Azure Stream Analytics:

Module 6, Lesson 6   
Real-time fraud detection Hands-On Lab

## Overview

In this lab, students will learn how to create an end-to-end solution for real-time fraud detection with Azure Stream Analytics. Students with bring events into an Azure event hub, write Stream Analytics queries for aggregation or alerting, and send the results to an output sink to gain insight over data with real-time processing. Real time anomaly detection for telecommunications as shown in this example technique is equally suited for other types of fraud detection such as credit card or identity theft scenarios

## Objectives

In this hands-on lab you will learn how to:

* Setup data to input into a pipeline of stages that will result in a new model
* Setup the pipeline and execute it
* Setup test data and run the new model against the test data

## Prerequisites

The following are required to complete this hands-on lab:

* A web browser
* Download [TelcoGenerator.zip](http://download.microsoft.com/download/8/B/D/8BD50991-8D54-4F59-AB83-3354B69C8A7E/TelcoGenerator.zip) from the Microsoft Download Center

Note : The Azure portal is continually improved and changed. The steps in this exercise reflect the user interface of the Microsoft Azure portal at the time of writing, but may not match the latest design of portal.

## Exercises

This hands-on lab includes the following exercises:

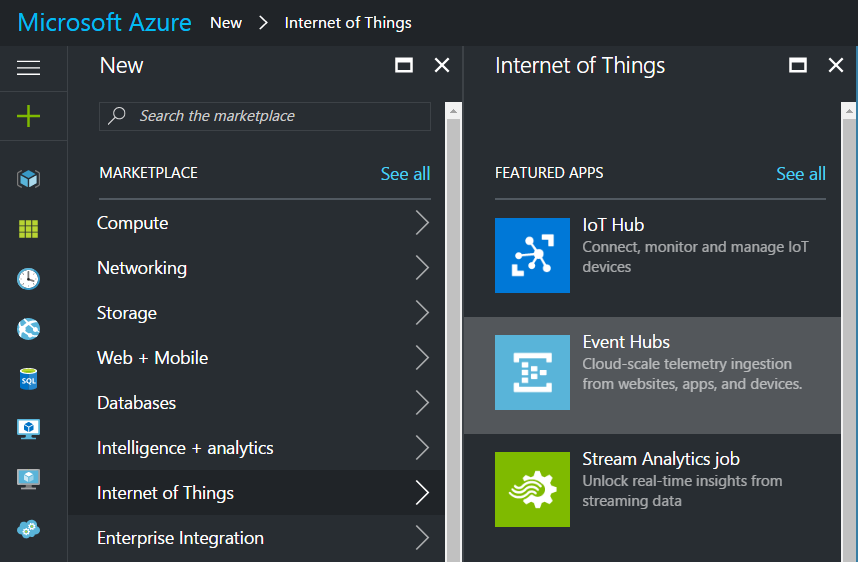
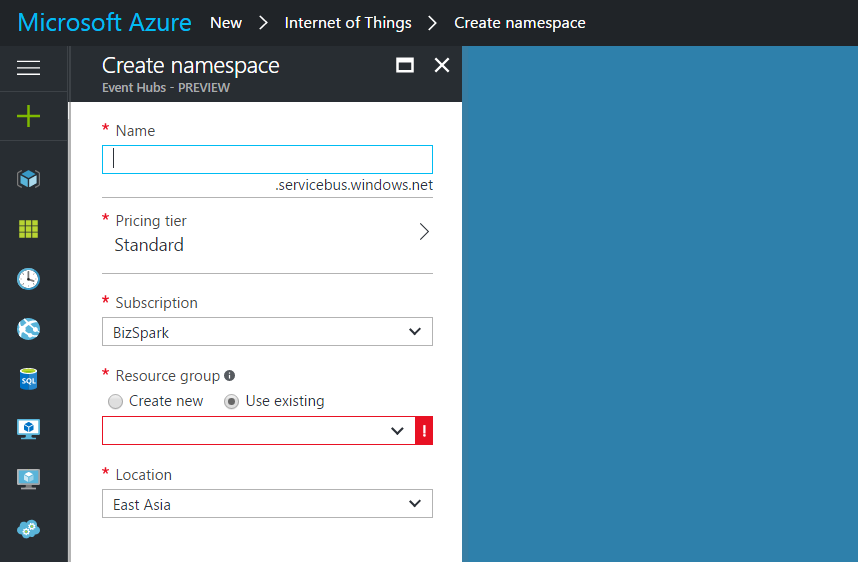
* Exercise 1: Create an Azure Event Hubs input
* Exercise 2: Configure and start event generator application
* Exercise 3: Create Stream Analytics Job
* Exercise 4: Create Stream Analytics Job Output

## Exercise 1: Create an Azure Event Hubs input

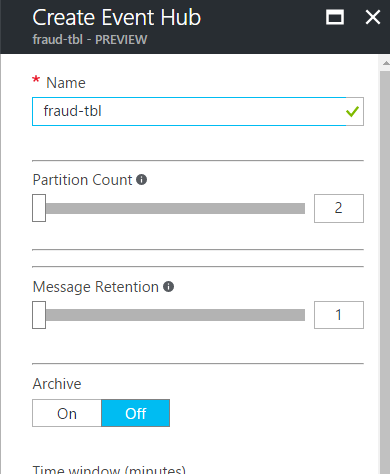
Students will generate events and push them to an Event Hub instance for real-time processing. Service Bus Event Hubs are the preferred method of event ingestion for Stream Analytics.

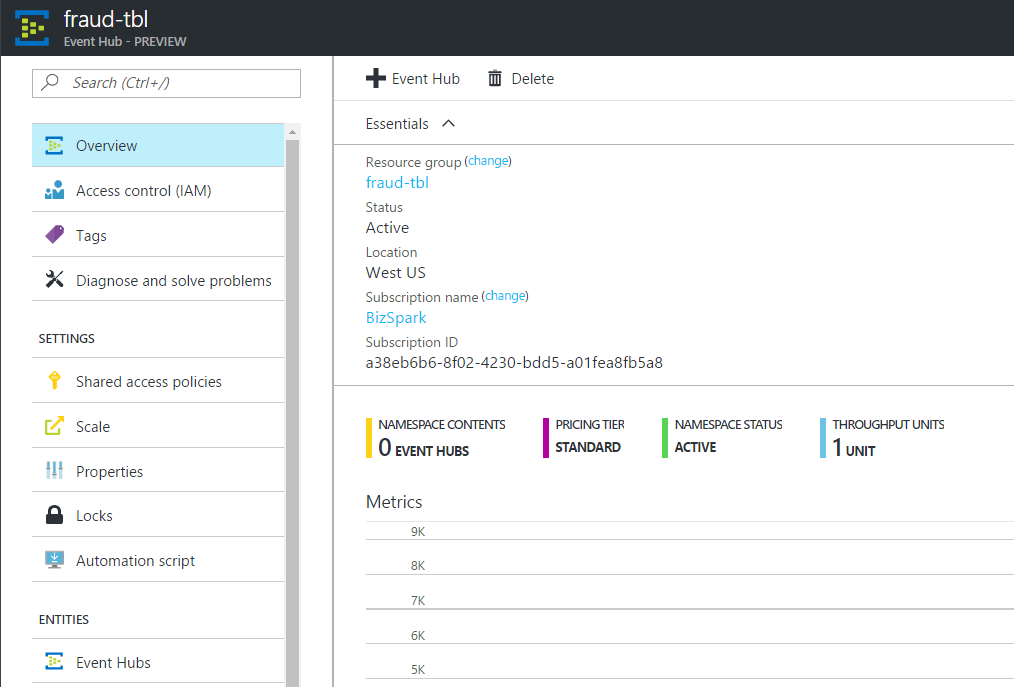
1. Create Event Hub Namespace

In the [Azure portal](https://portal.azure.com) click **New** > **Internet of Things** > **Event Hub** to create a new event hub namespace. Provide a friendly name for the namespace. Create or use an existing resource group. Resource groups allow you to group and manage resources together. Select an appropriate location (Region) to locate the Event Hub.



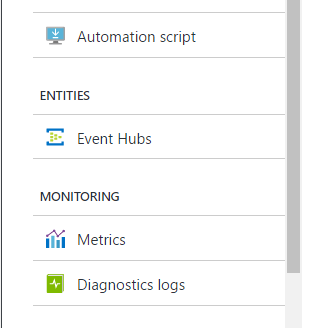
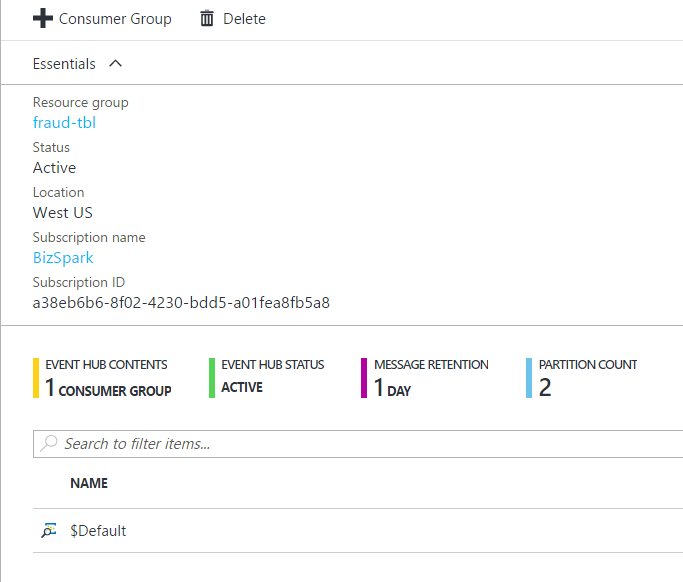
1. Create Event Hub

Click on **All resources** and select the newly created Event Hub from step 1.



Click on the **+ Event Hub** and fill in a friendly name for the new Event Hub, the click on **Create** at the bottom of the page to complete the operation.

1. Create Consumer Group

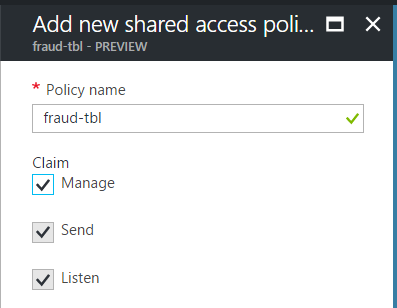


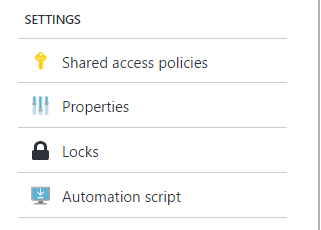
**From the menu on the left, select Entities – Event Hubs**. This will display all Event Hubs, including the one the student created in above step. Select the Event Hub created in above step. Once selected, click on **+ Consumer Group**. Fill in a friendly name for the new Consumer Group and click **Create** to complete the operation.

As a best practice, each Stream Analytics job should read from a single Event Hub Consumer Group.

1. Grant access to Event Hub

A shared access policy needs to be created to grant access to the Event Hub. From the menu on the left, select **Shared Access Policies** and then click **+ Add** to create a new policy with **Manage** permissions.

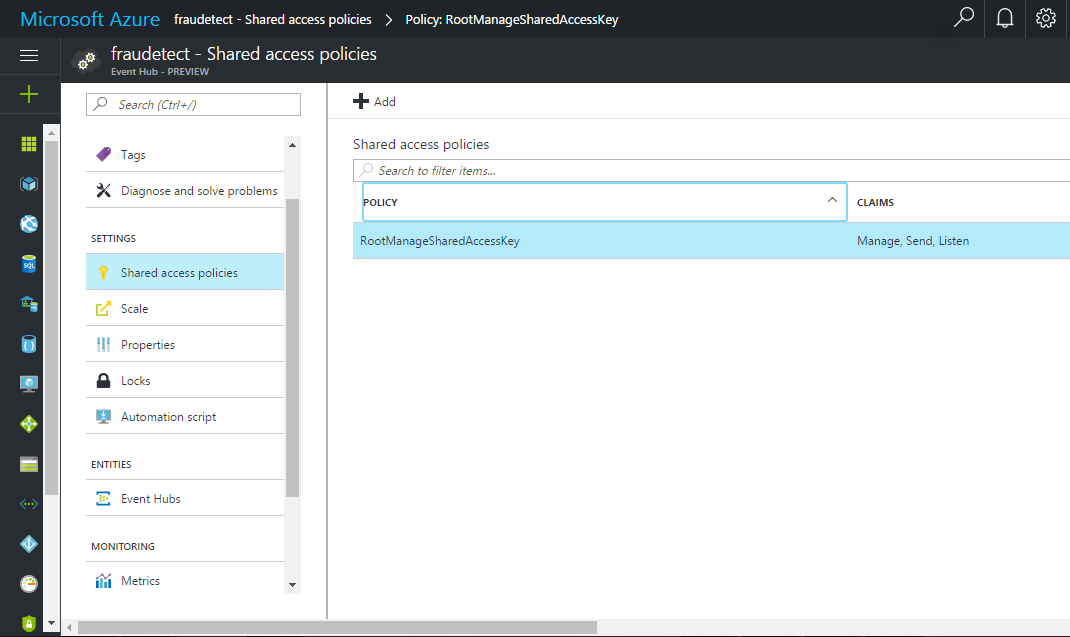


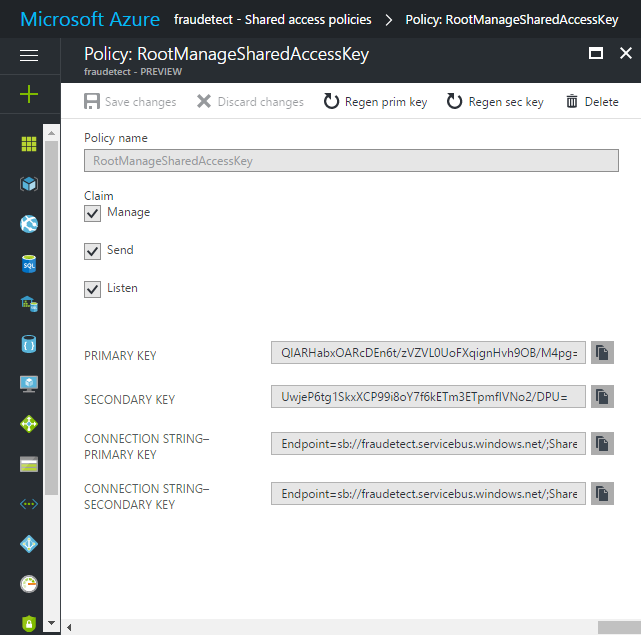


After entering a policy name and manage permissions, click **Create** at bottom of page.

1. Copy **Connection Information**

Return to the Event Hub namespace and under settings click **Shared access policies**. Click on **RootManagerSharedAccessKey** and copy the Connection String – Primary Key.





## Exercise 2: Configure and start event generator application

For this lab, students will use a client application that will generate sample incoming call metadata and push it to Event Hub. Follow the steps below to set up this application.

1. Download the [TelcoGenerator.zip file](http://download.microsoft.com/download/8/B/D/8BD50991-8D54-4F59-AB83-3354B69C8A7E/TelcoGenerator.zip). Then unzip it to a directory.

Windows may block the downloaded zip file. Right click the file and select properties. If the message "This file came from another computer and might be blocked to help protect this computer." then tick the "Unblock" box and click apply on the zip file.

1. Replace the Microsoft.ServiceBus.ConnectionString and EventHubName values in **telcodatagen.exe.config** with the Event Hub connection string and name.

The connection string copied from the Azure portal places the name of the connection at the end

1. Start the telcodatagen application.

The usage is as follows:

telcodatagen.exe [#NumCDRsPerHour] [SIM Card Fraud Probability] [#DurationHours]

telcodatagen.exe 1000 .2 2

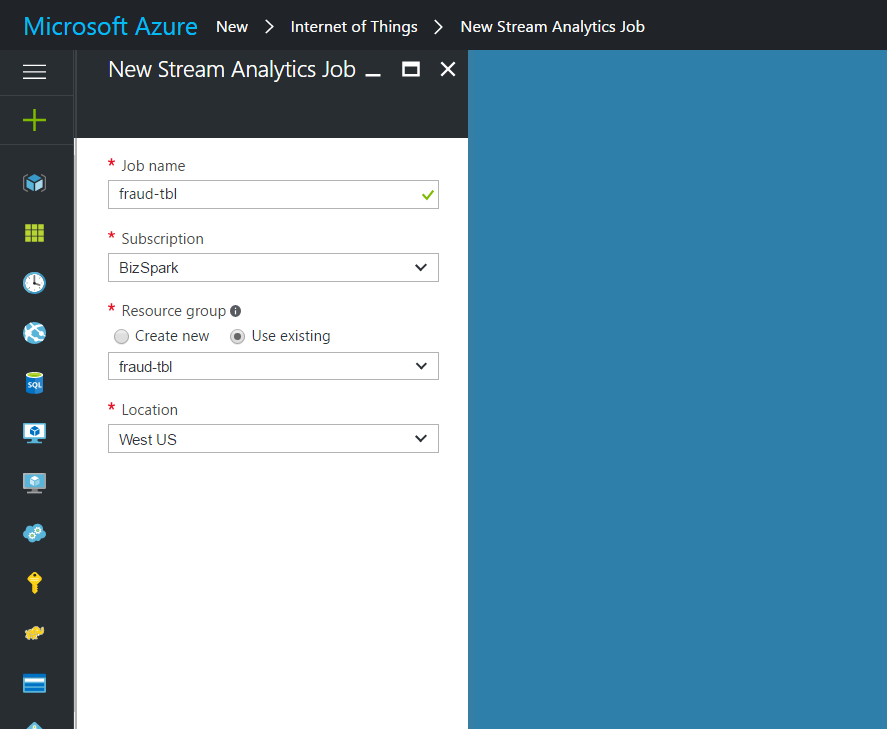
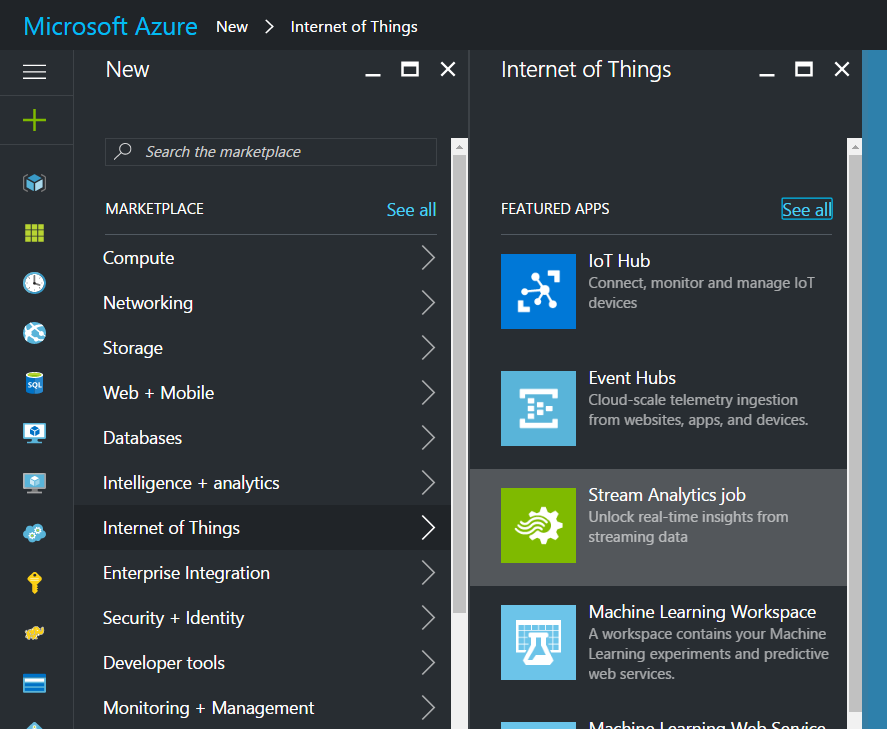
Students should see data being sent to the Event Hub. Some key fields that we will be using in this real-time fraud detection application are defined here:

| **Record** | **Definition** |
| --- | --- |
| CallrecTime | Timestamp for the call start time. |
| SwitchNum | Telephone switch used to connect the call. |
| CallingNum | Phone number of the caller. |
| CallingIMSI | International Mobile Subscriber Identity (IMSI). Unique identifier of the caller. |
| CalledNum | Phone number of the call recipient. |
| CalledIMSI | International Mobile Subscriber Identity (IMSI). Unique identifier of the call recipient. |

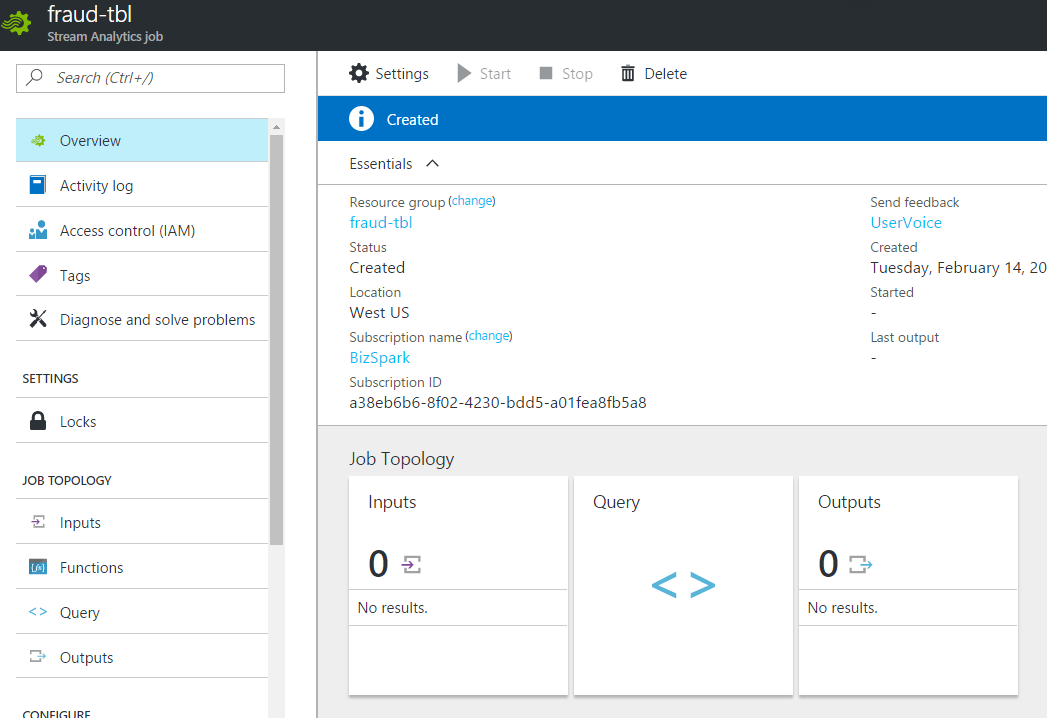
## Exercise 3: Create Stream Analytics Job

Students have just created an event hub where a stream of telecommunications events are being sent. The next step is to set up a Stream Analytics job to analyze these events in real time.

1. Provision a Stream Analytics Job

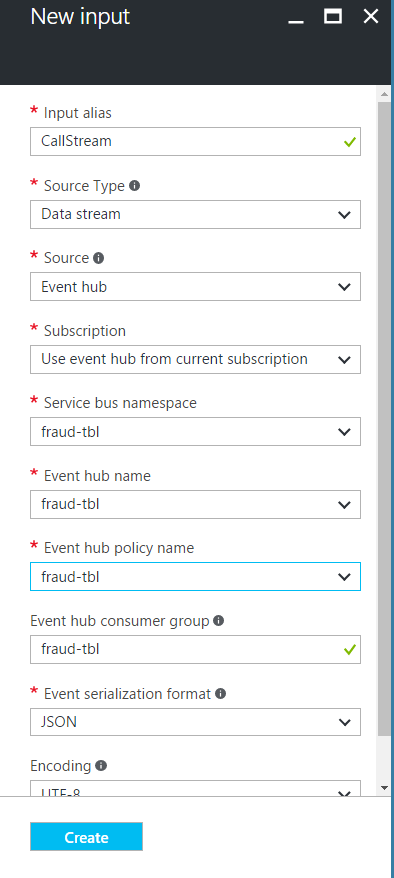
****In the Azure portal, click **New > Internet of Things > Stream Analytics job**

Specify the Job Name. Select a region where the job will run. It is best to choose the same region as where the Event Hub created in the previous exercise. This will ensure that data is not being transferred between regions at an additional cost. The easiest way to accomplish this is to use the same resource group that the student either created or used in Exercise 1. Finally, click on the **Create** button on the bottom to create a new Stream Analytics job.

From the main portal menu, click on **All resources** and select the Stream Analytics Job just created.

Notice that there are 0 Inputs and 0 outputs. In addition, the start button is disabled. This will be enabled once the job input, output, and query is specified.

1. Specify Job Inputs

Click **Inputs** from the job topology menu on the left, and then click **+ Add**. The dialog box that opens will walk students through a number of steps to set up the input.

In **Input Alias**, type in any friendly name. This name will be used later in queries.

In **SOURCE TYPE**, select Data Stream.

In **SOURCE** choose Event Hub.

In **SUBSCRIPTION**, student should be able to leave the default subscription from which student has been working.

In **SERVICE BUS NAMESPACE** choose the name of the namespace that the student created in Exercise 1.

In **EVENT HUB NAME** choose the name of the namespace that the student created in Exercise 1.

In **EVENT HUB POLICY NAME** choose the name of the namespace that the student created in Exercise 1.

In **EVENT HUB CONSUMER GROUP** type in the name of the Event Hub Consumer Group that the student created in Exercise 1. The student should see a green check indicating that the consumer group is valid.

In **EVENT SERIALIZATION** keep the default JSON format.

In **ENCODING** keep the default UTF-8 format.

Click the check button to add the specified input source and verify that Stream Analytics can connect to the specified Event Hub.

1. Specify Job Query

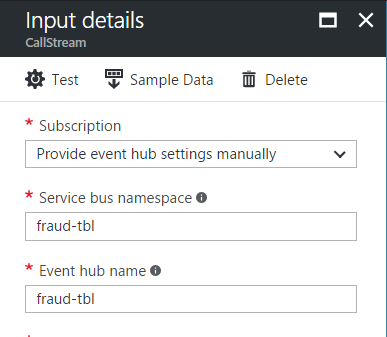
Stream Analytics supports a simple, declarative query model for describing transformations for real-time processing. To learn more about the language, see the [Azure Stream Analytics Query Language Reference](https://msdn.microsoft.com/library/dn834998.aspx). This tutorial will help students author and test several queries over the real-time stream of call data.

From the **Job Topology** Menu on the left, select **Query** to get started.

* 1. Create sample input data

The **Sample data from input** feature can be used to extract sample events on which users can validate queries.

Return to the **Inputs** screen by selecting it from the **Job Topology** menu. Select the Input the student created from step 2.



From the Input details screen, select the select the **Sample Data**. In the dialogue box that appears, specify a **START TIME** and **Duration**. Students should keep the default start time and adjust the Duration to about 2 minutes. Click **OK** to start sampling event data. Once the process is completed, a pop-up notification screen will allow the student to **Download** the sample data. Students can return to this notification anytime to download the sample data.

* 1. Create a Pass-through query

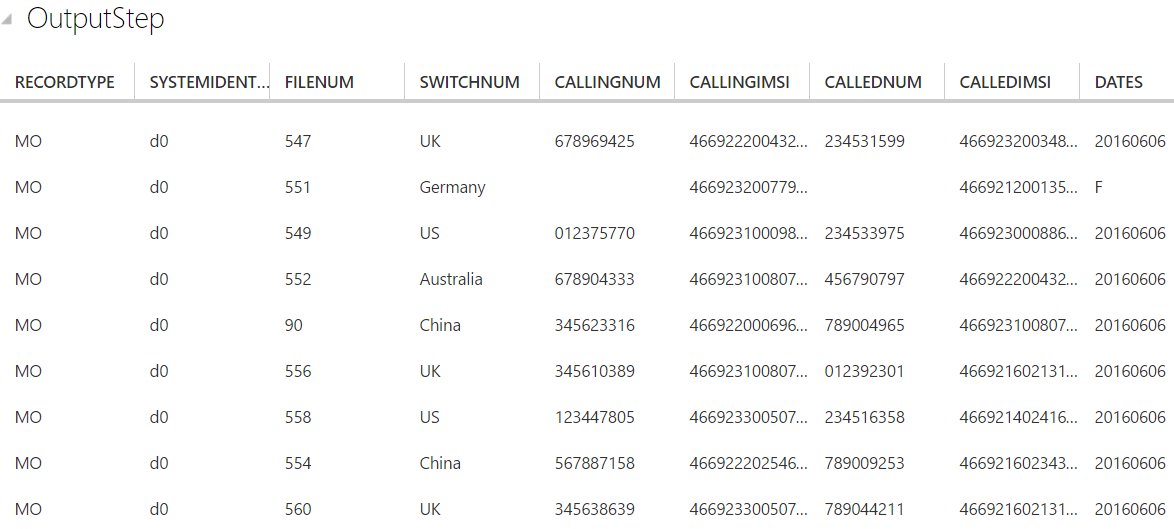
All events can be archived using a pass-through query to read all the fields of the event.

Click **Query** from the top of the Stream Analytics job page and add the following code.

Select \* from *CallStream*

*The example shows CallStream, however, students should substitute the INPUT ALIAS declared in the previous step*.

Click **TEST** from the query editor to execute the query.

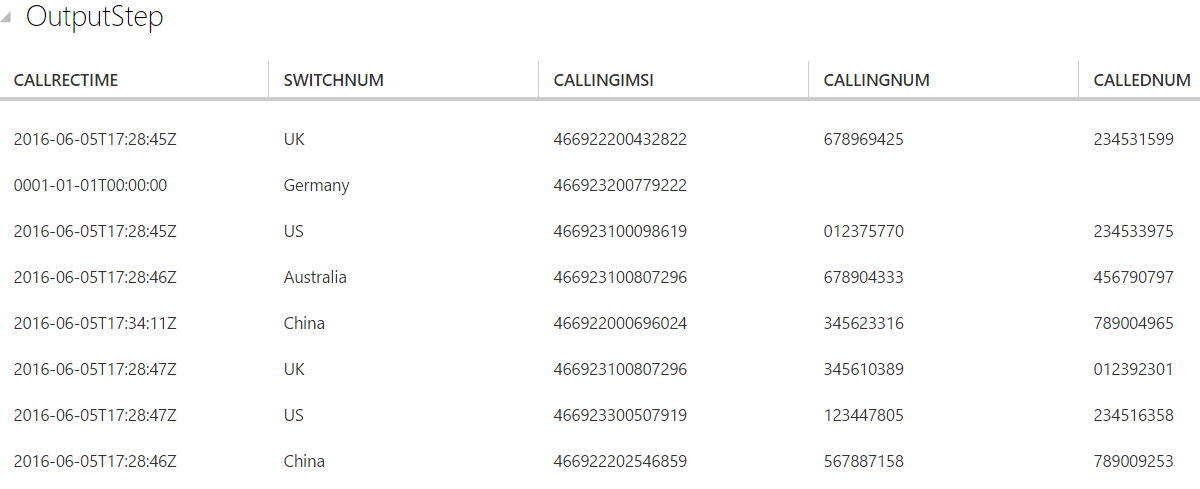
Click the check button to see the results of query displayed by scrolling down.

* 1. Create a column projection

Change the query to the code below and click **Test** to see results of query.

SELECT CallRecTime, SwitchNum, CallingIMSI, CallingNum, CalledNum

FROM CallStream



* 1. Create tumbling window with aggregation query

Create a query to count number of incoming calls by region. To compare the amount of incoming calls per region, students will use a tumbling window to get the count of incoming calls grouped by SwitchNum every 5 seconds.

Change the query to the code below and click **Test** to see results of query.

SELECT System.Timestamp as WindowEnd, SwitchNum, COUNT(\*) as CallCount

FROM CallStream TIMESTAMP BY CallRecTime

GROUP BY TUMBLINGWINDOW(s, 5), SwitchNum

This query uses the Timestamp By keyword to specify a timestamp field in the payload to be used in the temporal computation. If this field wasn't specified, the windowing operation would be performed using the time each event arrived at Event Hub.

* 1. Create SIM fraud detection with a Self-Join

To identify potentially fraudulent usage, we'll look for calls originating from the same user but in different locations in less than 5 seconds. We join the stream of call events with itself to check for these cases.

Change the query to the code below and click **Test** to see results of query.

SELECT System.Timestamp as Time, CS1.CallingIMSI, CS1.CallingNum as CallingNum1,

CS2.CallingNum as CallingNum2, CS1.SwitchNum as Switch1, CS2.SwitchNum as Switch2

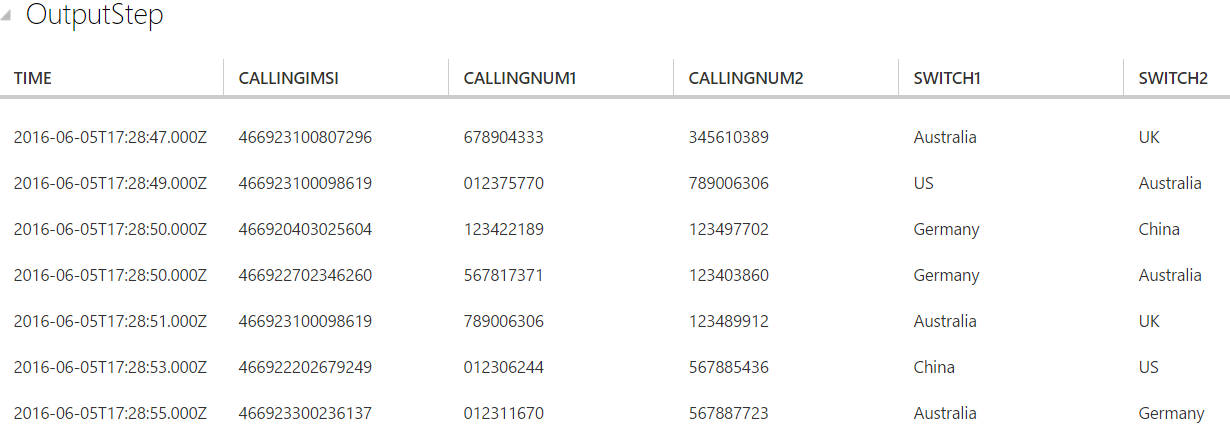
FROM CallStream CS1 TIMESTAMP BY CallRecTime

JOIN CallStream CS2 TIMESTAMP BY CallRecTime

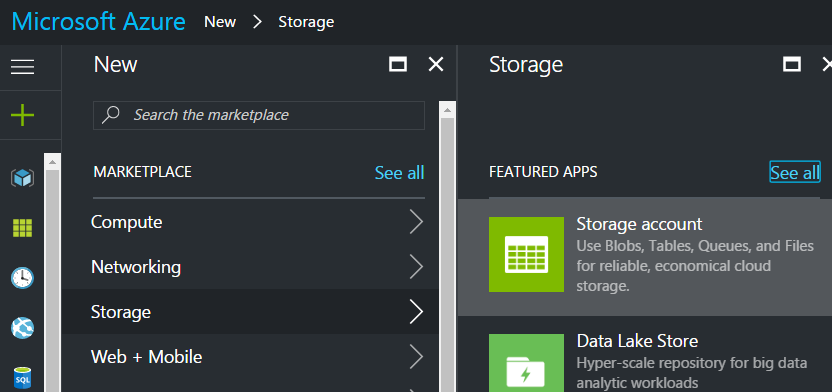
ON CS1.CallingIMSI = CS2.CallingIMSI

AND DATEDIFF(ss, CS1, CS2) BETWEEN 1 AND 5

WHERE CS1.SwitchNum != CS2.SwitchNum

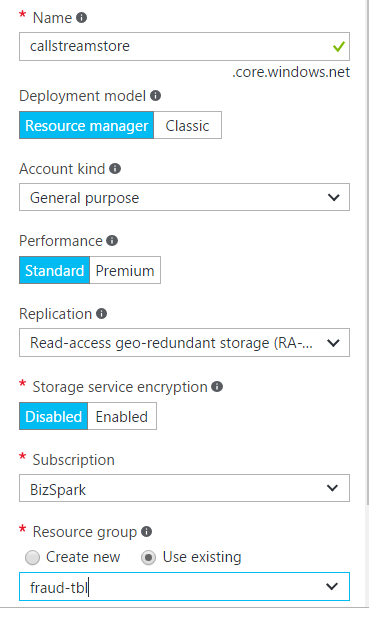


## Exercise 4: Create Stream Analytics Job Output

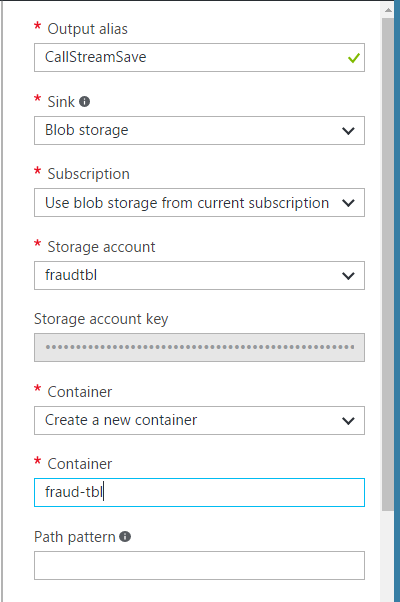
Students have defined an event stream, created an Event Hub input to ingest the events, and queries to perform transformation over the event stream. Now, the output of these Stream Analytics will be defined.

1. Create a Storage Account

Use an existing storage account or create a new storage account by clicking **NEW > STORAGE > STORAGE ACCOUNT** and following the instructions

Enter a friendly name for storage account. Leave all the other parameters to the default setting as shown here. For the **Resource Group**, the student can either create a new or use an existing one. As previously mentioned, it is often easier to manage resources for a project by placing all the resources in the same resource group. Finally, click on **Create** the storage account.

1. Specify Job Outputs

Return to the Stream Analytics job to create an Output for the project.

Click **Outputs** from the job topology menu, and then click **Add**.

In **OUTPUT Alias**, type in any friendly name.

In **SINK**, select Blob Storage.

In **SUBSCRIPTION**, student should be able to leave the default subscription from which student has been working.

In **STORAGE ACCOUNT** select the storage account created in step 1. Make sure the account is located in the same Region as the Event Hub. Otherwise, there will be an additional charge to move the data. If the student has been using the same Resource Group, all the resources should be in a single region.

In **CONTAINER** either create a new container or use an already existing one

In **EVENT SERIALIZATION** keep the default JSON format.

In **ENCODING** keep the default UTF-8 format.

Click the **Create** button to add this Output and to verify that Stream Analytics can successfully connect to the storage account.

1. Start job for real time processing

From the job dashboard, click start at the bottom of the page.

In the dialog box that appears, select **JOB START TIME** and then click the check button on the bottom of the dialog box. The job status will change to **Starting** and will shortly move to **Running**

## Summary

In this hands-on lab, you learned how to:

* Create a Stream Analytics Job by working through the 3 major steps:
  + Connecting to a stream of events as Input
  + Creating a query for transformation of the events
  + Creating an Output to save the results of the Stream Analytics job