**Analyze fraudulent call data with Stream Analytics and visualize results in Power BI dashboard**

Tutorial link- <https://docs.microsoft.com/en-us/azure/stream-analytics/stream-analytics-real-time-fraud-detection>

Contents

[1 Tasks 2](#_Toc66797506)

[2 Prerequisites 2](#_Toc66797507)

[3 Create an Azure Event Hub 2](#_Toc66797508)

[4 Grant access to the event hub and get a connection string 4](#_Toc66797509)

[5 Start the event generator application 6](#_Toc66797510)

[6 Create a Stream Analytics job 8](#_Toc66797511)

[7 Configure job input 9](#_Toc66797512)

[8 Configure job output 11](#_Toc66797513)

[9 Create queries to transform real-time data 12](#_Toc66797514)

[10 Start the job and visualize output 15](#_Toc66797515)

[11 Embedding your Power BI Dashboard in a Web Application 19](#_Toc66797516)

# Tasks

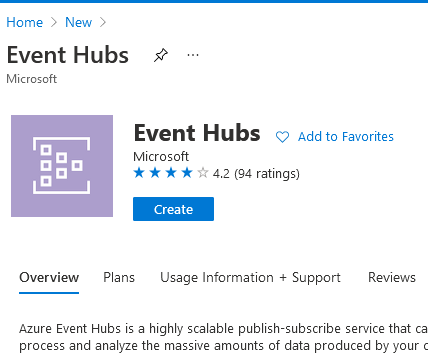
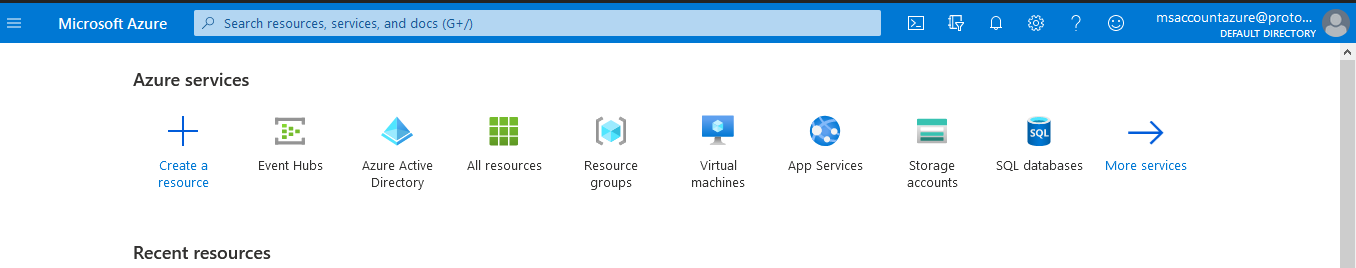
* Generate sample phone call data and send it to Azure Event Hubs.
* Create a Stream Analytics job.
* Configure job input and output.
* Define queries to filter fraudulent calls.
* Test and start the job.
* Visualize results in Power BI.

# Prerequisites

1. Create Azure account (<https://azure.microsoft.com/free/>). It requires credit card information when registering for an account.
2. Download the phone call event generator app [TelcoGenerator.zip](https://download.microsoft.com/download/8/B/D/8BD50991-8D54-4F59-AB83-3354B69C8A7E/TelcoGenerator.zip) from the Microsoft. Source code - [GitHub](https://aka.ms/azure-stream-analytics-telcogenerator).
3. Create Power BI account (<https://powerbi.microsoft.com/en-us/landing/signin/>) using work email address. **Personal emails like- xyz@gmail.com do not work**.

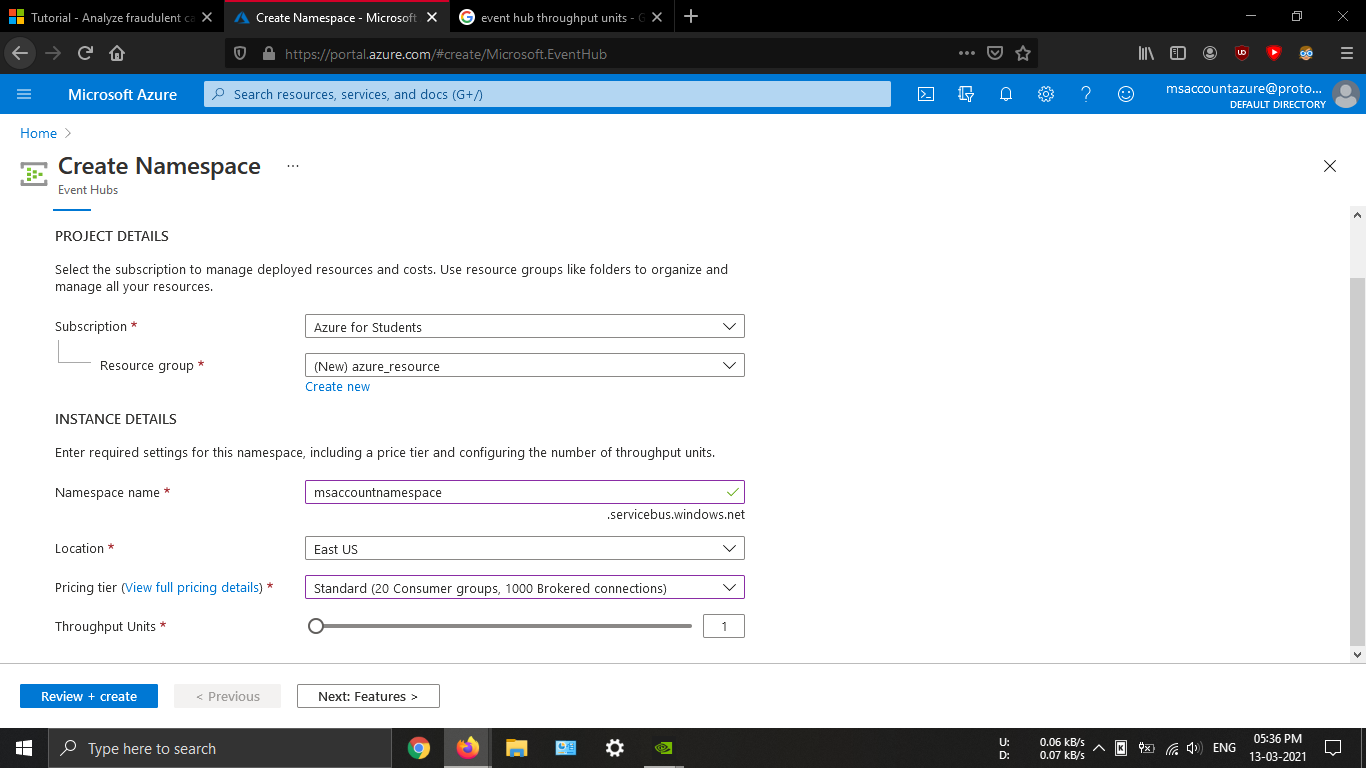
# Create an Azure Event Hub

1. Sign in to [azure](https://portal.azure.com/) after creating an account.
2. Click on **+ create a resource**. Search for “Event Hubs” and click “create”.

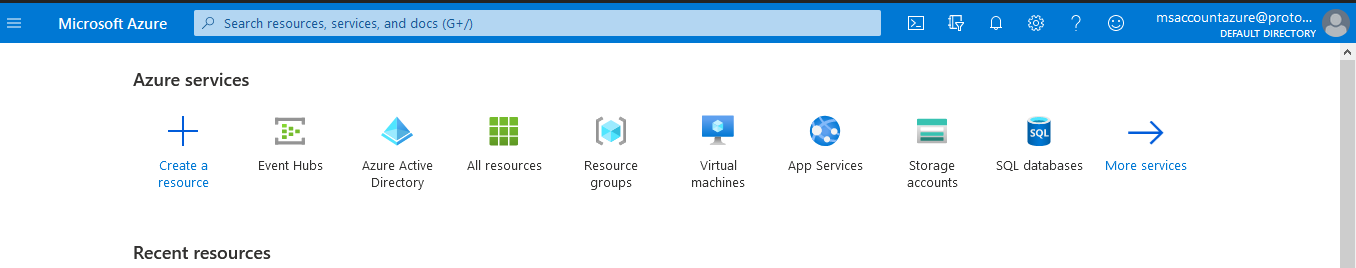


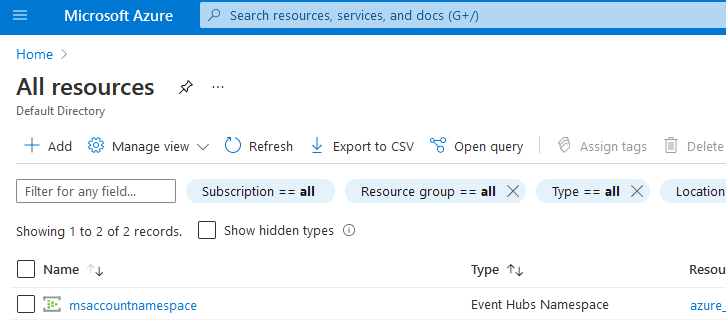
1. Fill out the Create Namespace pane with required details:

* Name: A unique name to identify the event hub namespace.
* Subscription: Select an Azure subscription where you want to create the event hub.
* Resource group: Select **Create New** and enter a new resource-group name for your account.
* Location: Location where the event hub namespace can be deployed.

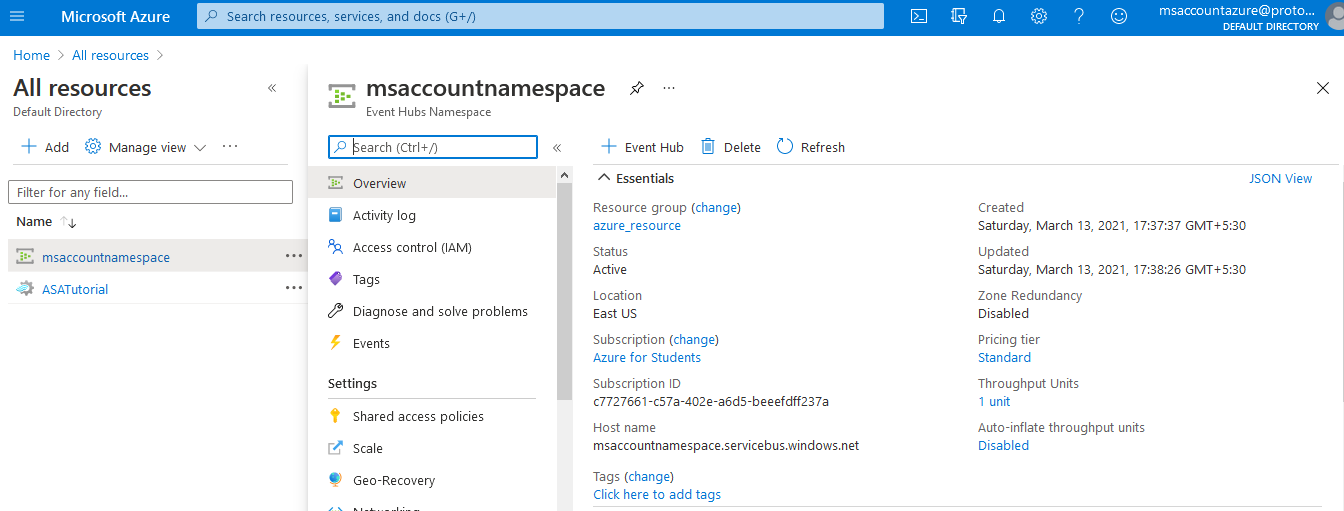


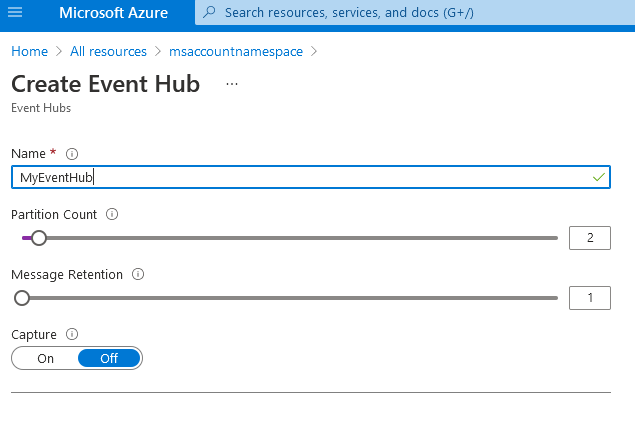
1. When the namespace has finished deploying, go to **All resources** and find your namespace in the list of Azure resources. Select <namespace> (here, msaccountnamespace) to open it.





1. Next select **+Event Hub** and enter a **Name** for the Event Hub. Set the **Partition Count** to 2. Use the default options in the remaining settings and select **Create**. Then wait for the deployment to succeed.

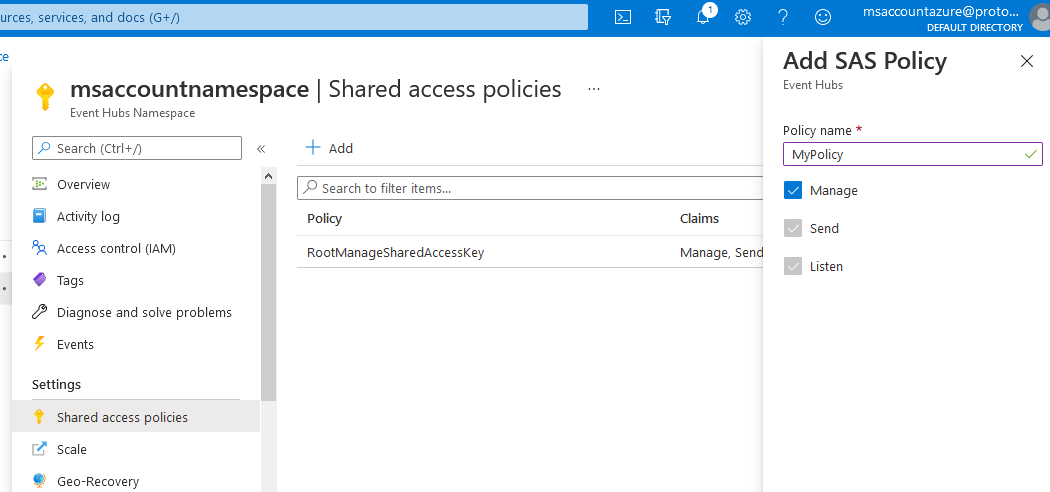


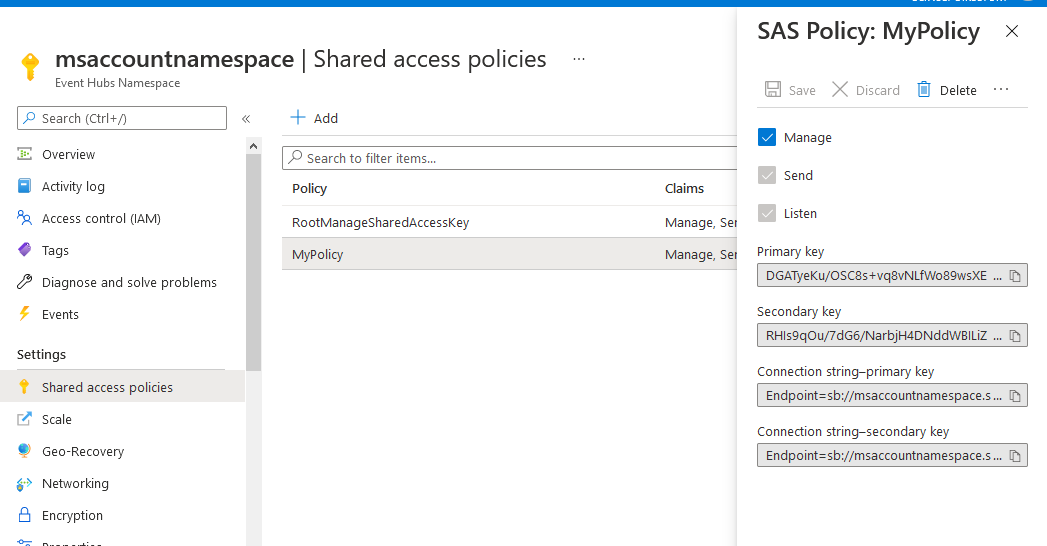


# Grant access to the event hub and get a connection string

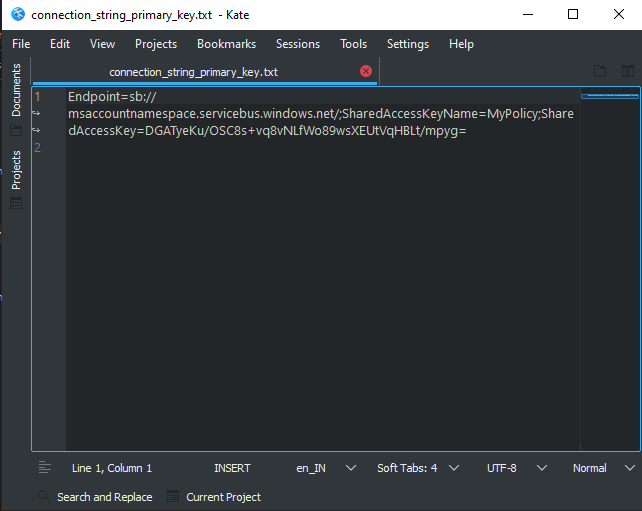
Before an application can send data to Azure Event Hubs, the event hub must have a policy that allows access. The access policy produces a connection string that includes authorization information.

1. Navigate to the event hub you created in the previous step, “MyEventHub”. Select Shared access policies under Settings, and then select **+ Add**.
2. Name the policy “MyPolicy” and ensure “Manage” is checked. Then select **Create**.





Here, **we will save “Connection string–primary key” in a text file** for later use.

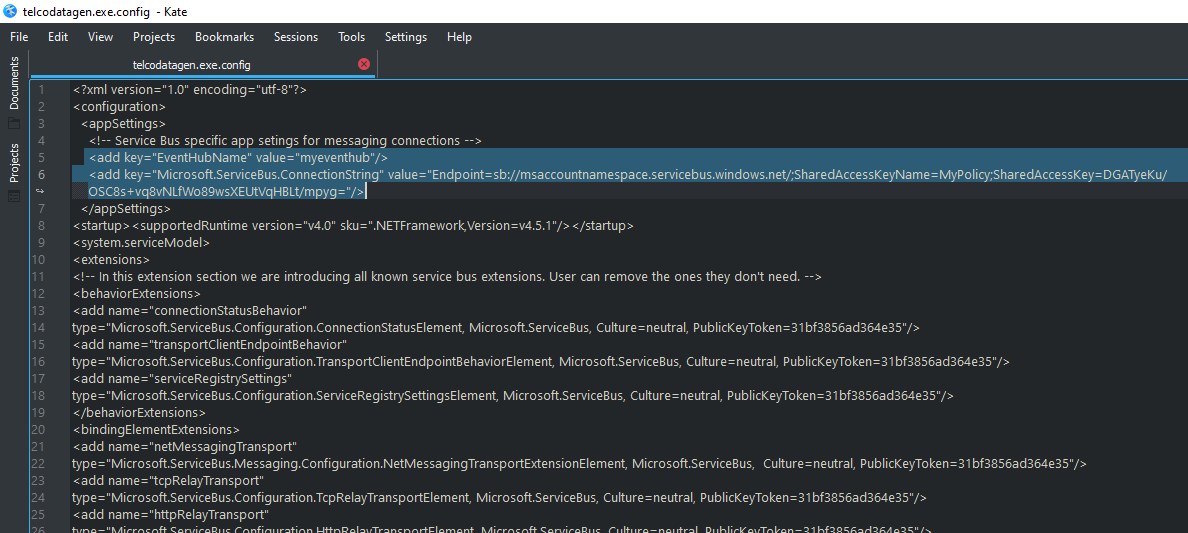


# Start the event generator application

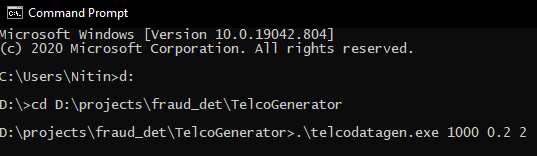
Before you start the TelcoGenerator app, you should configure it to send data to the Azure Event Hubs you created earlier.

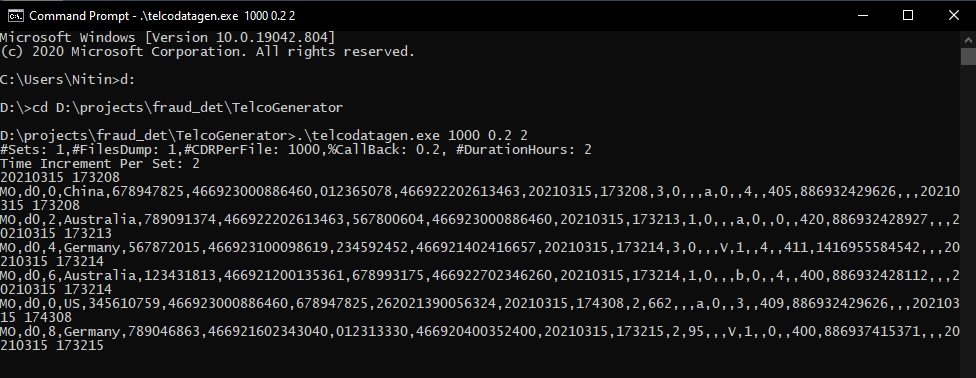
1. Extract the contents of [TelcoGenerator.zip](https://download.microsoft.com/download/8/B/D/8BD50991-8D54-4F59-AB83-3354B69C8A7E/TelcoGenerator.zip) file.
2. Open the TelcoGenerator\TelcoGenerator\telcodatagen.exe.config file in a text editor of your choice There is more than one .config file, so be sure that you open the correct one.
3. Update the <appSettings> element in the config file with the following details:

* Set the value of the EventHubName key as whatever event hub name was defined (“MyEventHub” in this case).
* Set the value of the Microsoft.ServiceBus.ConnectionString key from the text file saved earlier.



* Save the file.
* Next open a command window and change to the folder where you unzipped the TelcoGenerator application. Then enter the following command in extracted folder:





If you get any error when streaming data, make sure eventhub name and string is correct in the config file.

The command provided above takes the following parameters:

* Number of call data records per hour.
* Percentage of fraud probability, which is how often the app should simulate a fraudulent call. The value 0.2 means that about 20% of the call records will look fraudulent.
* Duration in hours, which is the number of hours that the app should run. You can also stop the app at any time by ending the process (Ctrl+C) at the command line.

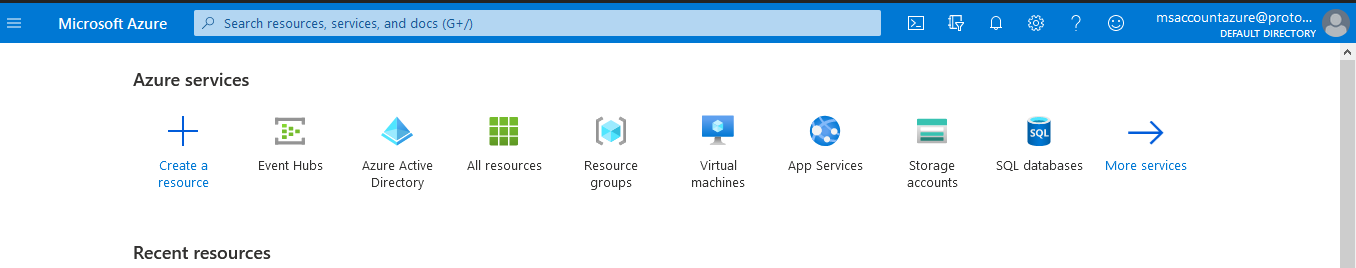
After a few seconds, the app starts displaying phone call records on the screen as it sends them to the event hub. The phone call data contains the following fields:

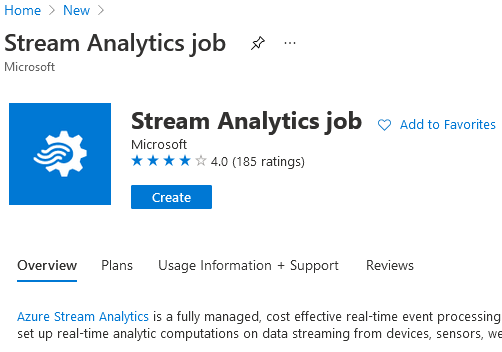
* CallrecTime: The timestamp for the call start time.
* SwitchNum: The telephone switch used to connect the call. For this example, the switches are strings that represent the country/region of origin (US, China, UK, Germany, or Australia).
* CallingNum: The phone number of the caller.
* CallingIMSI: The International Mobile Subscriber Identity (IMSI). It's a unique identifier of the caller.
* CalledNum: The phone number of the call recipient.
* CalledIMSI: International Mobile Subscriber Identity (IMSI). It's a unique identifier of the call recipient.

# Create a Stream Analytics job

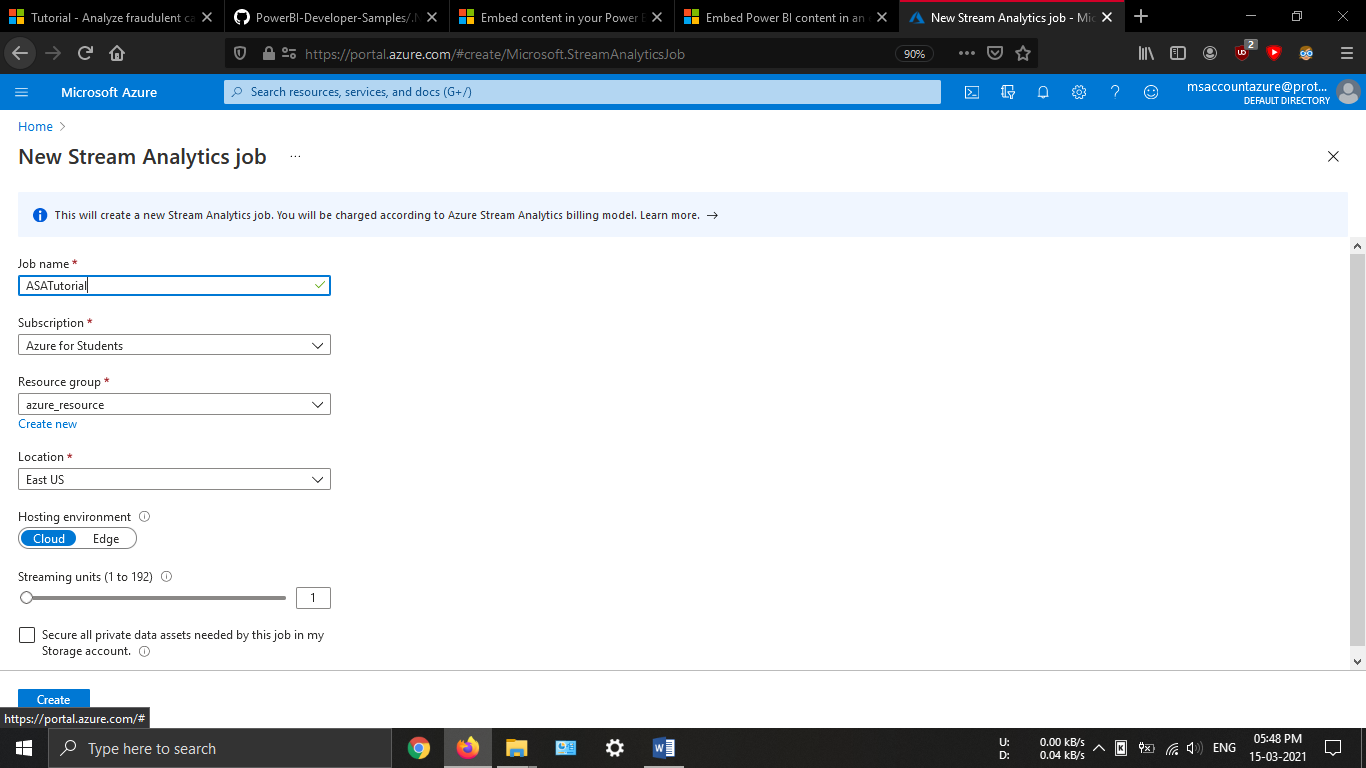
Now that you have a stream of call events, you can create a Stream Analytics job that reads data from the event hub.

1. Go to azure. Select **+** **Create a resource** and search for **Stream Analytics job**. Select the Stream Analytics job tile and select **Create**.





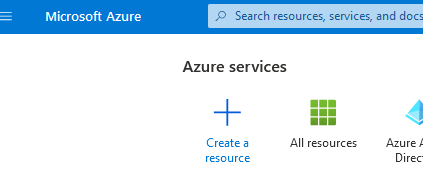
1. Fill out the New Stream Analytics job form.

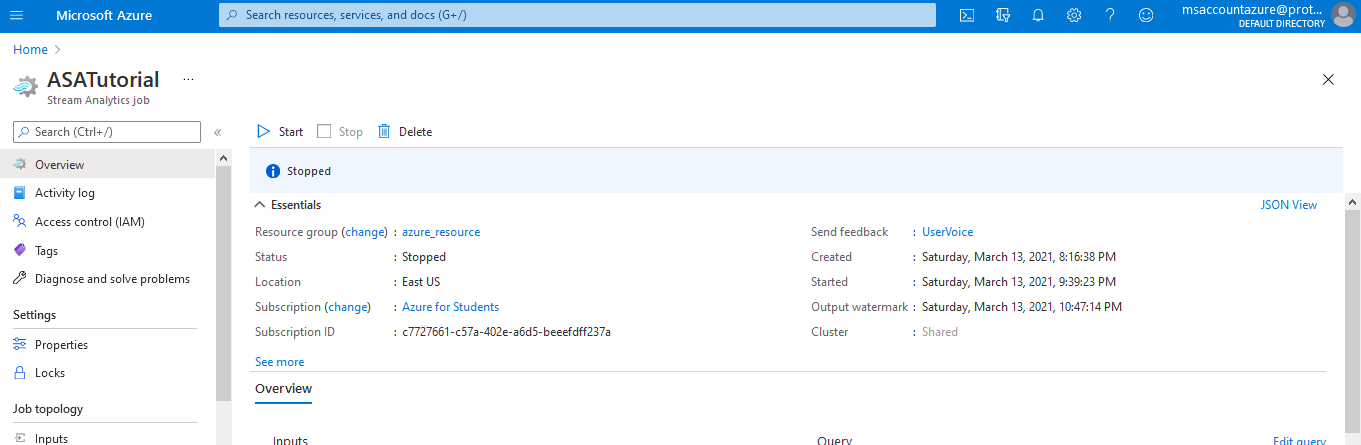


# Configure job input

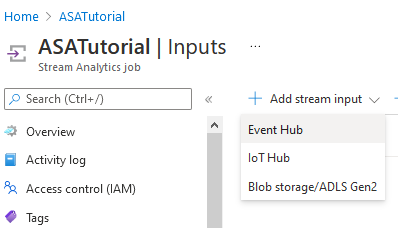
The next step is to define an input source for the job to read data using the event hub you created in the previous section.

1. From the Azure portal, open the **All resources** page, and find the **ASATutorial Stream Analytics job.**

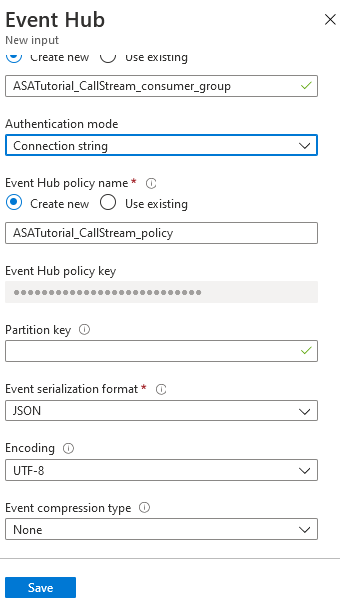
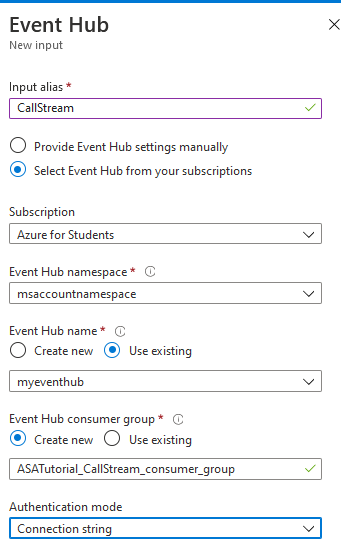




1. In the **Job Topology** section of the **Stream Analytics job**, select **Inputs**.
2. Select **+ Add stream input** and **Event hub**.



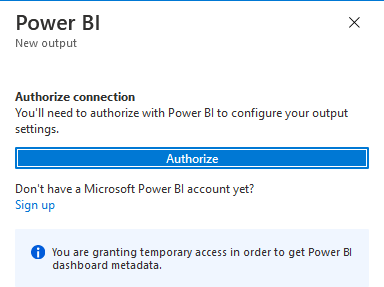
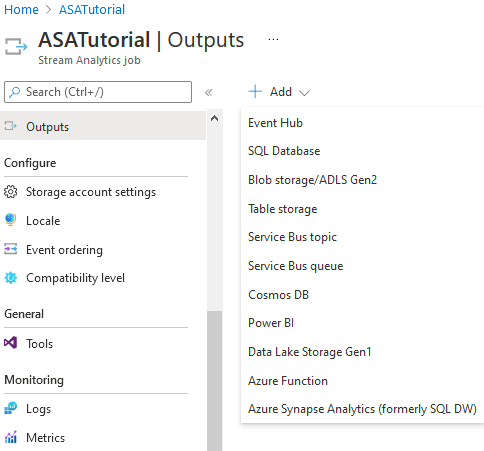
1. Fill out the input form:

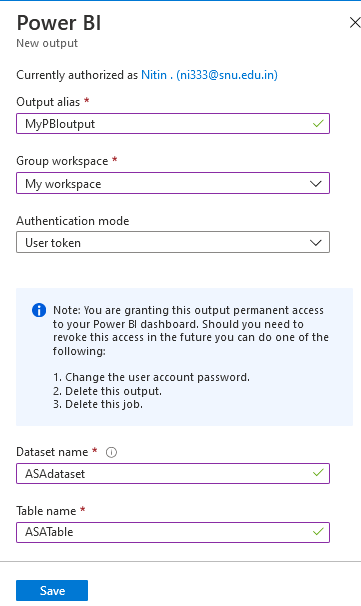


# Configure job output

The last step is to define an output sink where the job can write the transformed data. In this tutorial, you output and visualize data with Power BI.

1. From the Azure portal, open **All resources**, and select the ASATutorial **Stream Analytics job**.
2. In the **Job Topology** section of the **Stream Analytics job**, select the **Outputs** option.
3. Select **+ Add > Power BI**. Then, select Authorize and follow the prompts to authenticate Power BI. You will need a work (or education/school) account to login to power bi.

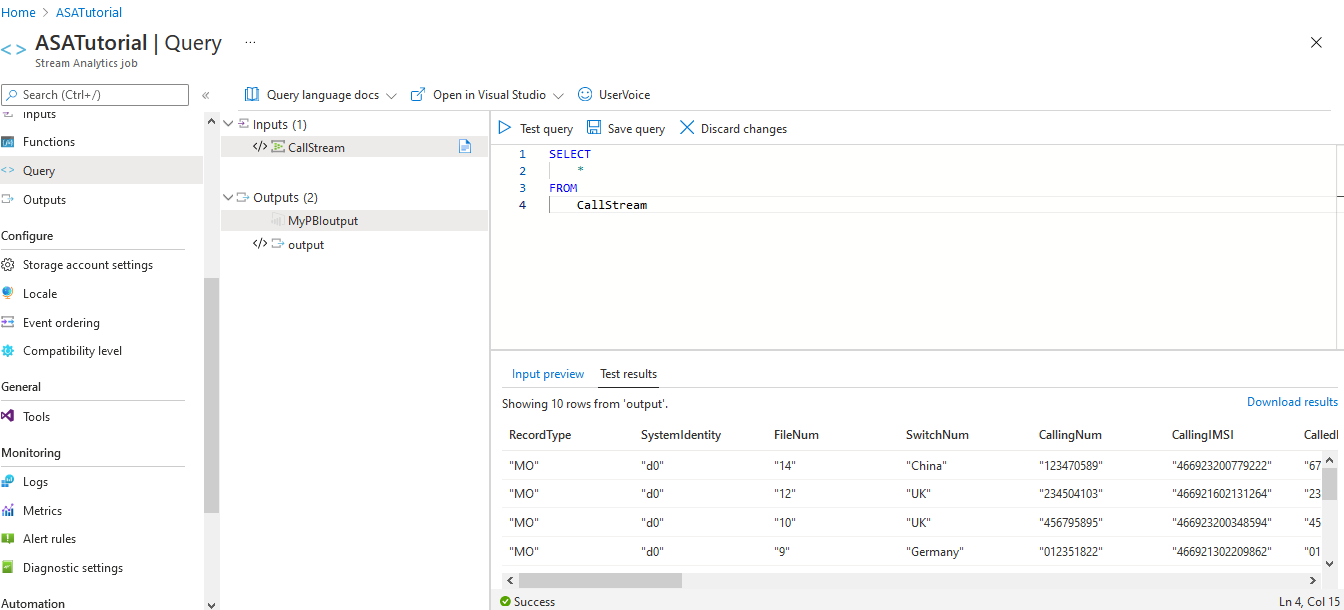




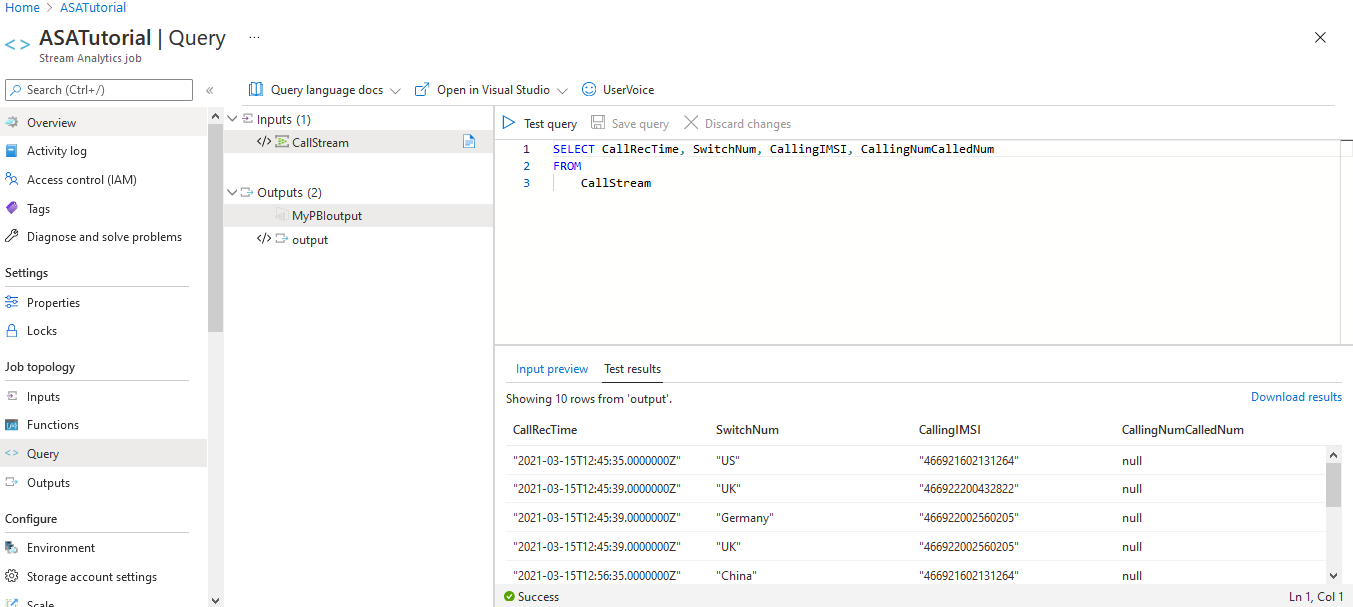
# Create queries to transform real-time data

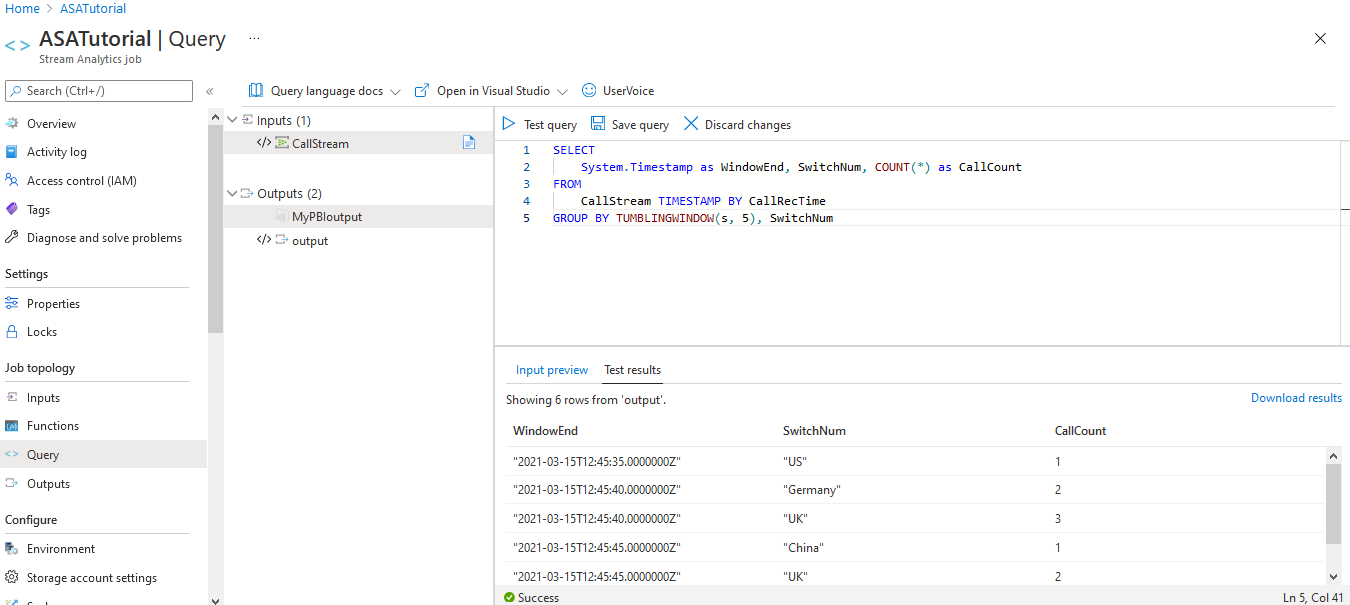
At this point, you have a Stream Analytics job set up to read an incoming data stream. The next step is to create a query that analyzes the data in real time. The queries use a SQL-like language that has some extensions specific to Stream Analytics.

1. Navigate to your **Stream Analytics job** in the Azure portal and select **Query** under **Job topology**.
2. Type/test your queries:

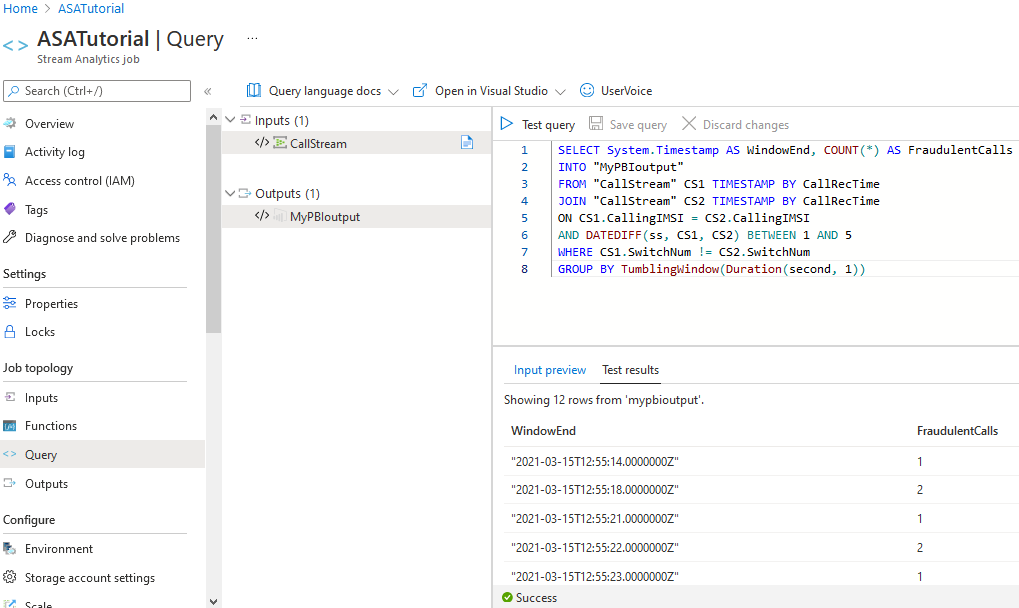


Reduce the number of fields using a column projection:



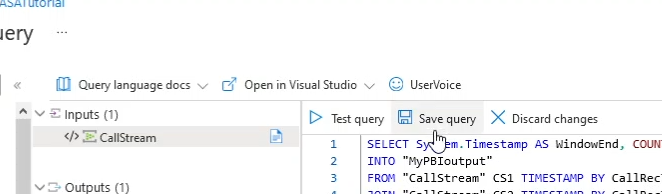
Count incoming calls by region: Tumbling window with aggregation:

Detect SIM fraud using a self-join:



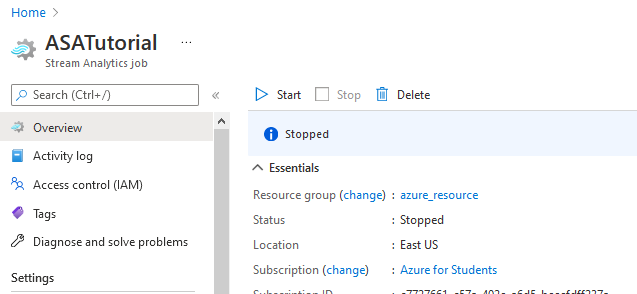
Source- <https://docs.microsoft.com/en-us/azure/stream-analytics/stream-analytics-real-time-fraud-detection#detect-sim-fraud-using-a-self-join>

Select **Test query**. Review the output, and then select **Save query to save this query so we can visualize results in Power BI.**

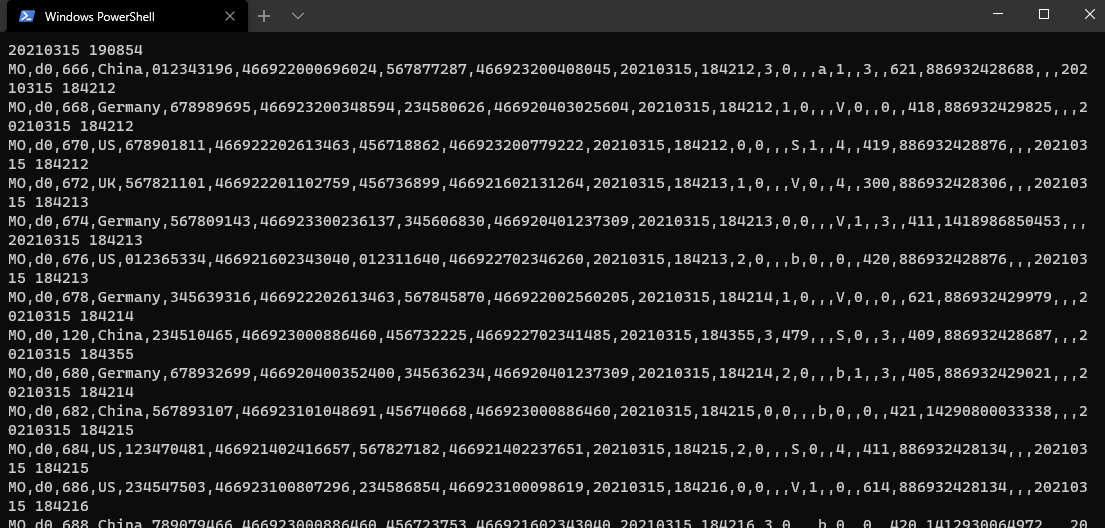


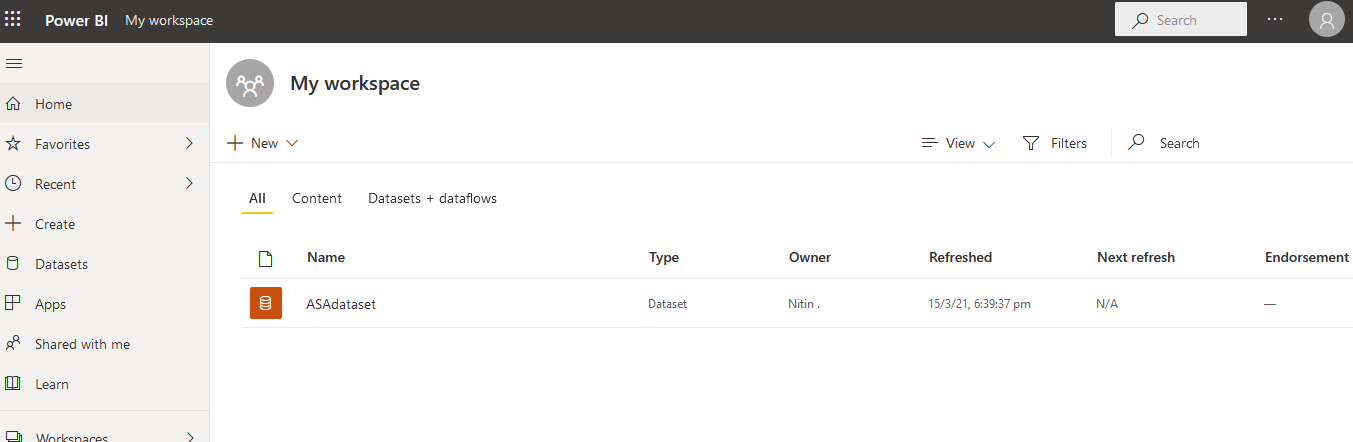
# Start the job and visualize output

1. To start the job, navigate to the job **Overview** and select **Start**.

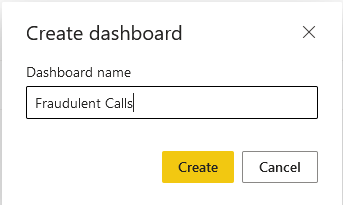
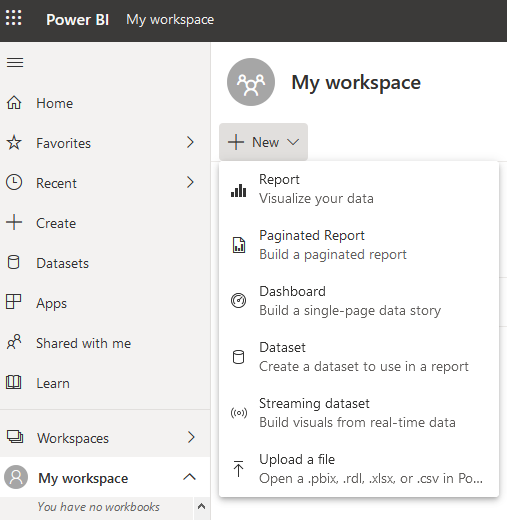


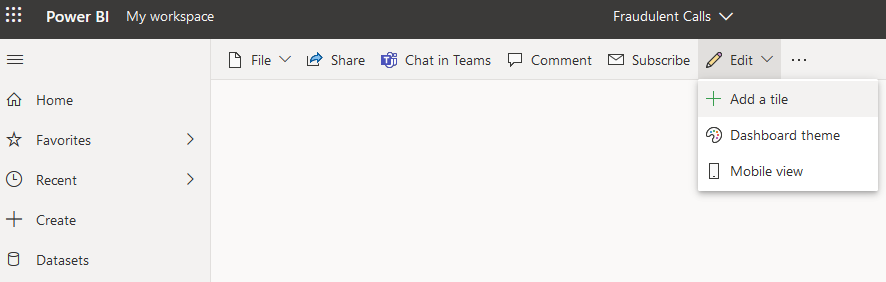
1. Select Now for job output start time and select **Start**. You can view the job status in the notification bar.
2. Once the job succeeds, navigate to [Power BI](https://powerbi.com/) and sign in with your work or school account. If the Stream Analytics job query is outputting results, the ASAdataset dataset you created exists under the Datasets tab.

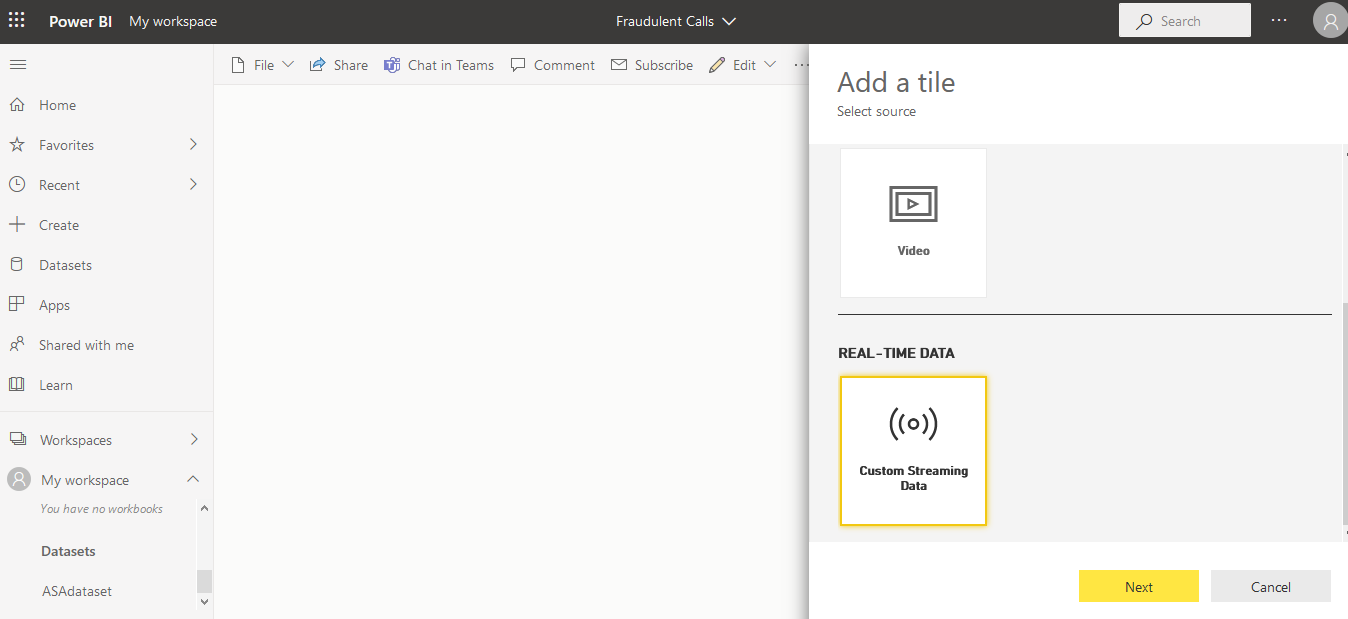


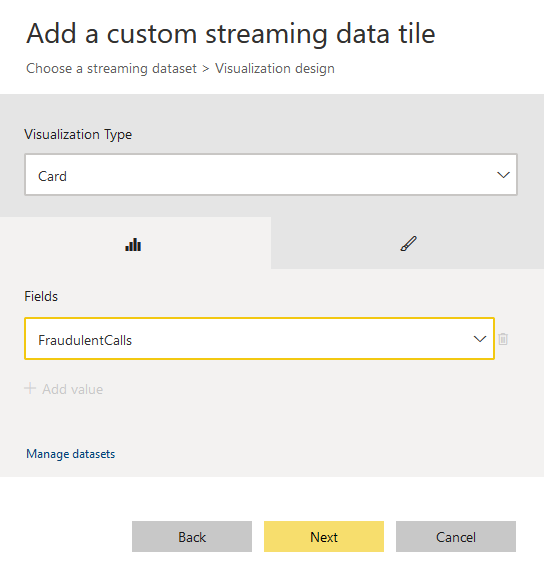
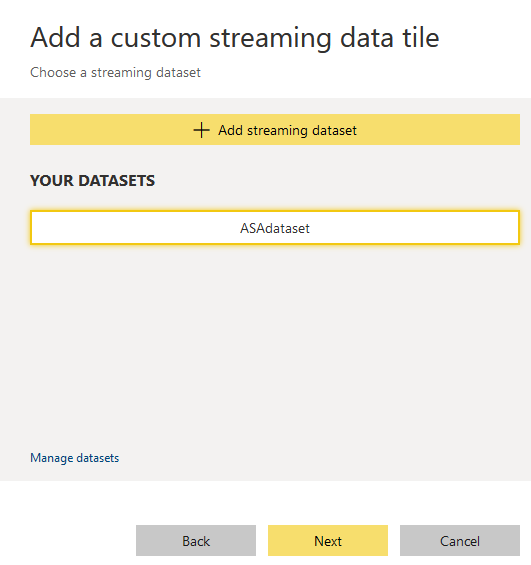


1. From your Power BI workspace, select **+ New** to create a new **Dashboard** named Fraudulent Calls. Then Follow steps as shown in images below.

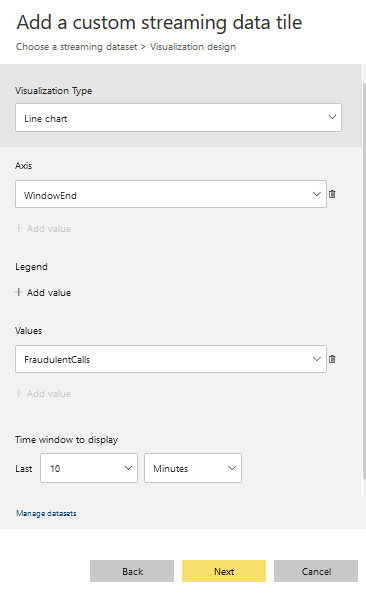


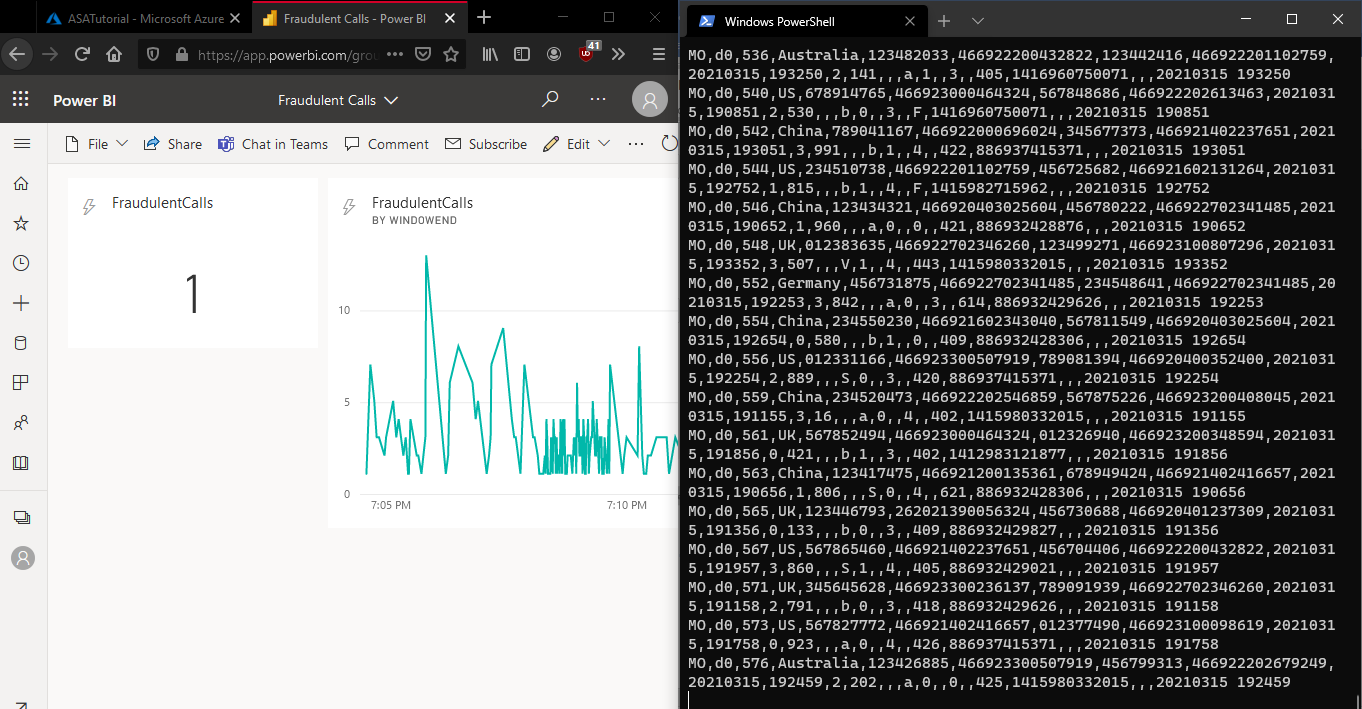






Add another tile for stream data in Line chart form.

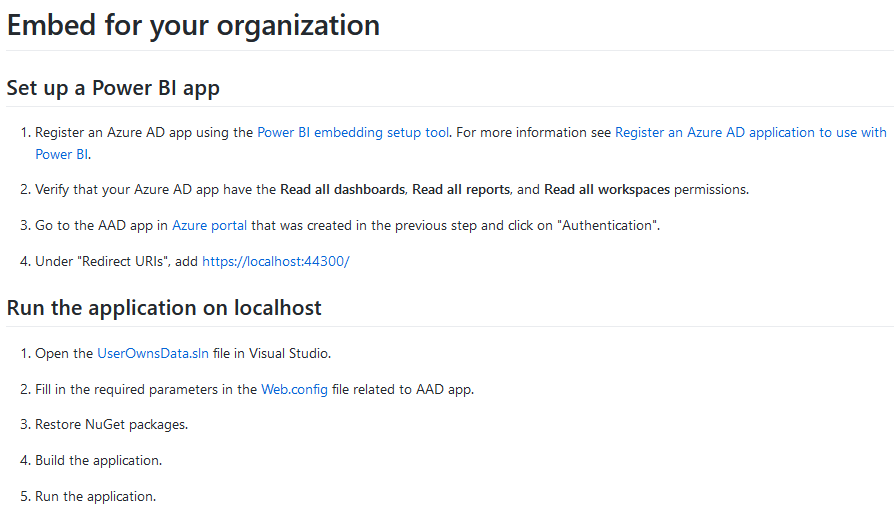




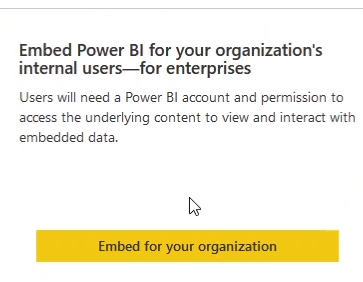
# Embedding your Power BI Dashboard in a Web Application

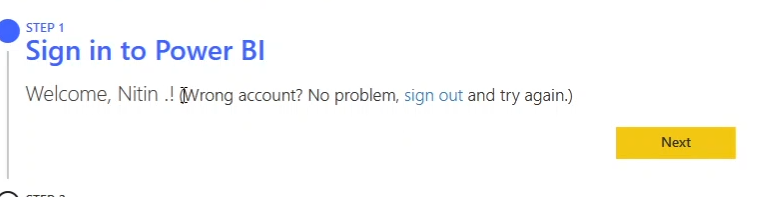
Now that we can get data inside Power BI, we will now use a sample web application to show the visualizations on web.

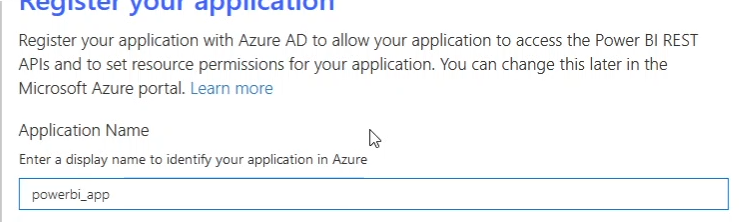
1. Go to [ASP.NET](https://dotnet.microsoft.com/download) and download .NET Core 3.1 and install it. Then download [.NET Framework 4.8 Developer Pack](https://dotnet.microsoft.com/download/dotnet-framework/net48) and install it. Also, we will need an IDE to run web app, so [download Visual Studio 2019](https://visualstudio.microsoft.com/).
2. Now we will have to follow steps as shown in the [github repo](https://github.com/microsoft/PowerBI-Developer-Samples/tree/master/.NET%20Framework) for sample web application for section “Embed for your organization”.

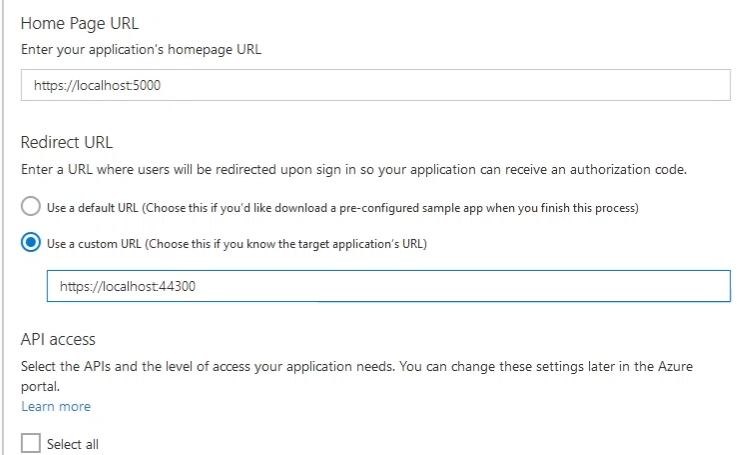


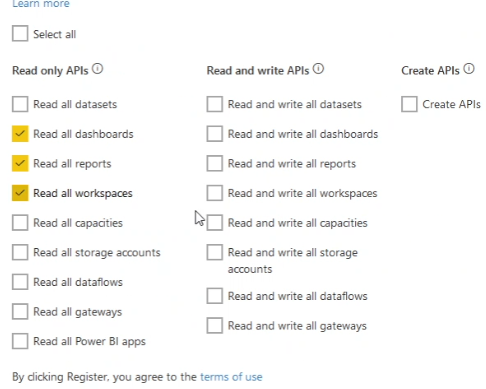
1. Next, go to <https://app.powerbi.com/embedsetup> an select “Embed for your organization”, then follow the steps to get required ids-

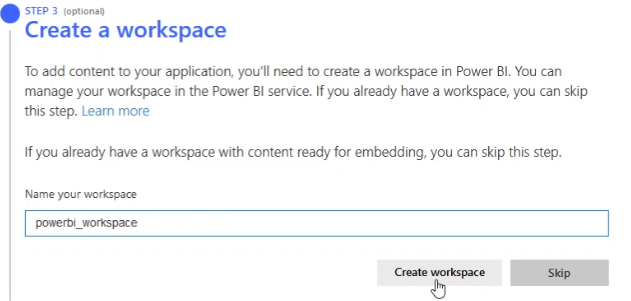




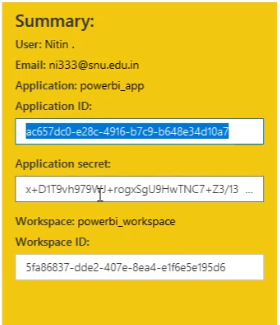


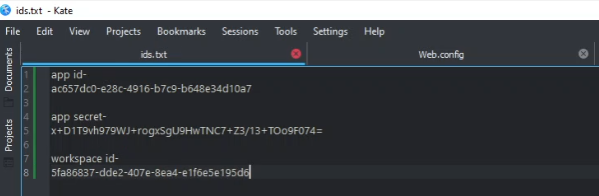




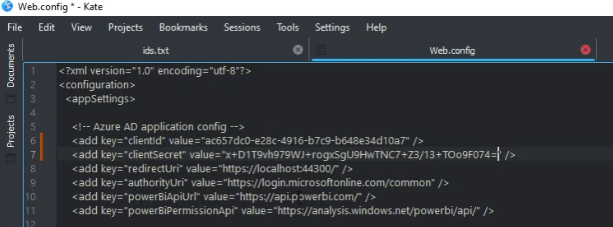


Now copy **application id** and **application secret id** to a text file that we will use later-



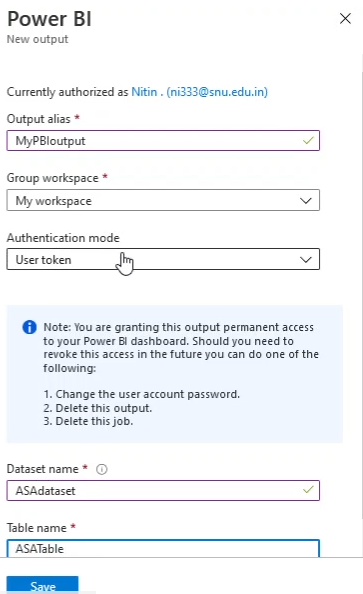


1. Copy “application id” and paste it “clientid” line in “Web.config” inside sample web project folder (D:\fruad\_det\Embed for your organization\UserOwnsData\Web.config). Similarly, copy “application secret” and paste it “clientsecret” line in “Web.config”.



Now, we will not get any output since streaming data is only available in “MyWorkspace” inside power bi account. We will need to add output job inside “ASATutorial Stream Analytics Job”.

1. Go to **Outputs** in **Job Topology** inside **Stream Analytics Job**. Either delete previous one and create new one, or just create new one with another name. We will create new one.



Save the file and run the program (ctrl+F5) inside visual studio by opening “UserOwnsData.sln” file with visual studio. Login inside web browser with account and you will be able to get the result in web. Then start the **stream analytics job**.

**So basically, repeat step 10 (Start the job and visualize output) and create visualizations again for our new workspace (powerbi\_workspace).**

1. After we are done run the code in Visual Studio, then login with bi account, select dashboard and workspace, click on embed and you will be able to see result inside web browser.

