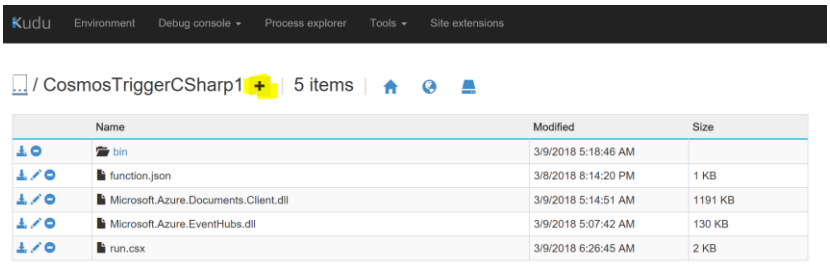
Open Kudu from platform features



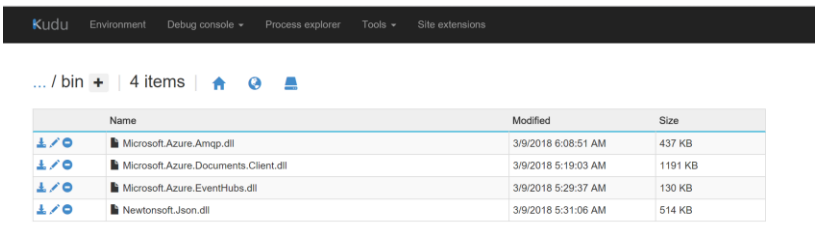
Click on Debug Console



Navigate to function directory by double clicking on wwwroot and then create a bin directory using the "+" sign.



Go to Bin directory by double clicking on it and then add the following DLL. Get these DLL from nuget



Microsoft.Azure.DocumentDB Version="1.21.0"

https://www.nuget.org/packages/Microsoft.Azure.DocumentDB/

Microsoft.Azure.EventHubs Version="1.1.0"

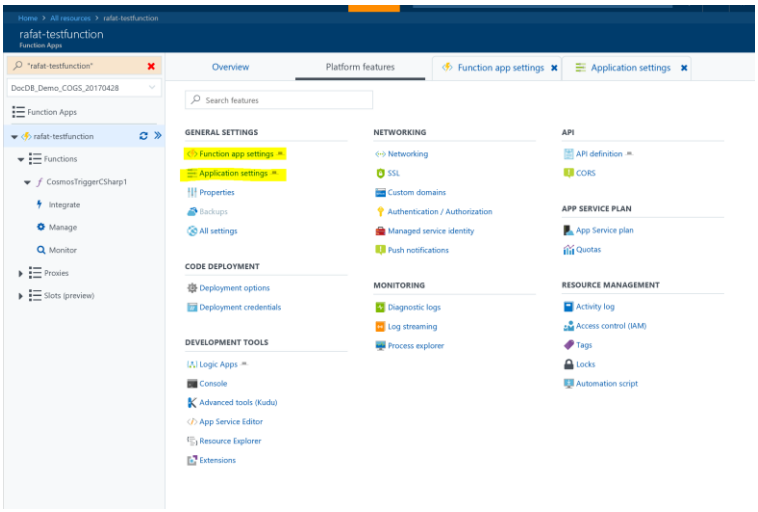
https://www.nuget.org/packages/Microsoft.Azure.EventHubs/

https://www.nuget.org/packages/Microsoft.Azure.Amqp/

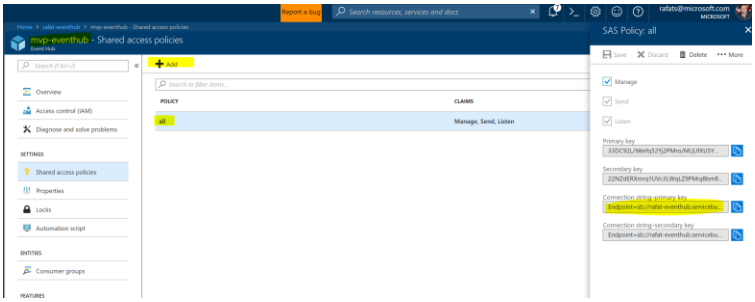
https://www.nuget.org/packages/Newtonsoft.Json

AppSettings

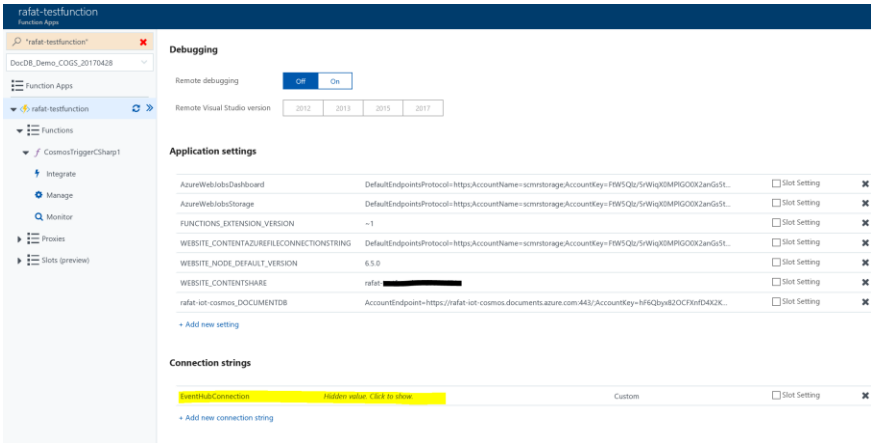
Click on AppSetting



On another tab open the EventHub and Get the EventHub connection String

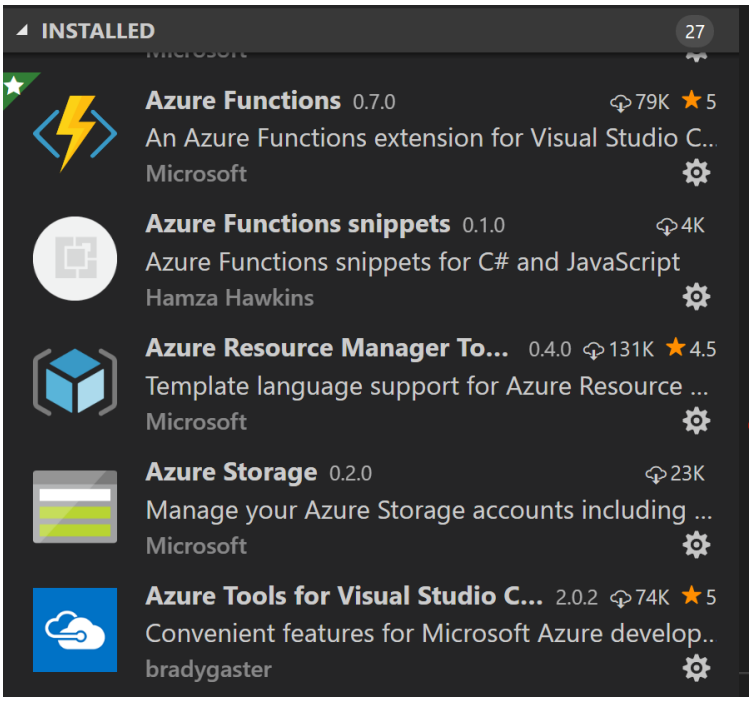


Goto Application Setting and update the EventHub Setting

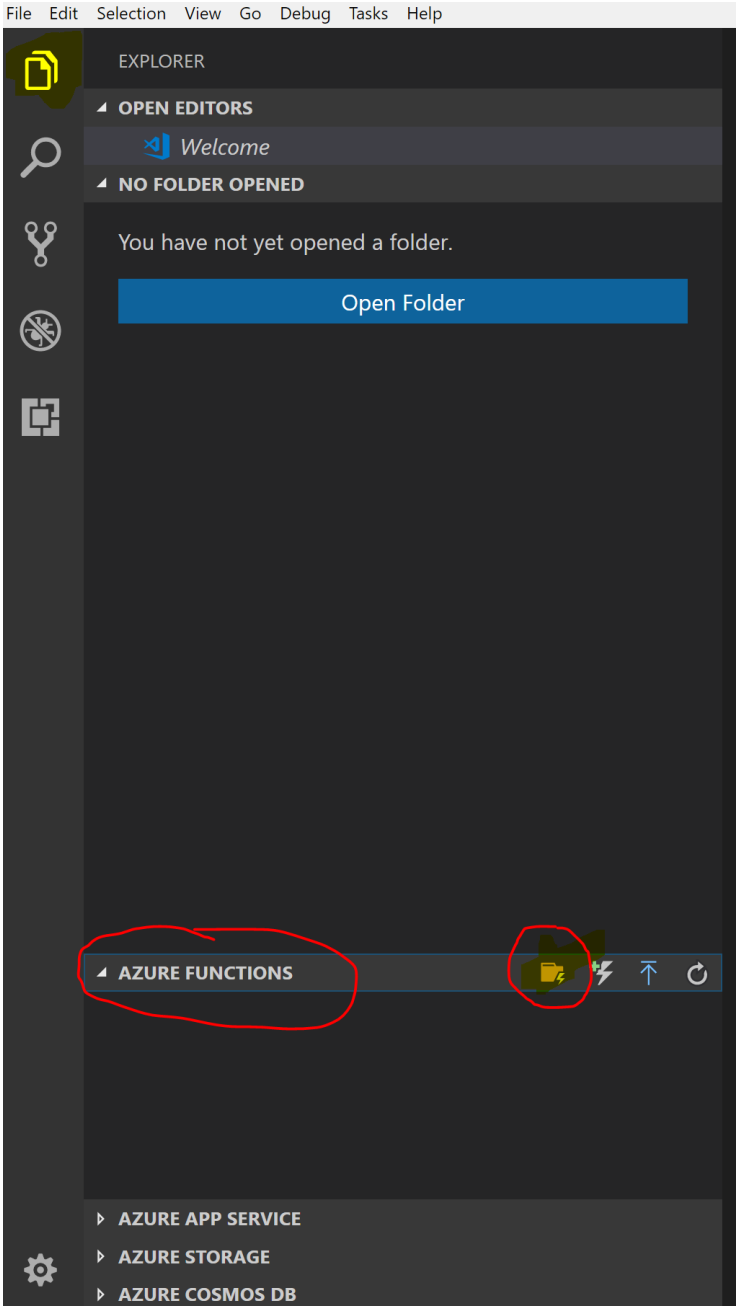


Azure Function using VS Code

Open code and install the Azure Functions extensions:

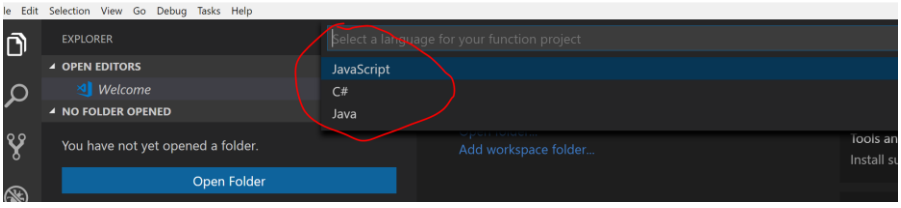


Click on Explorer on left hand top menu item

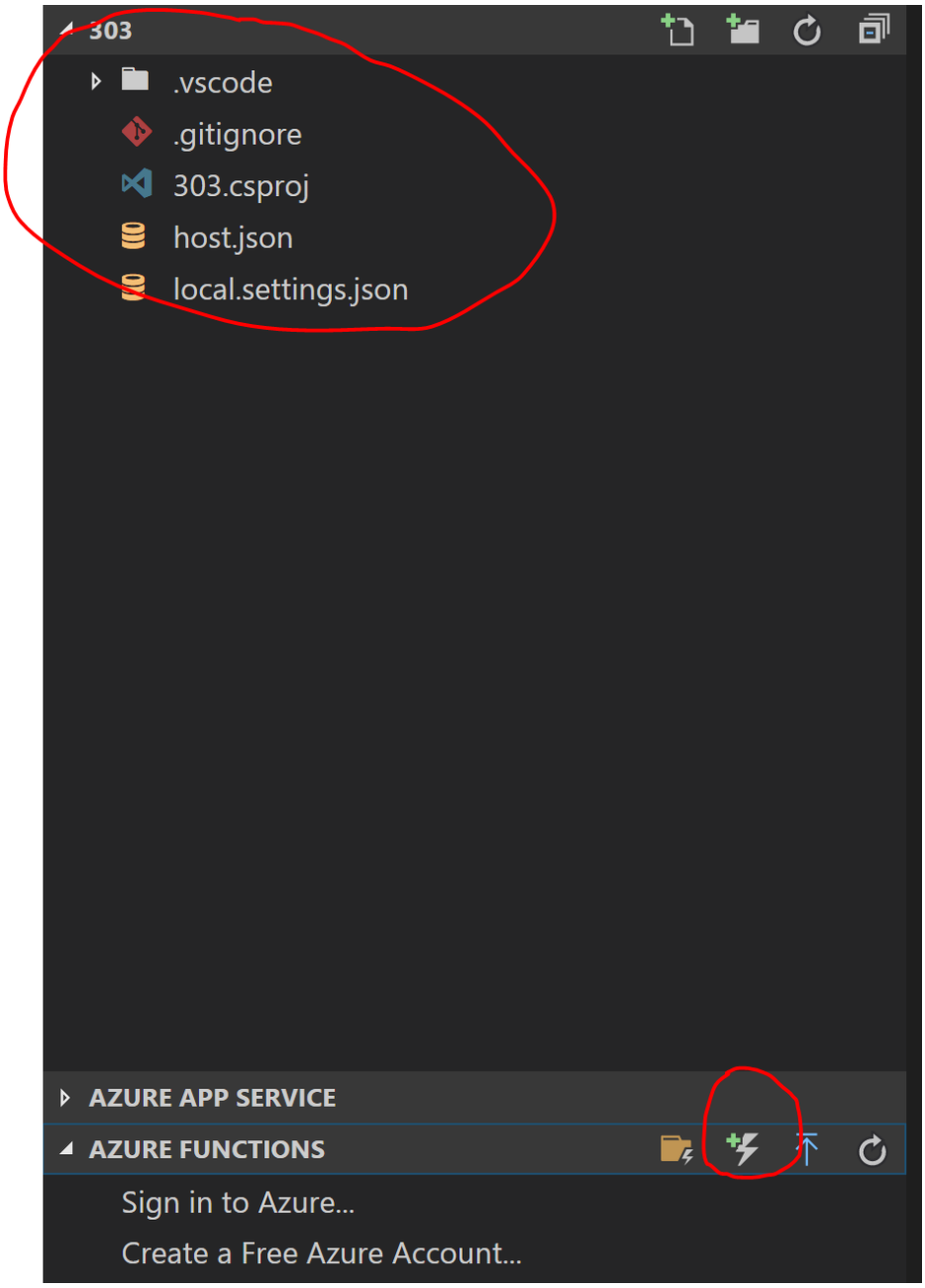


Select Azure Function and click on folder. Choose a new folder where you would like to keep the project.

Then choose the language you would like to use



After this you will see the project in folder, and then go ahead and choose to create the function



Choose Http Trigger and then later on in code you can change it to CosmosDB Trigger.

Make sure the code is as follows:

```
using System.Collections.Generic;

using System.Configuration;

using System.Text;

using Microsoft.Azure.Documents;

using Microsoft.Azure.EventHubs;
```

```
using Microsoft.Azure.WebJobs;
using Microsoft.Azure.WebJobs.Host;
using Microsoft.Azure.WebJobs.Extensions.DocumentDB;



namespace Company.Function
{
    public static class SendDataToEventHub
  {
      [FunctionName("SendDataToEventHub")]
      public static void Run([CosmosDBTrigger(
         databaseName: "mvpdb",
         collectionName: "mvpcoll",
         ConnectionStringSetting = "DBConnection",
         LeaseCollectionName = "mvpleases")]IReadOnlyList<Document> input, TraceWriter log)
     {
         // string connectionString = ConfigurationManager
          //   .ConnectionStrings["EventHubConnection"].ConnectionString;
         string connectionString = ConfigurationSettings.AppSettings["EventHubConnection"];

         log.Verbose(connectionString);
         var connectionStringBuilder = new EventHubsConnectionStringBuilder(connectionString)
         {
            EntityPath = "mvp-eventhub"
         };
         var client = Microsoft.Azure.EventHubs.EventHubClient.CreateFromConnectionString(connectionStringBuilder.ToString());

         foreach (var doc in input)
         {
            string json = string.Format("{{\"iotid\":\"{0}\",\"temp\":{1}}}", doc.GetPropertyValue<string>("iotid"),
doc.GetPropertyValue<string>("temp"));
            EventData data = new Microsoft.Azure.EventHubs.EventData(Encoding.UTF8.GetBytes(json));
            client.SendAsync(data);
         }
     }
```

```
  }
}
```

Local.setting.json

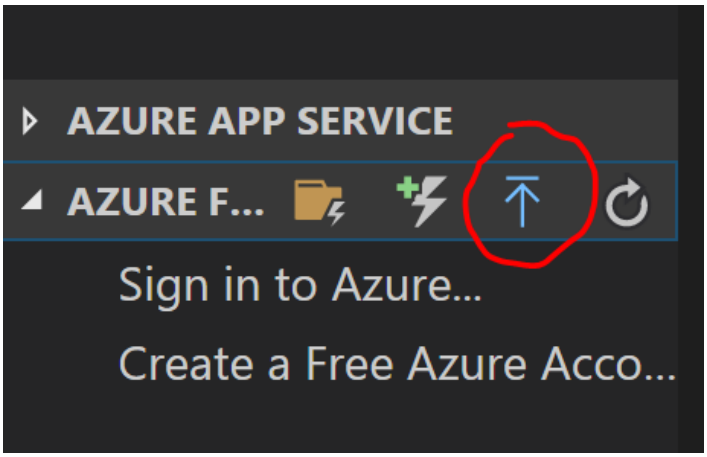
```
{
   "IsEncrypted": false,
 "Values": {
   "AzureWebJobsStorage": "DefaultEndpointsProtocol=https;;EndpointSuffix=core.windows.net",
   "AzureWebJobsDashboard": "DefaultEndpointsProtocol=;EndpointSuffix=core.windows.net",
   "DBConnection": "AccountEndpoint=https://rafat-iot-cosmos.",
   "EventHubConnection": "Endpoint=sb://rafat- "
 }
}
```

Update the vscode.proj file with following references

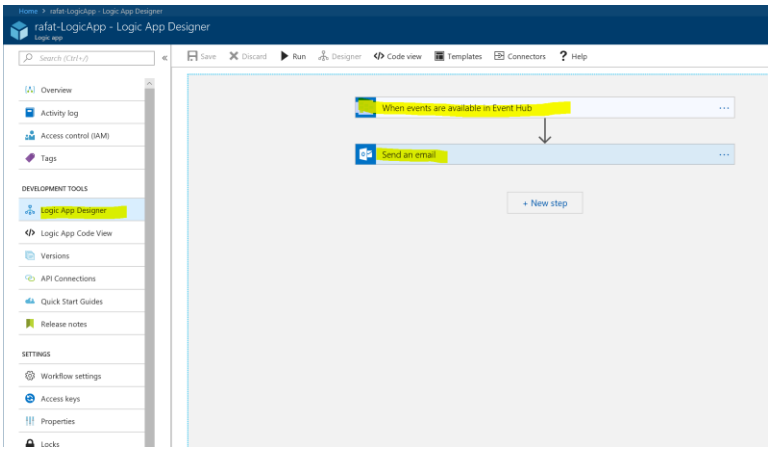
```
<Project Sdk="Microsoft.NET.Sdk">
  <PropertyGroup>
    <TargetFramework>net461</TargetFramework>
  </PropertyGroup>
  <ItemGroup>
   <PackageReference Include="Microsoft.NET.Sdk.Functions" Version="1.0.7" />
    <PackageReference Include="Microsoft.Azure.EventHubs" Version="1.1.0" />
    <PackageReference Include="Microsoft.Azure.WebJobs.Extensions.DocumentDB"
Version="1.1.0-beta4" />
  </ItemGroup>
  <ItemGroup>
    <Reference Include="Microsoft.CSharp" />
    <Reference Include="System.Configuration" />
  </ItemGroup>
```

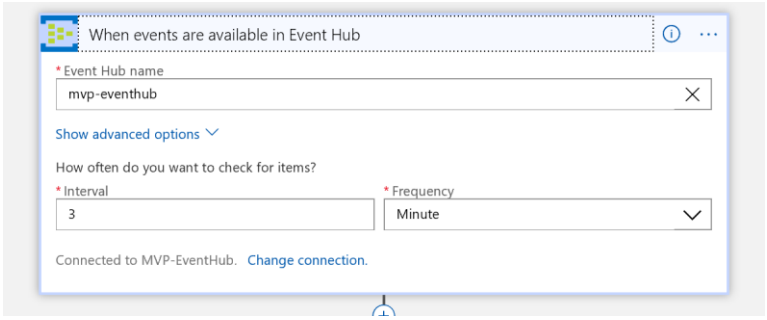
That's it Hit F5 to run the code.

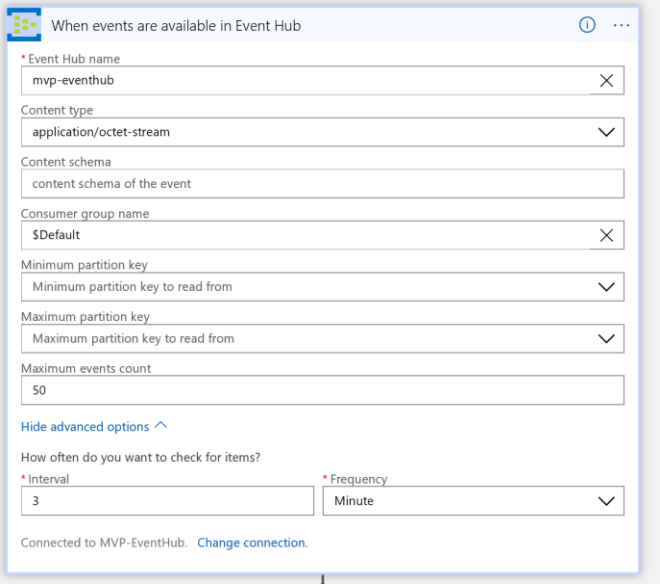
Afterwards upload the function to cloud

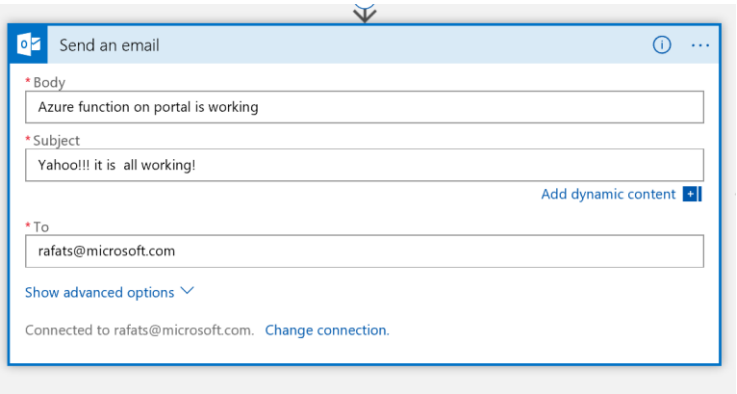

AZURE APP SERVICE
AZURE F...
Sign in to Azure...
Create a Free Azure Acco...

Logic App


rafat-LogicApp - Logic App Designer
When events are available in Event Hub
Send an email
+ New step


When events are available in Event Hub
* Event Hub name
mvp-eventhub
Show advanced options
How often do you want to check for items?
* Interval
3
* Frequency
Minute
Connected to MVP-EventHub. Change connection.





Links:

https://blog.todotnet.com/2017/07/building-azure-functions-in-c-with-visual-studio-code/

Debugging in Code

https://github.com/Microsoft/vscode-azurefunctions/blob/master/docs/func64bit.md

Nuget package in Visual Studio Code

https://stackoverflow.com/questions/40675162/install-a-nuget-package-in-visual-studio-code

How to add assembly refence in Azure Function

https://blogs.msdn.microsoft.com/benjaminperkins/2017/04/13/how-to-add-assembly-references-to-an-azure-function-app/