

Stream Analytics - *Fraud Detection*
From Phone Call Data with Azure
Event Hubs, Azure Stream Analytics
and Power BI

Things to note!!!

- **Project Focus:** To discuss in detail the key components involved in real time streaming process in Azure using Azure Event Hubs and Azure Stream Analytics while solving the problem of fraud detection.
- **Scenario:** The phone call data, generated by a client application, contains fraudulent calls, which are detected by the Stream Analytics job.
- **Solution:** Process a continuous flow of telemetry data seamlessly in real time utilizing a serverless architecture.

Requirements for this project

- Azure subscription
- TelcoGenerator - Call event generator app from Microsoft
- Visual Studio – interphase to edit the code
- Power BI Account – Premium or Pro Subscription

Key Components of the Project:

- Azure Event Hubs
- Azure Stream Analytics
- Power BI
- Shared Access Signatures
- Azure Event Bus
- Azure Service Grid
- Zone Redundancy
- User Token
- Managed Identity
- Partition Count
- Retention Time
- Window Functions

STEP 1: Creation of Event Hubs Namespace & Resource Group

- Sign into your Azure portal & create an Event Hubs Namespace
- Create a new resource group (if you don't want to use an existing one) to house the Event Hub Namespace
- **Event Hubs Namespace:** is a management container for event hubs. The namespace allows for effective management of event hubs with related artifacts within the Azure environment.

Home > Create a resource > Marketplace > Event Hubs >

Create Namespace

Event Hubs

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription * Azure subscription 1

Resource group * (New) stellastream
[Create new](#)

Instance Details

Enter required settings for this namespace, including a price tier and configuring the number of units (capacity).

Namespace name * stellastreamhub ✓
.servicebus.windows.net

Location * East US
The region selected supports Availability zones. Your namespace will have Availability Zones enabled. [Learn more.](#)

Pricing tier * Standard (~\$22 USD per TU per Month)
[Browse the available plans and their features](#)

Throughput Units * 1

Enable Auto-Inflate ⓘ ☐

[Review + create](#) < Previous Next: Advanced >

Create Namespace

Event Hubs

✓ Validation succeeded.

Event Hubs Namespace by Microsoft

Basics

Namespace name	stellastreamhub
Subscription	Azure subscription 1
Resource group	stellastream
Location	East US
Pricing tier	Standard
Throughput Units	1
Availability Zones (Zone Redundancy)	Enabled
Auto-Inflate Maximum Throughput Units	Disabled

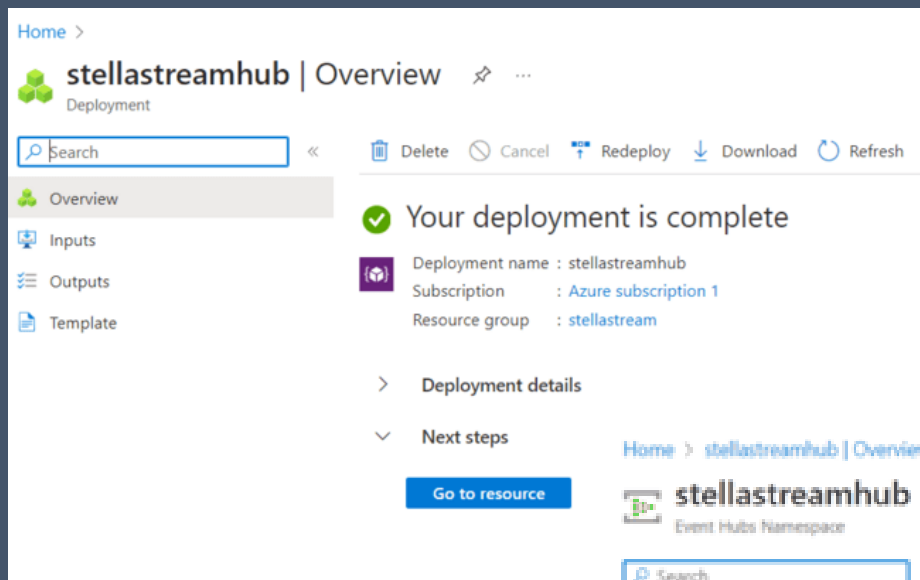
Networking

Connectivity method	Public access
---------------------	---------------

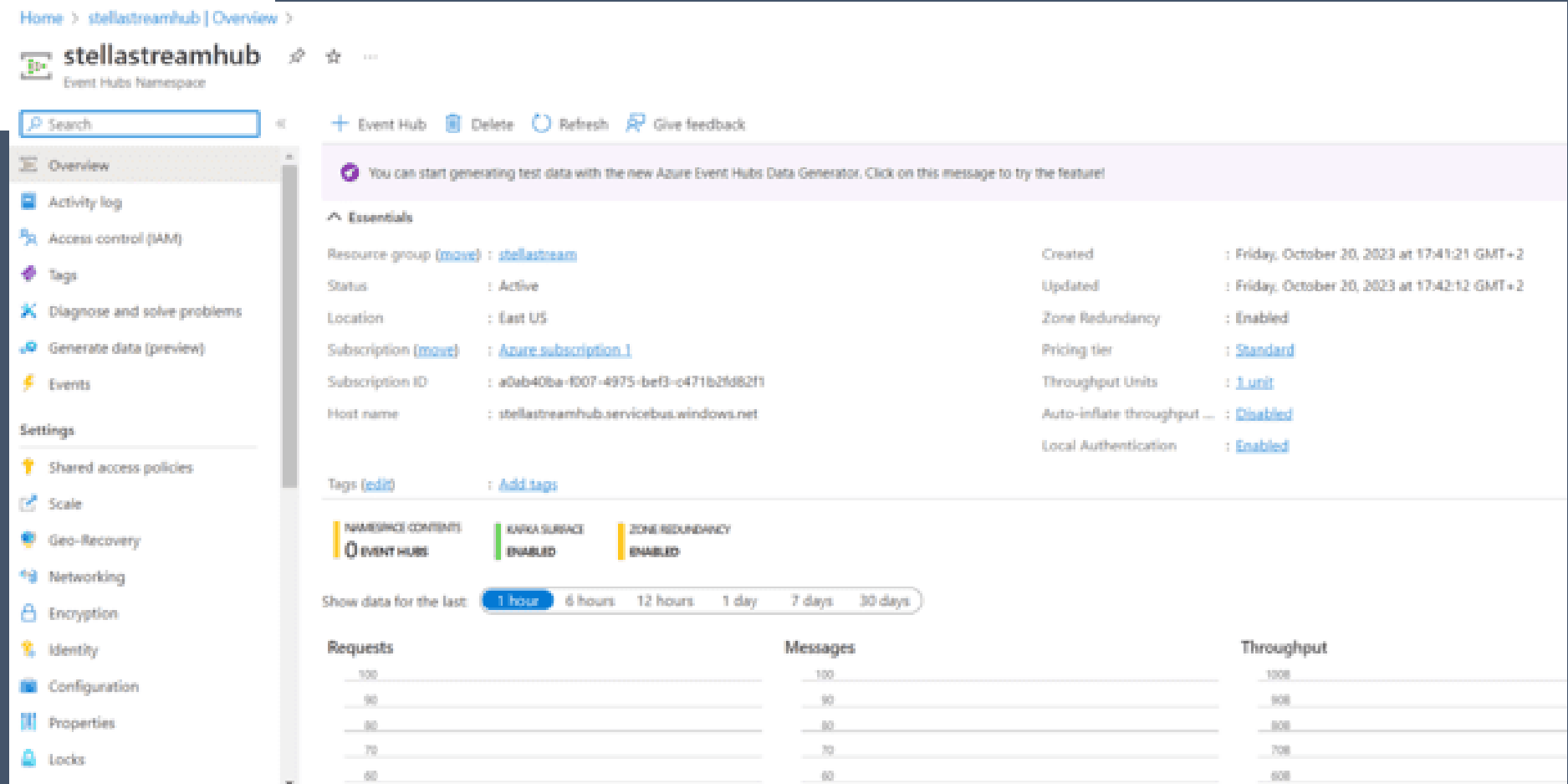
Security

Minimum TLS version	1.2
Local Authentication	Enabled

The event hubs namespace for this project is "stellastreamhub".



Below is an overview of the just created event hubs namespace **"stellastreamhub"** within the **"stellastream"** resource group. The namespace contents indicate that no event hubs has been created yet!



Also worthy to note is the **'Zone Redundancy'** which is set to 'Enabled' and therefore means that data for the East US region will be replicated synchronously across 3 Azure availability zones. In case of data outage in the primary data center, we can switch to the next available data center within the region

Diagram showing an Event Hubs Namespace housing a few event hubs

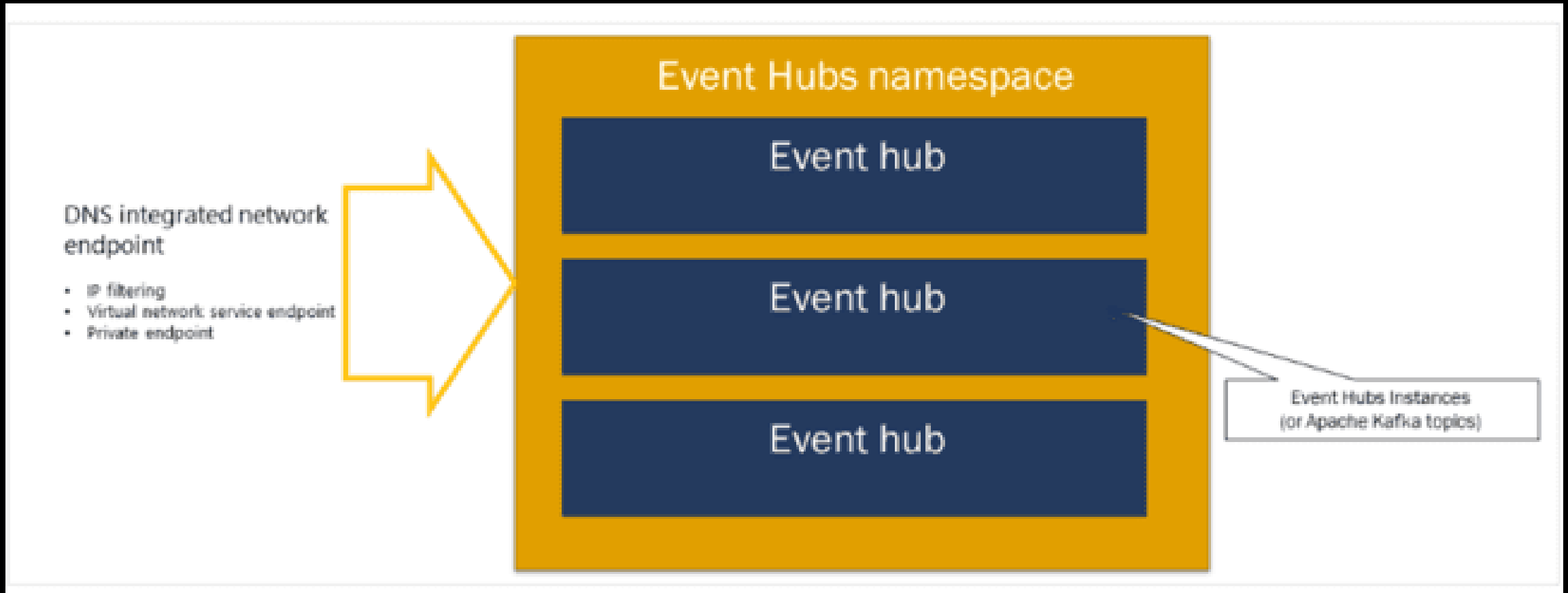


Image Credit: Microsoft

Step 2: Creation of the Event Hub 'stellastreamhub'

Home > stellastreamhub | Overview > stellastreamhub >

Create Event Hub ...


Event Hubs

Basics Capture Review + create

Event Hub Details

Enter required settings for this event hub, including partition count and message retention.

Name * ⓘ StellaEventHub ✓

Partition count ⓘ  2

Retention

Configure retention settings for this Event Hub. [Learn more](#)

Cleanup policy ⓘ Delete ▼

Retention time (hrs) * ⓘ 1

min. 1 hour, max. 168 hours (7days)

Partition Count 'PC'

Partition counts are ordered logs of events used to facilitate the parallel processing of data and events in event hubs. As event hubs are subject to high influx of streaming data, the partition counts offer horizontal scalability by distributing data across multiple partitions for efficient data ingestion and processing.

Partition Counts also allow for independent and parallel processing of real time event streams making it easier for multiple consumers to read the data simultaneously. They also help to prevent bottlenecks by ensuring that the system can handle high data traffic by employing effective load balancing techniques without compromising on performance.

Step 2 contd

Partition Counts:

- A few points to consider when determining the number of partition counts to allocate for a streaming event are
 - i. **The volume of the incoming data** and the rate at which it needs to be ingested and processed
 - ii. **Scalability Requirements:** Ensure the selected partition count can support future scalability and evenly distribute the processing load in the event of growing workloads and sudden spikes in events.
 - iii. **Consumers:** The number of consumers that would be reading data from the partitions simultaneously
 - iv. **Cost:** Although partitioning enhances the performance of the event hubs, it also impacts the cost. Cost therefore should be put into consideration whilst also evaluating the trade-offs with scalability and performance

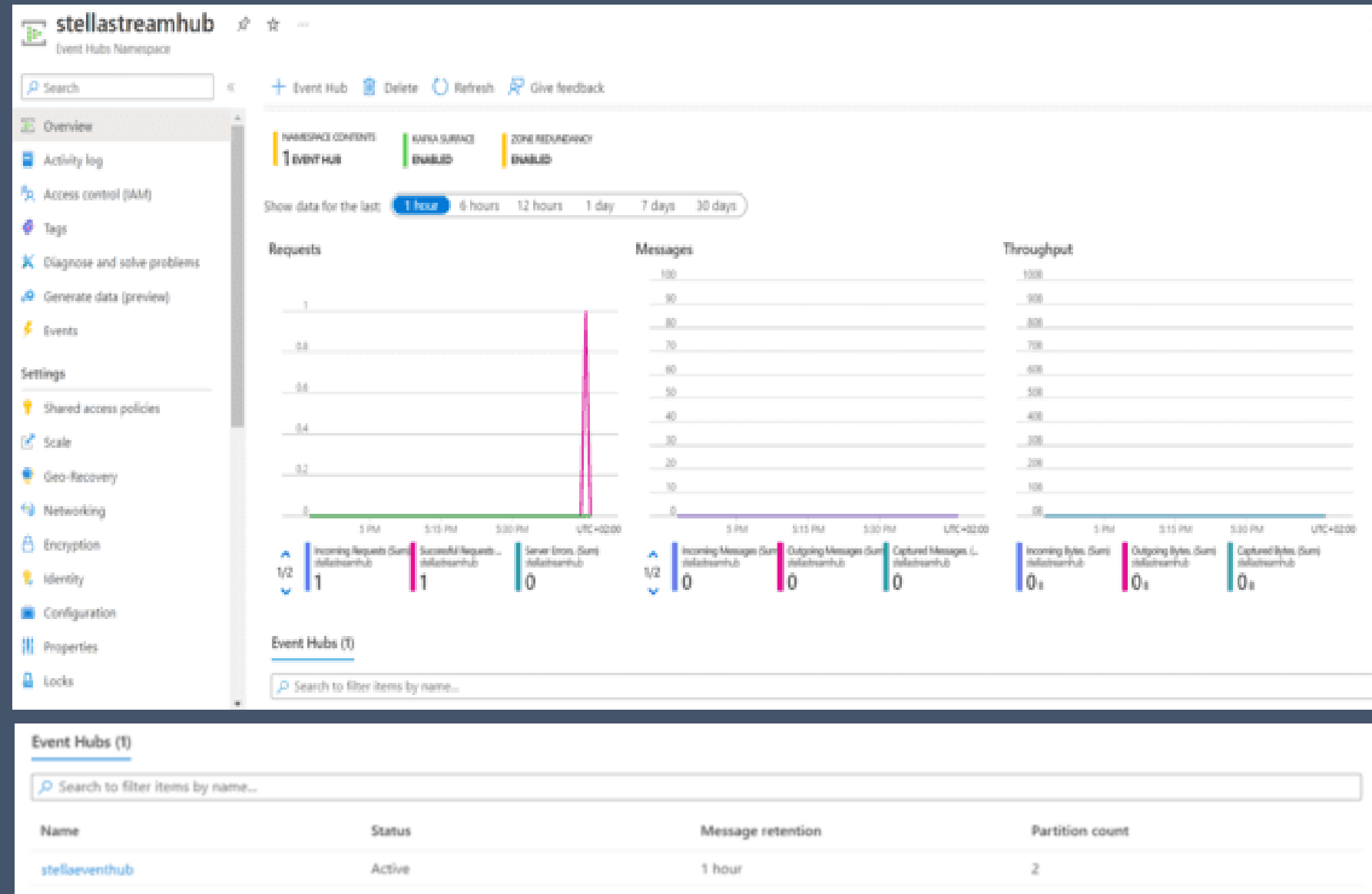
*For this project, the **partition count was set to '2'** meaning there were 2 partitions running simultaneously to ingest data and enhance the performance of the whole process. This is a sample project, and the cost of the partition count was the most important factor considered...lol!*

Step 2 contd

Retention Period

This is the specified period within which data is retained and made available for consumption, replay and troubleshooting purposes amongst other things. After the specified retention time, the data is permanently deleted.

The retention period for this project was set to one hour.



Step 3: Shared Access Signature Policies

Home > stellastreamhub | Overview > stellastreamhub

stellastreamhub | Shared access policies ☆ ...
Event Hubs Namespace

Search

ACCESS CONTROL (IAM)

Tags

Diagnose and solve problems

Generate data (preview)

Events

Settings

Shared access policies

Search to filter items...

Policy

RootManageSharedAccessKey

MyPolicy

Shared access policies ☆ ...

Search to filter items...

Policy

RootManageSharedAccessKey

MyPolicy

Claims

Manage

Manage

SAS Policy: MyPolicy

Save Discard Delete Regenerate Primary Key ...

☒ Manage

☒ Send

☒ Listen

Primary key

pRhNvdXzK0AcJT06@1ZK/qxkKy/PMATbo+AEhOvDm+c=

Secondary key

IDHixj2InquhvhmQbcKWF+lwSTNzqC+AEhOCdn4g=

Copy Copied

Connection string-primary key

Endpoint=sb://stellastreamhub.servicebus.windows.net/;SharedAccessKeyName=MyPol...

Connection string-secondary key

Endpoint=sb://stellastreamhub.servicebus.windows.net/;SharedAccessKeyName=MyPol...

SAS Policy ARM ID

/subscriptions/a0ab40ba-f007-4975-bef3-c471b2fd82f1/resourcegroups/stellastream/p...

In Azure Event Hubs, the **Shared Access Signature Policy** is a security feature that allows you to control and manage access to the Event Hub entities. It works by defining a set of permissions and constraints for a specific application or user identity, enabling them to perform specific actions on the Event Hubs. These policies are associated with an Azure Active Directory identity or a shared access key.

The **Shared Access Signature Policy** consists of a **connection string-primary key** that facilitates the connection between the event hub and event generator.

Step 3 contd – How Shared Access Signature Policies Work

- **Creation of Policies:** You can create multiple shared access policies for an Event Hub namespace or a specific Event Hub. Each policy can have its own set of permissions/ authorisation rules, such as sending, receiving, managing, or listening to events.
- **Permission Management:** Once created, the shared access policies can be assigned specific permissions (send, receive or manage or all) that define what actions the application or user can perform. This also aids the regulation of access levels.
- **Access Key Management:** Shared access policies are associated with a primary and secondary access key, which are used to authenticate and authorize the application or user to interact securely with the Event Hub.
- **Expiration and Revocation:** Shared access policies can be configured with an expiration date, providing an added layer of security by ensuring that the access rights are automatically revoked after a specified period which helps to prevent unauthorized access over an extended period.

****Shared Access Policy Permissions****

- With the **"Send"** permission, authorized users can send messages to a queue within the Service Bus for further processing or distribution to other connected services.
- The **"Receive"** permission enables users to receive and process messages from a queue or a subscription for processing and handling based on specific business logic.
- The **"Manage"** permission provides users with the ability to manage the Service Bus entities, including creating or deleting queues, topics, subscriptions, and other related resources. Users with this permission can perform administrative tasks such as configuring settings, setting up filters, or adjusting access control rules for the Service Bus entities.

For this project, I created an SAP called 'MyPolicy' with the permissions - managing, sending, and receiving.

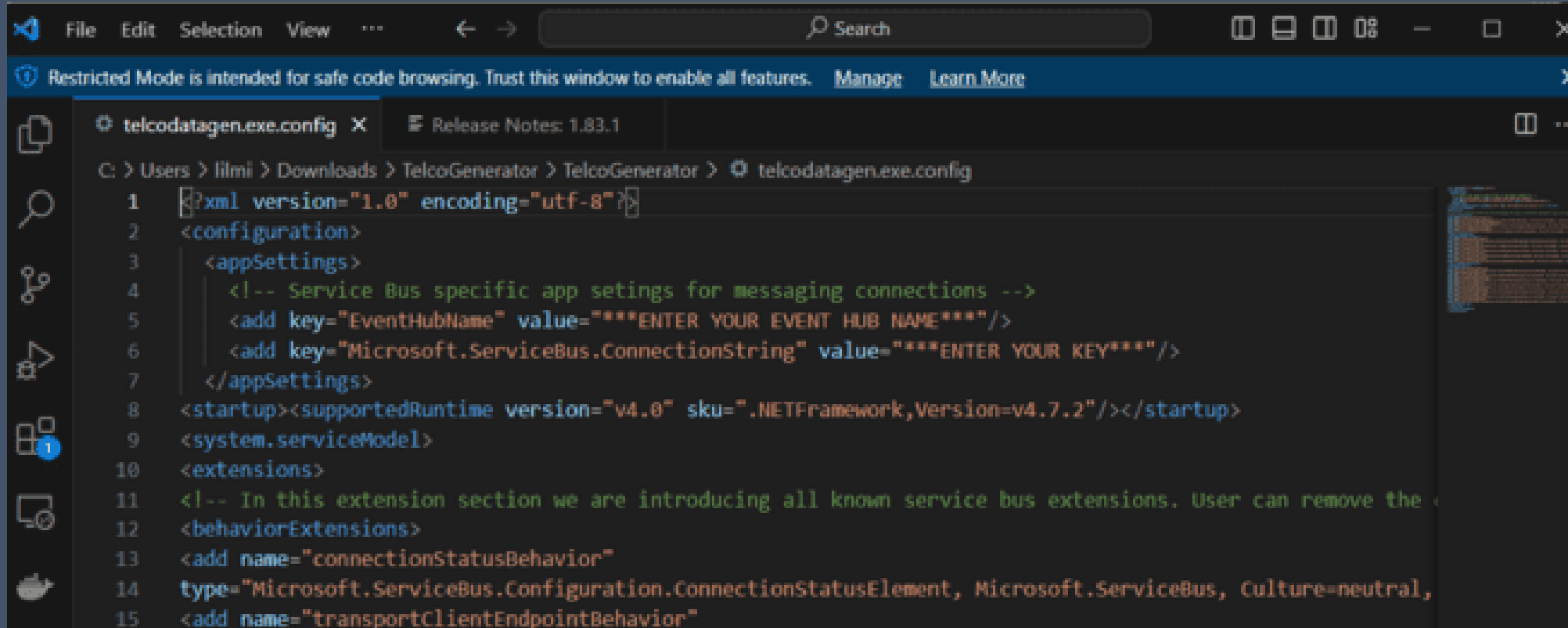
Step 4: Generating Data – TelcoGenerator & Visual Studio

Name	Date modified	Type	Size
▼ Today			
telcodatagen.pdb	20/10/2023 18:00	PDB File	38 KB
TelcoGenerator	20/10/2023 18:00	File folder	
telcodatagen.exe	20/10/2023 18:00	CONFIG File	
telcodatagen	20/10/2023 18:00	Application	
Newtonsoft.Json.xml	20/10/2023 18:00	Microsoft Edge HT...	
(Show all 12)			
▼ Earlier this week			
TelcoGenerator	16/10/2023 10:47	Compressed (zipp...	

TelcoGenerator				
← → ↑ ↺ > Downloads > TelcoGenerator > TelcoGenerator				
+ New ✂ 📄 📁 📄 📄 🗑️ ⬆️ Sort ▾ ≡ View ▾ ⋮				
> Adedotun - Pers				
▼ Today				
Newtonsoft.Json.dll	20/10/2023 18:00	Application extens...	502 KB	
Newtonsoft.Json.xml	20/10/2023 18:00	XML File	483 KB	
telcodatagen	20/10/2023 18:00	Application	30 KB	
telcodatagen.exe	20/10/2023 18:00	Configuration Sou...	4 KB	
telcodatagen.pdb	20/10/2023 18:00	PDB File	38 KB	
LICENSE	20/10/2023 18:00	Text Document	2 KB	
Microsoft.ServiceBus.dll	20/10/2023 18:00	Application extens...	3,660 KB	
Microsoft.ServiceBus.xml	20/10/2023 18:00	XML File	1,408 KB	
Microsoft.WindowsAzure.Configuration.dll	20/10/2023 18:00	Application extens...	19 KB	
10 items 1 item selected 3.50 KB				

The **TelcoGenerator** is an App that generates phone call streams streaming. It can either be downloaded on a computer or the source code copied from Github

Step 4 contd: Visual Studio & Microsoft Service Bus



```
1 <?xml version="1.0" encoding="utf-8"?>
2 <configuration>
3   <appSettings>
4     <!-- Service Bus specific app settings for messaging connections -->
5     <add key="EventHubName" value="***ENTER YOUR EVENT HUB NAME***/>
6     <add key="Microsoft.ServiceBus.ConnectionString" value="***ENTER YOUR KEY***/>
7   </appSettings>
8 <startup><supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.7.2"/></startup>
9 <system.serviceModel>
10 <extensions>
11 <!-- In this extension section we are introducing all known service bus extensions. User can remove the
12 <behaviorExtensions>
13 <add name="connectionStatusBehavior"
14 type="Microsoft.ServiceBus.Configuration.ConnectionStatusElement, Microsoft.ServiceBus, Culture=neutral,
15 <add name="transportClientEndpointBehavior"
```

The **Visual studio** provides an interphase by which we can input the connection string values. This information enables the **Service Bus** to transport information from the TelcoGenerator to the Event hub.

Connection String Sample: Endpoint=sb://stellastreamhub.servicebus.windows.net/;SharedAccessKeyName=MyPolicy;SharedAccessKey=pRhNvdXzK0AcJT06B1ZK/qxEky/PMATbo+AEhOvDm+c=

Step 4: Event Bus

```
<configuration>
  <appSettings>
    <!-- Service Bus specific app settings for messaging connections -->
    <add key="EventHubName" value="stellaeventhub"/>
    <add key="Microsoft.ServiceBus.ConnectionString" value="Endpoint=sb://stellastreamhub.servicebus.windows.net/>
  </appSettings>
  <startup><supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.7.2"/></startup>
  <system.serviceModel>
```

*In Azure, an **event bus** refers to a communication channel or messaging infrastructure that facilitates the transfer of event data between different applications or services within the Azure ecosystem. It enables various components of a system to communicate and interact in a decoupled manner.*

*Examples of event buses provided by Azure are **Azure Service Bus** and **Azure Event Grid**.*

Azure Service Bus is a fully managed enterprise integration message broker that enables communication between applications and services. It supports both cloud and on-premises solutions and provides a reliable platform for independent transfer of data between different components of a distributed system.

On the other hand, **Azure Event Grid** is a fully managed event routing service that enables the seamless integration of various Azure services and third-party applications, allowing them to react to events and take necessary actions in real time.

*For this project, **Azure** deployed "**Azure Service Bus**" and this is because TelcoGenerator is a Microsoft product. If it were to be a third-party product, then Azure event grid would have been deployed as the means of transport.*

Sample of PhoneStreams generated by TelcoGenerator

```
Command Prompt x Windows PowerShell x + v
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\lilmi> cd "C:\Users\lilmi\Downloads\TelcoGenerator\TelcoGenerator"
PS C:\Users\lilmi\Downloads\TelcoGenerator\TelcoGenerator>
> .\telcodatagen.exe 1000 0.2 2
#Sets: 1,#FilesDump: 1,#CDRPerFile: 1000,%CallBack: 0.2, #DurationHours: 2
Time Increment Per Set: 2
20231020 183412
MO,d0,0,Australia,456722620,466921200135361,789066693,466923200779222,20231020,183412,0,0,,,b,1,,4,,425,141298312187
7,,,20231020 183412
MO,d0,2,Germany,234566001,466923000886460,567858330,466921302209862,20231020,183424,1,0,,,V,1,,0,,421,886932428306,,
,20231020 183424
MO,d0,4,UK,456776271,466923000464324,234520325,466922702346260,20231020,183426,2,0,,,a,1,,0,,425,886932429155,,,2023
1020 183426
MO,d0,6,Australia,234552476,466923101048691,345613685,466923000464324,20231020,183427,3,299,,,b,1,,4,,623,1416955584
542,,,20231020 183427
MO,d0,7,China,,466922202679249,,466922000696024,F,F,1,,,V,1,,3,,411,886932429155,,,F F
MO,d0,0,UK,234539056,466921200135361,456722620,466923300236137,20231020,190112,3,321,,,V,1,,3,,420,1412983121877,,,2
0231020 190112
MO,d0,8,China,456748353,466921602131264,567818869,466923200408045,20231020,183429,0,0,,,a,1,,0,,422,886932429626,,,2
0231020 183429
MO,d0,10,Australia,345681743,466922202546859,789060425,466921602343040,20231020,183430,1,0,,,S,1,,0,,422,14169555845
42,,,20231020 183430
MO,d0,12,China,345635507,466922200432822,345686347,466923300236137,20231020,183430,2,0,,,b,0,,3,,614,886932429979,,,
20231020 183430
MO,d0,14,China,678946438,466923300507919,234575017,466923100807296,20231020,183430,1,0,,,S,0,,0,,425,886932428688,,,
20231020 183430
```

The phonestreams will be fed into event hubs which will in turn be fed into azure streams analytics

Step 5: Time to Involve Azure Stream Analytics

Create a new stream analytics job
“*stellastreamanalytics*” in the resource
group.

Home > stellastream > Marketplace >

New Stream Analytics job ...

Basics Storage Tags Review + create

[View automation template](#)

Basics

Subscription	Azure subscription 1
Resource group	stellastream
Name	stellastreamanalytics
Location	East US
Hosting environment	Cloud
Streaming units	1

Home >

StreamAnalyticsJob | Overview

Deployment

Search

Delete Cancel Redeploy Download Refresh

Overview

Inputs

Outputs

Template

✓ Your deployment is complete

Deployment name : StreamAnalyticsJob
Subscription : Azure subscription 1
Resource group : stellastream

> Deployment details






✓ Next steps

[Go to resource](#)

Give feedback

[Tell us about your experience with deployment](#)

Step 5 contd

Resources		
<div>Recent</div> <div>Favorite</div>		
Name	Type	Last Viewed
 stellastreamanalytics	Stream Analytics job	4 minutes ago
 stellastream	Resource group	4 minutes ago
 stellastreamhub	Event Hubs Namespace	an hour ago
 Azure subscription 1	Subscription	2 days ago
 Azure for Students Starter	Subscription	3 days ago
See all		

Azure Stream Analytics will be used to process the streaming data and perform aggregate functions before it is been set to the process output function like Power BI.

Input stream (Event Hubs) -> Process stream (Azure Stream Analytics) -> Output stream (Power BI)

Step 5 contd: Overview of Stream Analytics Dashboard

The screenshot displays the Azure Stream Analytics dashboard for a job named "stellastreamanalytics". The interface includes a left-hand navigation pane with sections for "Job topology" (Inputs, Functions, Query, Outputs) and "Settings" (Environment, Storage account settings, Scale, Locale, Event ordering). The main content area shows the job's "Created" state and a table of properties. Below the properties, there are tabs for "Get started", "Properties", "Monitoring", and "Tutorials". A prominent heading reads "Build an end-to-end serverless streaming pipeline with just a few clicks", followed by a descriptive paragraph about Azure Stream Analytics. At the bottom, a "No-code editor" section is visible, featuring a "No-code editor (preview)" button and a brief description of its functionality.

stellastreamanalytics ☆ ...
Stream Analytics job

Search

Start job Delete Move Refresh Share feedback

Created

Essentials

Resource group (move)	: stellastream	Created	: Friday, October 20, 2023 6:39 PM
Location	: East US	Started	:
Status	: Created	Output watermark	:
Subscription (move)	: Azure subscription 1	Cluster	: Shared
Subscription ID	: a0ab40ba-f007-4975-bef3-c471b2d82f1	Hosting environment	: Cloud
Pricing plan	: StandardV2 (manage)	Virtual Network	: Disabled
Tags (edit)	:		

Get started Properties Monitoring Tutorials

Build an end-to-end serverless streaming pipeline with just a few clicks

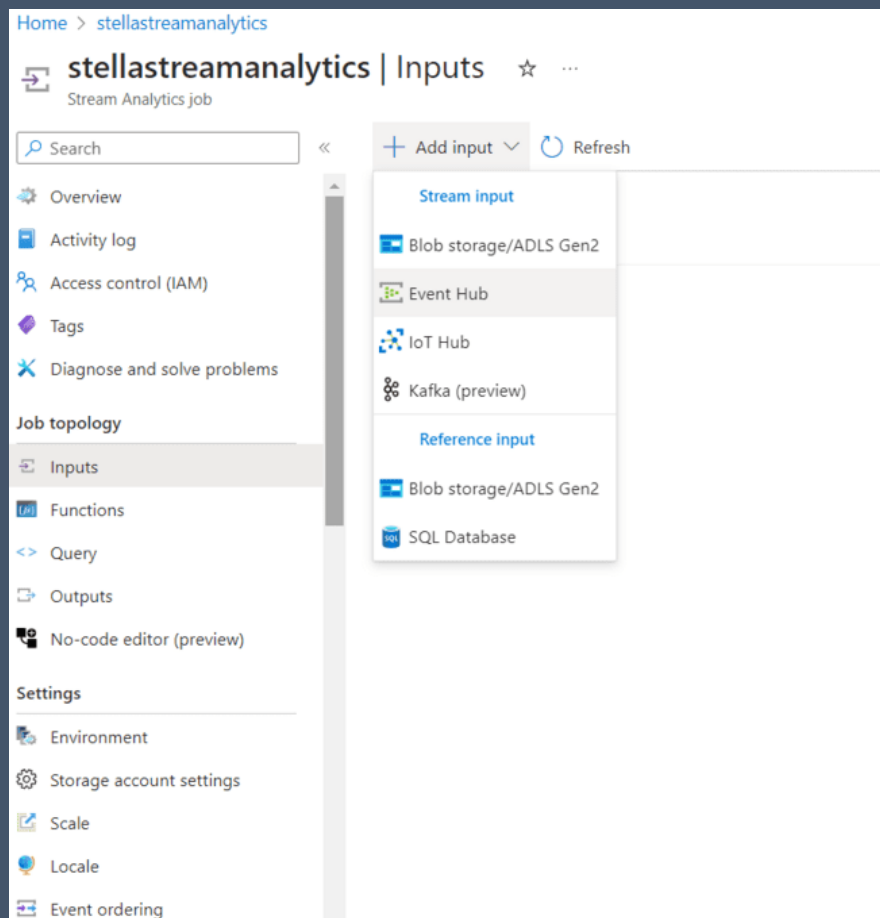
Azure Stream Analytics is a fully managed, real-time stream processing service designed to help you tackle scenarios such as streaming ETL to ADLS Gen2 or Synapse SQL, real time apps with Cosmos DB or SQL DB, live dashboarding with Power BI, or real-time alerting with Azure Functions. [Learn more](#)

No-code editor

No-code editor (preview)
Use no-code editor to build your Stream Analytics job without a single line of code.

Step 5 contd: Configuring ASA

- Specifying the job input channel as Event Hub



Event Hub

New input

Input alias *
Phonestream ✓

☐ Provide Event Hub settings manually
☒ Select Event Hub from your subscriptions

Subscription
Azure subscription 1

Event Hub namespace * ⓘ
stellastreamhub

Event Hub name * ⓘ
☐ Create new ☒ Use existing
stellaeventhub

Event Hub consumer group * ⓘ
☒ Create new ☐ Use existing
stellastreamanalytics_Phonestream_con... ✓

Authentication mode
Connection string

Event Hub

New input

Authentication mode
Connection string

Event Hub policy name * ⓘ
☐ Create new ☒ Use existing
MyPolicy

Event Hub policy key ⓘ
.....

Partition key ⓘ

Event serialization format * ⓘ
JSON

Encoding ⓘ
UTF-8

Event compression type ⓘ
None

Schema registry (preview)
None

Step 5 contd: Configuring ASA

- Job Processing Channel: Azure Stream Analytics

Home >

All resources



Montpellier Business School - Production (montpellier-bs.com)

+ Create Manage view Refresh Export to CSV Open query Assign tags Delete

Filter for any field... Subscription equals all Resource group equals all Type equals all Location equals all Add filter

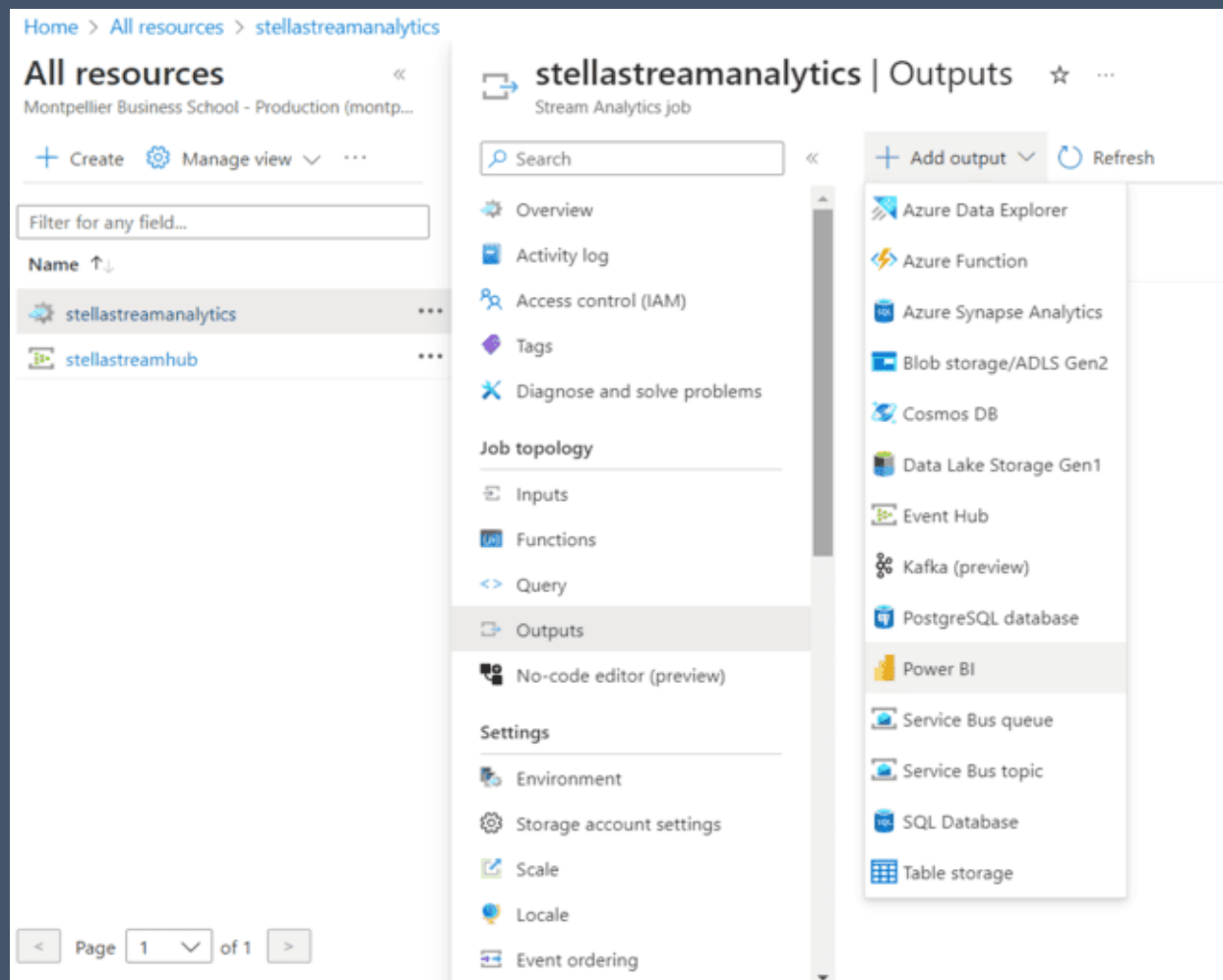
0 Recommendations 0 Unsecure resources

No grouping List view

<input type="checkbox"/> Name ↑	Type ↑	Resource group ↑	Location ↑	Subscription ↑
<input type="checkbox"/>  stellastreamanalytics	Stream Analytics Job	stellastream	East US	Azure subscription 1
<input type="checkbox"/>  stellastreamhub	Event Hubs Namespace	stellastream	East US	Azure subscription 1

Step 5 contd: Configuring ASA

- Specifying the job output channel as Power BI



Power BI

New output

Output alias *

Phonestreamoutput ✓

☐ Provide Power BI settings manually

☒ Select Power BI from your subscriptions

Group workspace *

My workspace

Authentication mode

User token

Dataset name * ⓘ

streamDataset ✓

Table name *

streamTable ✓

Authorize connection

You'll need to authorize with Power BI to configure your output settings.

[Sign up](#)

[Authorize](#)

[Save](#)

The authentication mode selected is "User Token" which is system generated and will expire after a while. Another option which is safer is the "Managed Identity" but will only work with a **Power BI Pro License**.





Overview of Input and Output

stellastreamanalytics | Inputs ☆ ...

Stream Analytics job

Search < + Add input Refresh

Overview
Activity log
Access control (IAM)
Tags
Diagnose and solve problems
Job topology



Alias ↑	Source type	Type	A...	R...
Phonestream	Stream	Event Hub	Connr	   

stellastreamanalytics | Outputs ☆ ...

Stream Analytics job

Search < + Add output Refresh

Overview
Activity log
Access control (IAM)
Tags
Diagnose and solve problems

Alias ↑	Type	Authentication ...	R...
Phonestreamoutput	Power BI	User token	 

Step 6: Test the Query to confirm that the connection is active

The screenshot shows the Azure Stream Analytics Query Editor interface. The top bar displays the job name 'stellastreamanalytics | Query' and a status 'Job ready to start'. The left sidebar shows the job structure with inputs, outputs, and functions. The main area contains a query editor with a test query and an input preview table.

Query Editor:

```
1 /*
2 Here are links to help you get started with Stream Analytics Query Language:
3 Common query patterns - https://go.microsoft.com/fwlink/?linkID=619153
4 Query language - https://docs.microsoft.com/stream-analytics-query/query-language-elements-azure-stream-analytics
5 */
6 SELECT
7     *
8 INTO
9     [Phonestreamoutput]
10 FROM
11     [Phonestream]
```

Input preview:

No data was found for preview from 'Phonestream'. Make sure the input has recently received data and the correct format of those events has been selected.

RecordType	SystemIdentity	FileNum	SwitchNum	CallingNum	CallingMSI	CalledNum	Call
string	string	string	string	string	string	string	string
"MO"	"d0"	"727"	"Australia"	"466717757"	"466923300236137"	"234593576"	"466"
"MO"	"d0"	"729"	"US"	"012365084"	"466923201502759"	"012325089"	"466"

Literal meaning of the query:

*Select all the results from input/Event Hub “**Phonestream**” and send to output/Power BI “**Phonestreamoutput**”*

Step 6 contd: Result of the Query

Home > All resources > stellastreamanalytics

stellastreamanalytics | Query

Stream Analytics job

Start job Query language docs Open in VS Code Diagnostic settings Refresh Share feedback Tutorial Job ready to start

Inputs (1) +
Phonestream

Outputs (1) +
Phonestream

Functions (0) +

Test query Save query Discard changes

```
1  
2 SELECT  
3 *  
4 FROM  
5 [Phonestream]
```

Input preview Test results Job simulation (preview)

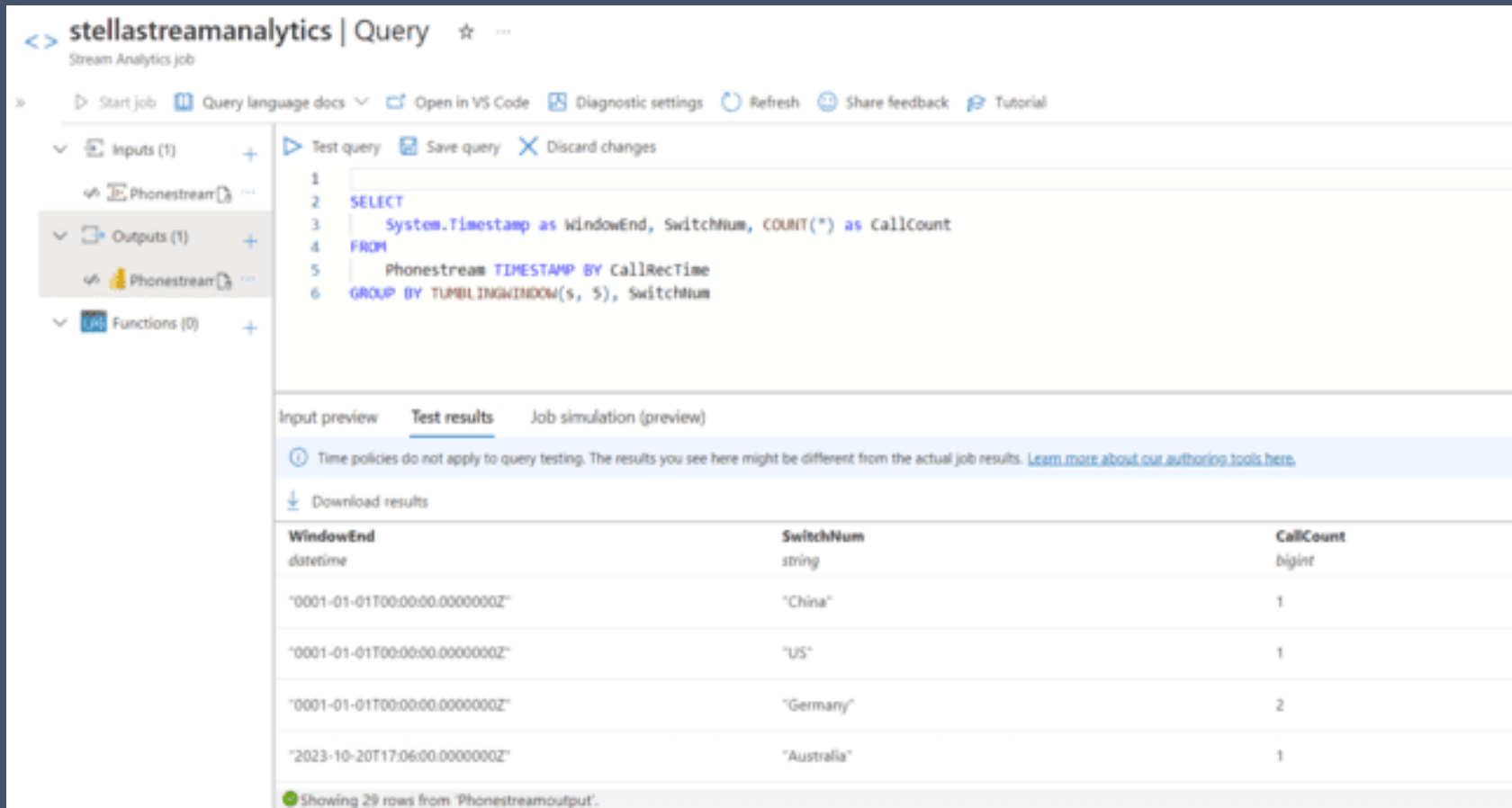
Download results

RecordType	SystemIdentity	FileNum	SwitchNum	CallingNum	CallingIMSI	CalledNum	CalledIMSI
string	string	string	string	string	string	string	string
"MO"	"80"	"727"	"Australia"	"456717757"	"466923300236137"	"234593576"	"466920400352400"
"MO"	"80"	"729"	"US"	"012365084"	"466922201102759"	"012321089"	"466921200135361"
"MO"	"80"	"735"	"Germany"	"567849619"	"466921402416657"	"345659273"	"466922702346260"
"MO"	"80"	"737"	"US"	"678995286"	"466921402237651"	"345681474"	"466922201102759"
"MO"	"80"	"741"	"UK"	"789069946"	"466922000696024"	"456716808"	"466922201102759"
"MO"	"80"	"753"	"Australia"	"234581792"	"466921602131264"	"567852810"	"466922201102759"

Showing 50 rows from 'Phonestreamoutput'.

Ln 5, Col 18

Step 7: Analysis of 2nd Query



The screenshot shows the Azure Stream Analytics Query Editor interface. The query is as follows:

```
1  
2 SELECT  
3   System.Timestamp as WindowEnd, SwitchNum, COUNT(*) as CallCount  
4 FROM  
5   Phonestream TIMESTAMP BY CallRecTime  
6 GROUP BY TUMBLINGWINDOW(s, 5), SwitchNum
```

The results tab shows the following data:

WindowEnd	SwitchNum	CallCount
datetime	string	bigint
"0001-01-01T00:00:00.0000000Z"	"China"	1
"0001-01-01T00:00:00.0000000Z"	"US"	1
"0001-01-01T00:00:00.0000000Z"	"Germany"	2
"2023-10-20T17:06:00.0000000Z"	"Australia"	1

Showing 29 rows from 'Phonestreamoutput'.

System.Timestamp: Every event at every stage of the query in Azure Stream Analytics has a timestamp associated with it.

`System.Timestamp()` is a system property that can be used to retrieve the event's timestamp.

System.Timestamp format =
'yyyy-MM-dd'T'HH:mm:ss.SSSSSS' where 'SSSSSS' is milliseconds

Timestamp By: On the other hand, `Timestamp By` allows us to specify custom timestamp values. We can select a particular timestamp field in the input stream that'll be used to define the tumbling window. If no timestamp field is selected, it automatically records the time the event was received by the Event Hub or IOT hub.

Step 7: Analysis of 2nd Query Contd

- **Tumbling Window:** is one of the *window functions* in azure stream that enables us to perform aggregations such as count or group by on streaming data. Other window functions are session, hopping, sliding and snapshot and all these window functions use the “group by” aggregation.
- According to Microsoft documentation, Tumbling window functions are used to segment a data stream into distinct time segments and perform a function against them. The key differentiators of a Tumbling window are that they repeat, do not overlap, and an event cannot belong to more than one tumbling window.
- *Syntax for tumbling window = Tumbling window (Duration(timeunit, windowsize)).* Where time unit can be days ‘d’, hours ‘h’, seconds ‘ss’, milliseconds ‘ms’ & microseconds ‘mcs’ with 7days being the absolute maximum.
- **T/N:** By default, tumbling windows are inclusive in the end of the window and exclusive in the beginning – for example 12:00 PM – 1:00 PM window will include events that happened exactly at 1:00 PM but will not include events that happened at 12:00PM (these events will be part of 11:00 AM – 12:00 PM window).

For this project, the system.timestamp is stored in the variable “WindowEnd” which is basically a timestamp for the end of each 5 seconds window of the ‘CallRecTime’. The aggregation factor are the countries in the “SwitchNum” column and the tumbling window is an interval of “5 seconds” – this essentially means the calls will be grouped by country/region and counted every 5 seconds interval without repetition.

Step 8: 3rd Query to detect SIM Fraud

Home > All resources > stellastreamanalytics

stellastreamanalytics | Query

Stream Analytics job

» ▶ Start job 📖 Query language docs 🔗 Open in VS Code 🛠 Diagnostic settings 🔄 Refresh 😊 Share feedback 📖 Tutorial

▼ Inputs (1) +
 🔗 Phonestream ⌵

▼ Outputs (1) +
 🔗 Phonestream ⌵

▼ Functions (0) +

▶ Test query 📄 Save query ✕ Discard changes

```
1  
2 SELECT System.Timestamp AS WindowEnd, COUNT(*) AS FraudulentCalls  
3     INTO "Phonestreamoutput"  
4     FROM "Phonestream" CS1 TIMESTAMP BY CallRecTime  
5     JOIN "Phonestream" CS2 TIMESTAMP BY CallRecTime  
6     ON CS1.CallingIMSI = CS2.CallingIMSI  
7     AND DATEDIFF(ss, CS1, CS2) BETWEEN 1 AND 5  
8     WHERE CS1.SwitchNum != CS2.SwitchNum  
9     GROUP BY TumblingWindow(Duration(second, 1))
```

Input preview **Test results** Job simulation (preview)

ⓘ Time policies do not apply to query testing. The results you see here might be different from the actual job results. [Learn more about our authoring.](#)

⬇ Download results

WindowEnd <small>datetime</small>	FraudulentCalls <small>bigint</small>
"2023-10-20T17:06:04.0000000Z"	1
"2023-10-20T17:06:05.0000000Z"	3
"2023-10-20T17:06:10.0000000Z"	1
"2023-10-20T17:06:12.0000000Z"	1

🟢 Showing 13 rows from 'Phonestreamoutput'.

Remember to save the query before running it. It is also **important to save the query in other to be able to 'start the job'** in stream analytics so it can stream the results of the query to Power BI

The result of the query is displayed in "Fraudulent calls"

Step 8: Analysis of 3rd Query Contd

```
Test query Save query Discard changes
1
2 SELECT System.Timestamp AS WindowEnd, COUNT(*) AS FraudulentCalls
3 INTO "Phonestreamoutput"
4 FROM "Phonestream" CS1 TIMESTAMP BY CallRecTime
5 JOIN "Phonestream" CS2 TIMESTAMP BY CallRecTime
6 ON CS1.CallingIMSI = CS2.CallingIMSI
7 AND DATEDIFF(ss, CS1, CS2) BETWEEN 1 AND 5
8 WHERE CS1.SwitchNum != CS2.SwitchNum
9 GROUP BY TumblingWindow(Duration(second, 1))
```

In essence, the code will detect if a user is using the same phone number to dial 2 or more countries at the same time which is illegal. *You've made it this far, you may want to go back and read the words in yellow as a whole sentence!*

SELECT System.Timestamp AS WindowEnd, COUNT() AS FraudulentCalls (Output of the entire query is found in this column)*

INTO "Phonestreamoutput"

FROM "PhoneStream" CS1 TIMESTAMP BY CallRecTime

JOIN "PhoneStream" CS2 TIMESTAMP BY CallRecTime (It's a self-join and we're basically comparing values in the stream against themselves)

ON CS1.CallingIMSI = CS2.CallingIMSI (to check if a phone number appeared more than once at the same time)

AND DATEDIFF(ss, CS1, CS2) BETWEEN 1 AND 5 (within a space of 5 seconds)

WHERE CS1.SwitchNum != CS2.SwitchNum (especially if the countries from which the phone number originated are not the same)

stellastreamanalytics | Query ☆ ...
Stream Analytics job

» Start job Query language docs Open in VS Code Diagnostic settings Refresh Share feedback Tutorial

Inputs (1) +

Phonestream

Outputs (1) +

Phonestream

Functions (0) +

Test query Save query Discard changes

```
1
2 SELECT System.Timestamp AS WindowEnd, COUNT(*) AS FraudulentCalls
3     INTO "Phonestreamoutput"
4     FROM "Phonestream" CS1 TIMESTAMP BY CallRecTime
5     JOIN "Phonestream" CS2 TIMESTAMP BY CallRecTime
6     ON CS1.CallingIMSI = CS2.CallingIMSI
7     AND DATEDIFF(ss, CS1, CS2) BETWEEN 1 AND 5
8     WHERE CS1.SwitchNum != CS2.SwitchNum
9     GROUP BY TumblingWindow(Duration(second, 1))
```

Diagnostic settings Refresh Share feedback Tutorial

Job ready to start

Discard changes

```
stamp AS WindowEnd, COUNT(*) AS FraudulentCalls
eamoutput"
nam" CS1 TIMESTAMP BY CallRecTime
eam" CS2 TIMESTAMP BY CallRecTime
IMSI = CS2.CallingIMSI
```

Now that the query has been saved, the “**Start job**” button comes on. Once the button is tapped, the “**job ready to start**” is seen.

The job has already started, and we can now see the “Stop job” button has come on. Once the button is tapped, the job will be stopped, and the data streams will stop being transmitted.

The screenshot displays the 'stellastreamanalytics' Stream Analytics job interface. The left sidebar contains navigation links: Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Job topology, Inputs, Functions, and Query. The main content area shows the job status as 'Running' with a clock icon. Below this, the 'Essentials' section lists job details in two columns. The top toolbar includes buttons for 'Stop job', 'Delete', 'Move', 'Refresh', and 'Share feedback'. A search bar is located at the top left of the main content area.

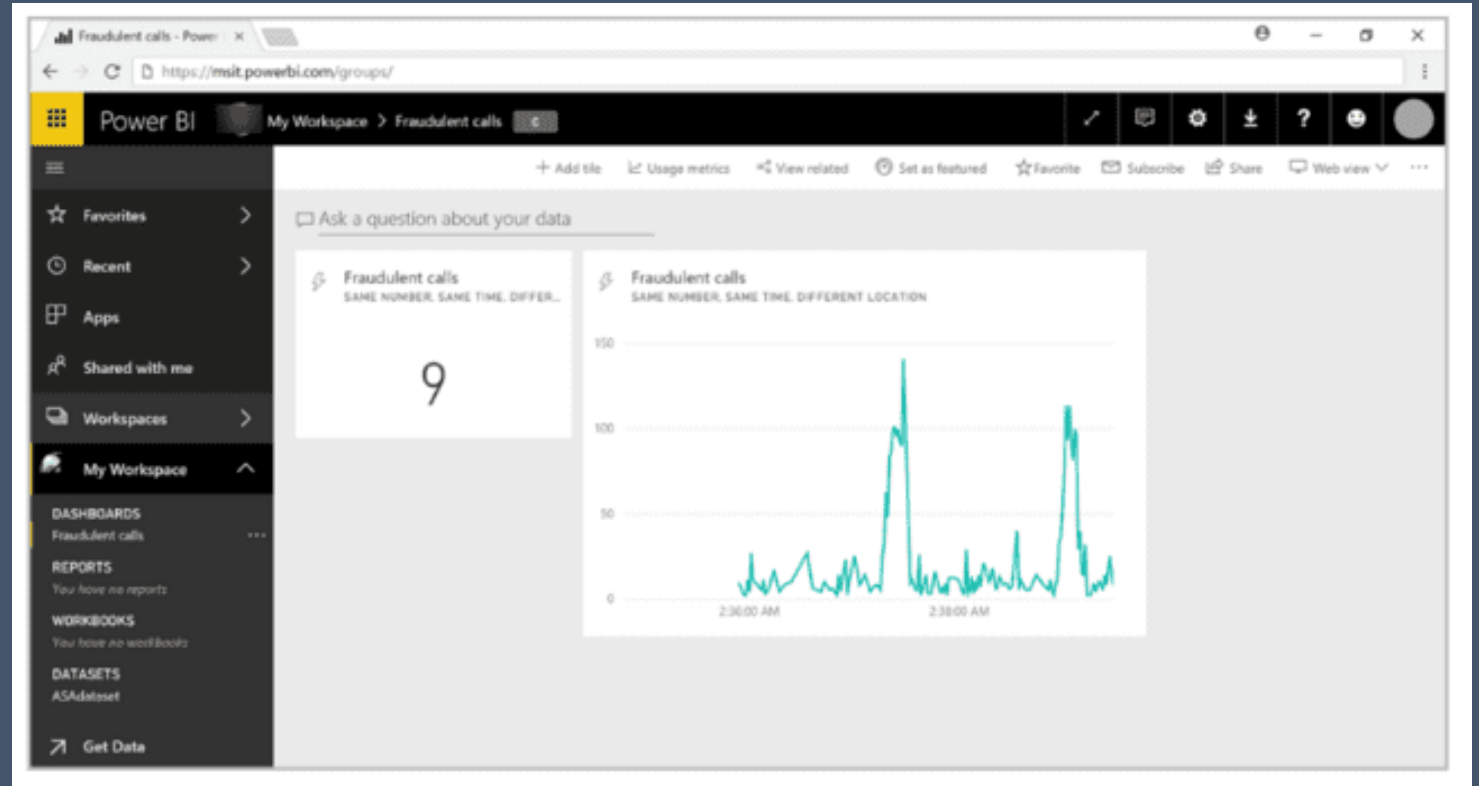
Job Details	
Resource group (move)	: stellastream
Location	: East US
Status	: Running
Subscription (move)	: Azure subscription 1
Subscription ID	: a0ab40ba-f007-4975-bef3-c471b2fd82f1
Pricing plan	: StandardV2 (manage)
Tags (edit)	:
Created	: Friday, October 20, 2023 6:39 PM
Started	: Friday, October 20, 2023 7:41 PM
Output watermark	:
Cluster	: Shared
Hosting environment	: Cloud
Virtual Network	: Disabled

Step 9: Visualising the results in Power BI

PROCESS

Now that the job has started running in Azure Streams Analytics, you can navigate to Power BI and sign in. If the Stream Analytics job query is outputting results, the ASAdataset dataset that was created will be found under the Datasets tab.

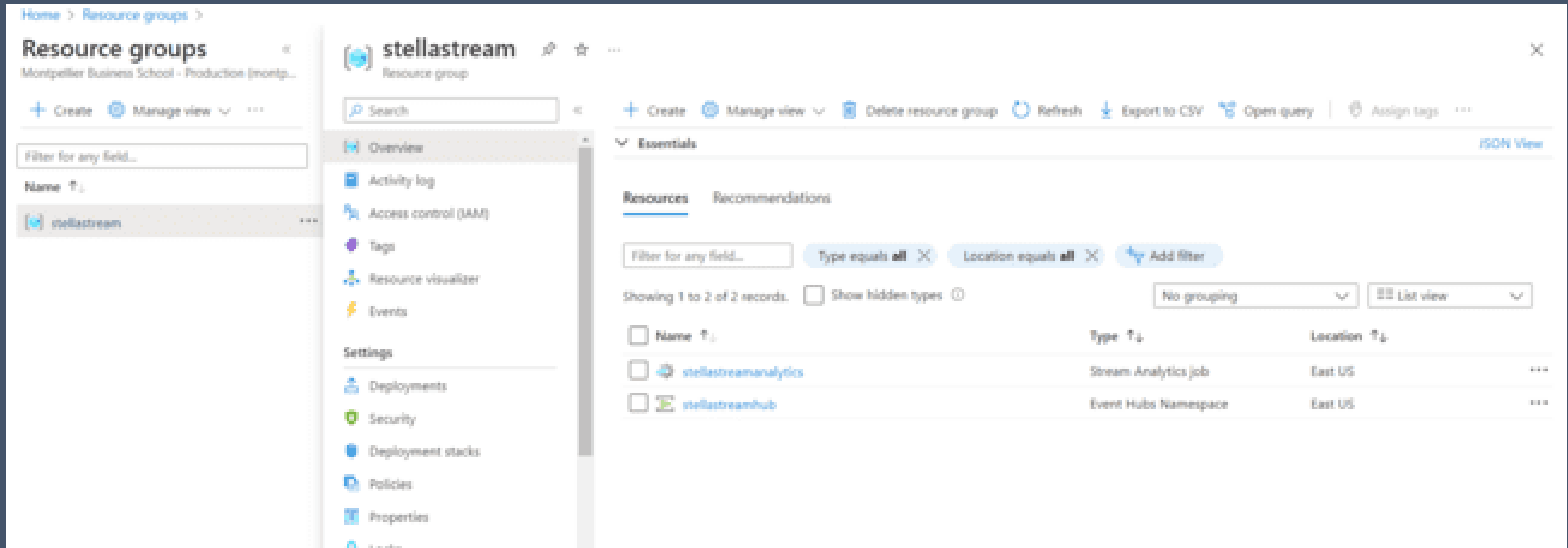
- From your Power BI workspace, select **+ Create** to create a new dashboard named Fraudulent Calls.
- At the top of the window, select **Edit and Add tile**.
- Then select Custom Streaming Data and Next.
- Choose the **ASAdataset** (yours should be the name you gave it when you configured Power BI in Azure Stream Analytics) under Your Datasets.
- Select **Card** from the Visualization type dropdown and add fraudulent calls to Fields.
- Select Next to enter a name for the tile, and then select Apply to create the tile.



Follow the step 5 again with the following options:

- When you get to Visualization Type, select Line chart.
- Add an axis and select **windowend**.
- Add a value and select **fraudulent calls**.
- For **Time window to display**, select the last 10 minutes

Step 9: Delete Resource group “stellastream”



The screenshot displays the Azure portal interface for the 'stellastream' resource group. The left sidebar shows the 'Resource groups' list with 'stellastream' selected. The main pane shows the 'Resources' tab with a table of resources:

Name	Type	Location
stellastreamanalytics	Stream Analytics job	East US
stellastreamhub	Event Hubs Namespace	East US

Home >>> Resource groups >>> Select the resources displayed >>> Select “Delete resource group” >>> Confirm the deletion of the resource group

Step 9 contd

The screenshot shows the AWS Management Console interface for a resource group named 'stellastream'. The left sidebar contains navigation links for Overview, Activity log, Access control (IAM), Tags, Resource visualizer, Events, Settings, Deployments, Security, Deployment stacks, Policies, Properties, Locks, Cost Management, Cost analysis, Cost alerts (preview), and Budgets. The main content area displays the 'Delete a resource group' dialog. The dialog title is 'Delete a resource group'. Below the title, it states: 'The following resource group and all its dependent resources will be permanently deleted.' The 'Resource group to be deleted' section shows the 'stellastream' resource group. The 'Dependent resources to be deleted (2)' section lists two resources: 'stellastreamanalytics' (Stream Analytics job) and 'stellastreamhub' (Event Hubs Namespace). At the bottom of the dialog, there is a text input field labeled 'Enter resource group name to confirm deletion *' with the value 'stellastream' entered. Below the input field are 'Delete' and 'Cancel' buttons.

Delete a resource group

The following resource group and all its dependent resources will be permanently deleted.

Resource group to be deleted

stellastream

Dependent resources to be deleted (2)

All dependent resources, including hidden types, are shown

Name	Resource type
stellastreamanalytics	Stream Analytics job
stellastreamhub	Event Hubs Namespace

Enter resource group name to confirm deletion *

stellastream

Delete Cancel

Delete a resource group

The following resource group and all its dependent resources will be permanently deleted.

Resource group to be deleted

stellastream

Dependent resources to be deleted (2)

All dependent resources, including hidden types, are shown

Name	Resource type
stellastreamanalytics	Stream Analytics job
stellastreamhub	Event Hubs Namespace

Enter resource group name to confirm deletion *

stellastream

Delete

Cancel

Step 9 contd: Confirm that the resources have been deleted

Resource groups ⚙️ ...


Montpellier Business School - Production (montpellier-bs.com)

+ Create ⚙️ Manage view ▾ 🔄 Refresh ⬇️ Export to CSV 🔗 Open query | 🏷️ Assign tags

Filter for any field... Subscription equals all Location equals all ✕ ➕ Add filter

Showing 0 to 0 of 0 records.

No grouping

Name ↑↓	Subscription ↑↓	Location ↑↓
<div></div> <p>No resource groups to display</p> <p>Try changing or clearing your filters.</p> <p>Create resource group</p> <p>Learn more</p>		

If you do not delete the resource group, it will run up a lot of cost!