Get started using Azure Stream Analytics: Real-time fraud detection

By Jeff Stokes
Last updated: 12/04/2015

In this article:

Learn how to create an end-to-end solution for real-time fraud detection with Azure Stream Analytics. 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 is covered but the example technique is equally suited for other types of fraud detection such as credit card or identity theft scenarios.

Stream Analytics is a fully managed service providing low-latency, highly available, scalable complex event processing over streaming data in the cloud. For more information, see Introduction to Azure Stream Analytics.

Scenario: Telecommunications and SIM fraud detection in real-time

A telecommunications company has a large volume of data for incoming calls. The company needs the following from its data: * Pare this data down to a manageable amount and obtain insights about customer usage over time and geographical regions. * Detect SIM fraud (multiple calls coming from the same identity around the same time but in geographically different locations) in real-time so that they can easily respond by notifying customers or shutting down service.

In canonical Internet of Things (IoT) scenarios there is a ton of telemetry or sensor data being generated – and customers want to aggregate them or alert over anomalies in real-time.

Prerequisites

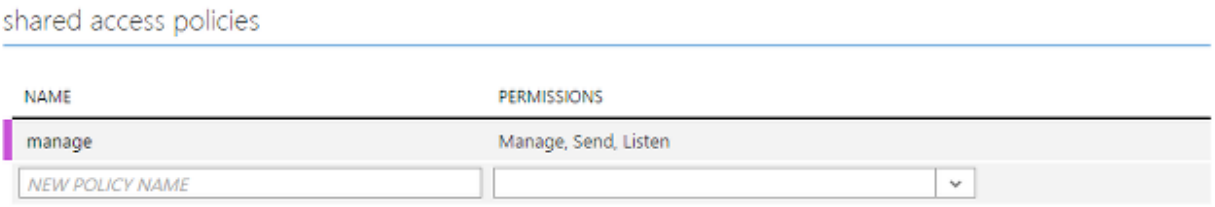
This scenario leverages an event generator located on GitHub. Download it here and follow the steps in this tutorial to set up your solution.

Create an Azure Event Hubs input and Consumer Group

The sample application 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 and you can learn more about Event Hubs in Azure Service Bus documentation.

To create an Event Hub:

1. In the Azure portal click **New** > **App Services** > **Service Bus** > **Event Hub** > **Quick Create**. Provide a name, region, and new or existing namespace to create a new Event Hub.
2. As a best practice, each Stream Analytics job should read from a single Event Hub Consumer Group. We will walk you through the process of creating a Consumer Group below, and you can learn more about Consumer Groups. To create a Consumer Group, navigate to the newly created Event Hub and click the **Consumer Groups** tab, then click **Create** on the bottom of the page and provide a name for your Consumer Group.
3. To grant access to the Event Hub, we will need to create a shared access policy. Click the **Configure** tab of your Event Hub.
4. Under **Shared Access Policies**, create a new policy with **Manage** permissions.



5. Click **Save** at the bottom of the page.
6. Navigate to the **Dashboard** and click **Connection Information** at the bottom of the page, and then copy and save the connection information.

Configure and start event generator application

We have provided 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 solution from https://github.com/Azure/azure-stream-analytics/tree/master/DataGenerators/TelcoGenerator.
2. Replace the Microsoft.ServiceBus.ConnectionString and EventHubName values in App.Config with your Event Hub connection string and name.
3. Build the solution to trigger the download of required nuget packages.
4. Start the application. The usage is as follows:

```
telcodatagen [#NumCDRsPerHour] [SIM Card Fraud Probability] [#DurationHours]
```

The following example will generate 1000 events with a 20 percent probability of fraud over the course of 2 hours.

```
TelcoDataGen.exe 1000 .2 2
```

You will see records being sent to your 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.

Record	Definition
CalledIMSI	International Mobile Subscriber Identity (IMSI). Unique identifier of the call recipient.

Create Stream Analytics job

Now that we have a stream of telecommunications events, we can set up a Stream Analytics job to analyze these events in real-time.

Provision a Stream Analytics job

1. In the Azure portal, click **New** > **Data Services** > **Stream Analytics** > **Quick Create**.
2. Specify the following values, and then click **Create Stream Analytics Job**:
   - **Job Name**: Enter a job name.
   - **Region**: Select the region where you want to run the job. Consider placing the job and the event hub in the same region to ensure better performance and to ensure that you will not be paying to transfer data between regions.
   - **Storage Account**: Choose the Azure storage account that you would like to use to store monitoring data for all Stream Analytics jobs running within this region. You have the option to choose an existing storage account or to create a new one.
3. Click **Stream Analytics** in the left pane to list the Stream Analytics jobs.

4. The new job will be shown with a status of **Created**. Notice that the **Start** button on the bottom of the page is disabled. You must configure the job input, output, and query before you can start the job.

Specify job input

1. In your Stream Analytics job click **Inputs** from the top of the page, and then click **Add Input**. The dialog box that opens will walk you through a number of steps to set up your input.
2. Select **Data Stream**, and then click the right button.
3. Select **Event Hub**, and then click the right button.
4. Type or select the following values on the third page:
   - **Input Alias**: Enter a friendly name for this job input such as *CallStream*. Note that you will be using this name in the query later.

- **Event Hub**: If the Event Hub you created is in the same subscription as the Stream Analytics job, select the namespace that the event hub is in.

  If your event hub is in a different subscription, select **Use Event Hub from Another Subscription** and manually enter information for **Service Bus Namespace**, **Event Hub Name**, **Event Hub Policy Name**, **Event Hub Policy Key**, and **Event Hub Partition Count**. * **Event Hub Name**: Select the name of the Event Hub. * **Event Hub Policy Name**: Select the event-hub policy created earlier in this tutorial. * **Event Hub Consumer Group**: Type the Consumer Group created earlier in this tutorial.

5. Click the right button.
6. Specify the following values:
   - **Event Serializer Format**: JSON
   - **Encoding**: UTF8
7. Click the check button to add this source and to verify that Stream Analytics can successfully connect to the event hub.

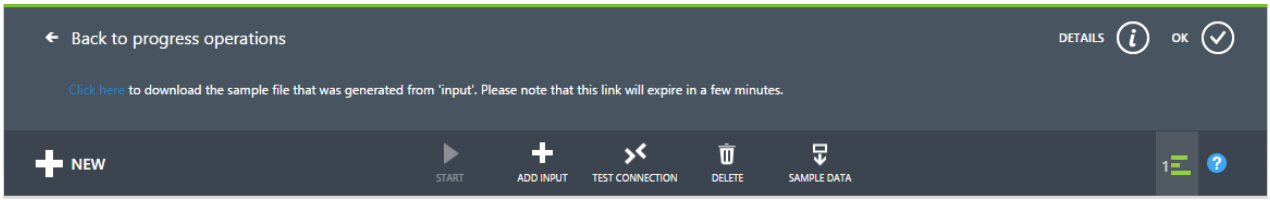
Specify job query

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. This tutorial will help you author and test several queries over your real-time stream of call data.

Optional: Sample input data

To validate your query against actual job data, you can use the **Sample Data** feature to extract events from your stream and create a .JSON file of the events for testing. The following steps show how to do this and we have also provided a sample Telco.json file for testing purposes.

1. Select your Event Hub input and click **Sample Data** at the bottom of the page.
2. In the dialog box that appears, specify a **Start Time** to start collecting data from and a **Duration** for how much additional data to consume.
3. Click the check button to start sampling data from the input. It can take a minute or two for the data file to be produced. When the process is completed, click **Details** and download and save the .JSON file that is generated.



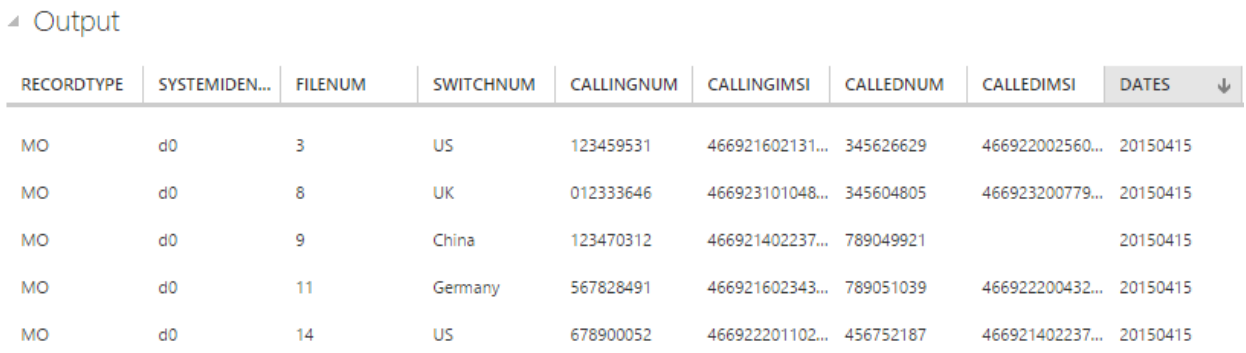
Passthrough query

If you want to archive every event, you can use a passthrough query to read all the fields in the payload of the event or message. To start with, do a simple passthrough query that projects all the fields in an event.

1. Click **Query** from the top of the Stream Analytics job page.
2. Add the following to the code editor:

   ```
   SELECT * FROM CallStream
   ```

   Make sure that the name of the input source matches the name of the input you specified earlier.
3. Click **Test** under the query editor.
4. Supply a test file, either one that you created using the previous steps or use Telco.json.
5. Click the check button and see the results displayed below the query definition.



Column projection

We'll now pare down the returned fields to a smaller set.

1. Change the query in the code editor to:

   ```
   SELECT CallRecTime, SwitchNum, CallingIMSI, CallingNum, CalledNum
   FROM CallStream
   ```
2. Click **Rerun** under the query editor to see the results of the query.

Output

CALLRECTIME	SWITCHNUM	CALLINGIMSI	CALLINGNUM	CALLEDNUM
2015-04-15T17:39:29Z	US	466921602131264	123459531	345626629
2015-04-15T17:39:31Z	UK	466923101048691	012333646	345604805
2015-04-15T17:39:31Z	China	466921402237651	123470312	789049921
2015-04-15T17:39:31Z	Germany	466921602343040	567828491	789051039
2015-04-15T17:39:31Z	US	466922201102759	678900052	456752187

Count of incoming calls by region: Tumbling window with aggregation

To compare the amount that incoming calls per region we'll leverage a TumblingWindow to get the count of incoming calls grouped by SwitchNum every 5 seconds.

1. Change the query in the code editor to:

```
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. See "Arrival Time Vs Application Time" in the Stream Analytics Query Language Reference.

   Note that you can access a timestamp for the end of each window by using the **System.Timestamp** property.

2. Click **Rerun** under the query editor to see the results of the query.

Output

WINDOWEND	SWITCHNUM	CALLCOUNT
2015-04-14T01:20:05.000Z	Australia	1
2015-04-14T01:20:05.000Z	China	1
2015-04-14T01:20:05.000Z	Germany	1
2015-04-14T01:20:05.000Z	UK	2
2015-04-14T01:20:10.000Z	China	1

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.

1. Change the query in the code editor to:

```
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
```

2. Click **Rerun** under the query editor to see the results of the query.


Output

TIME	CALLINGIMSI	CALLINGNUM1	CALLINGNUM2	SWITCH1	SWITCH2
2015-04-15T18:05:58.000Z	466921602131264	234560397	678957514	US	Australia
2015-04-15T17:40:59.000Z	466922200432822	678933469	789097907	Australia	Germany
2015-04-15T17:40:58.000Z	466921602343040	234523973	345692262	US	UK
2015-04-15T17:40:58.000Z	466920401237309	234556280	678963520	US	Australia
2015-04-15T17:40:56.000Z	466920401237309	234556280	345669096	US	Australia

Create output sink

Now that we have defined an event stream, an Event Hub input to ingest events, and a query to perform a transformation over the stream, the last step is to define an output sink for the job. We'll write events for fraudulent behavior to Blob storage.

Follow the steps below to create a container for Blob storage if you don't already have one.

1. Use an existing storage account or create a new storage account by clicking **NEW** > **DATA SERVICES** > **STORAGE** > **QUICK CREATE** and following the instructions.
2. Select the storage account, click **CONTAINERS** at the top of the page, and then click **ADD**.
3. Specify a **NAME** for your container and set its **ACCESS** to Public Blob.

Specify job output

1. In your Stream Analytics job click **OUTPUT** from the top of the page, and then click **ADD OUTPUT**. The dialog box that opens will walk you through a number of steps to set up your output.
2. Select **BLOB STORAGE**, and then click the right button.
3. Type or select the following values on the third page:
   - **OUTPUT ALIAS**: Enter a friendly name for this job output.

- **SUBSCRIPTION**: If the Blob storage you created is in the same subscription as the Stream Analytics job, select **Use Storage Account from Current Subscription**. If your storage is in a different subscription, select **Use Storage Account from Another Subscription** and manually enter information for **STORAGE ACCOUNT**, **STORAGE ACCOUNT KEY**, **CONTAINER**.
- **STORAGE ACCOUNT**: Select the name of the storage account.
- **CONTAINER**: Select the name of the container.
- **FILENAME PREFIX**: Type in a file prefix to use when writing blob output.

4. Click the right button.
5. Specify the following values:
   - **EVENT SERIALIZER FORMAT**: JSON
   - **ENCODING**: UTF8
6. Click the check button to add this source and to verify that Stream Analytics can successfully connect to the storage account.

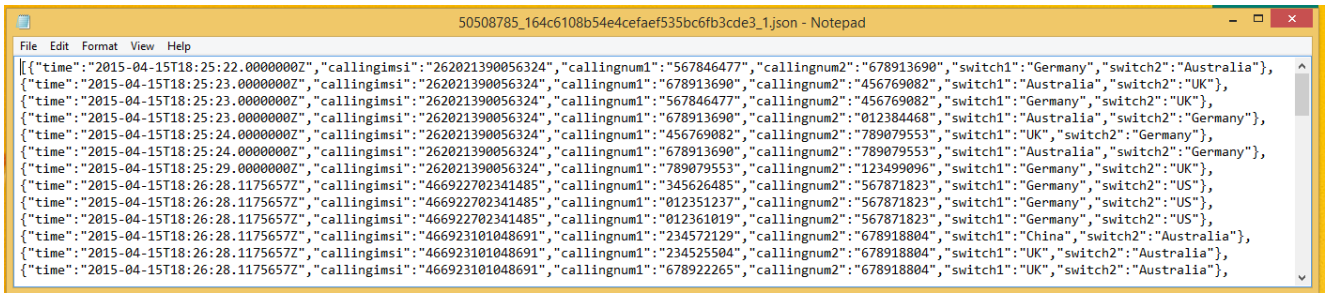
Start job for real time processing

Since a job input, query, and output have all been specified, we are ready to start the Stream Analytics job for real-time fraud detection.

1. From the job **DASHBOARD**, click **START** at the bottom of the page.
2. 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**.

View fraud detection output

Use a tool like Azure Storage Explorer or Azure Explorer to view fraudulent events as they are written to your output in real-time.



Get support

For further assistance, try our Azure Stream Analytics forum.

Next steps

- Introduction to Azure Stream Analytics
- Get started using Azure Stream Analytics
- Scale Azure Stream Analytics jobs
- Azure Stream Analytics Query Language Reference
- Azure Stream Analytics Management REST API Reference

Need help?

Go to an MSDN forum ▶

StackOverflow discussion ▶

• Reply • Share ›

santoshb(msft) • 3 days ago

Hi ,Thanks for remind, however; there aren't any message on dashboard until that data is really start working(in this mean time, my database will have data input), so I didn't think it's just dashboard update delay..It's just like whole data delayed but I don't know the problem is on event hub or ASA...

• Reply • Share ›

Bayan Alghuraybi • 2 months ago

In the Configure and start event generator application step

by doing the Download the TelcoGenerator solution from
https://github.com/Azure/azure...
Do you mean downloading in our local system?

2- wghat does this mean Build the solution to trigger the download of required nuget packages?

• Reply • Share ›

Jocelyn • 2 months ago

I'm having the same problem - I'm creating events and sending them. There is no error. But I don't see these in the eventhub dashboard until the next day and in the stream analytics input I've set up it comes up with "no events found". Perhaps I need to wait 24hours for these to appear here. Also I set the Subscription to "Current subscription" and it all fills in with the details of my event hub fine. But then i save and reopen and its changed the subscription to "another subscription" - I cant seem to get "current subscription" to stick.

• Reply • Share ›

santosh Jocelyn • 2 months ago

Hi Jocelyn,
The data from the event generator goes to Event Hub immediately (no delays), there is a maximum of 10 min delay in the dashboard in Event Hub getting updated.
Questions on Stream analytics -
What are you doing to see the message 'no events found'?
Did you start the job with the event generator running for getting real time results in the blob output?
Or when you started the stream analytics job, did you point the job start time to the time that you uploaded the events to event hub?

btw. as a test please do try the end to end with your event generator pushing data to event hub, and the stream analytics job running at the same time.

see more

• Reply • Share ›

Eduardo Rosas Osorno • 4 months ago

I get an error when trying to get data from the event hub "No events where found for the specified time range" but in my event hub I clearly see messabes being succesfully received. Can you please help me with this? I haven't been able to identify the problem.

• Reply • Share ›

Janet Yeilding [janety@MSFT] Eduardo Rosas Osorno • 4 months ago

Hi Eduardo, would you be more specific where this error message is surfaced? Is this in the client app or somewhere in the Azure portal? Additionally, the best way to get support from the ASA team is via our forum- would you create a post there? https://social.msdn.microsoft....

• Reply • Share ›

Ajay Bawa Janet Yeilding [janety@MSFT] • 4 months ago

I am facing the same error i.e. "No events were found for the specified time range". Eduardo did you find a solution to this issue?

• Reply • Share ›

Arne Brun • 7 months ago

Why do I have to download sample data? Why can't I just run a test query directly. There's a lot of unnecessary steps in the testing procedures. It would be much easier if I could click Test and then select the data input directly, instead of downloading first and then uploading.

• Reply • Share ›

Ziv Kasperski [zivk@MSFT] Arne Brun • 7 months ago

Hi Arne,
Thank you for your feedback.
We are currently working on a number of updates to our service based on customer feedback we have received since our GA and we will definitely take this feedback into consideration as we move forward.
Thanks!
Ziv.

• Reply • Share ›

Carla Sabotta • 8 months ago

When I run the queries shown in the article, am I seeing data stored in the Telco.json file, or am I seeing data that client application that is being sent to the event hub?

• Reply • Share ›

Janet Yeilding [janety@MSFT] Carla Sabotta • 8 months ago

Hi Carla, it depends on how you are running the queries. If you are using the "Test" functionality in the browser, then the query is just being executed over a local JSON file. If you have started a job then the query will run over the events

in the input you configured. In the scope of this tutorial, the input is your Event Hub and the client application is sending events to it.

Is it safe to assume that you have resolved the previous two issue you posted about? If not, please let me know.

• Reply • Share ›

Carla Sabotta ➚ Janet Yeilding [janety@MSFT] • 8 months ago

Thanks Janet. At what point in the above tutorial is the query running over the events in the input? Or in this tutorial, am I always just executing over a local JSON file?

• Reply • Share ›

Janet Yeilding [janety@MSFT] ➚ Carla Sabotta • 8 months ago

In this tutorial you are testing over a local JSON file until the section titled Start Job.

• Reply • Share ›

Carla Sabotta ➚ Janet Yeilding [janety@MSFT] • 8 months ago

Does the Telco generator app need to be running while you're running the steam analytics job? I had assumed that when I ran the app yesterday, the data would be stored in the event hub; so when I ran the job again today I could access the data. However, the Azure Explorer (connected to my storage acct), shows no data.

see more

• Reply • Share ›

Janet Yeilding [janety@MSFT] ➚ Carla Sabotta • 8 months ago

When you start the job you will be prompted to specify a time to start consuming events from. If the Telco generator app is currently running, pick Job Start Time. That's the easiest route and it's what the instructions in this tutorial guide you to do. If you want you consume historical data as you described, pick Custom Time and

specify a timeframe in the past that corresponds to when you were running the generator.

For continued support with ASA, please leverage our forum: https://social.msdn.microsoft....

see more

• Reply • Share ›

Carla Sabotta • 8 months ago

I get the following error message when I try to open the solution in Visual Studio 2010 -- "This project is incompatible with the current version of Visual Studio" . I also tried opening it using Visual STudio 2012

• Reply • Share ›

Carla Sabotta • 8 months ago

I'm not able to download the generator. There appears to be no download command on the page linked to from the tutorial. Also, following the instructions in the tutorial, I'm not able to access Connection Information. When I click the Connection Information link, the only connection string provided is "SAS".

• Reply • Share ›

jkth • 9 months ago

I encountered a problem running thru the tutorial when specifying the job output. Upon clicking the check mark to create the output the error I got was:
"The JSON provided in the request body is invalid. Property 'server' value 'jt' is not acceptable. Activity Id: '35decfe7-91e4-4225-a96e-9f0faeef6408-2015-04-22 20:09:18Z'."
This is very strange because I didn't actually enter a value for the server, it was selected for me when I chose the database (from the same subscription_ that I wanted to insert into.

Figuring there was a bug somewhere I went through the job output wizard again but this time I chose to "Use SQL Database from another subscription" as this allowed me to enter the fully-qualified name of the server which I duly entered ("jt.database.windows.net"), plus all the rest of the required information (here's a screenshot). Upon hitting the check mark I got the same error:

"The JSON provided in the request body is invalid. Property 'server' value 'jt.database.windows.net' is not acceptable. Activity Id: 'a3b3143d-0a81-42dd-b793-73ca00bb670d-2015-04-22 20:18:34Z'."

see more

David Dasher • a year ago

Hello

I've setup a small test using one of our event hubs which captures application telemetry data from our software. Input events seems to be recording correctly, however output events are always 0 and no data is being recorded.

For the output we have tried:-

SQL Azure database and Blob Storage

We have tested the query and the results look fine.

Can anybody help? All connections are showing as ok.

Nabarun ↗ David Dasher • 10 months ago

I am not sure if it got resolved, I landed up in same situation. Can you check the operation logs?

jeffki • a year ago

I get nothing but a stream of exceptions when I try this app.

Jonathan Gao [jgao@MSFT] ↗ jeffki • a year ago

Can you share the details of the exceptions?

Hans Geurtsen • a year ago

Tried to follow this entire getting started tutorial (multiple times), but failed to load data into the SQL database.

Everything seems fine, but the initial (historical) load from the event hub, doesn't seem to do anything. There is no receive events in the operations log. The EventHub dashboard shows the reception of 200 messages. A subsequen run of BasicEventHubSample gives a failure on the event hub creation with a conflict error (which is strange, as the CreateEventHubIfNotExistsAsync method is being called), but events seem to be received correctly, according to the EventHub dashboard. At that time, the Streaming Analytics job gives an error on the Receive Events operation in the operations log: "A receiver with epoch '1' already exists. A new receiver with epoch null cannot be created." and the message "The Event Hub is configured to only allow one event receiver at a given time and an existing receiver is already connected to this Event Hub".

Am I doing something wrong, is there an error in this Getting Started tutorial, or am I running into a bug in the Stream Analytics service?

see more

• Reply • Share ›

Janet Yeilding [janety@MSFT] Hans Geurtsen • a year ago

Hi Hans, the error is likely caused because the Stream Analytics job and the client application that created the Event Hub are both using the Default Consumer Group to read from the Event Hub. Without additional configuration, there can only be one receiver at a time (this is detailed as a limitation here: http://azure.microsoft.com/en-... under the title "Jobs use Default Consumer Group for Event Hub"). Try stopping the client code and restarting the Stream Analytics job. If that doesn't resolve the issue, please let us know via the team forum: https://social.msdn.microsoft....

• Reply • Share ›

Leonid Ganeline Janet Yeilding [janety@MSFT] • a year ago

Hi Janet,

I'm getting the same error. I've found some explanation of this mythical 'epoch' here http://blogs.msdn.com/b/gyan/a.... Why don't include this information in the Documentation???

In my case the Stream Analytics receives events from EventHub and output results in the same EventHub. Client sends events and receive them with EventHubHost.
When I restart the client, the events from SA received by this client. And they only received in this time, when I restart the client. Maybe SA waits something when client stopped.
Everything looks very unpredictable...

see more

• Reply • Share ›

Janet Yeilding [janety@MSFT] Leonid Ganeline • a year ago

Hi Leonid, sorry that you ran into this issue. We recently pushed out an update to Stream Analytics that should help. You can now associate a Stream Analytics job with its own Consumer Group, enabling your job to have its own view over the input stream, independent of other event readers. A Consumer Group is specified via the optional property consumerGroupName in the Create Input request and will be surfaced in the portal in a future update.

see more

• Reply • Share ›

uday kiran • a year ago

Guys, this article is pretty good, but it would be great if you can post the example talked in the video explaining streaming analytics where sentiment analysis was done on twitter content.

2 • Reply • Share ›

Janet Yeilding [janety@MSFT] uday kiran • a year ago

Thanks for the feedback, uday. Additional samples and walkthroughs are coming soon.

• Reply • Share ›

Janet Yeilding [janety@MSFT] Janet Yeilding [janety@MSFT] • 9 months ago

Uday, check out this sample for Twitter sentiment analysis that was published today: http://blogs.msdn.com/b/stream...

• Reply • Share ›

Go Social

Microsoft Azure +

Community +

Support +

Account +

Trust Center +

Hello from Seattle.

English (US)

Euro (€)

Contact Us Feedback Trademarks Privacy & Cookies

Microsoft