05/03/2019

Real-Time Data Ingestion with Azure Stream Analytics

Roy, Sandip (Cognizant)
COGNIZANT

Table of Contents

1. Introduction

This document outlines how to ingest real-time streaming data (structured or semi structured) using Azure Stream Analytics and store them in persistent storage (e.g. Azure SQL DB) for any future analytical purposes.

2. Solution Components

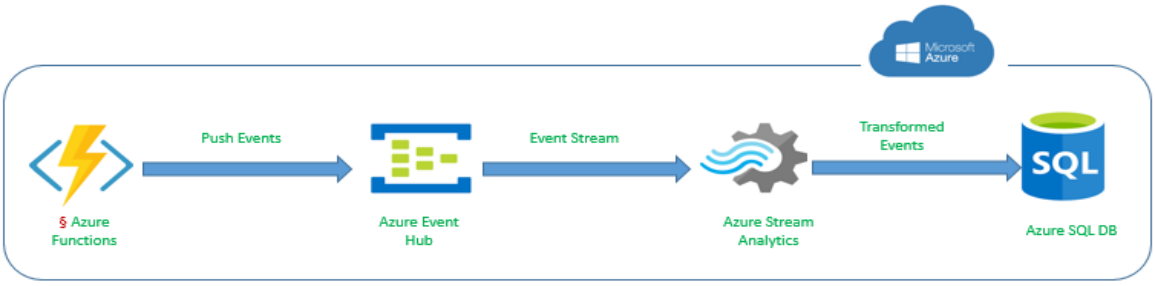
Azure Event Hubs is a big data streaming platform and event ingestion service. It can receive and process millions of events per second. Data sent to an event hub can be transformed and stored by using any real-time analytics provider or batching/storage adapters.

Azure Stream Analytics is an event-processing engine that allows you to examine high volumes of data streaming from devices. Incoming data can be from devices, sensors, web sites, social media feeds, applications, and more. It also supports extracting information from data streams, identifying patterns, and relationships.

Azure SQL Database is a managed cloud database provided as part of Microsoft Azure. A cloud database is a database that runs on a cloud computing platform, and access to it is provided as a service. Managed database services take care of scalability, backup, and high availability of the database.

Azure Function App is a solution for easily running small pieces of code, or "functions," in the cloud. You can write just the code you need for the problem at hand, without worrying about a whole application or the infrastructure to run it.

3. Solution



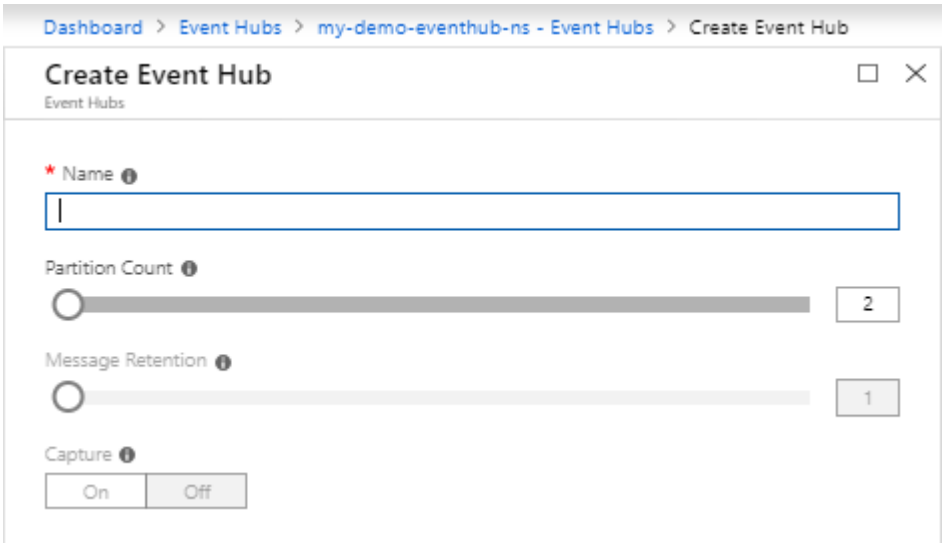
§ Used in this demo for simulation only. In practical cases, we can have different producer pushing events into Event Hub

4. Detailed Steps

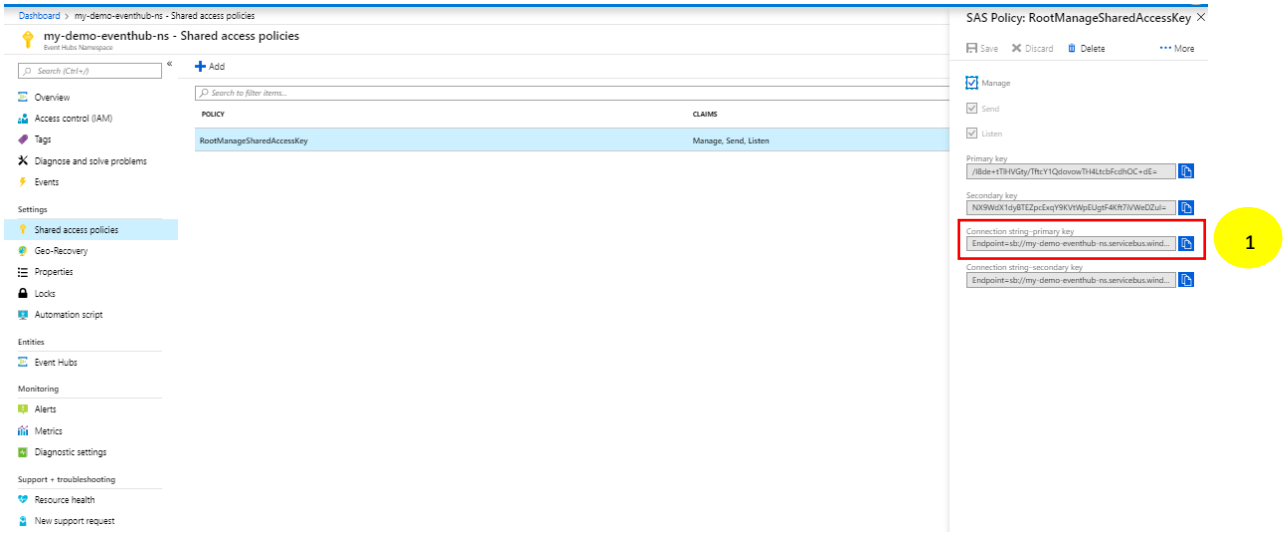
Step 1 – Create an Azure Event Hub.

Firstly, create Namespace/Service Bus component under which your Event Hub will reside.

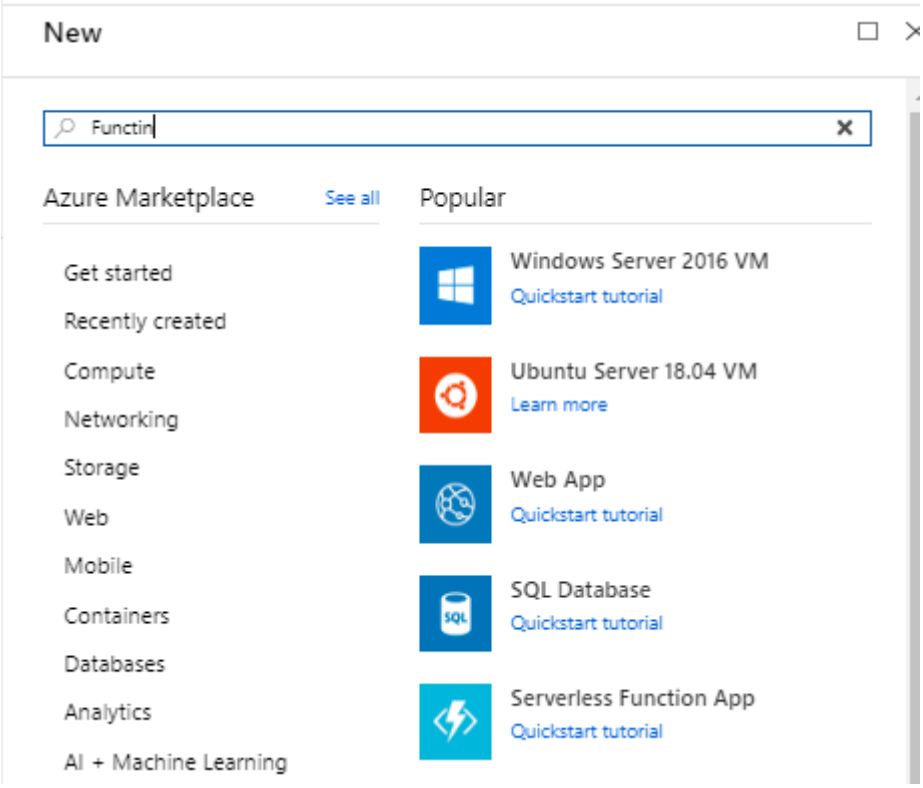
Now create Event Hub under Namespace created above.

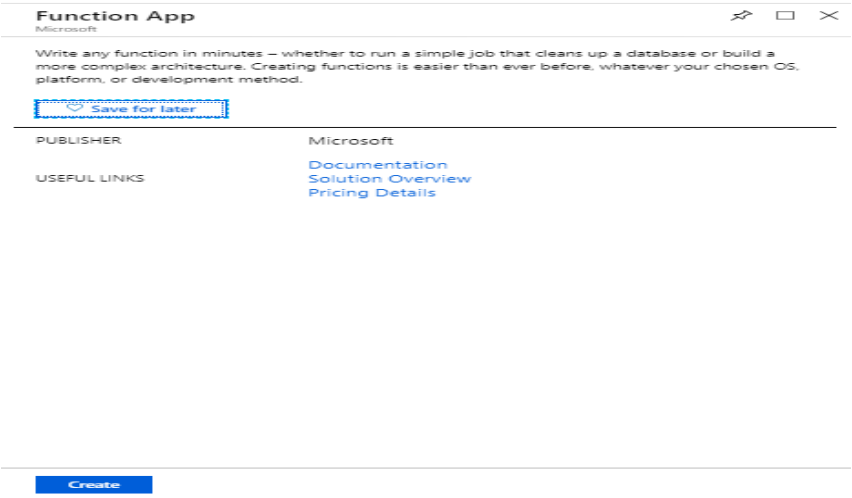


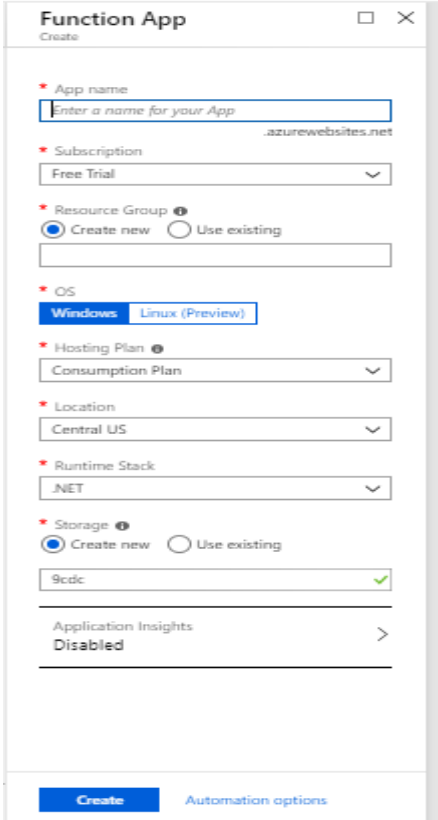
Once Event Hub is created go to parent namespace and check Shares access policies under Settings to take a note of the SAS (Shared Access Signature) policy of the namespace – this is required in order to access the namespace and its decedent resources (Event Hub) with required permissions. Make sure you have send and listen permission on to this as per requirement.

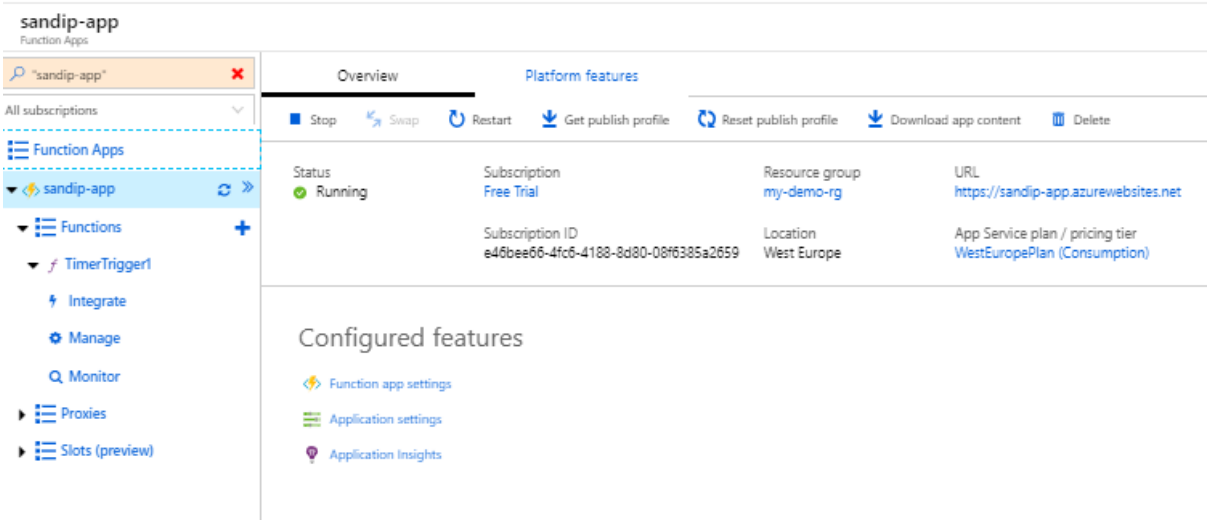


Step 2 – Create an Azure Function (Function App) which would send JSON stream of event payload into Azure Event Hub in periodic manner (let say every 2 minutes or so).

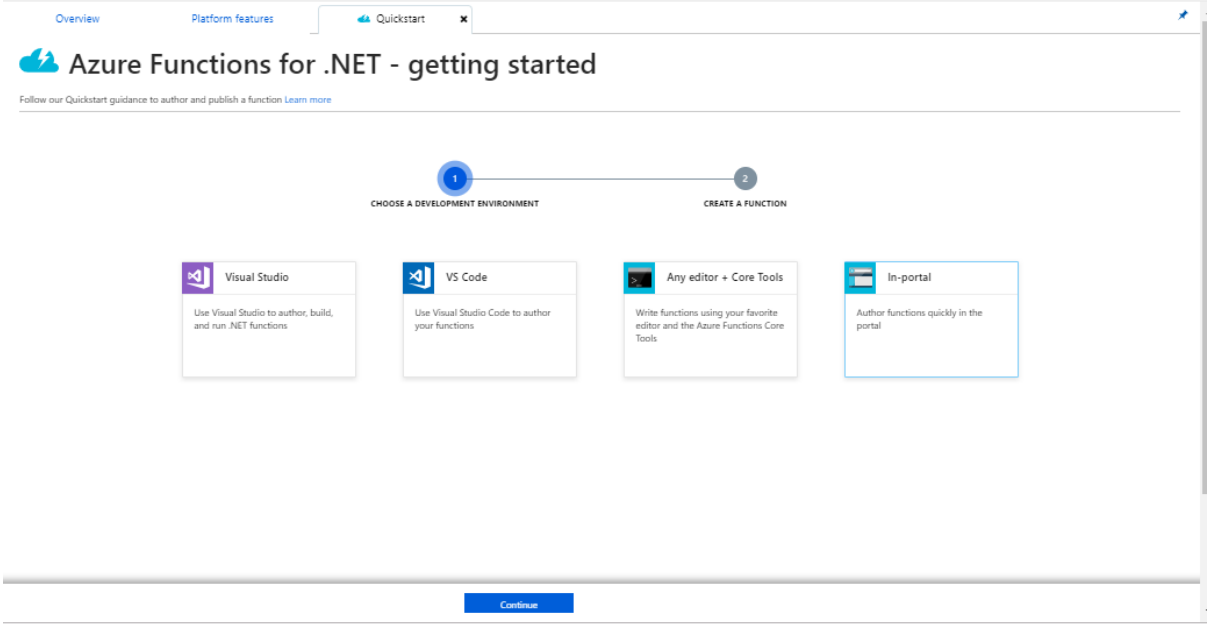



Function App
Microsoft
Write any function in minutes – whether to run a simple job that cleans up a database or build a more complex architecture. Creating functions is easier than ever before, whatever your chosen OS, platform, or development method.
Save for later
PUBLISHER
Microsoft
USEFUL LINKS
Documentation
Solution Overview
Pricing Details
Create

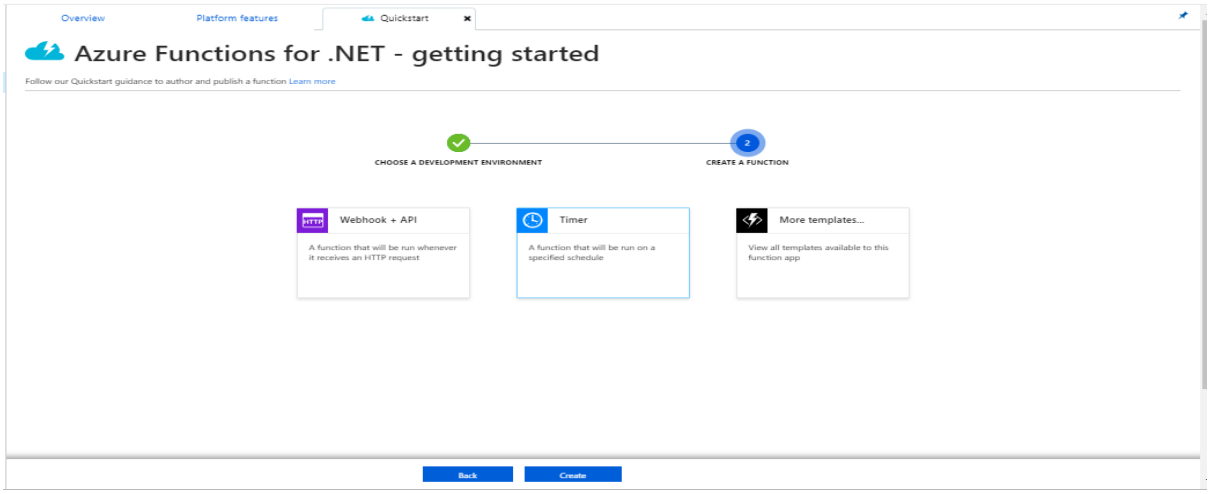

Function App
Create
* App name
Enter a name for your App
.azurewebsites.net
* Subscription
Free Trial
* Resource Group
Create new
Use existing
* OS
Windows
Linux (Preview)
* Hosting Plan
Consumption Plan
* Location
Central US
* Runtime Stack
.NET
* Storage
Create new
Use existing
9cdc
Application Insights
Disabled
Create
Automation options



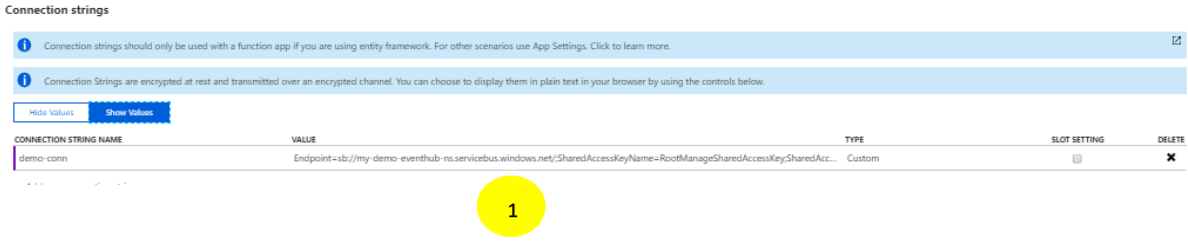
Select the in-portal option to develop the simple C# (you can Node JS or Java script method also) method.



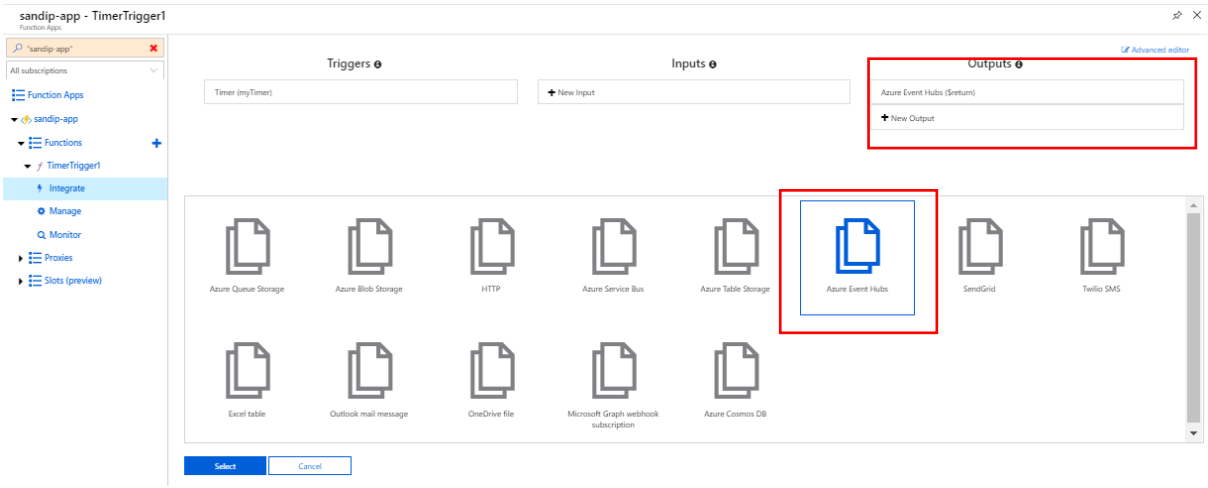
Select the required template for the method. We will go for Timer option in case as we would like to get our method executed in certain duration automatically so that it works as simulator for producing real-time data into Event Hub.

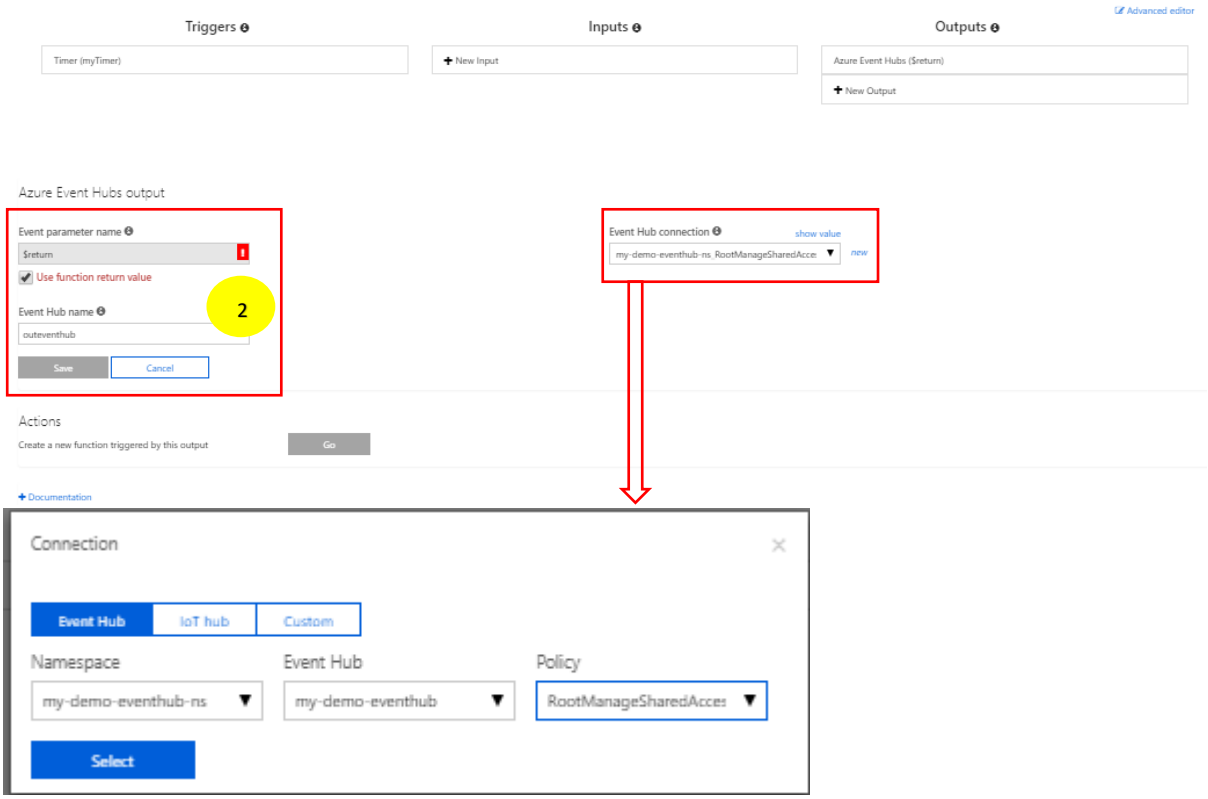


Once the Function is created (it is created with basic Function body), we will configure the Event Hub endpoint (i.e. SAS Primary Key that we highlighted above) as connection string in Azure Functions.

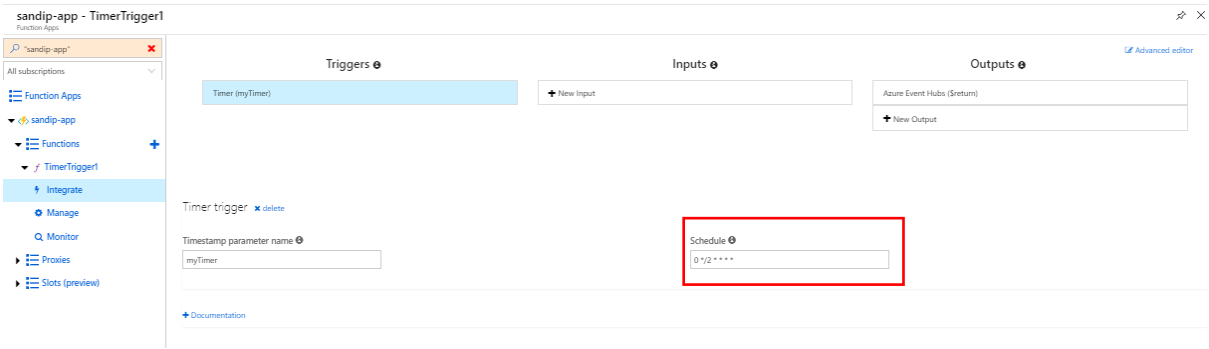


Now go to Integrate option and select the input/output option for this Function App. We will add Azure Event Hub as outputs. Make sure you use return value from Azure Function as Event Hub input (yellow circled as 2).

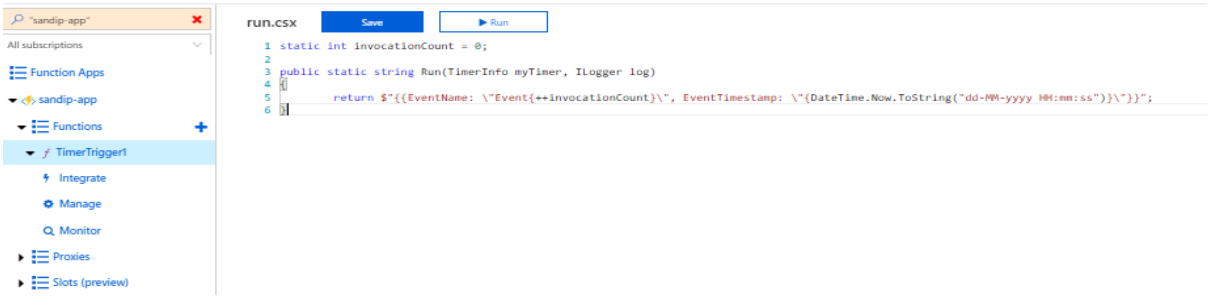




Update the execution schedule for this timer based Function App. E.g. Here it is chosen 2 minutes' interval using CRON expression.

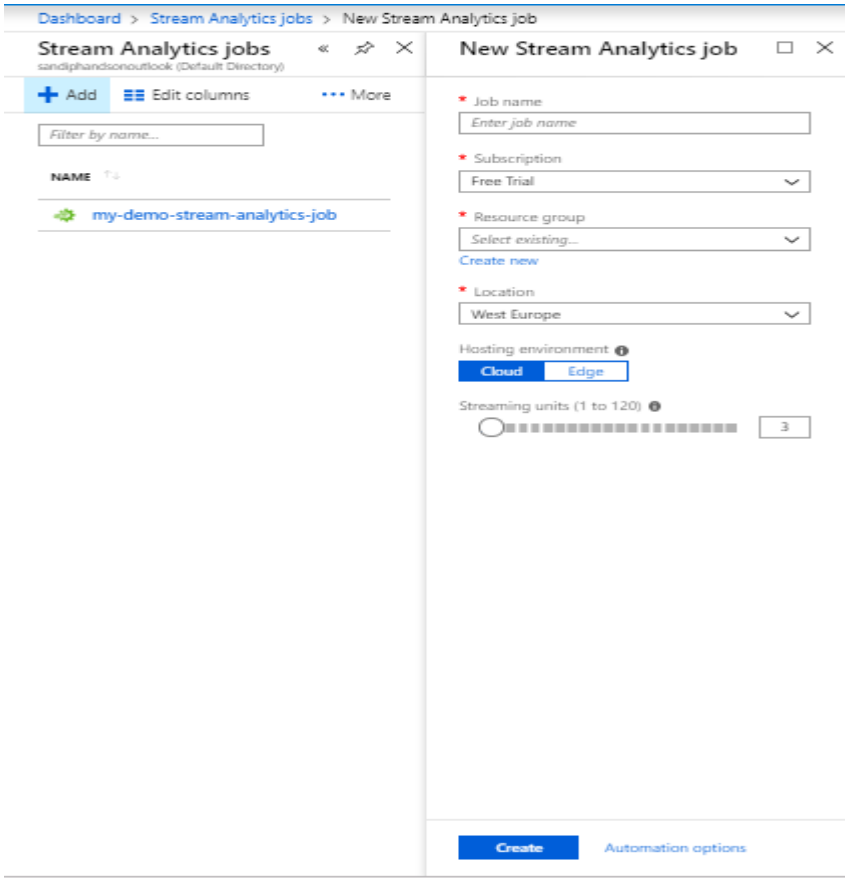


Finally update the Azure Function to implement your code as per requirement. E.g. Below is simple code to return hard-coded JSON string which will be eventually sent to Event Hub as Events.

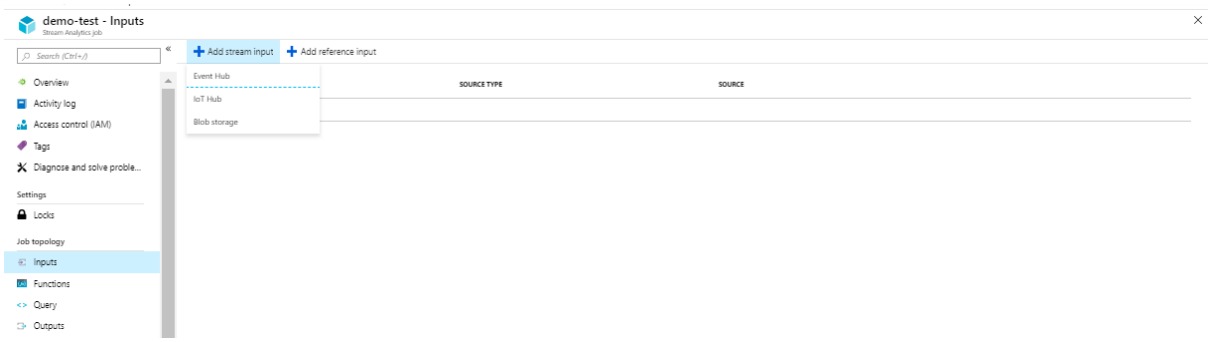


Sample message: {"EventName":"EVENT1","EventTimestamp":"18-02-2019 12:38:00"}

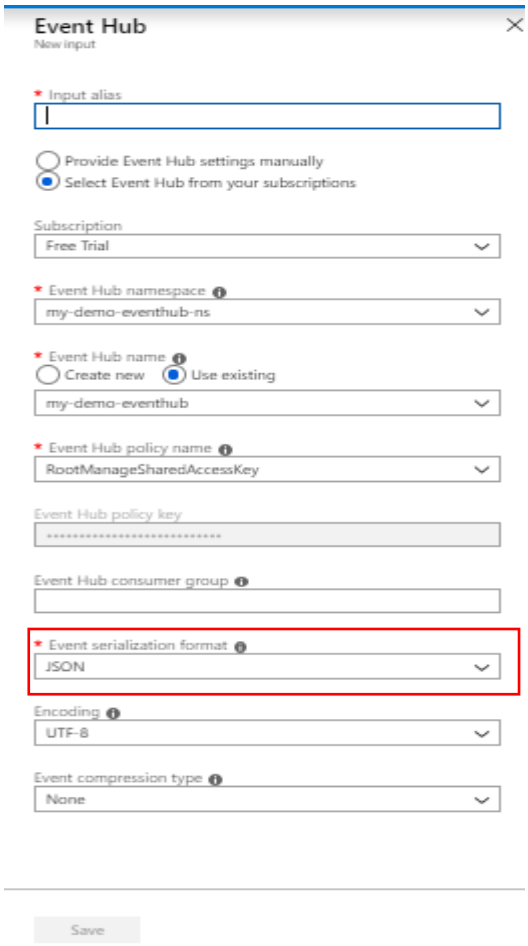
Step 3 – Create the Stream Analytics Job.

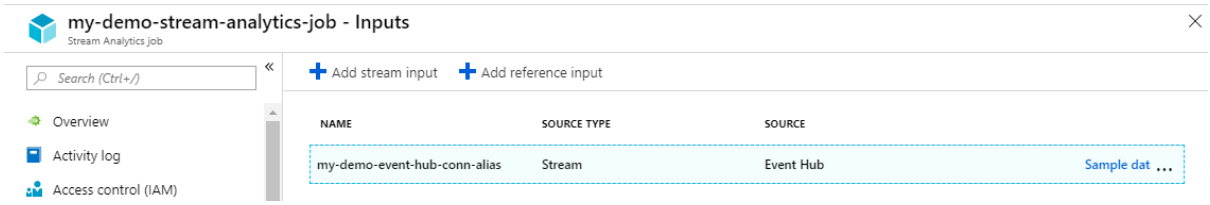


Once the Stream Analytics job is created, add Event Hub as stream input.

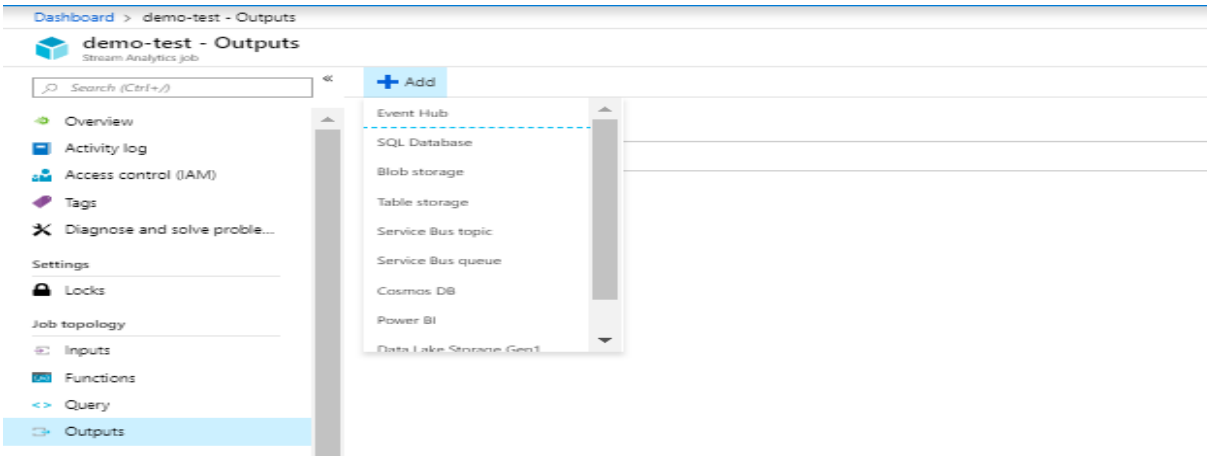


Make sure you choose Event message serialization format correctly (please note Stream Analytics job supports seriation format like JSON, Avro, CSV etc.) otherwise we would get error retrieving the message from Event Hub.

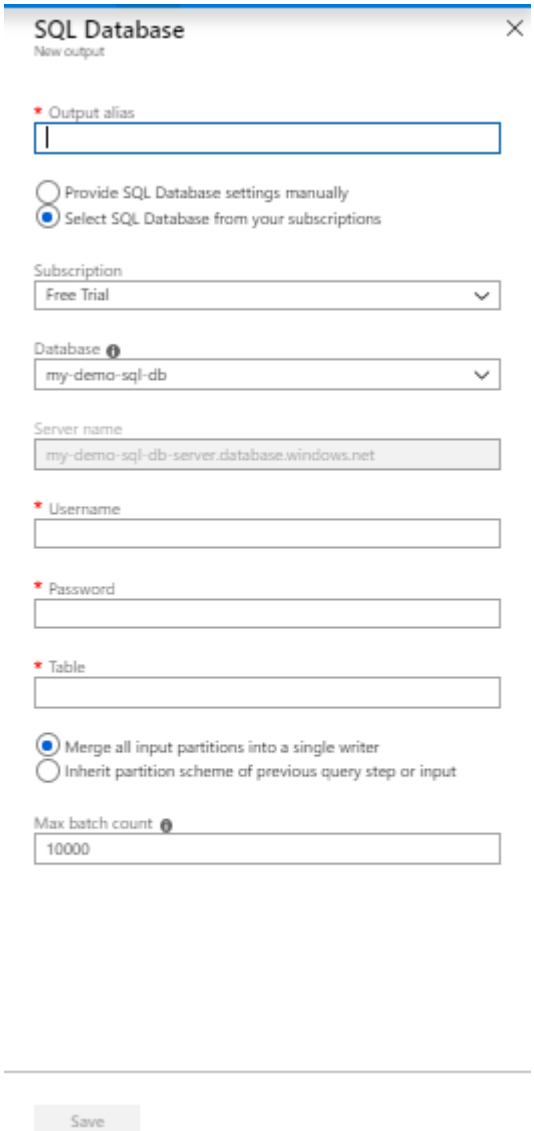


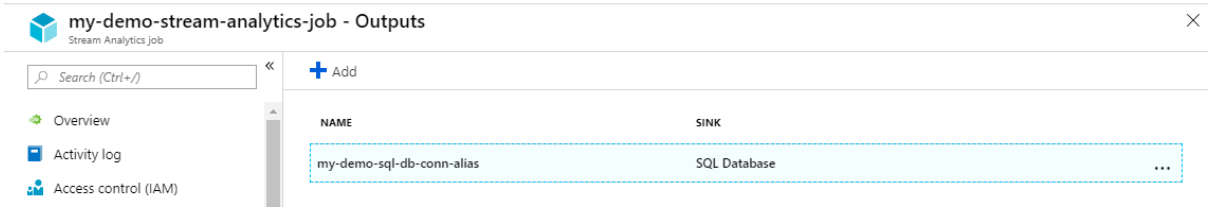


Add outputs to Stream Analytics job. In our case we would have Azure SQL DB as target.

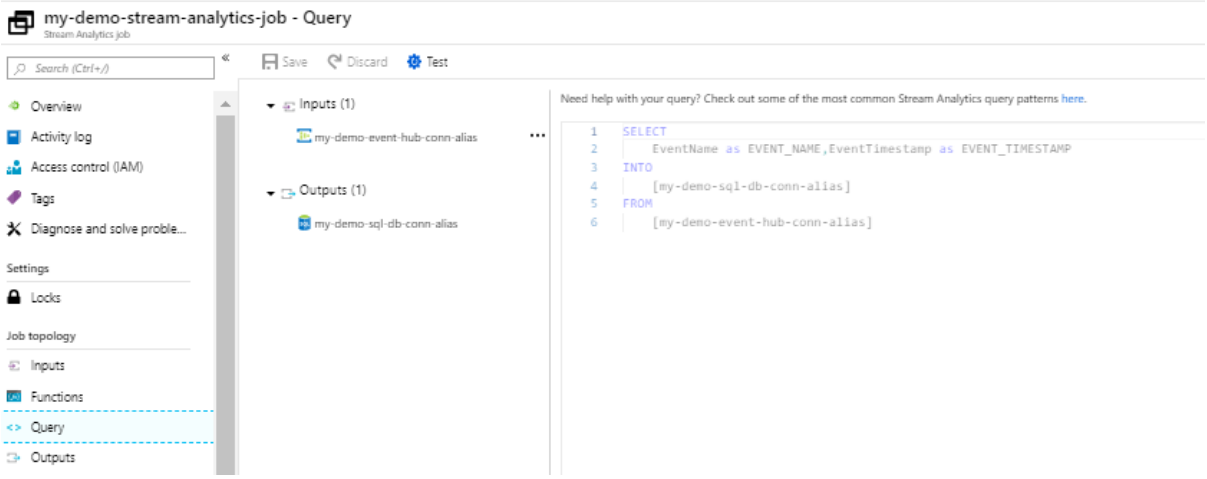


Specify the DB details (database, credentials, target table etc.)

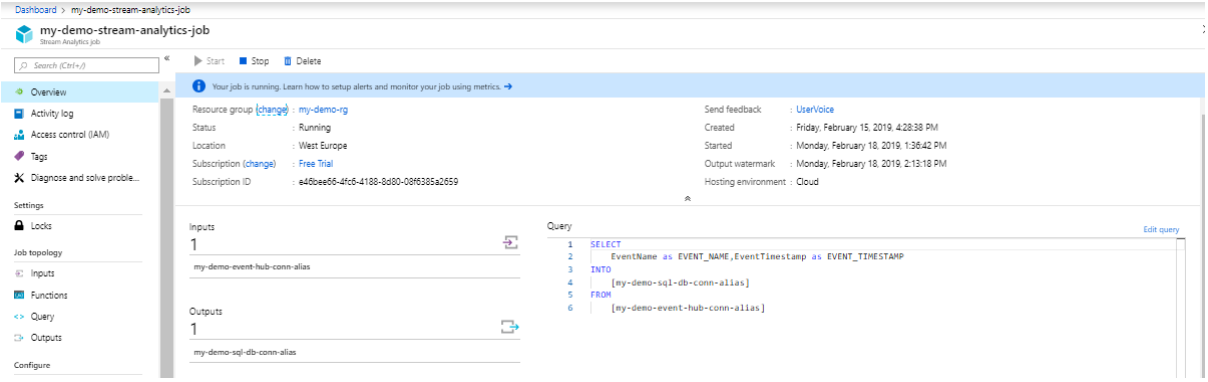




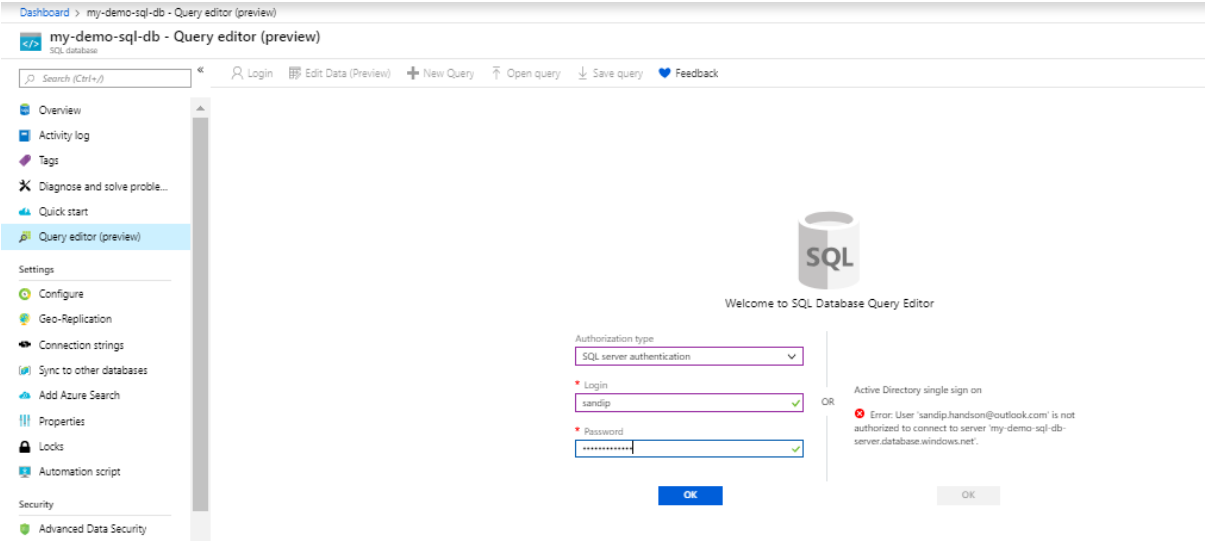
Once both input and outputs are defined for Stream Analytics job, it's time to write the query (actual business logic/mapping/transformation to populate target – to insert records into SQL DB). Please note that Azure Stream Analytics offers a SQL-like query language for performing transformations and computations over streams of events.

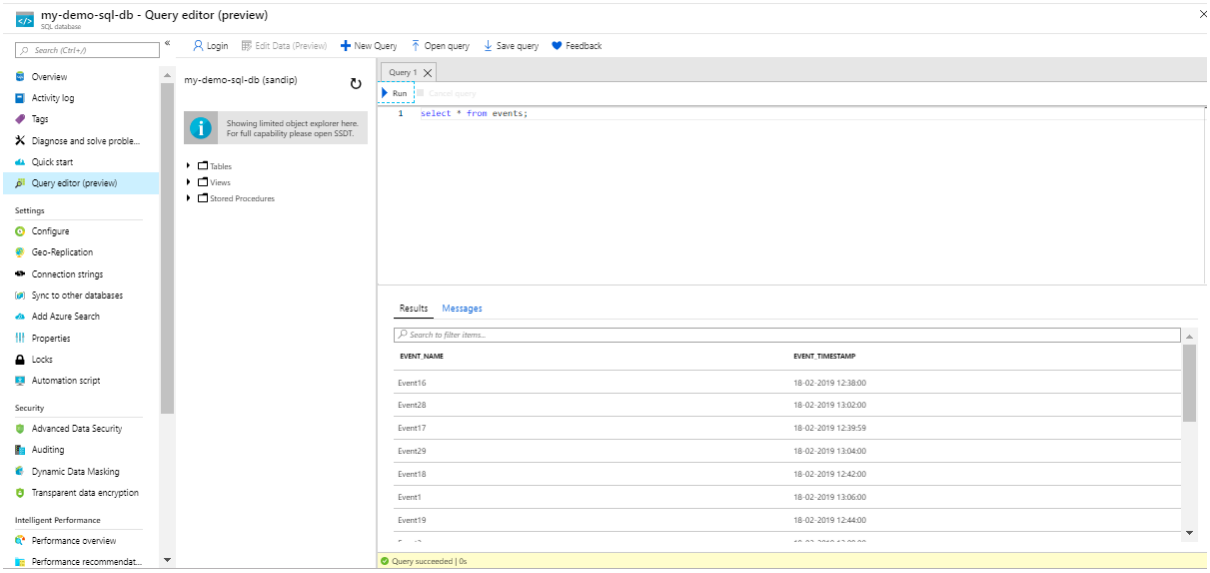


Once the job is ready, just start it.



Step 4 – Go to SQL DB and notice that the streaming data is getting logged into target table.





5. References

https://docs.microsoft.com/en-us/azure/azure-functions/functions-create-scheduled-function

https://docs.microsoft.com/en-us/azure/event-hubs/event-hubs-create

https://docs.microsoft.com/en-us/azure/stream-analytics/stream-analytics-quick-create-portal

https://docs.microsoft.com/en-us/azure/sql-database/sql-database-single-database-get-started