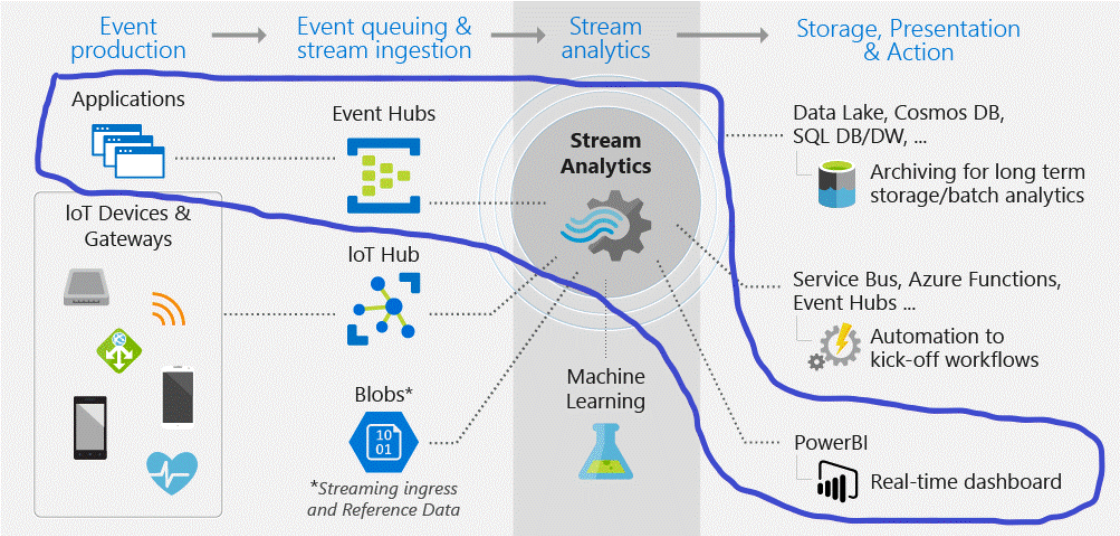
Demo Introduction

This setup guide will step you through creating an end-to-end demo of Python writing to an Azure Event Hub, Azure Stream Analytics mulling from Azure Event Hub and finally creating a real time line chart in Power BI.

The section of this architecture diagram outlined in blue represent the architecture of this demo.



Prerequisites

A computer with python installed.

The first thing you will need is somewhere to run the sample Python code that will simulate a sine wave.

Go to https://www.python.org/downloads/ download and install the latest version of Python.

If the machine you have does not have the module to include service bus calls you can install it using the following command.

pip install azure-eventhub==1.3.*

You will need to make minor changes to the python code to include the name of your event hub and your event hub credentials. This can be done in notepad or you could install Visual Studio or Visual Studio Code.

An Azure Account

If you have an Azure account, you can skip this step. If you do not have an Azure account you can sign up for a free trial here: https://azure.microsoft.com/en-us/free/search/?&OCID=AID2000128_SEM_XuZaUwAAAGT3OwZj:20200614171147:s&msclkid=024061b7e2a71698aefd3f51b8e83491&ef_id=XuZaUwAAAGT3OwZj:20200614171147:s&dclid=CJnflr7mgeoCFYGHaQod36cHcw

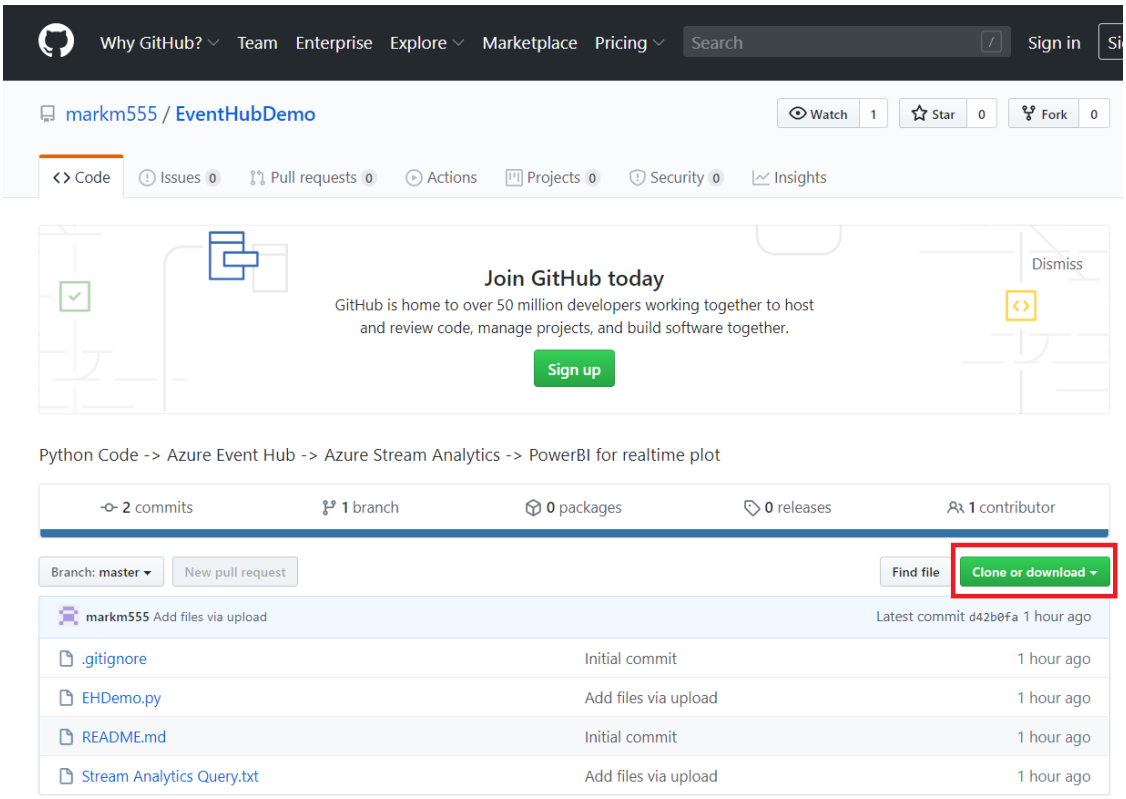
A PowerBI account

If you have a Power BI account you can skip this step. If you do not have a Power BI account you and sign up for a free trial here: https://powerbi.microsoft.com/en-us/

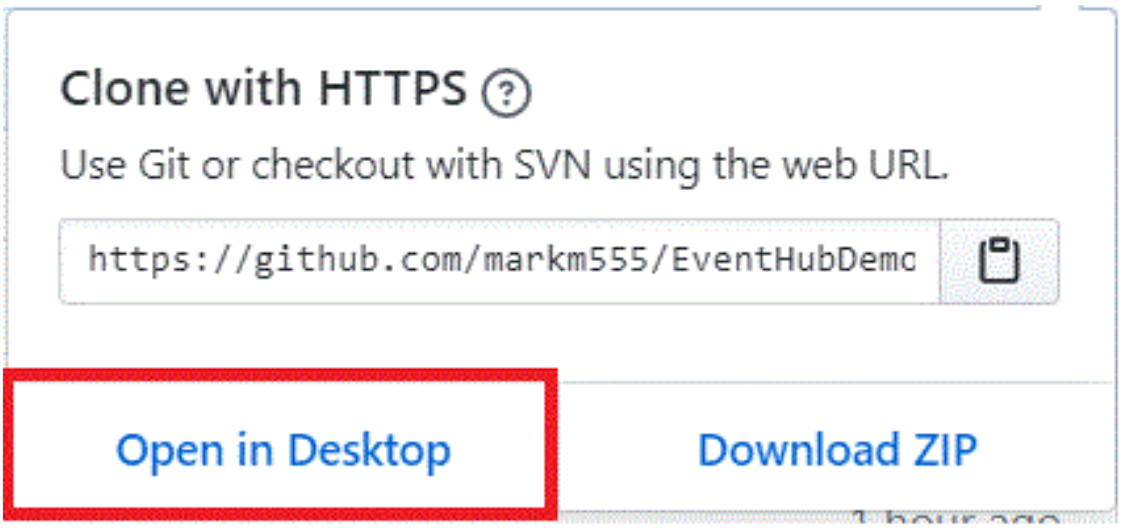
Clone the Git Hub repository

From the machine with python installed go to https://www.github.com/markm555/EventHubDemo and clone the repository to your machine.

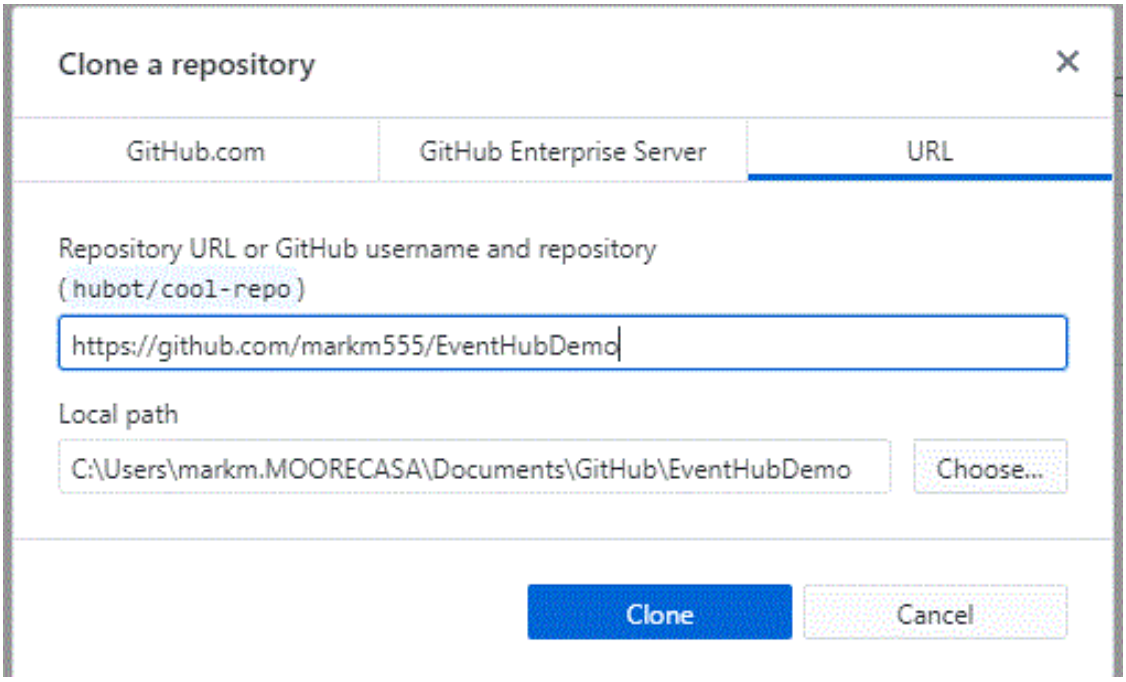
Click on Clone or download and the following dialog box will display:



Click on Open in Desktop



You may be prompted to install Github desktop. If you are click okay and the following dialog box will appear:



You can accept the default local path or change it to something different. The path you choose is where we will do the remainder of the work for this demo.

Setting up the Demo

Log into Azure and begin setting up the resources we will need for this demo: In this section you will:

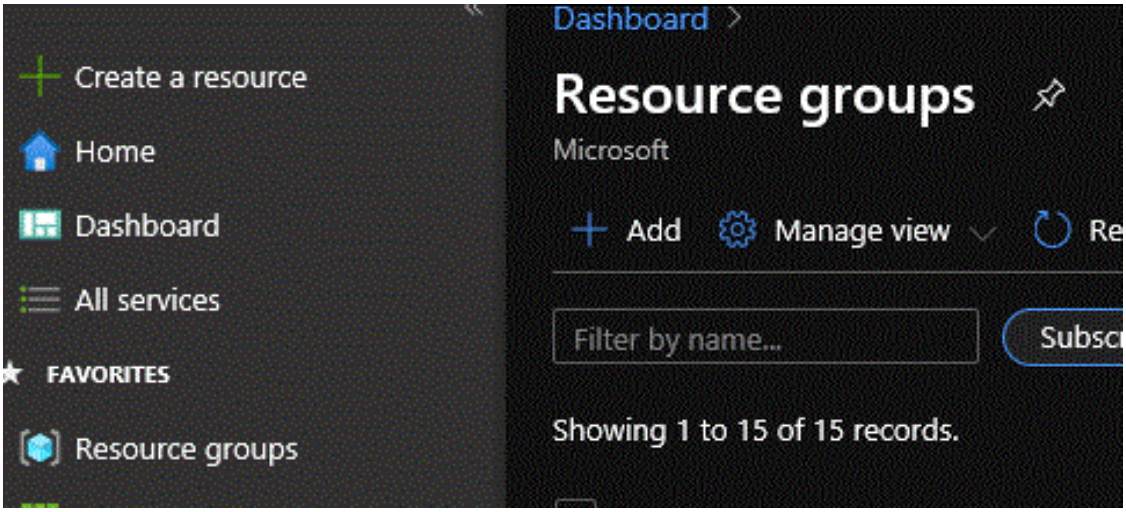
1. Create a Resource Group

2. Create an Event Hub
3. Create a Stream Analytics Job with the Event Hub as your input and Power BI as your output.
4. Create a Stream Analytics Query
5. Run the Stream Analytics Query

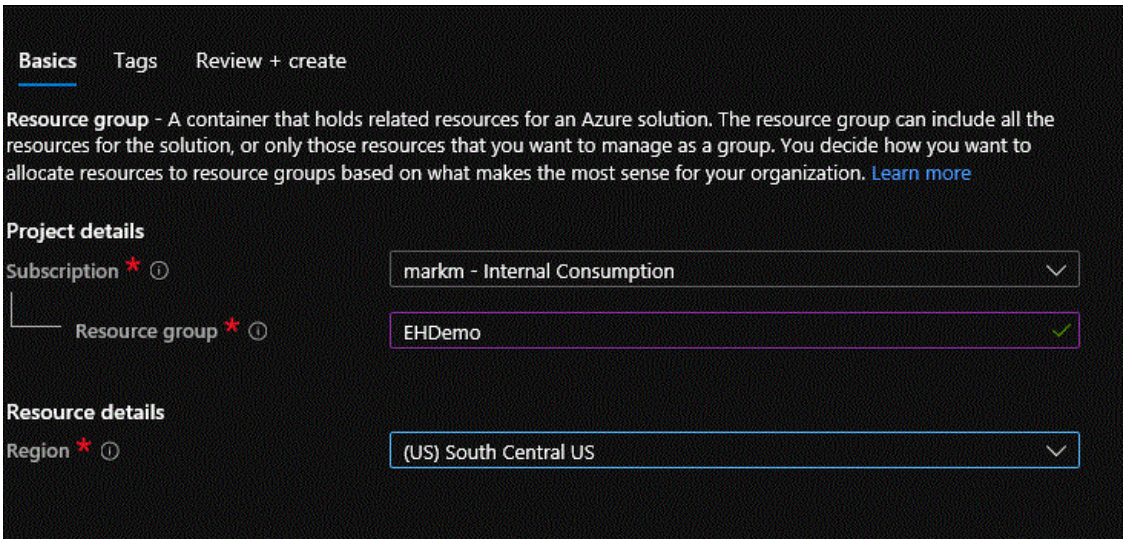
Go to https://portal.azure.com and let's get started.

Create a Resource Group

Once you have logged into Azure click on Resource Groups and Click on + Add

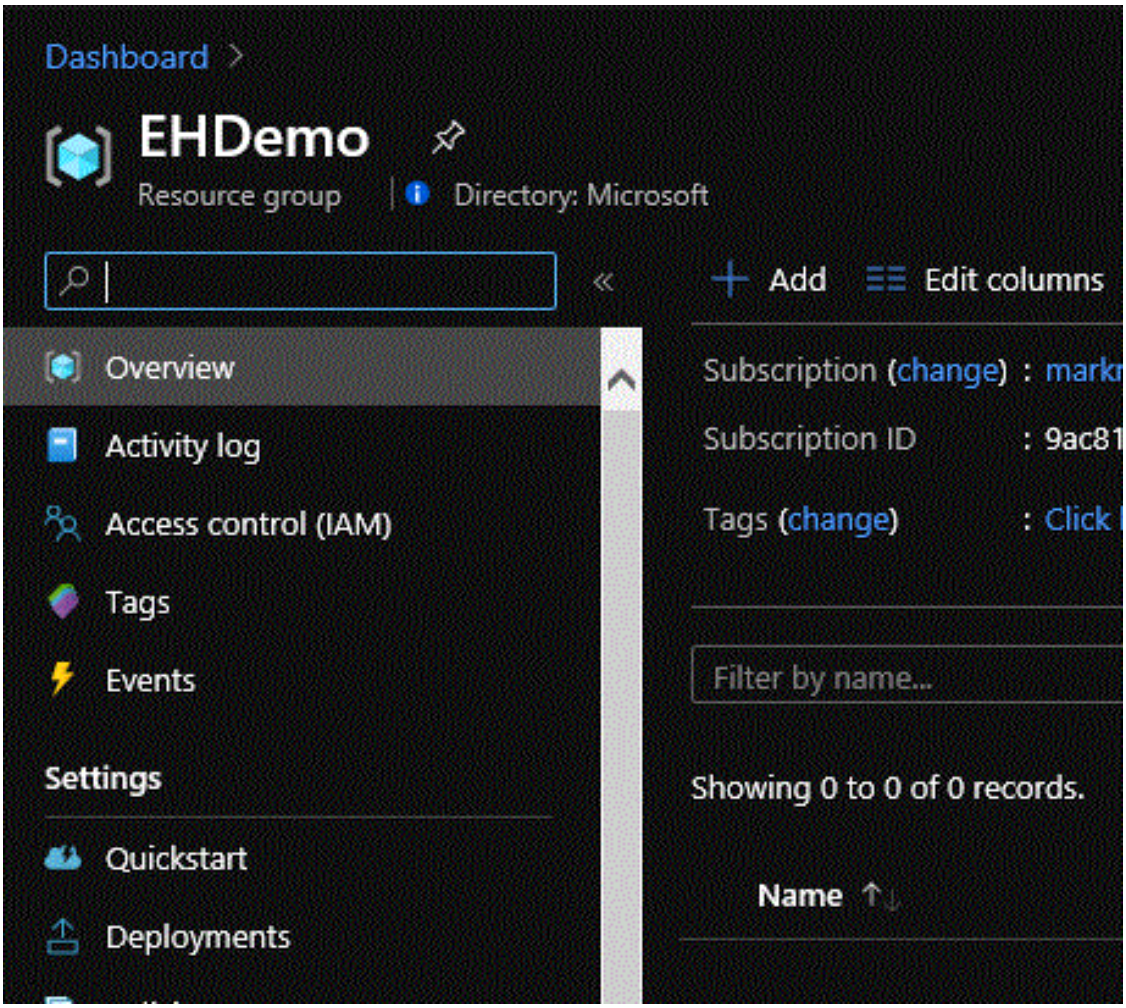


Chose a name for your resource group and choose a region for the resource group

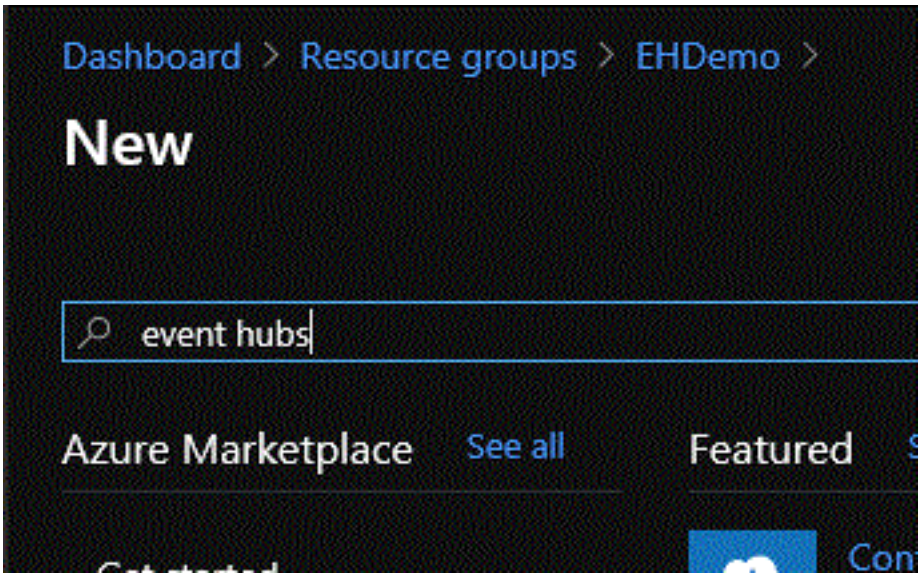


Creating an Event Hub

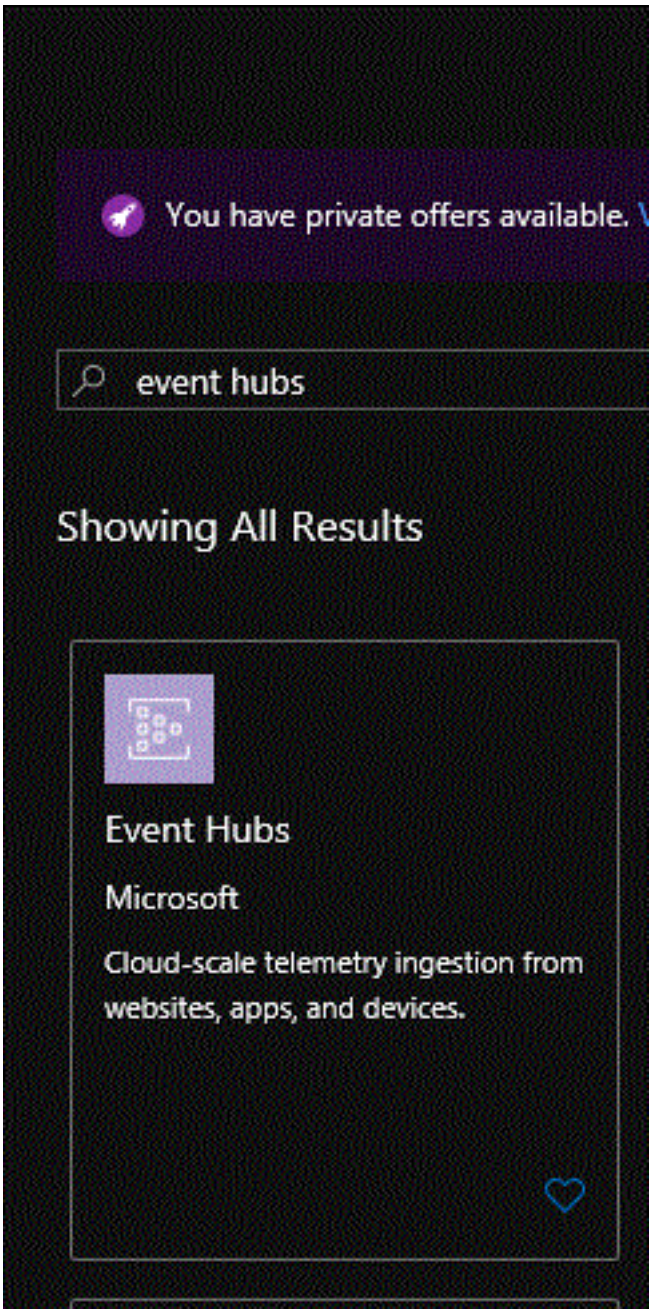
Click on your newly created resource group and click on + Add.



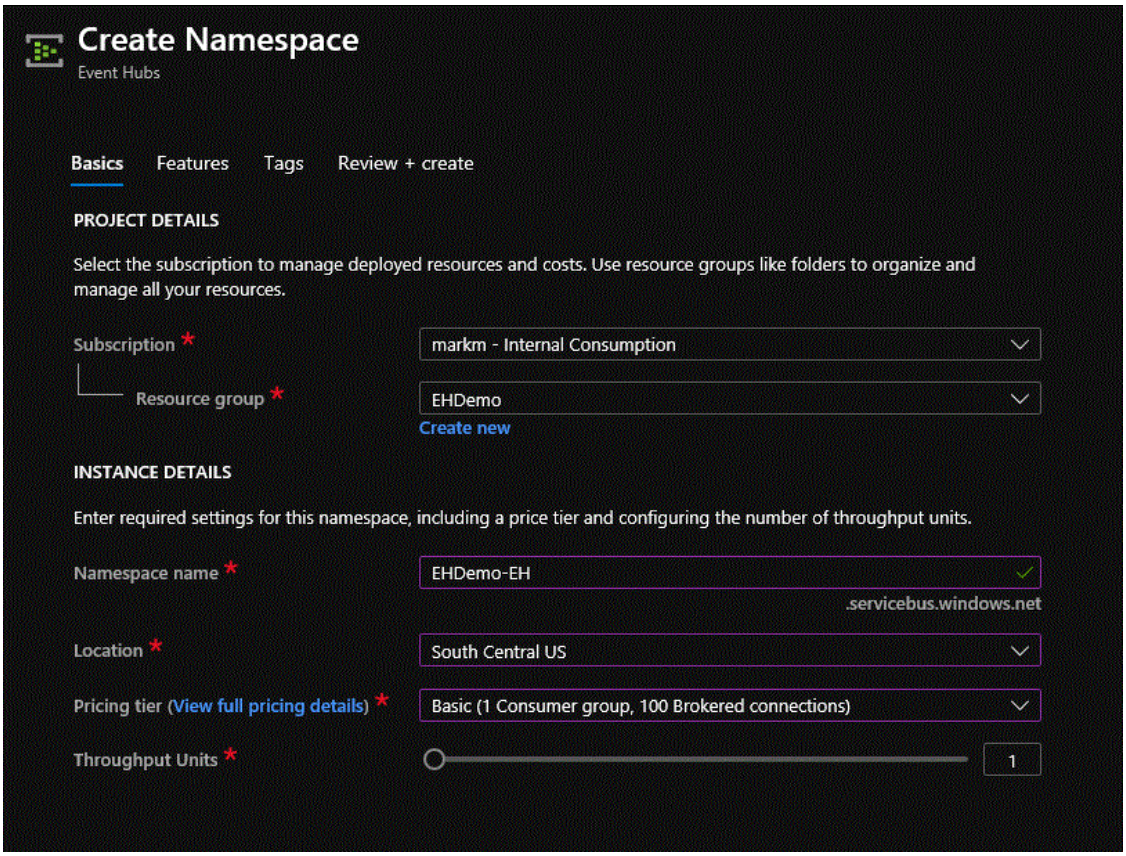
In the search box type “event hubs”



Select event hubs

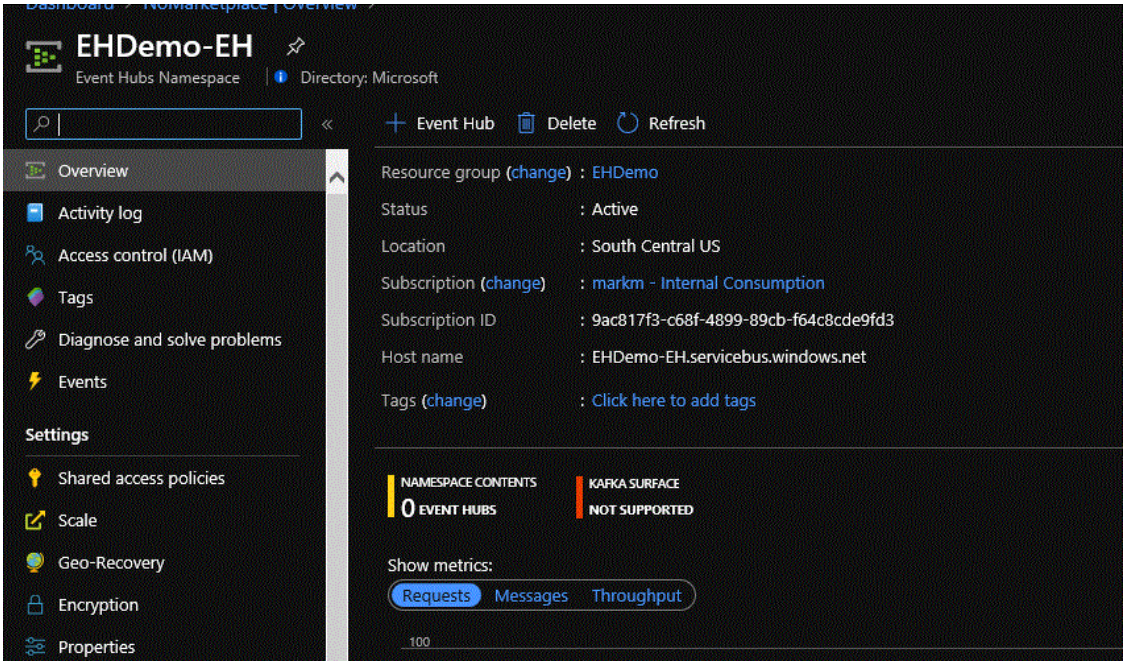


In this screen you will need to provide the resource group where you want the event hub, a namespace for the event hub, a region for the event hub, a pricing tier (basic works fine and is the least expensive).

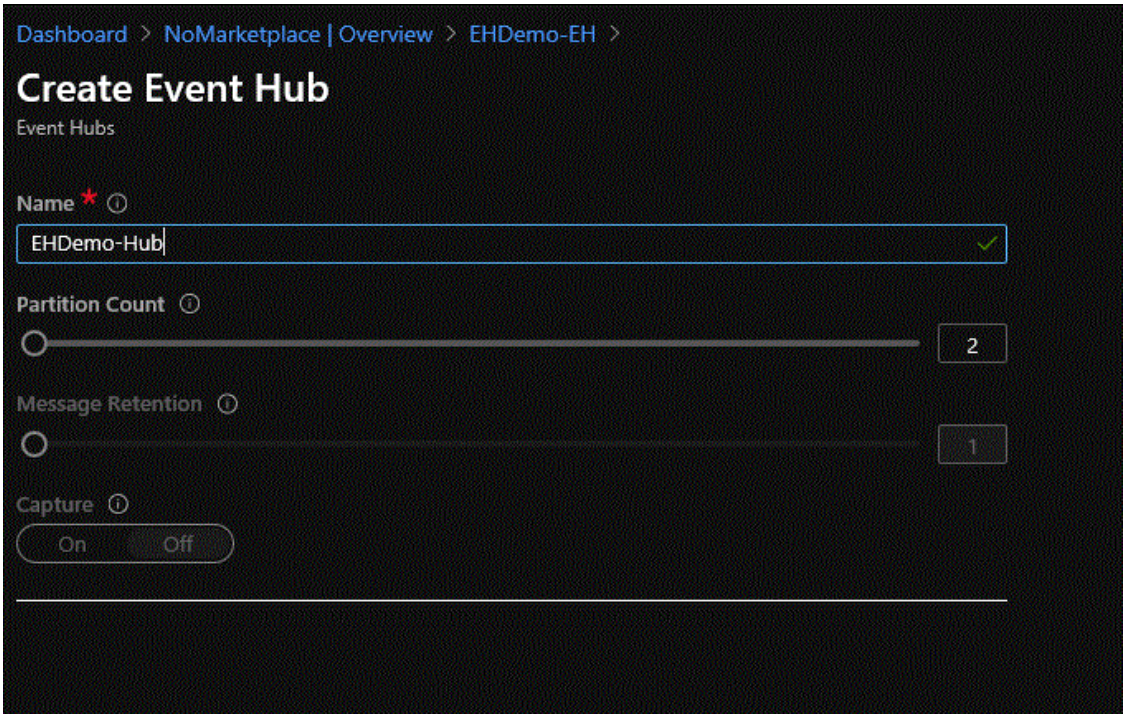


Click Create and once the event hub is created click on it in your resource group.

Click on + Event Hub to create an Event Hub.



Make a note of your Host name displayed above, you will need that for the python portion of this Demo.

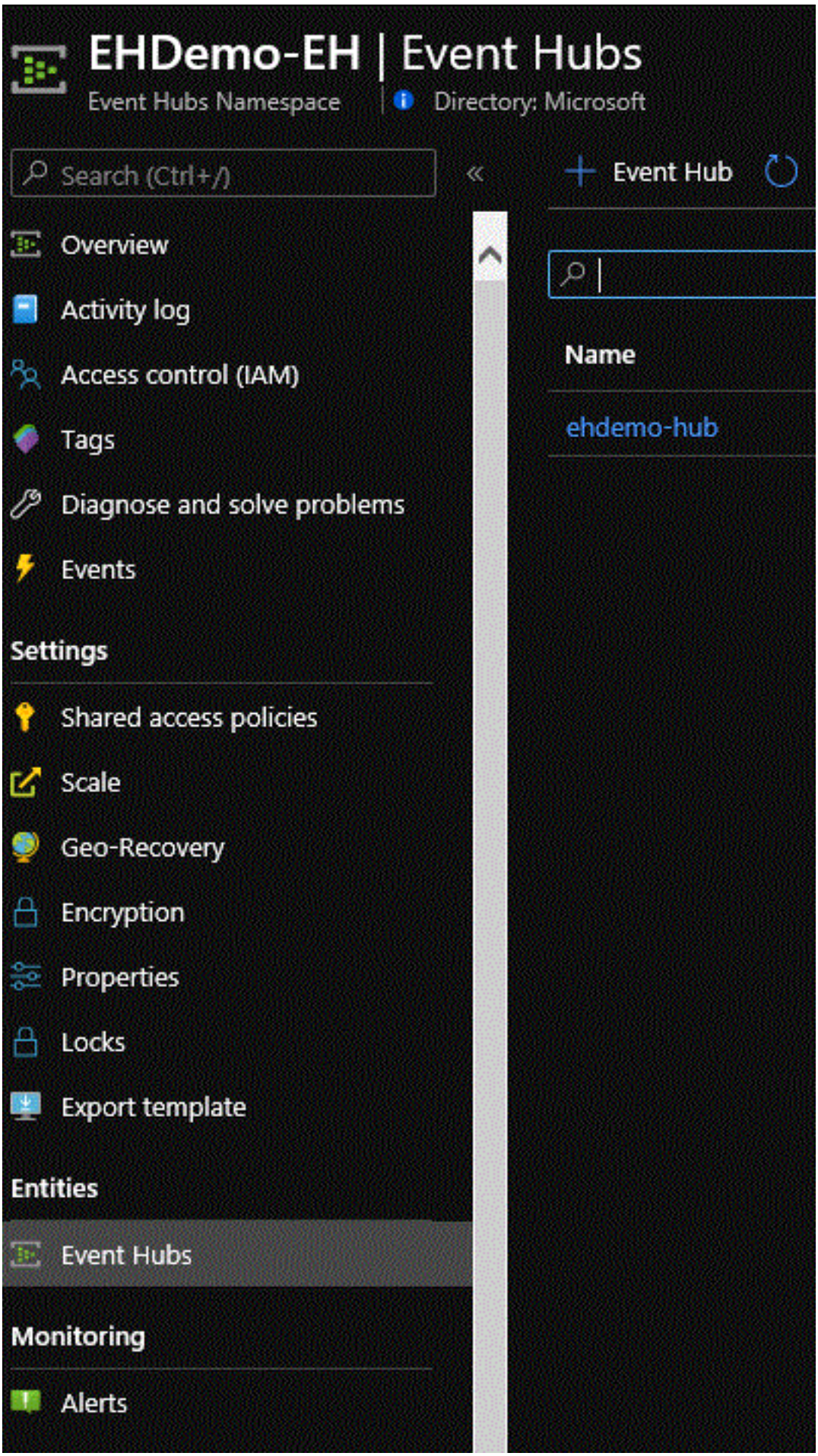


Choose a name for the Event Hub and make a note of this name as well, you will need that for the python portion.

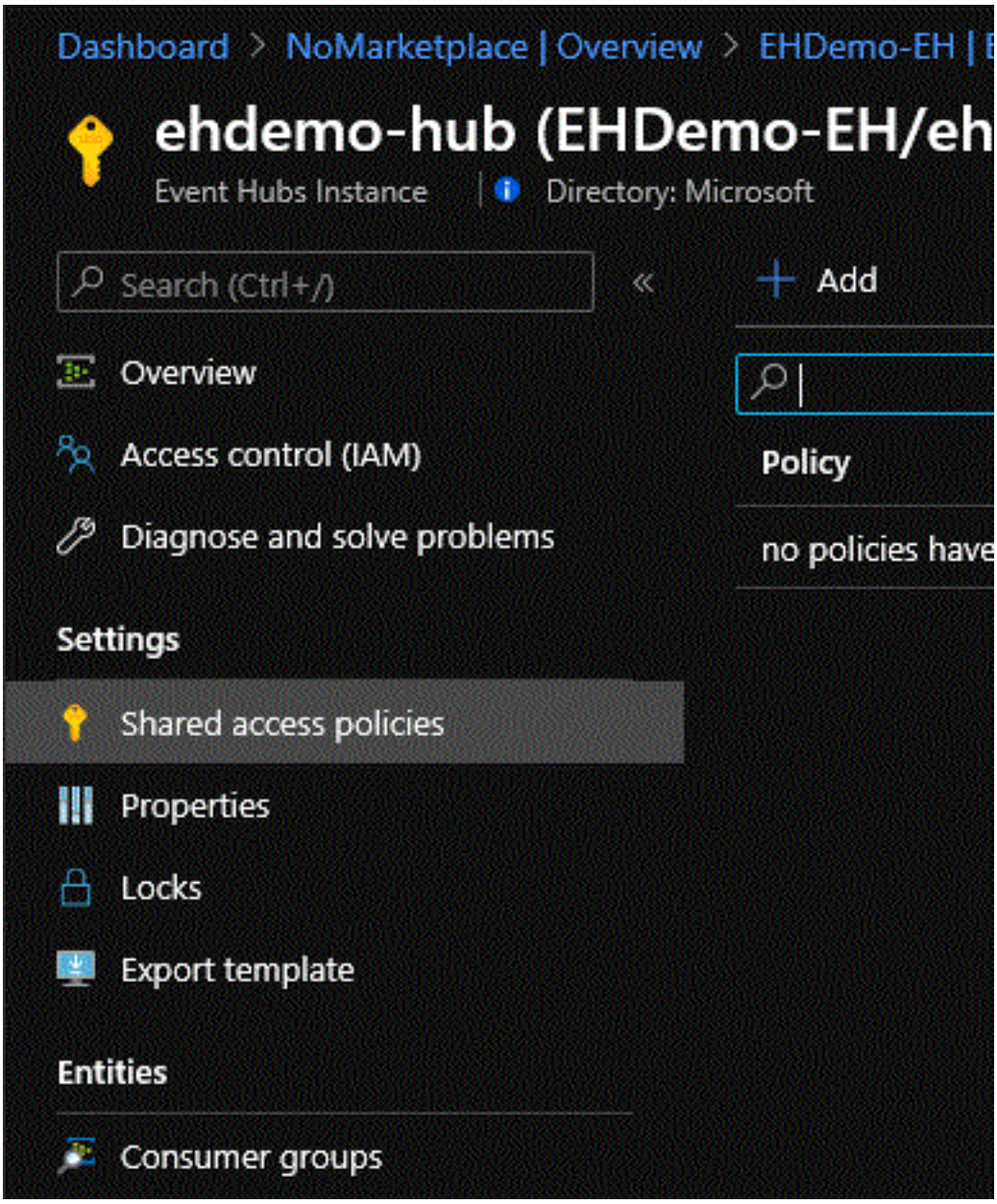
Click Create.

Once the Event Hub has been created you will need to create a policy and get a key.

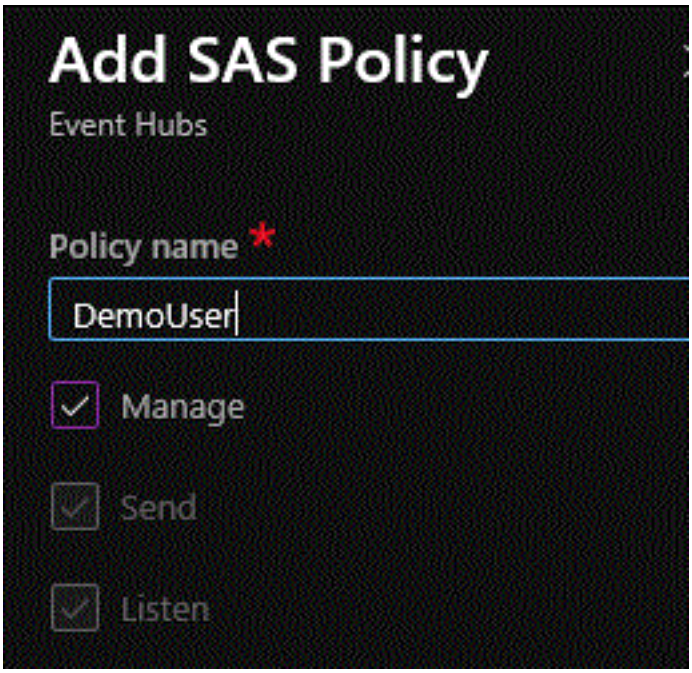
Under Entities click on Event Hubs and Click on the Event Hub you just created.


EHDemo-EH | Event Hubs
Event Hubs Namespace
Directory: Microsoft
Search (Ctrl+/)
Event Hub
Overview
Activity log
Access control (IAM)
Tags
Diagnose and solve problems
Events
Settings
Shared access policies
Scale
Geo-Recovery
Encryption
Properties
Locks
Export template
Entities
Event Hubs
Monitoring
Alerts
Name
ehdemo-hub

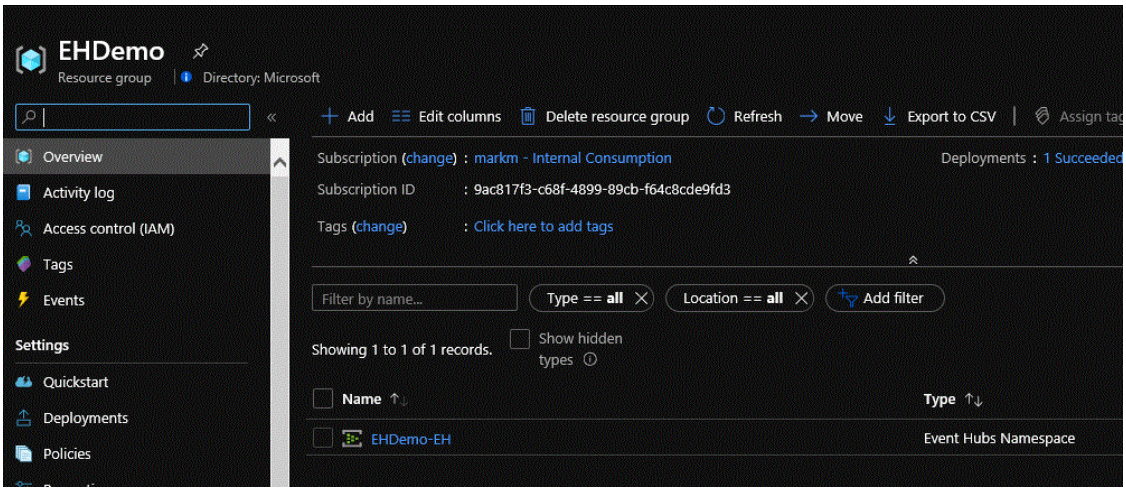
Click on Shared access policies and click + Add



Create a Policy name and select Manage. Send and Listen will fill in automatically.

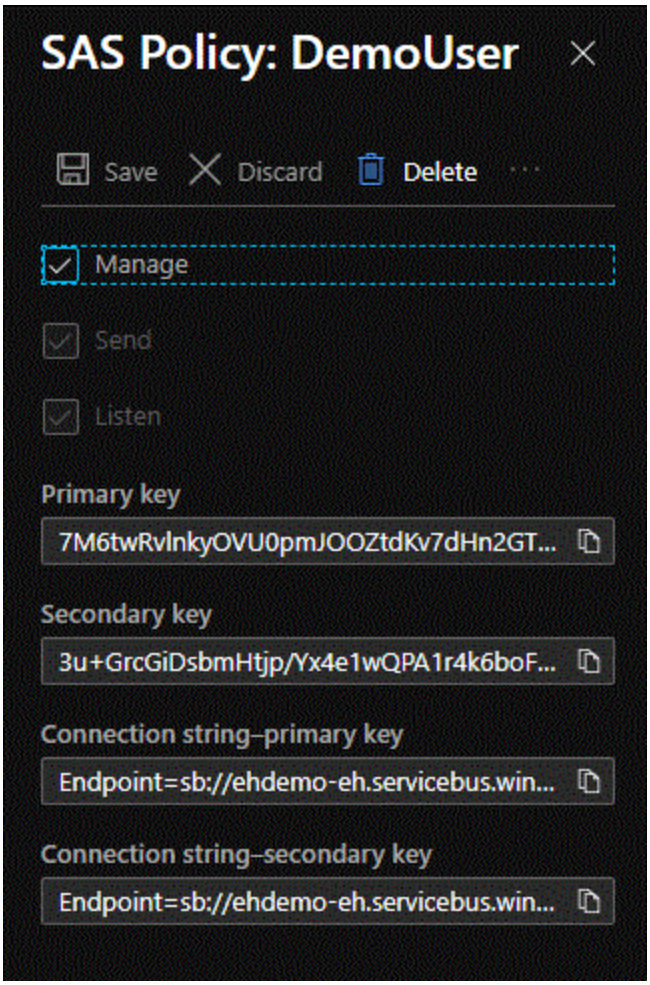


Make a note of the Policy name you will need this for the Python portion of the Demo setup.



Click on the newly created Share Access Policy and you will see you keys.

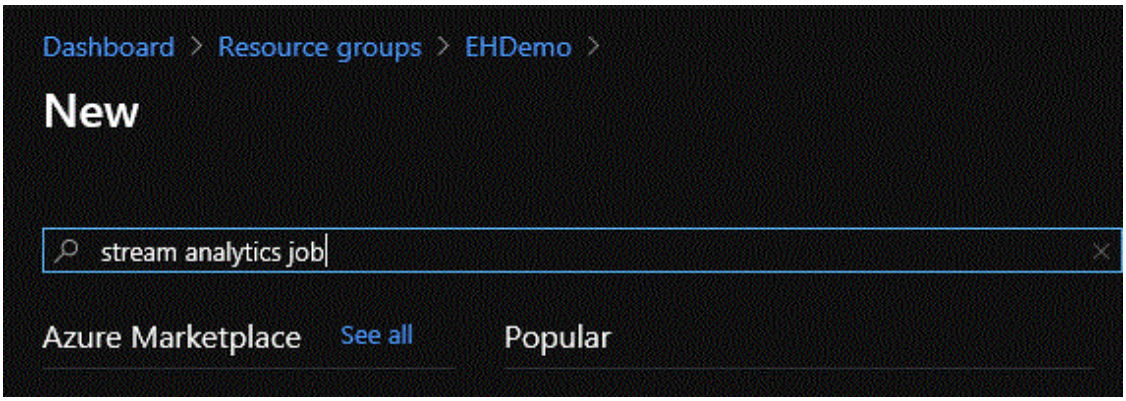
Copy your Primay Key and paste it where you have been keeping the other notes you have made for the Python portion of the demo. This is all the information you will need for the Python section.



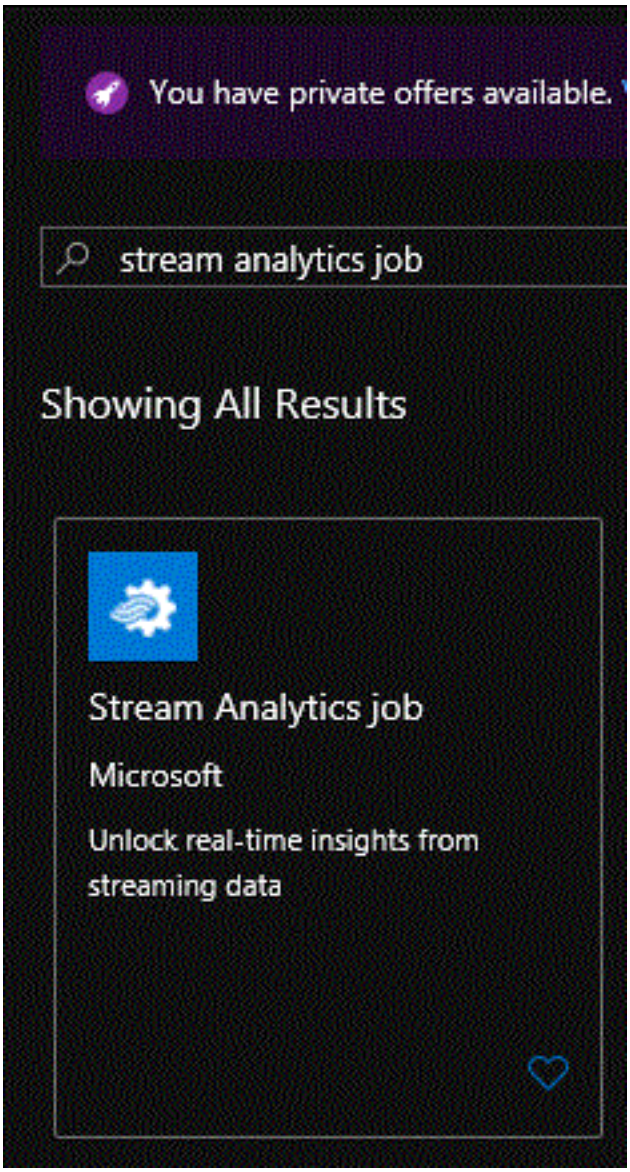
Create a Stream Analytics Job.

Go back to your resource group and click on + Add like you did in the previous step.

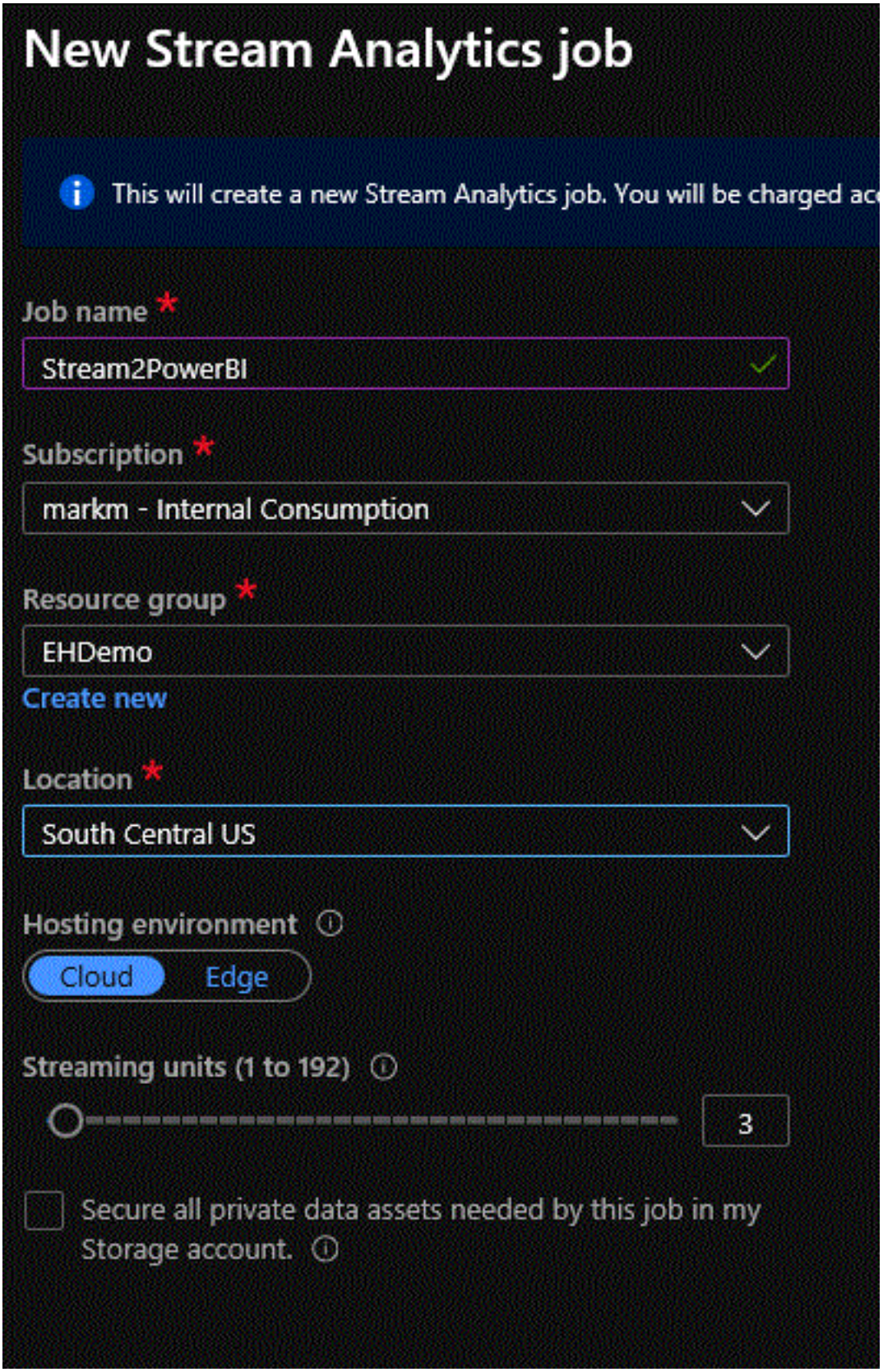
Search for “stream analytics job”



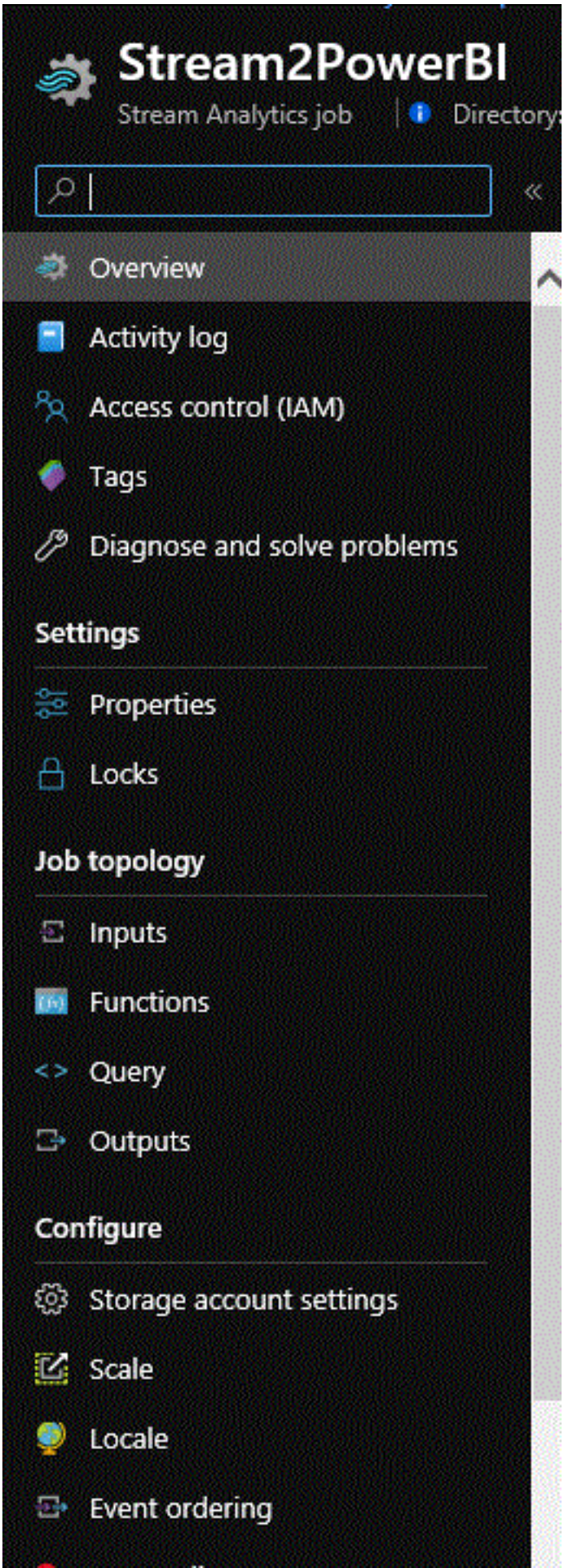
Click on Stream Analytics Job



You will need to provide a Job name, a resource group and region.

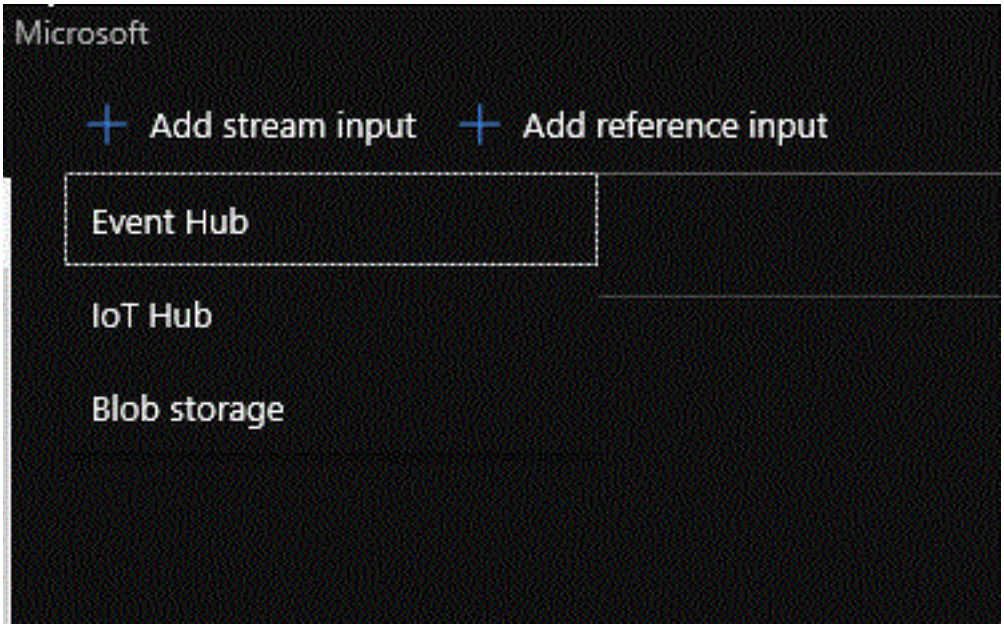


Click Create and go to the resource once it is created.

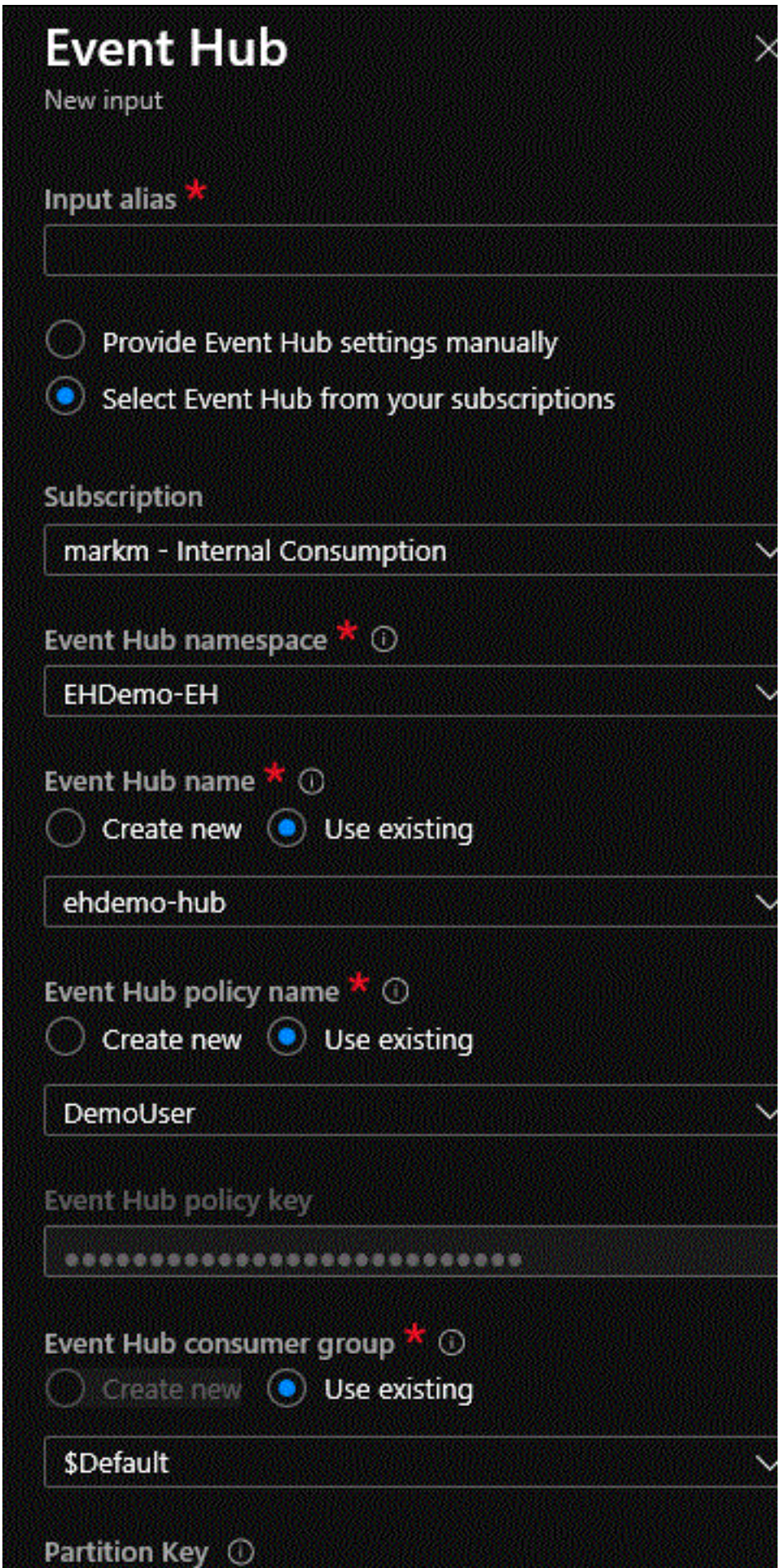

Stream2PowerBI
Stream Analytics job
Directory:
Overview
Activity log
Access control (IAM)
Tags
Diagnose and solve problems
Settings
Properties
Locks
Job topology
Inputs
Functions
Query
Outputs
Configure
Storage account settings
Scale
Locale
Event ordering

Under Job Topology click on input and then click + Add stream input

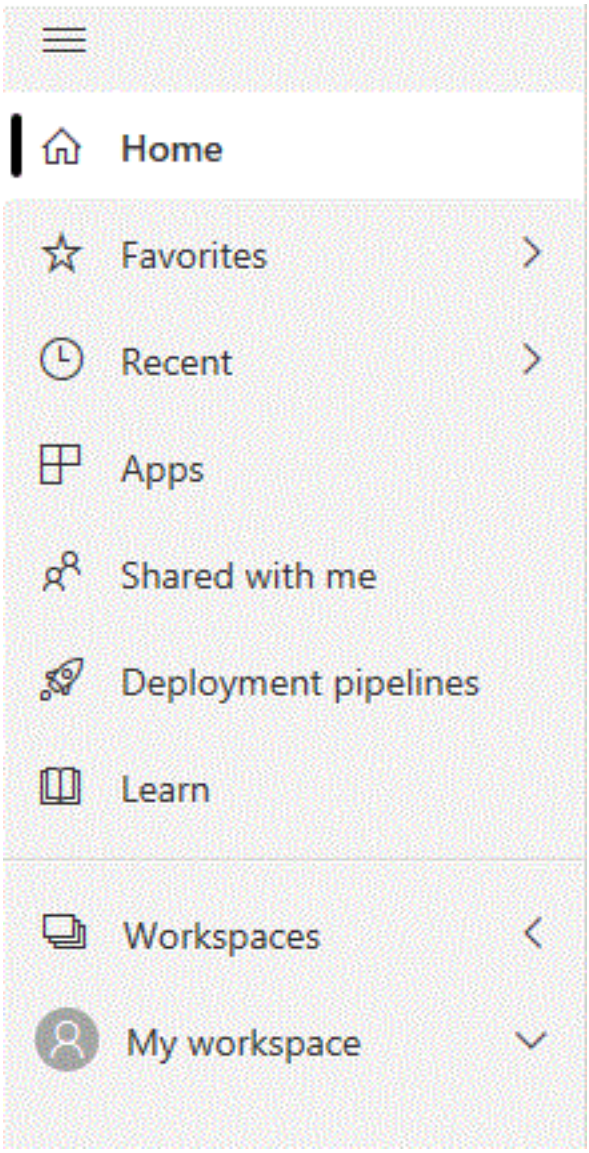
Select Event Hub as your stream input.



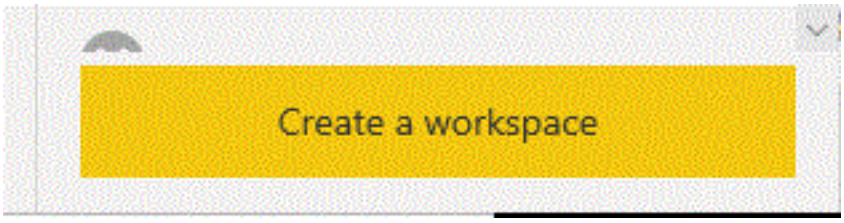
You will be asked to provide an input alias, namespace, event hub name and policy name. Select the ones you created while setting up your event hub.


Event Hub
New input
Input alias *
Provide Event Hub settings manually
Select Event Hub from your subscriptions
Subscription
markm - Internal Consumption
Event Hub namespace *
EHDemo-EH
Event Hub name *
Create new
Use existing
ehdemo-hub
Event Hub policy name *
Create new
Use existing
DemoUser
Event Hub policy key
Event Hub consumer group *
Create new
Use existing
$Default
Partition Key

We will need to create a new workspace in Power BI for our Demo. Log in to Power BI and on the left hand side of the screen select Workspaces.



Now select Create a workspace.



Provide the workspace with a name and click save.

Create a workspace

YOU'RE CREATING AN UPGRADED WORKSPACE

Enjoy new features, better sharing options, and improved security controls.

Revert to classic | Learn more

Workspace image

Upload

Delete

Workspace name

EHDemo

Available

Description

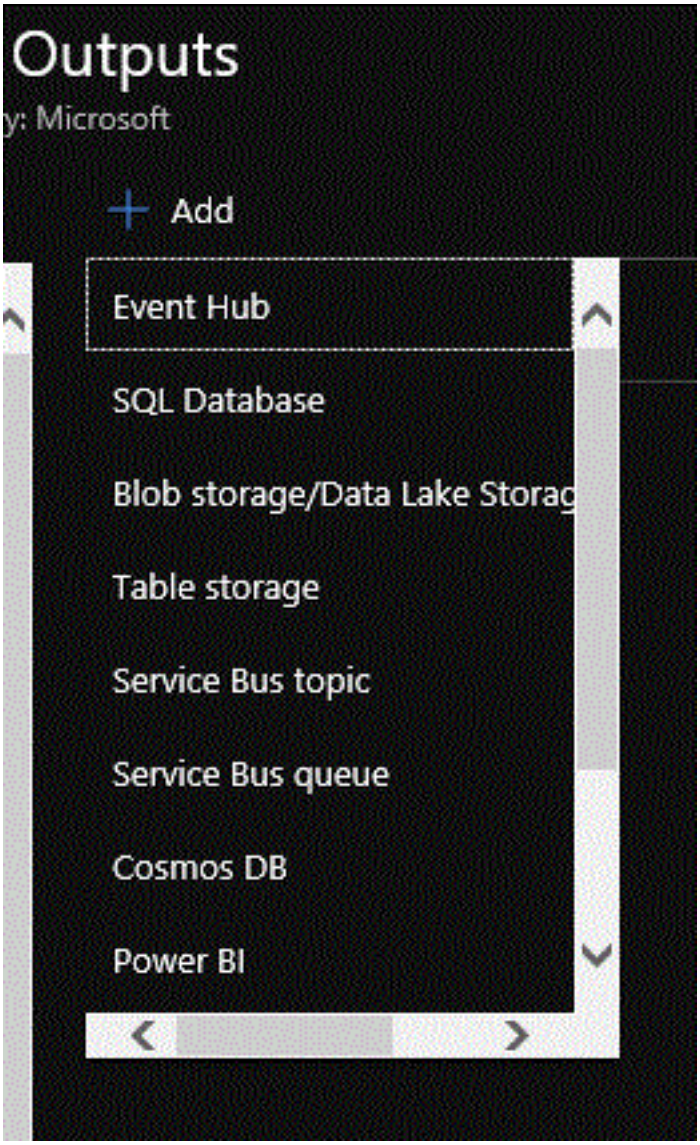
Describe this workspace

Learn more about workspace settings

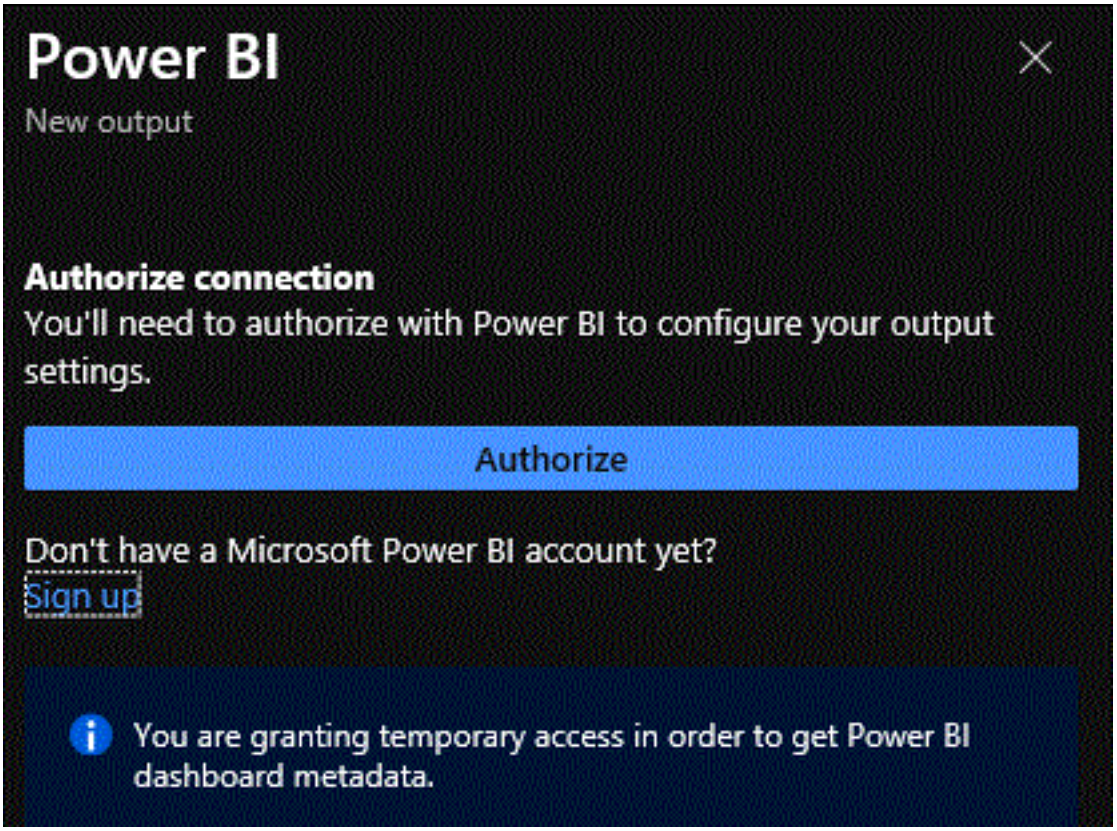
Advanced

Save Cancel

Now that we have a workspace to use as an output to Power BI, back in the Azure portal in your stream analytics job under Job Topology select Outputs. Click on + Add and select Power BI at the bottom of the list.



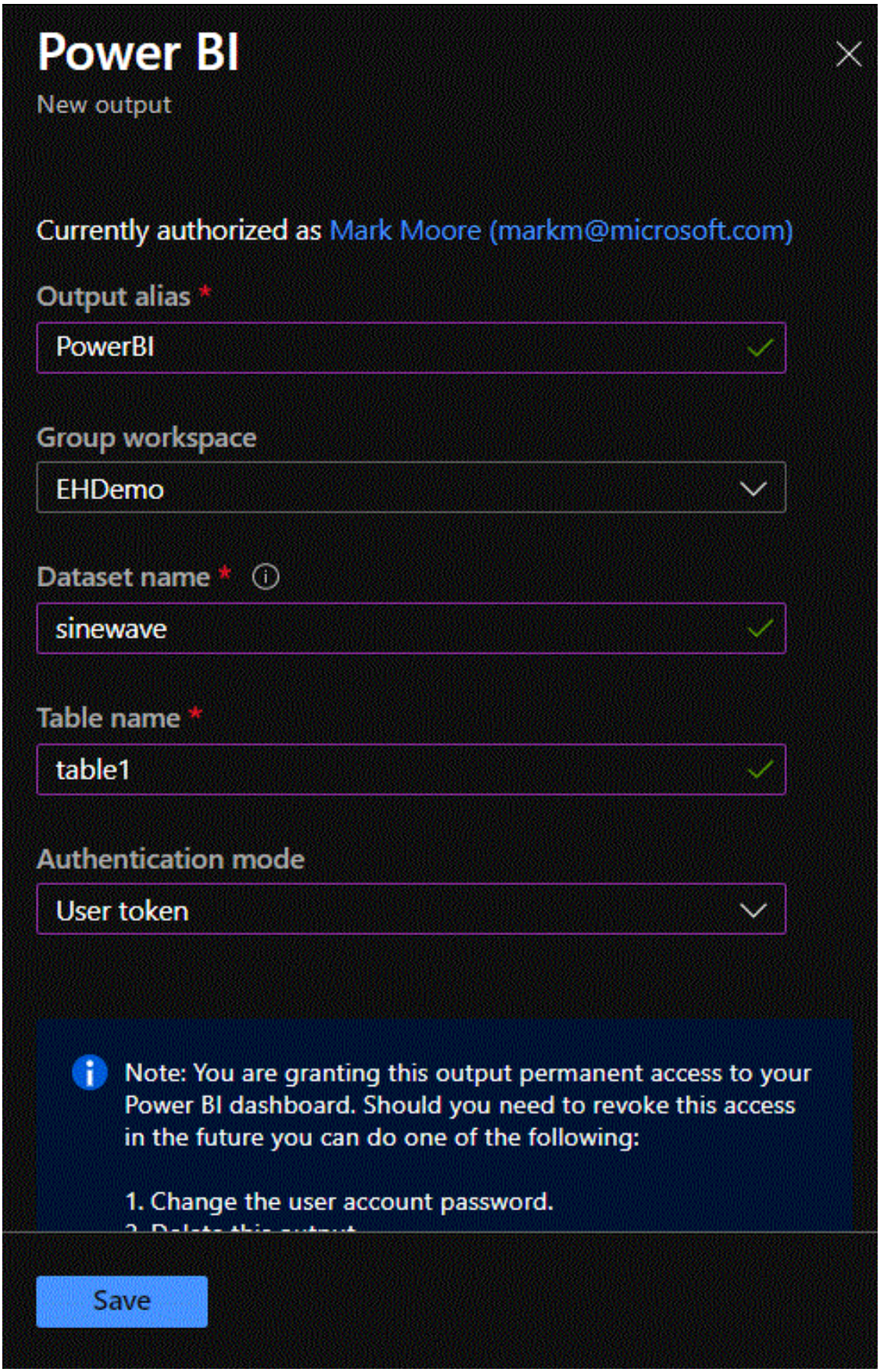
You will be prompted to Authorize, use the same credential to Authorize that you used to log into Power BI.



You will now be asked to provide an Output Alias. Make a note of the name you use you will need it for the Query.

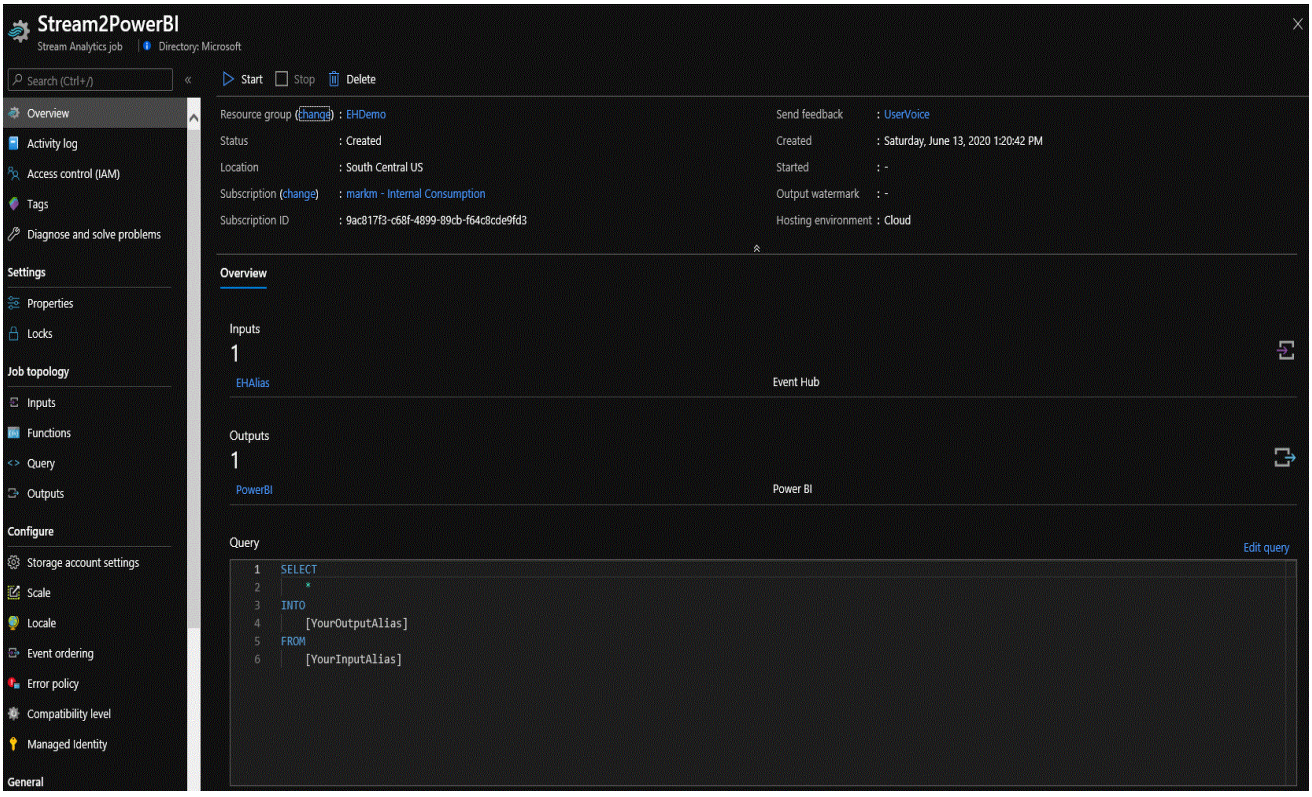
Select the newly created Workspace you just Created in Power BI.

Provide a dataset name and table name and click save.

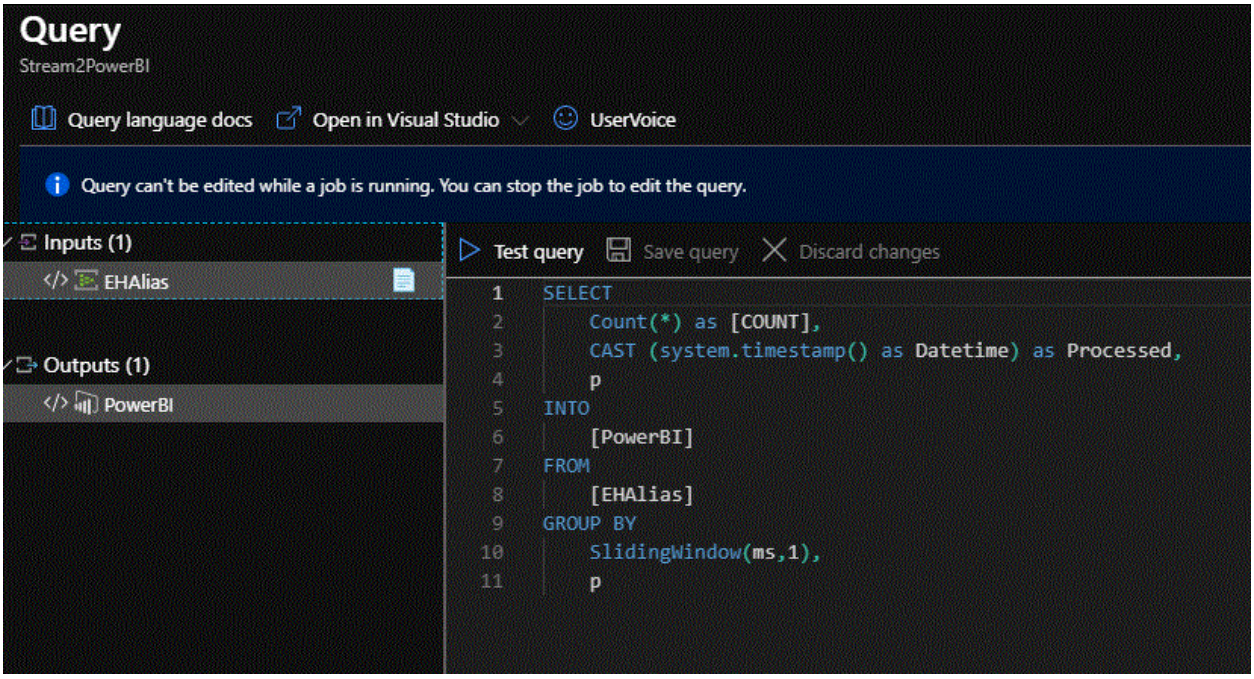

Power BI
New output
Currently authorized as Mark Moore (markm@microsoft.com)
Output alias *
PowerBI
Group workspace
EHDemo
Dataset name *
sinewave
Table name *
table1
Authentication mode
User token
Note: You are granting this output permanent access to your Power BI dashboard. Should you need to revoke this access in the future you can do one of the following:
1. Change the user account password.
Save

In the Overview section of your stream analytics Job you will see a blank query. Click Edit Query.

Select the sample query and paste over it with the contents of the sample query provided on git hub. If you chose different names for your Input and Output alias, insert them into the Stream Analytics Query replacing the ones provided in the sample query.



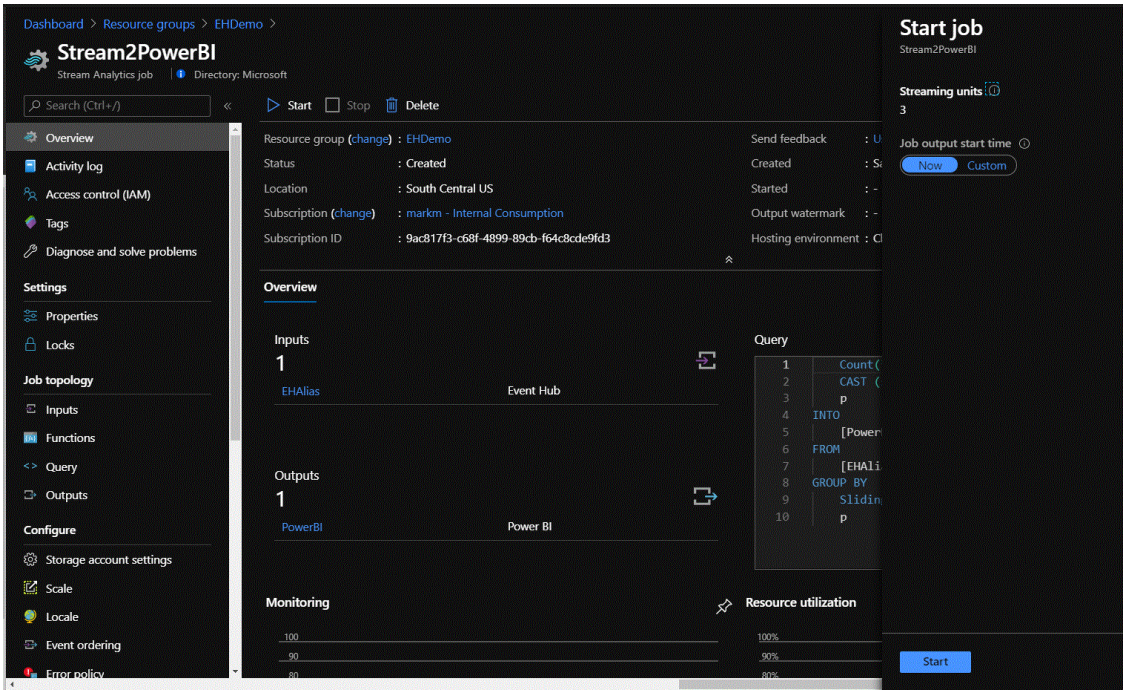
Your query should look like this:



Test the Query using the Test Query button. If the Test Query button is greyed out the syntax of your query is incorrect.

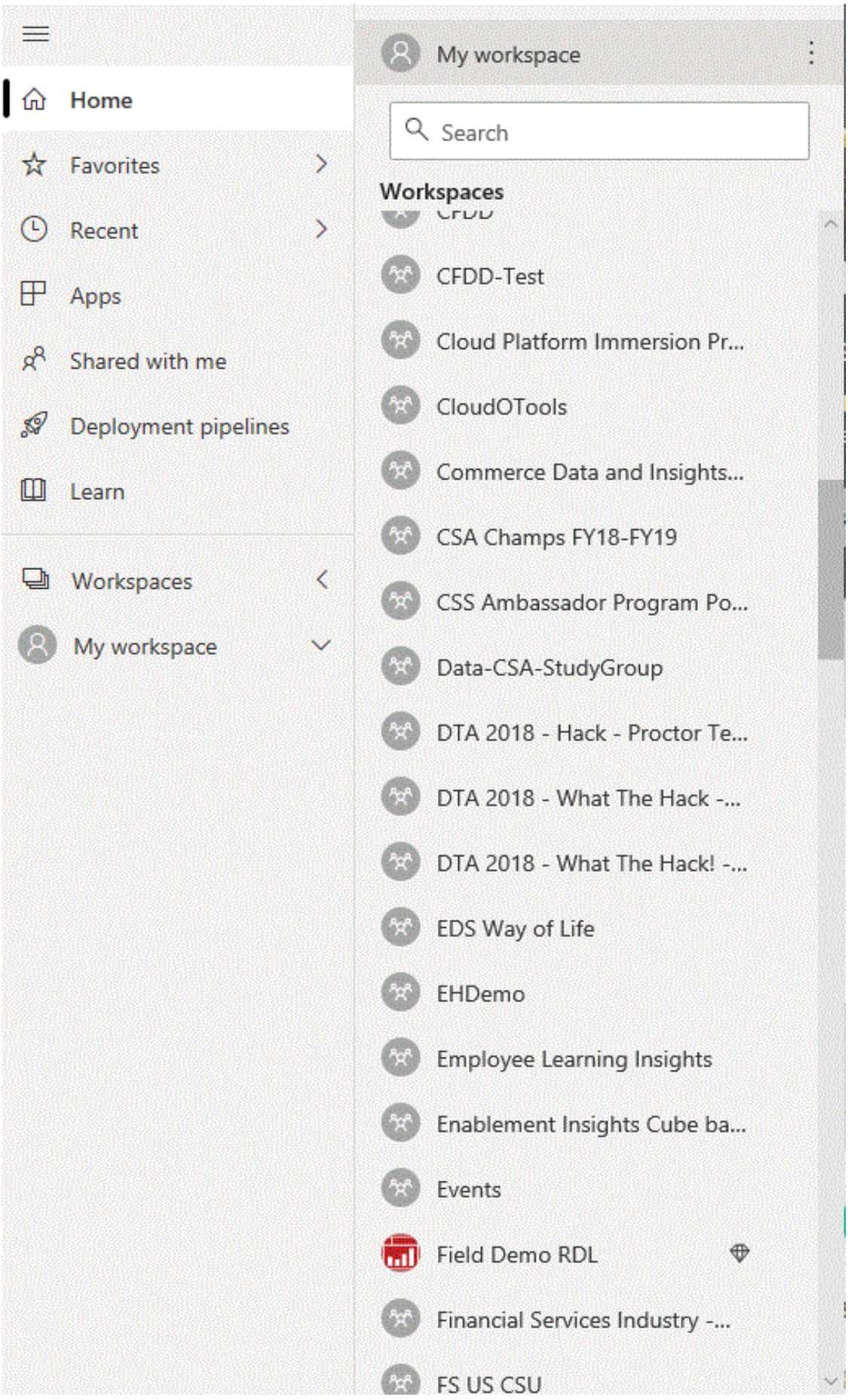
Once you have successfully tested your query save the query by clicking the Save query button and you will return to the overview page.

Click on the Start button to start your Stream Analytics Job. It may take a minute or two for the job to start.

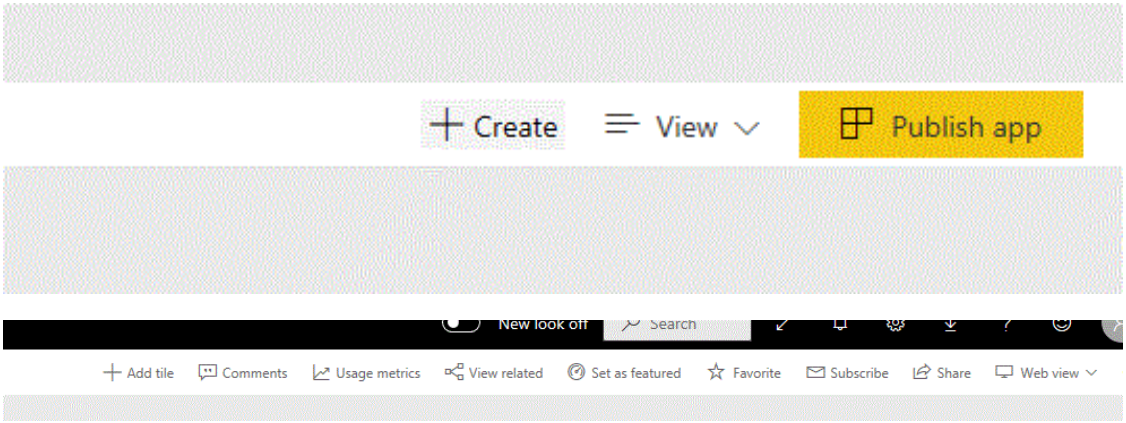


Once the stream analytics job has successfully started you are streaming data to your Power BI workspace. Now let's put together a simple Dashboard to view the output in real time.

Go back to Power BI and click on Workspace on the left-hand side of the screen. Select the workspace you previously created for this demo.


My workspace
Home
Search
Favorites
Workspaces
Recent
CFDD-Test
Apps
Cloud Platform Immersion Pr...
Shared with me
CloudOTools
Deployment pipelines
Commerce Data and Insights...
Learn
CSA Champs FY18-FY19
Workspaces
CSS Ambassador Program Po...
My workspace
Data-CSA-StudyGroup
DTA 2018 - Hack - Proctor Te...
DTA 2018 - What The Hack -...
DTA 2018 - What The Hack! -...
EDS Way of Life
EHDemo
Employee Learning Insights
Enablement Insights Cube ba...
Events
Field Demo RDL
Financial Services Industry -...
FS US CSU

Once you have selected your workspace click on Create.



In the next screen under Real-Time Data select Custom Streaming Data.

Add tile

Select source

MEDIA

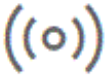
Web content

Image

Text box

Video

REAL-TIME DATA



Custom Streaming Data

Next

Cancel

Click Next.

Choose the dataset you created while setting up the Output in the stream analytics job.

Add a custom streaming data tile

Choose a streaming dataset

+ Add streaming dataset

YOUR DATASETS

sinewave

Manage datasets

Back | Next | Cancel

Select Line Chart and for the Axis select processed. This is the time axis and we are pulling system time from stream analytics in real-time as the query process each point. Under values select P.

Add a custom streaming data tile

Choose a streaming dataset > Visualization design

Visualization Type

Line chart

Axis

Processed

+ Add value

Legend

+ Add value

Values

p

+ Add value

Time window to display

Last 1 Minutes

Manage datasets

Back Next Cancel

You should see a preview of your sine wave being updated in real time. Click next and your demo is complete.