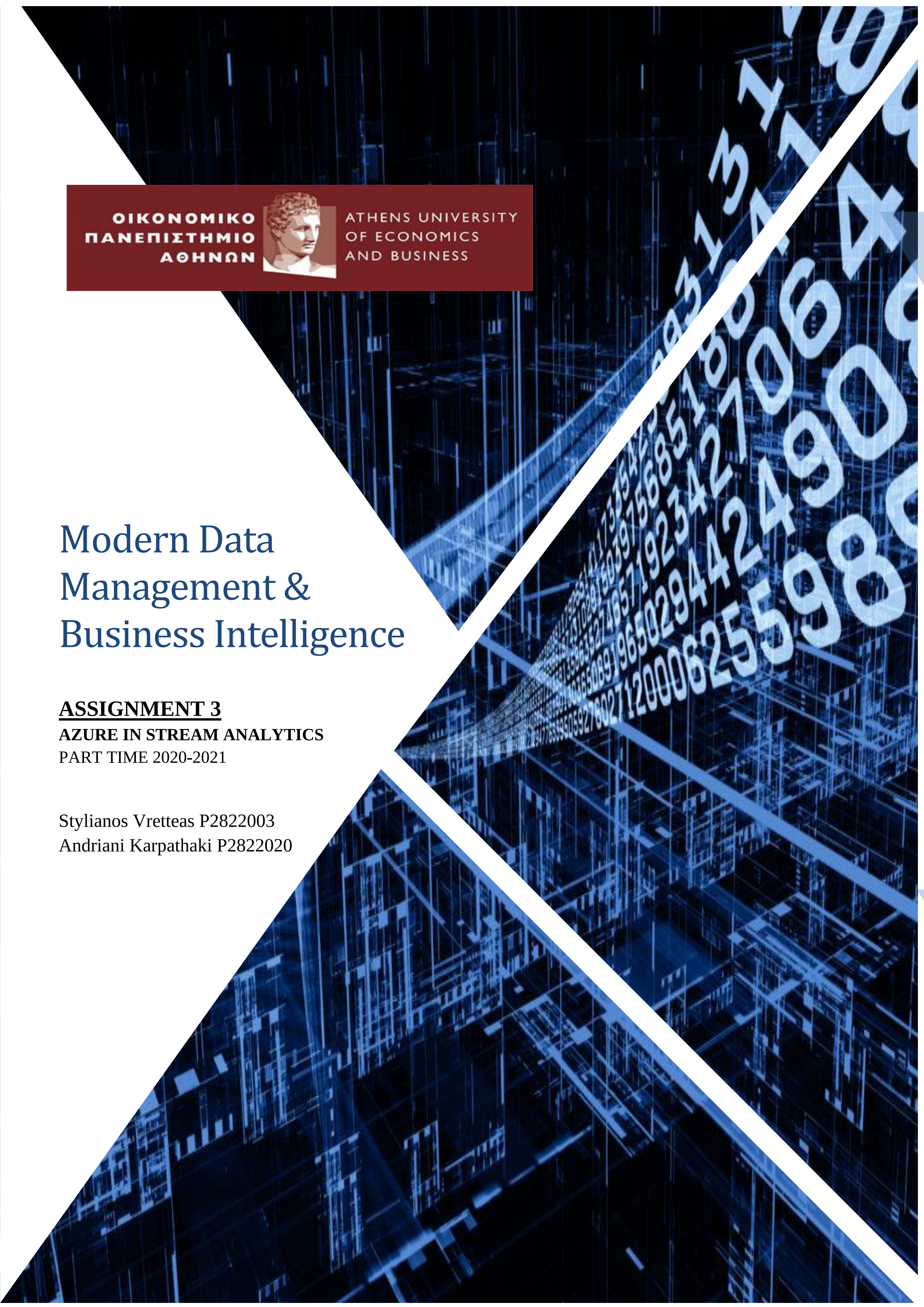


ΟΙΚΟΝΟΜΙΚΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΑΘΗΝΩΝ

ATHENS UNIVERSITY
OF ECONOMICS
AND BUSINESS

Modern Data Management & Business Intelligence

ASSIGNMENT 3

AZURE IN STREAM ANALYTICS

PART TIME 2020-2021

Stylianos Vretteas P2822003
Andriani Karpathaki P2822020

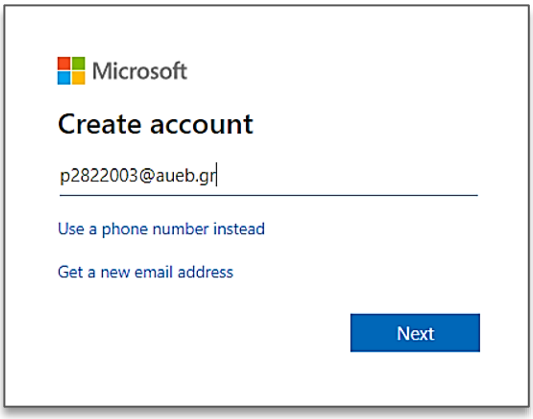
Contents

Abstract

The scope of this report was to use Azure Stream Analytics to process a data stream of ATM transactions and answer stream queries. The schema of the stream is: (ATMCode, CardNumber, Type, Amount). The steps followed in this procedure are represented below.

1. Create a student’s account at: https://azure.microsoft.com/en us/free/students/

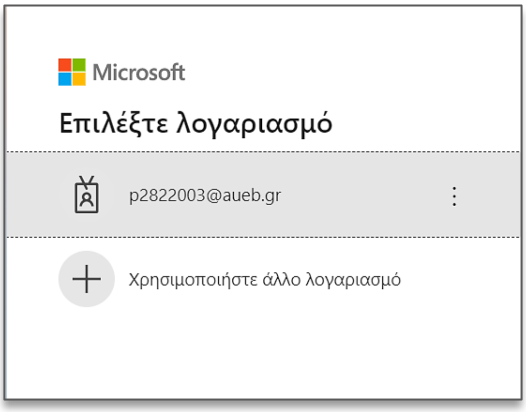
First of all, we created an Azure Student account (Figure 1) and then we signed in (Figure 2).



Create account

p2822003@aueb.gr

Figure 1 - Create an Azure Student Account



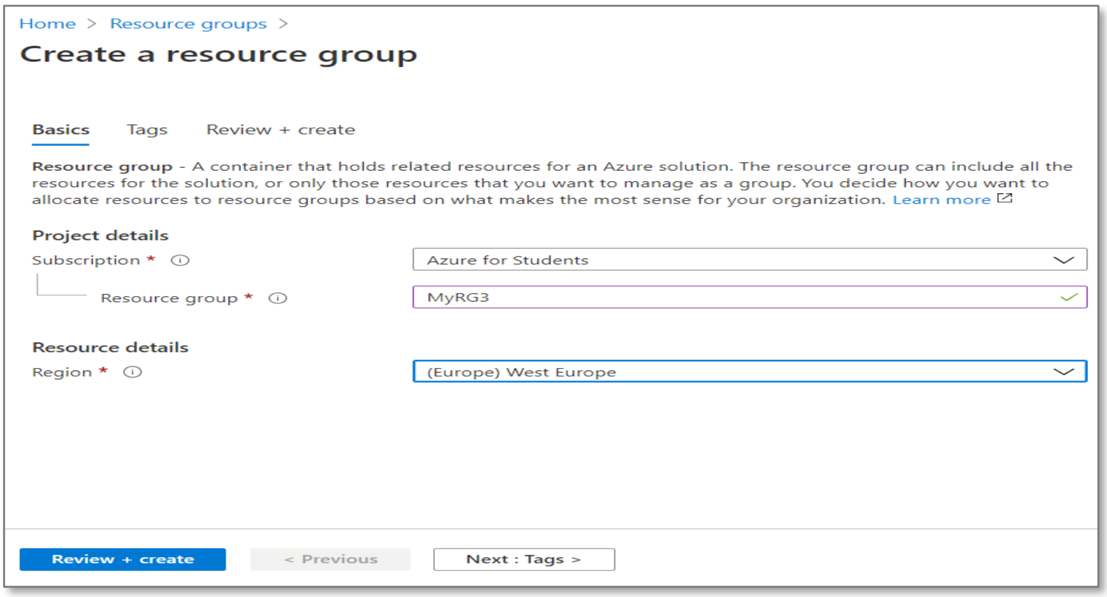
Επιλέξτε λογαριασμό

p2822003@aueb.gr

Figure 2 - Sign in Azure account

2. Setup an Event Hub.

We created a Resource group because it serves as a folder that holds related resources for an Azure solution. The resource group can include all the resources for the solution, or only those resources that are preferable to be managed as a group (Figure 3).



Home > Resource groups >

Create a resource group

Basics Tags Review + create

Resource group - A container that holds related resources for an Azure solution. The resource group can include all the resources for the solution, or only those resources that you want to manage as a group. You decide how you want to allocate resources to resource groups based on what makes the most sense for your organization. Learn more

Project details

Subscription * : Azure for Students

Resource group * : MyRG3

Resource details

Region * : (Europe) West Europe

Figure 3 - Create a resource group

Next step was to create an Azure Event Hub which is a big data streaming platform and event ingestion service. It can receive and process millions of events per second. Data sent to an event hub can be transformed and stored by using any real-time analytics provider or batching/storage adapters (Figure 4).

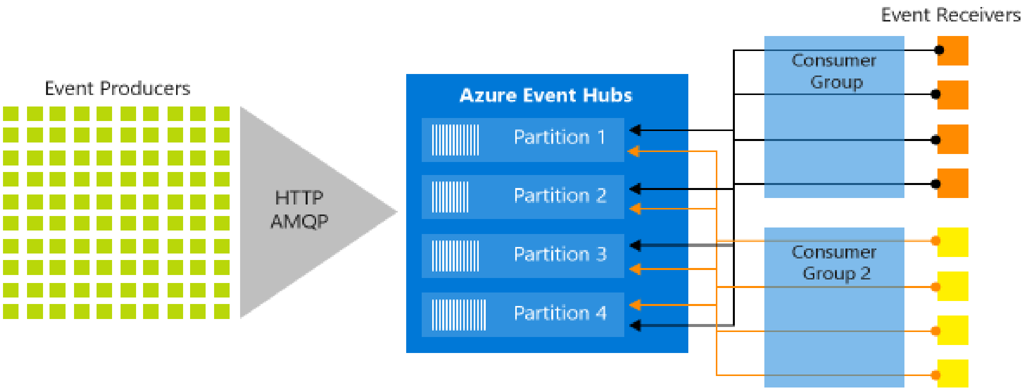
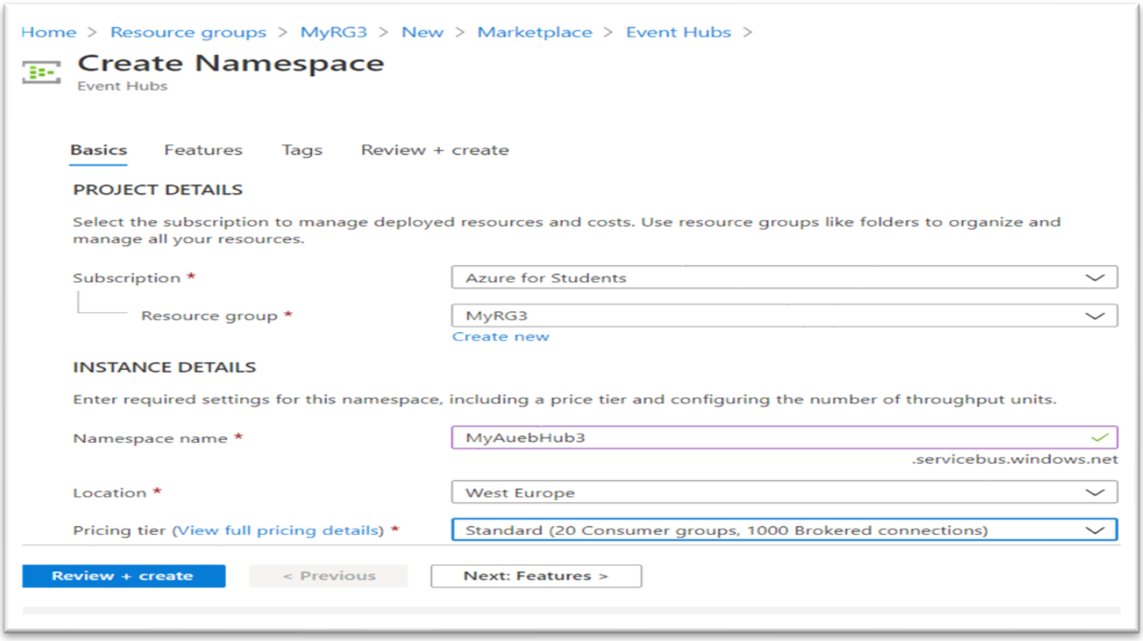


Figure 4 - Architecture Azure Event Hubs

Construction of Event Hub

At this point it necessary to create an Event Hubs Namespace. An Event Hubs namespace provides a unique scoping container, in which we can create one or more event hubs. For the scope of this paper, we have created one Event Hub, named MyAuebHub3 (Figure 5). The procedure was completed successfully (Figure 6).



Create Namespace

Event Hubs

PROJECT DETAILS

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

Subscription * Azure for Students

Resource group * MyRG3

INSTANCE DETAILS

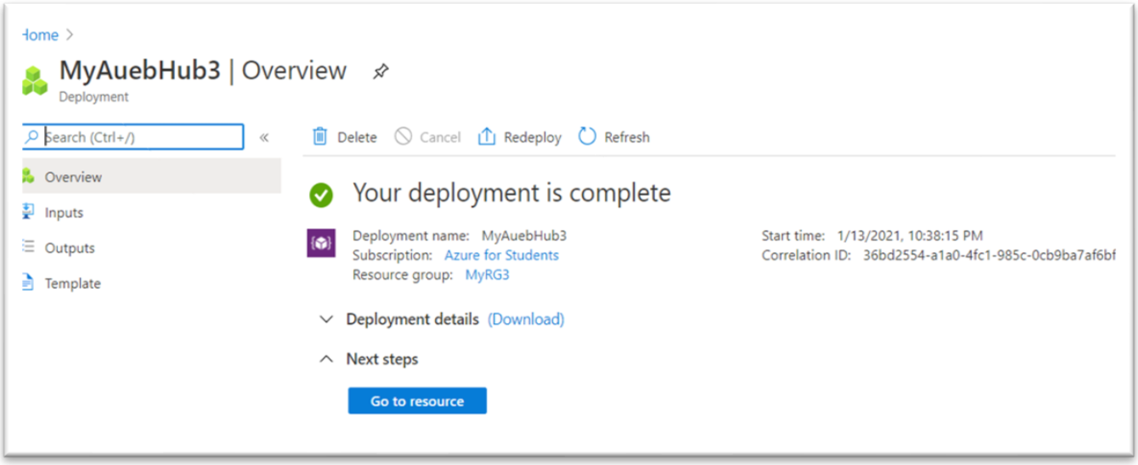
Enter required settings for this namespace, including a price tier and configuring the number of throughput units.

Namespace name * MyAuebHub3 .servicebus.windows.net

Location * West Europe

Pricing tier (View full pricing details) * Standard (20 Consumer groups, 1000 Brokered connections)

Figure 5 - Event Hub Creation



MyAuebHub3 | Overview

Deployment

Your deployment is complete

Deployment name: MyAuebHub3

Subscription: Azure for Students

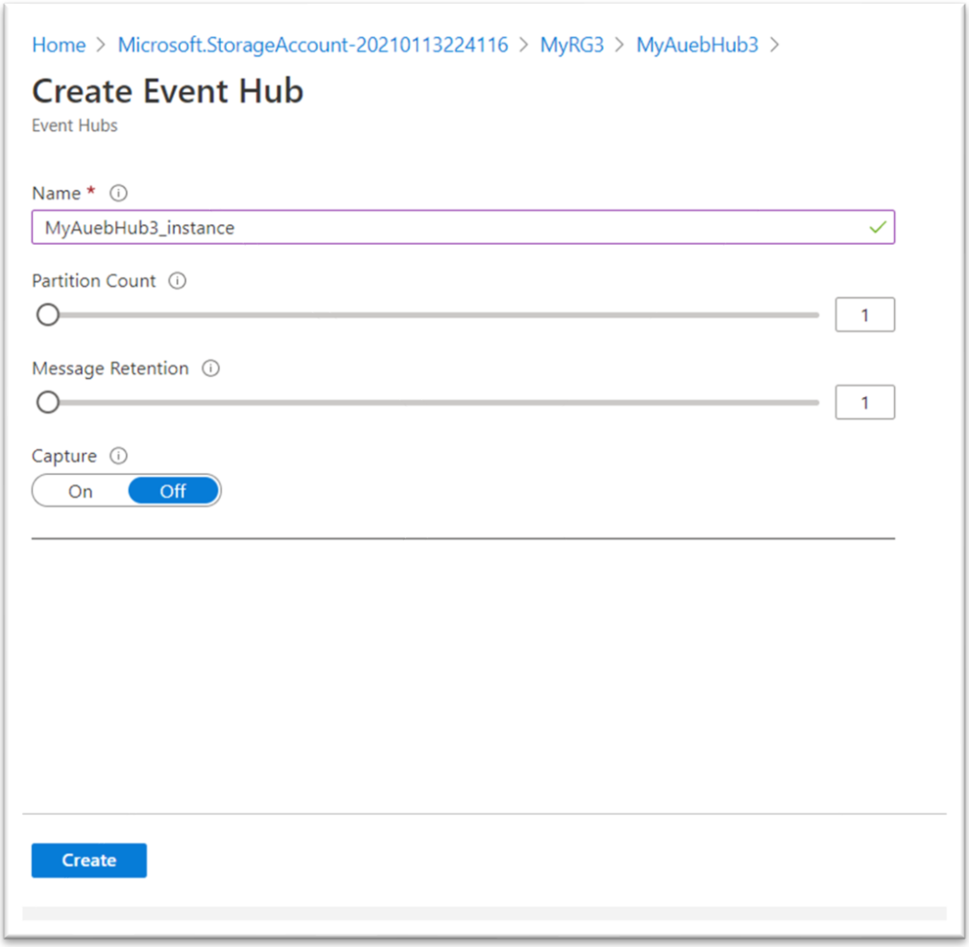
Resource group: MyRG3

Start time: 1/13/2021, 10:38:15 PM

Correlation ID: 36bd2554-a1a0-4fc1-985c-0cb9ba7af6bf

Figure 6 - Deployment completed for Event Hub creation

Next, we created an Event Hub instance (Figure 7).



Home > Microsoft.StorageAccount-20210113224116 > MyRG3 > MyAuebHub3 >

Create Event Hub

Event Hubs

Name * : MyAuebHub3_instance

Partition Count: 1

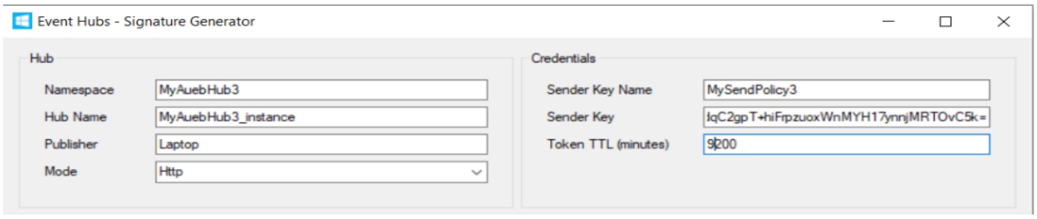
Message Retention: 1

Capture: On [✓ Off]

Figure 7 Event Hub Instance creation

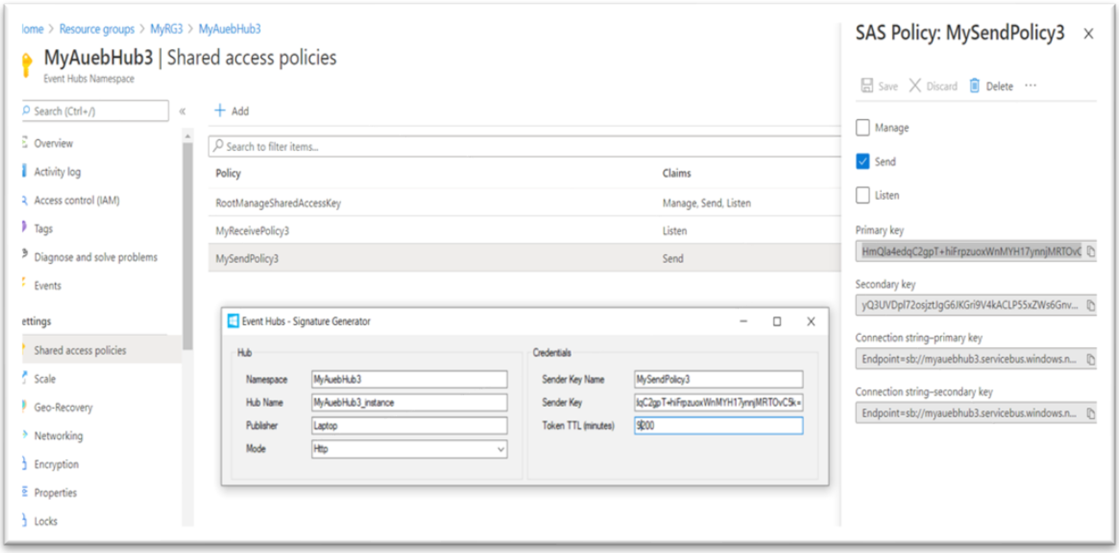
3. Generate a Security Access Signature (use a terminal with windows operationg system): https://github.com/sandrinodimattia/RedDog/releases

Next step was to generate a security access signature and use it in order to edit the data transmission generator configuration settings and connect it with the event hub (Figure 8 & 9).



Hub		Credentials	
Namespace	MyAuebHub3	Sender Key Name	MySendPolicy3
Hub Name	MyAuebHub3_instance	Sender Key	lqC2gpT+hiFrpzuoxWnMYH17ynnjMRTOvC5k=
Publisher	Laptop	Token TTL (minutes)	9200
Mode	Http		

Figure 8 - Signature Generator



lome > Resource groups > MyRG3 > MyAuebHub3

MyAuebHub3 | Shared access policies

Event Hubs Namespace

Policy	Claims
RootManageSharedAccessKey	Manage, Send, Listen
MyReceivePolicy3	Listen
MySendPolicy3	Send

Event Hubs - Signature Generator

Hub

Namespace: MyAuebHub3

Hub Name: MyAuebHub3_instance

Publisher: Laptop

Mode: Http

Credentials

Sender Key Name: MySendPolicy3

Sender Key: lqC2gpT+hiFrpzuoxWnMYH17ynnjMRTOvC5k=

Token TTL (minutes): 9200

SAS Policy: MySendPolicy3

□ Manage

☑ Send

□ Listen

Primary key: HmQla4edqC2gpT+hiFrpzuoxWnMYH17ynnjMRTOvC

Secondary key: yQ3UVDpl72osjztJgG6JKGri9V4kACLP55xZWs6Gnv...

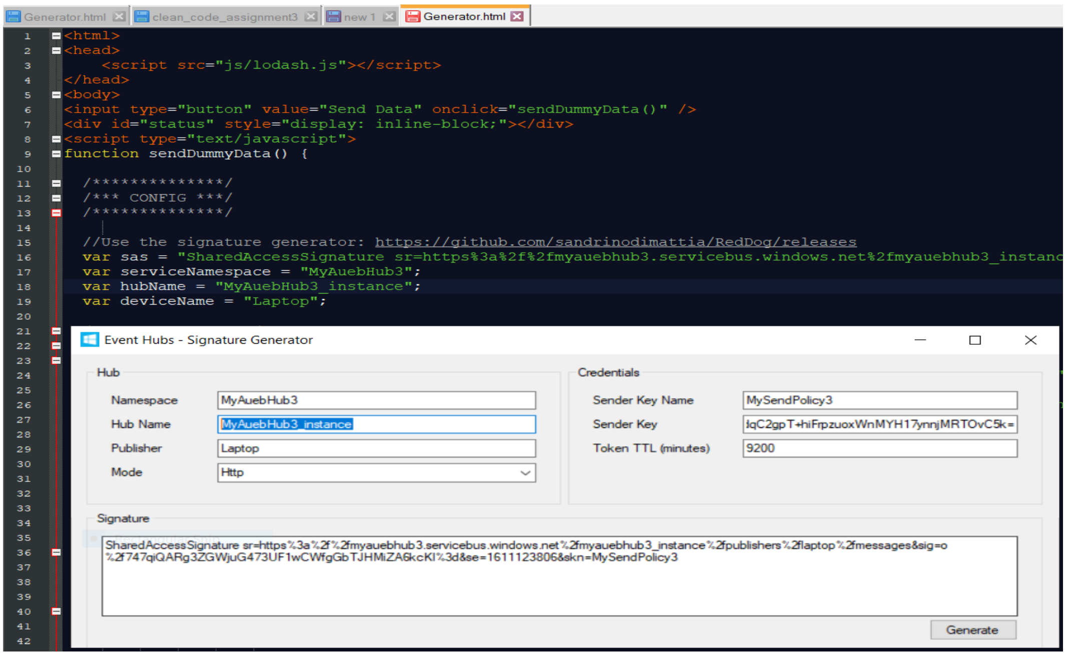
Connection string–primary key: Endpoint=sb://myauebhub3.servicebus.windows.n...

Connection string–secondary key: Endpoint=sb://myauebhub3.servicebus.windows.n...

Figure 9 - Creation of signature

4. Edit Generator.html (open with a text editor, e.g.: Sublime or Notepad++) and update the CONFIG variables. Keep the "js" folder in the same folder as the Generator.html file.

After creating the signature, we edited the html file via Notepad++ and updated the CONFIG variables (Figure 10).



```
<html>
<head>
    <script src="js/lodash.js"></script>
</head>
<body>
<input type="button" value="Send Data" onclick="sendDummyData()" />
<div id="status" style="display: inline-block;"></div>
<script type="text/javascript">
function sendDummyData() {

  /**************/
  /*** CONFIG ***/
  /**************/

  //Use the signature generator: https://github.com/sandrinodimattia/RedDog/releases
  var sas = "SharedAccessSignature sr=https%3a%2f%2fmyauebhub3.servicebus.windows.net%2fmyauebhub3_instanc
  var serviceNamespace = "MyAuebHub3";
  var hubName = "MyAuebHub3_instance";
  var deviceName = "Laptop";
```

Event Hubs - Signature Generator

Hub

Namespace: MyAuebHub3

Hub Name: MyAuebHub3_instance

Publisher: Laptop

Mode: Http

Credentials

Sender Key Name: MySendPolicy3

Sender Key: lqC2gpT+hiFrpzuoxWnMYH17ynnjMRTOvC5k=

Token TTL (minutes): 9200

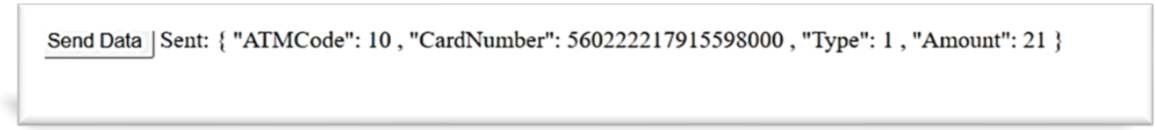
Signature

```
SharedAccessSignature sr=https%3a%2f%2fmyauebhub3.servicebus.windows.net%2fmyauebhub3_instance%2fpublishers%2flaptop%2fmessages&sig=o
%2f747qiQARg3ZGWjuG473UF1wCWfgGbTJHMiZA6kcKI%3d&se=1611123806&skn=MySendPolicy3
```

Figure 10 - Edit html file via Notepad++

5. Feed the Event Hub with the use of Generator.html (In order to start the Stream Generator, open the Generator.html with a web browser, e.g.: Chrome and press the “Send Data” button.)

After editing the html file, we opened the Generator.html via Chrome web browser in order to start the transmission of data (Figure 11).



Send Data Sent: { "ATMCode": 10 , "CardNumber": 560222217915598000 , "Type": 1 , "Amount": 21 }

Figure 11 - Start of the Stream Generator

6. Setup a Storage account.

Storage Account

Azure Blob storage is Microsoft's object storage solution for the cloud. More precisely, Blob storage is optimized for storing massive amounts of unstructured data and offers three types of resources (Figure 12):

- The storage account which provides a unique namespace in Azure for your data
- A container in the storage account
- An unlimited number of containers as a blob in a container

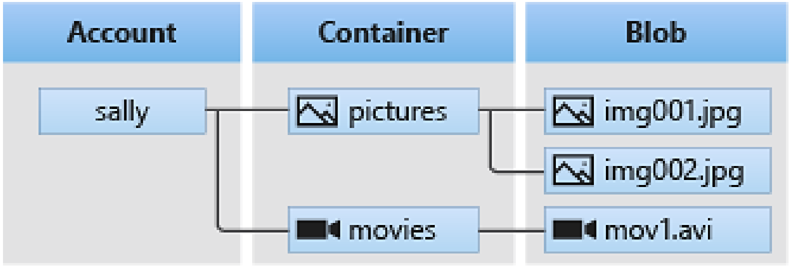
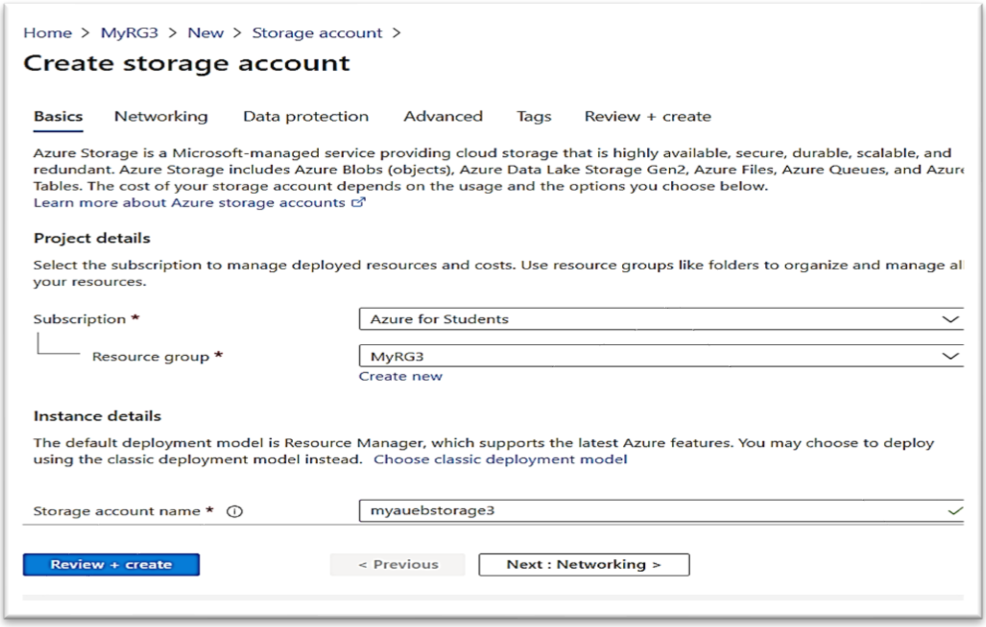


Figure 12 - Blob Storage Diagram

Construction of storage account and container

After creating the Event Hub, a storage account and container was built (Figure 13 & 15). To be noted that another container was created later in order to store the reference data there.



Home > MyRG3 > New > Storage account >

Create storage account

Azure Storage is a Microsoft-managed service providing cloud storage that is highly available, secure, durable, scalable, and redundant. Azure Storage includes Azure Blobs (objects), Azure Data Lake Storage Gen2, Azure Files, Azure Queues, and Azur Tables. The cost of your storage account depends on the usage and the options you choose below.
Learn more about Azure storage accounts

Project details

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

Subscription *: Azure for Students

Resource group *: MyRG3
Create new

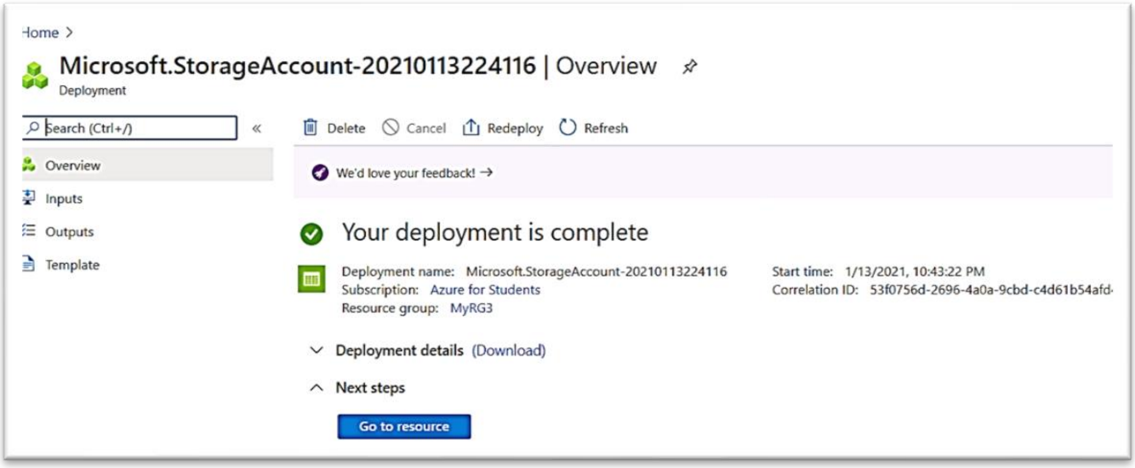
Instance details

The default deployment model is Resource Manager, which supports the latest Azure features. You may choose to deploy using the classic deployment model instead. Choose classic deployment model

Storage account name *: myauebstorage3

Figure 13 - Storage account creation

The procedure was completed successfully (Figure 14).



Home >

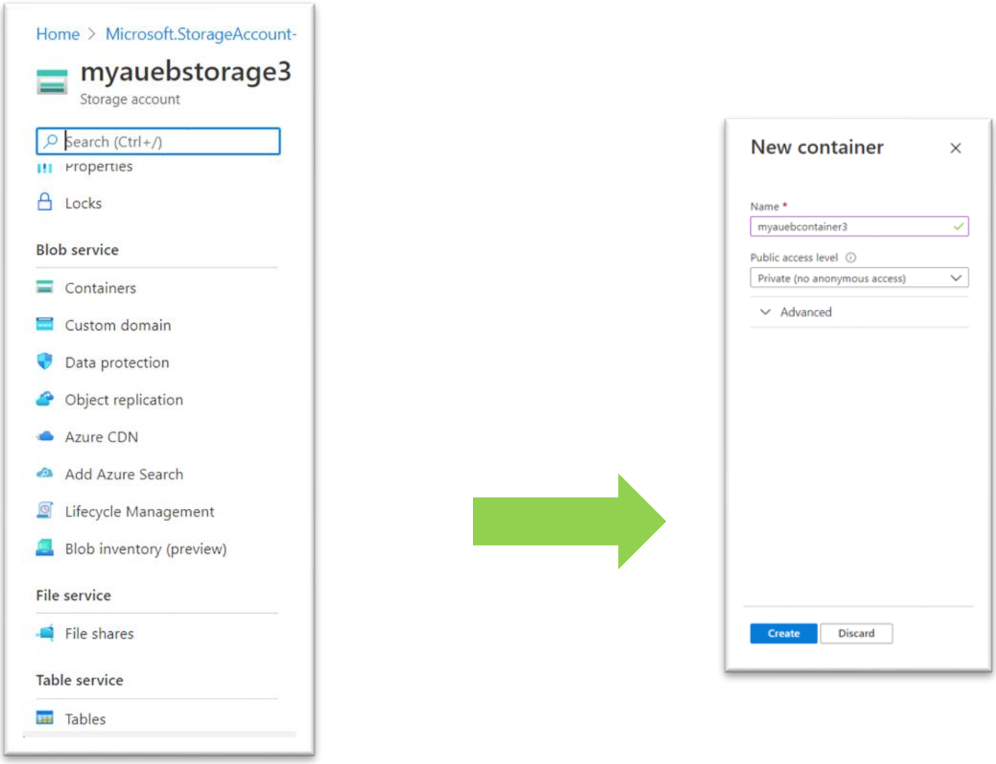
Microsoft.StorageAccount-20210113224116 | Overview
Deployment

Your deployment is complete

Deployment name: Microsoft.StorageAccount-20210113224116
Subscription: Azure for Students
Resource group: MyRG3

Start time: 1/13/2021, 10:43:22 PM
Correlation ID: 53f0756d-2696-4a0a-9cbd-c4d61b54afd

Figure 14 - Deployment completed for Storage Account creation



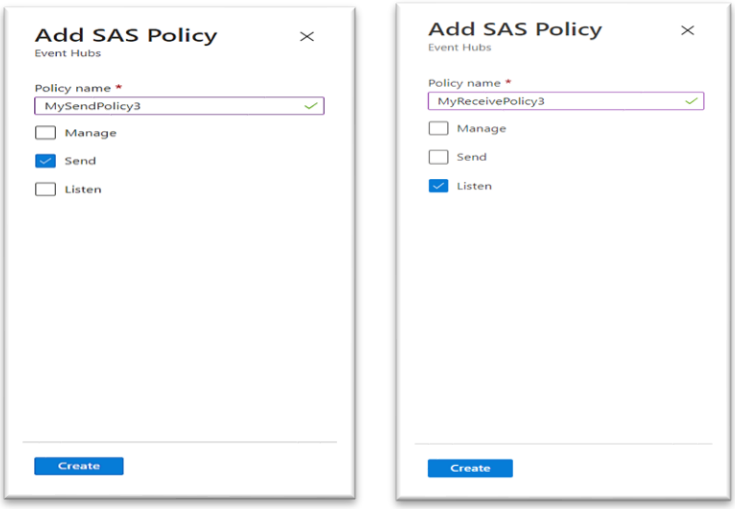
Name * myauebcontainer3

Public access level Private (no anonymous access)

Figure 15 - Container creation

Construction of Policies

Afterwards, we constructed policies in the event hub so as to send and receive events. Thus , we created the MyReceivePolicy3 and the MySendPolicy3 policies (Figure 16).



Policy name * MySendPolicy3

☐ Manage

☑ Send

☐ Listen

Policy name * MyReceivePolicy3

☐ Manage

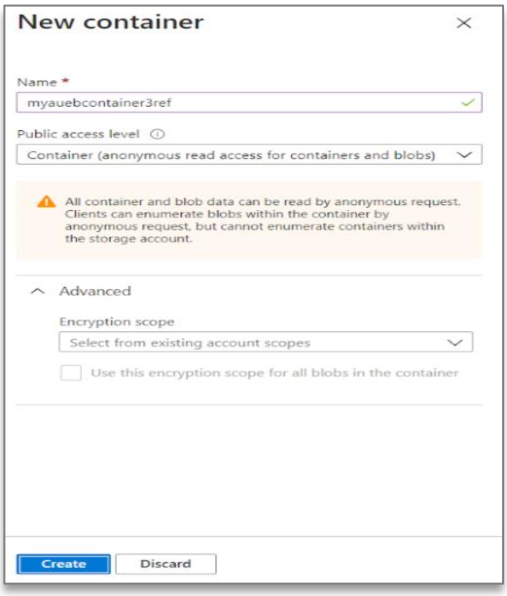
☐ Send

☑ Listen

Figure 16 Creation of policies

7. Upload the Reference Data files to your storage account.

In order to implement queries with joins, we had to upload reference data files that would serve as lookup tables between the stream input and reference input. A second container was built with the purpose of storing the reference data (Figure 17 & 18).



New container

Name *

myauebcontainer3ref

Public access level

Container (anonymous read access for containers and blobs)

All container and blob data can be read by anonymous request. Clients can enumerate blobs within the container by anonymous request, but cannot enumerate containers within the storage account.

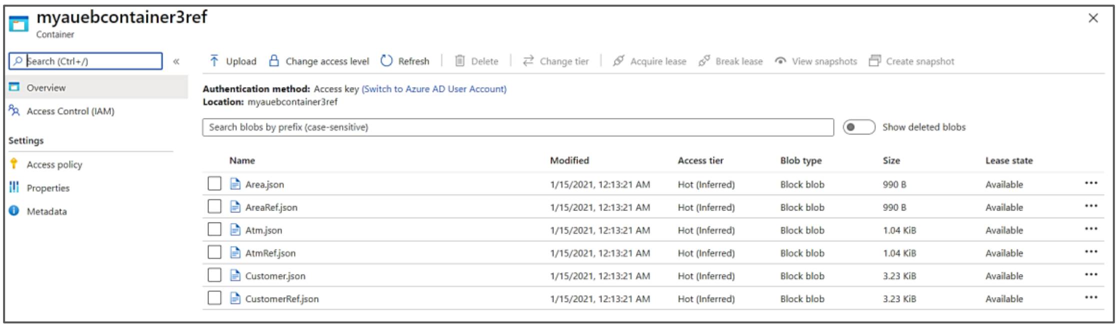
Advanced

Encryption scope

Select from existing account scopes

□ Use this encryption scope for all blobs in the container

Figure 17 - Creation of a container for storing the reference data



myauebcontainer3ref
Container

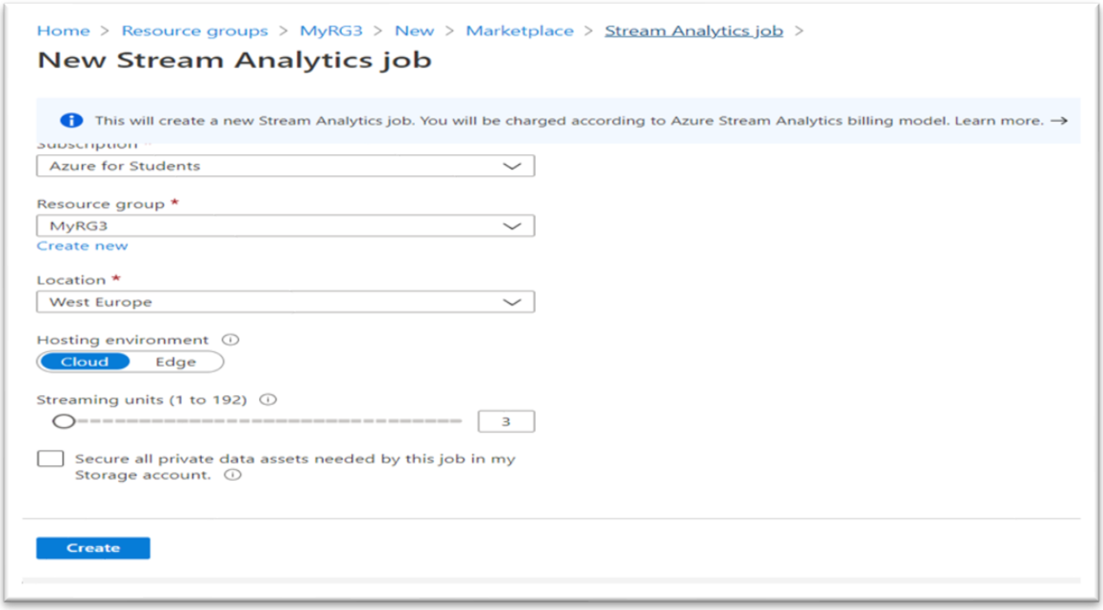
Authentication method: Access key (Switch to Azure AD User Account)
Location: myauebcontainer3ref

Name	Modified	Access tier	Blob type	Size	Lease state
Area.json	1/15/2021, 12:13:21 AM	Hot (Inferred)	Block blob	990 B	Available
AreaRef.json	1/15/2021, 12:13:21 AM	Hot (Inferred)	Block blob	990 B	Available
Atm.json	1/15/2021, 12:13:21 AM	Hot (Inferred)	Block blob	1.04 KiB	Available
AtmRef.json	1/15/2021, 12:13:21 AM	Hot (Inferred)	Block blob	1.04 KiB	Available
Customer.json	1/15/2021, 12:13:21 AM	Hot (Inferred)	Block blob	3.23 KiB	Available
CustomerRef.json	1/15/2021, 12:13:21 AM	Hot (Inferred)	Block blob	3.23 KiB	Available

Figure 18 - Uploaded Reference Data

8. Setup a Stream Analytics Job.

Next, we had to create the last Azure solution which is the Stream Analytics Job (Figure 19 & 20). An Azure Stream Analytics job consists of an input, query, and an output. Below we can see the overview of the solutions created (Figure 21).



Home > Resource groups > MyRG3 > New > Marketplace > Stream Analytics job >

New Stream Analytics job

This will create a new Stream Analytics job. You will be charged according to Azure Stream Analytics billing model. Learn more. →

Subscription: Azure for Students

Resource group *: MyRG3

Create new

Location *: West Europe

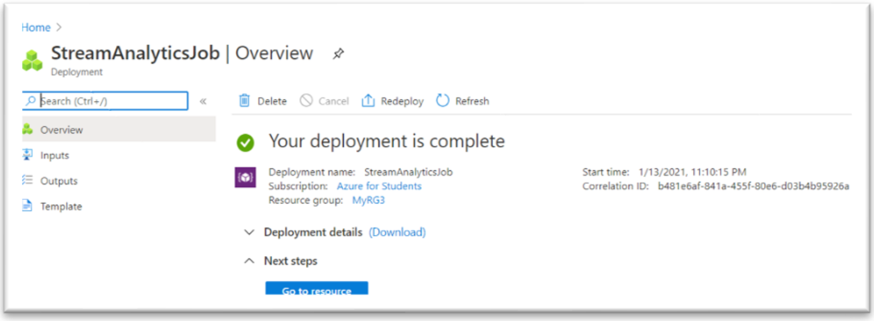
Hosting environment: Cloud | Edge

Streaming units (1 to 192): 3

□ Secure all private data assets needed by this job in my Storage account.

Create

Figure 19 - New Streams Analytics job



Home >

StreamAnalyticsJob | Overview

Deployment

Your deployment is complete

Deployment name: StreamAnalyticsJob

Subscription: Azure for Students

Resource group: MyRG3

Start time: 1/13/2021, 11:10:15 PM

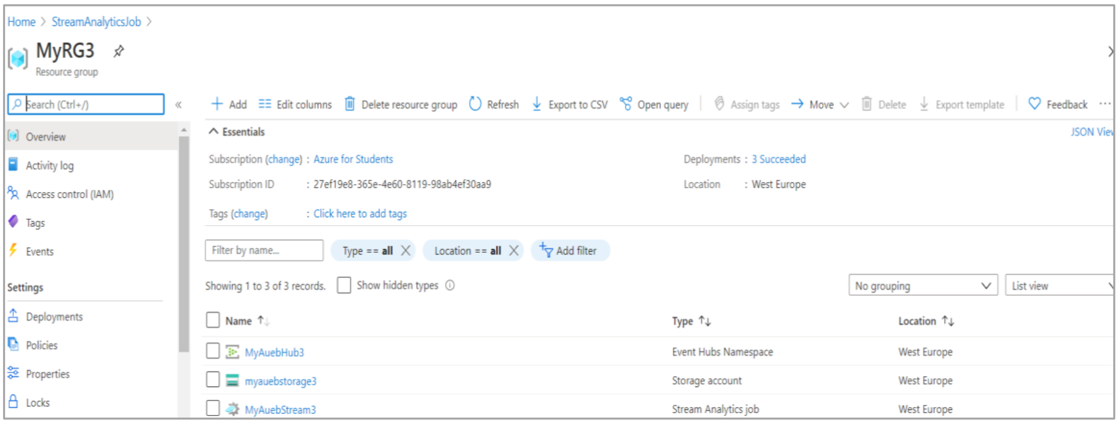
Correlation ID: b481e6af-841a-455f-80e6-d03b4b95926a

Deployment details (Download)

Next steps

Go to resource

Figure 20 - Creation of New Streams Analytics job completed



Home > StreamAnalyticsJob >

MyRG3

Resource group

Essentials

Subscription (change) : Azure for Students

Subscription ID : 27ef19e8-365e-4e60-8119-98ab4ef30aa9

Tags (change) : Click here to add tags

Deployments : 3 Succeeded

Location : West Europe

Filter by name... | Type == all | Location == all | Add filter

Showing 1 to 3 of 3 records. □ Show hidden types

No grouping | List view

Name	Type	Location
MyAuebHub3	Event Hubs Namespace	West Europe
myauebstorage3	Storage account	West Europe
MyAuebStream3	Stream Analytics job	West Europe

Figure 21 - Overview of the solutions created

9. Use the Event Hub + Reference Data Files as Input.

We had to define the input and output sources. There are two types of inputs:

Data stream input: A data stream is an unbounded sequence of events over time. Event Hubs are used to collect event streams from multiple devices and services.

Reference data input: Reference data is either completely static or changes slowly. It is typically used to perform correlation and lookups.

In our case for input we had one stream input regarding ATM transactions, three (3) reference inputs (Area, Atm, Customer) and one output (Figures 22-25).

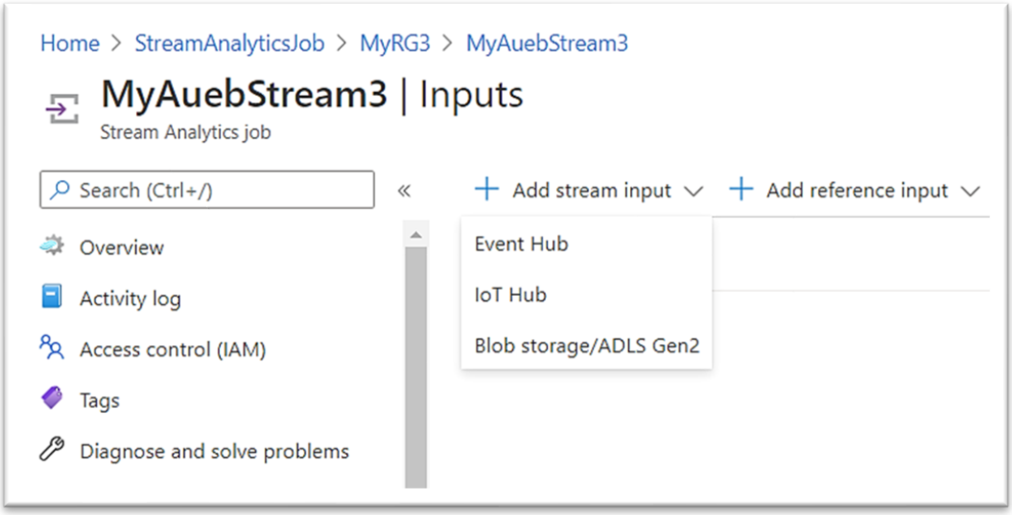
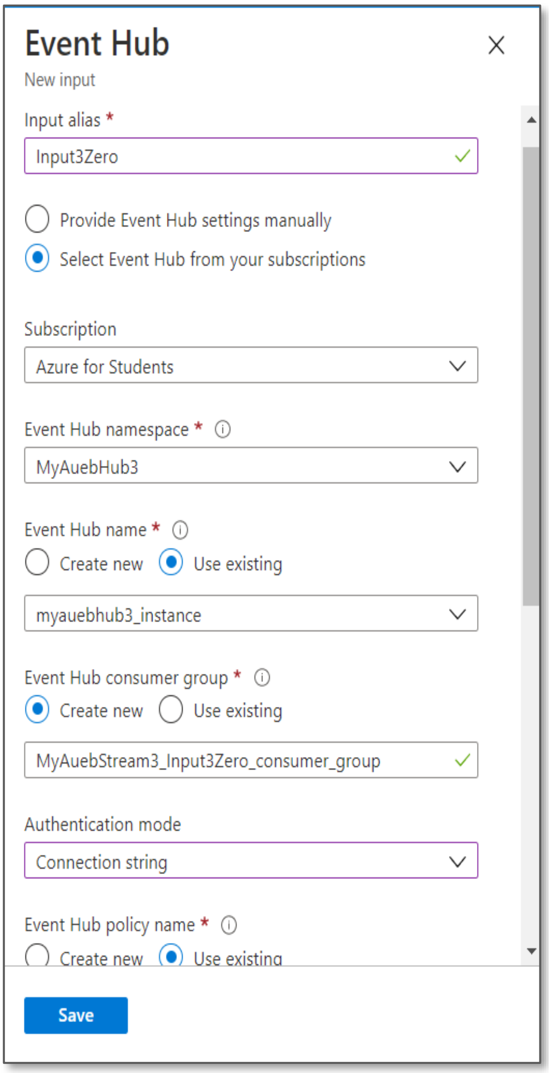


Figure 22 – Add stream input



Event Hub

New input

Input alias *

Input3Zero ✓

○ Provide Event Hub settings manually

● Select Event Hub from your subscriptions

Subscription

Azure for Students

Event Hub namespace *

MyAuebHub3

Event Hub name *

○ Create new ● Use existing

myauebhub3_instance

Event Hub consumer group *

● Create new ○ Use existing

MyAuebStream3_Input3Zero_consumer_group ✓

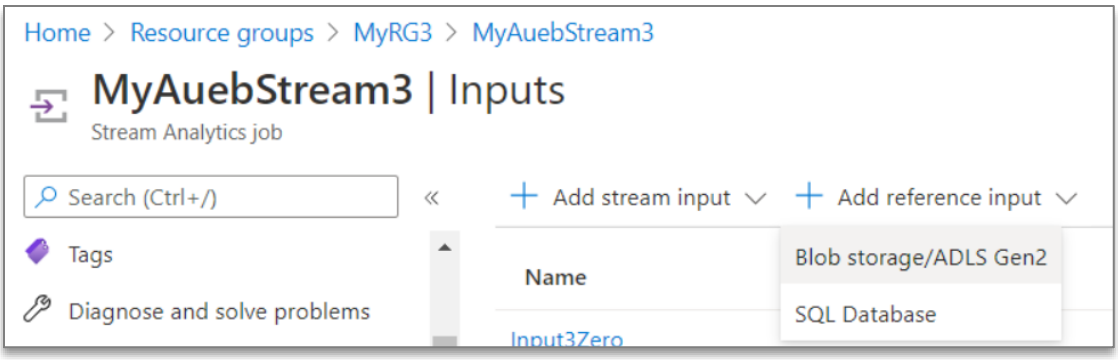
Authentication mode

Connection string

Event Hub policy name *

○ Create new ● Use existing

Figure 23 - Stream data input creation



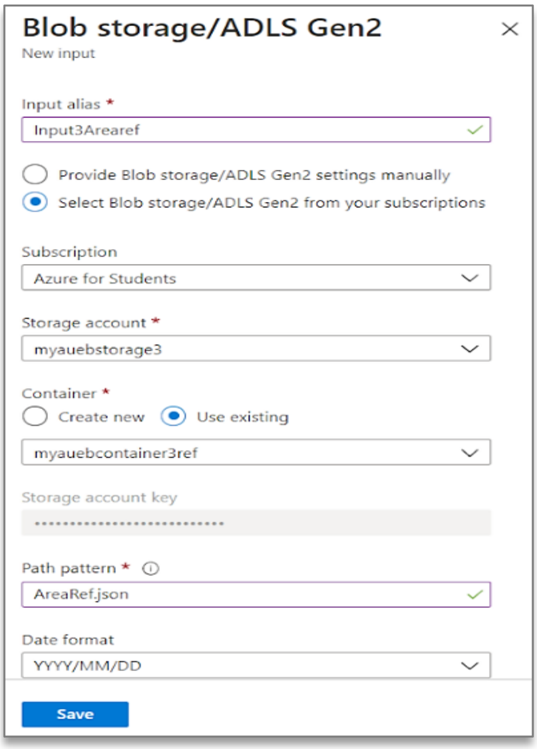
Home > Resource groups > MyRG3 > MyAuebStream3

MyAuebStream3 | Inputs

Stream Analytics job

Name
Input3Zero

Figure 24 - Add reference data input



Blob storage/ADLS Gen2

New input

Input alias *

Input3Arearef

○ Provide Blob storage/ADLS Gen2 settings manually

● Select Blob storage/ADLS Gen2 from your subscriptions

Subscription

Azure for Students

Storage account *

myauebstorage3

Container *

○ Create new ● Use existing

myauebcontainer3ref

Storage account key

••••••••••••••••••••••••••

Path pattern *

AreaRef.json

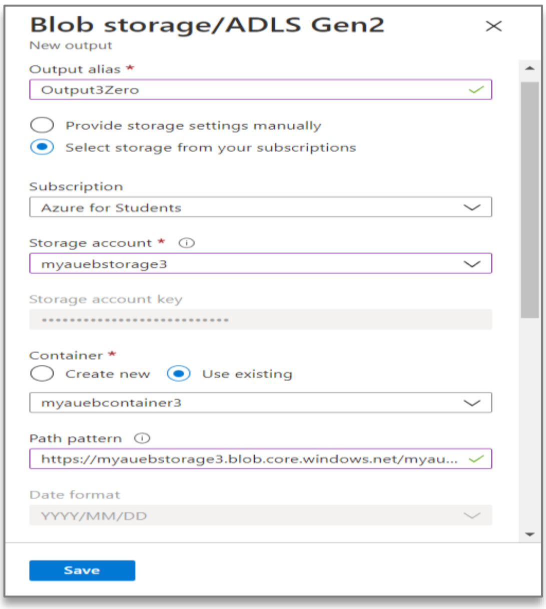
Date format

YYYY/MM/DD

Figure 25 - Reference data input creation

10. Create a Blob Storage Output.

There are several output types for sending data. In our case we constructed a single output per job and linked it with the instance container (Figure 26).



Blob storage/ADLS Gen2

New output

Output alias *

Output3Zero

○ Provide storage settings manually

● Select storage from your subscriptions

Subscription

Azure for Students

Storage account *

myauebstorage3

Storage account key

••••••••••••••••••••••••••

Container *

○ Create new ● Use existing

myauebcontainer3

Path pattern

https://myauebstorage3.blob.core.windows.net/myau...

Date format

YYYY/MM/DD

Figure 26 - Output creation

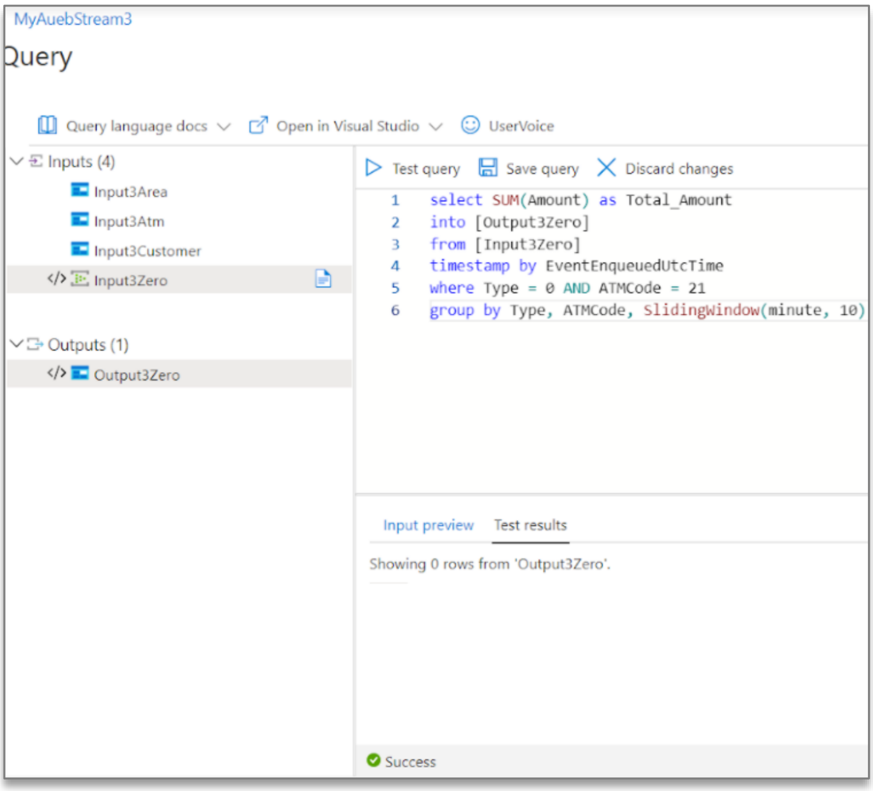
11. Queries

The SQL query language in Azure Stream Analytics gives the opportunity to carry out real-time analysis on streaming data. We created an Azure Analytics solution for the tasks listed in the “QUERIES” section, which comes as follows.

Query 1: Show the total “Amount” of “Type = 0” transactions at “ATM Code = 21” of the last 10 minutes. Repeat as new events keep flowing in (use a sliding window).

Query 1 - Code

```
select SUM(Amount) as Total_Amount
into [OutputZero]
from [InputZero]
timestamp by EventEnqueuedUtcTime
where Type = 0 AND ATMCode = 21
group by Type, ATMCode, SlidingWindow(minute, 10)
```

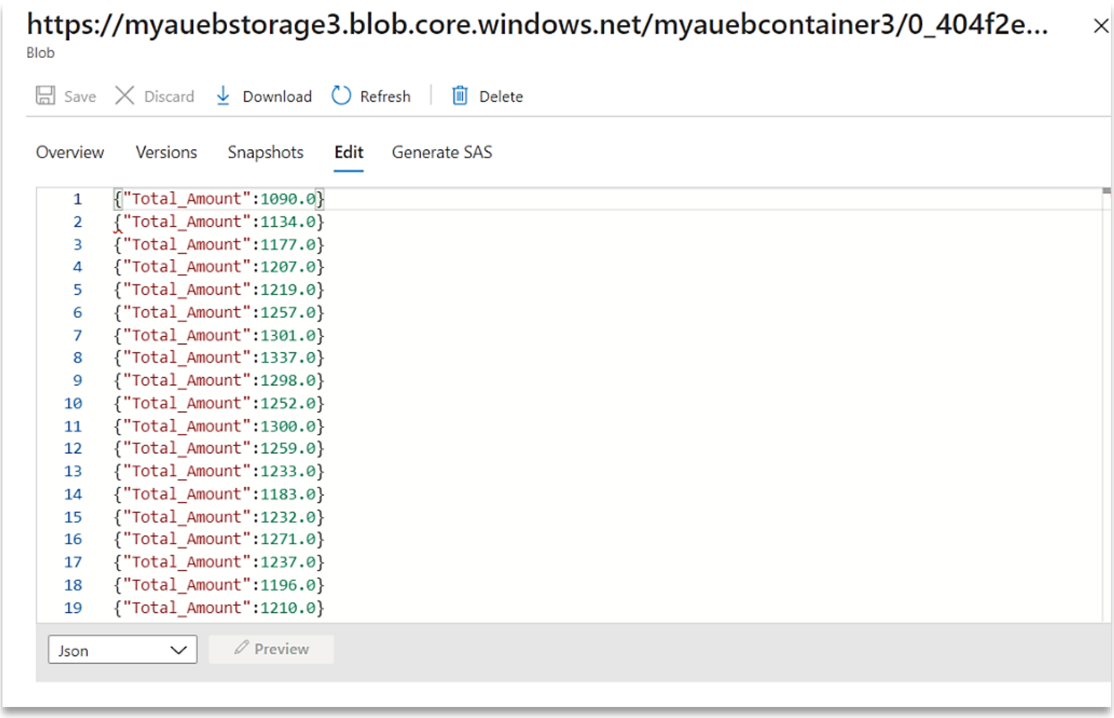


```
select SUM(Amount) as Total_Amount
into [Output3Zero]
from [Input3Zero]
timestamp by EventEnqueuedUtcTime
where Type = 0 AND ATMCode = 21
group by Type, ATMCode, SlidingWindow(minute, 10)
```

Showing 0 rows from 'Output3Zero'.

Success

Figure 27 - Query 1 Test Results



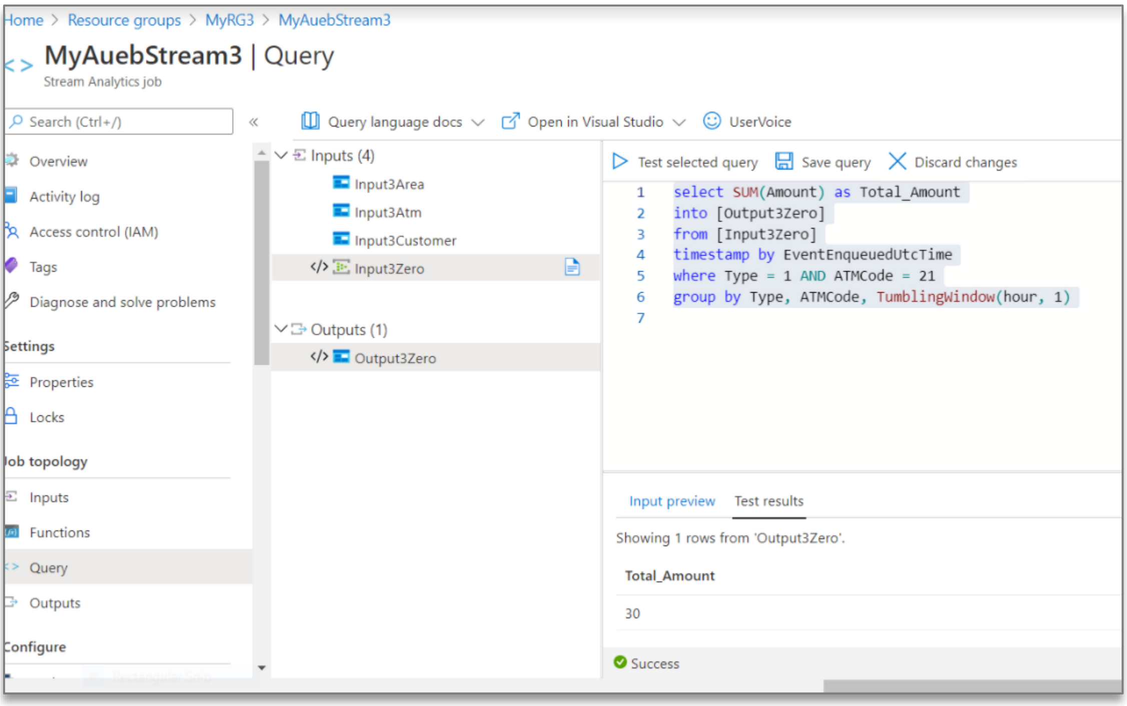
```
{"Total_Amount":1090.0}
{"Total_Amount":1134.0}
{"Total_Amount":1177.0}
{"Total_Amount":1207.0}
{"Total_Amount":1219.0}
{"Total_Amount":1257.0}
{"Total_Amount":1301.0}
{"Total_Amount":1337.0}
{"Total_Amount":1298.0}
{"Total_Amount":1252.0}
{"Total_Amount":1300.0}
{"Total_Amount":1259.0}
{"Total_Amount":1233.0}
{"Total_Amount":1183.0}
{"Total_Amount":1232.0}
{"Total_Amount":1271.0}
{"Total_Amount":1237.0}
{"Total_Amount":1196.0}
{"Total_Amount":1210.0}
```

Figure 28 - Query 1 Output

Query 2: Show the total "Amount" of "Type = 1" transactions at "ATM Code = 21" of the last hour. Repeat once every hour (use a tumbling window).

Query 2 – Code

```
select SUM(Amount) as Total_Amount
into [Output3Zero]
from [Input3Zero]
timestamp by EventEnqueuedUtcTime
where Type = 1 AND ATMCode = 21
group by Type, ATMCode, TumblingWindow(hour, 1)
```



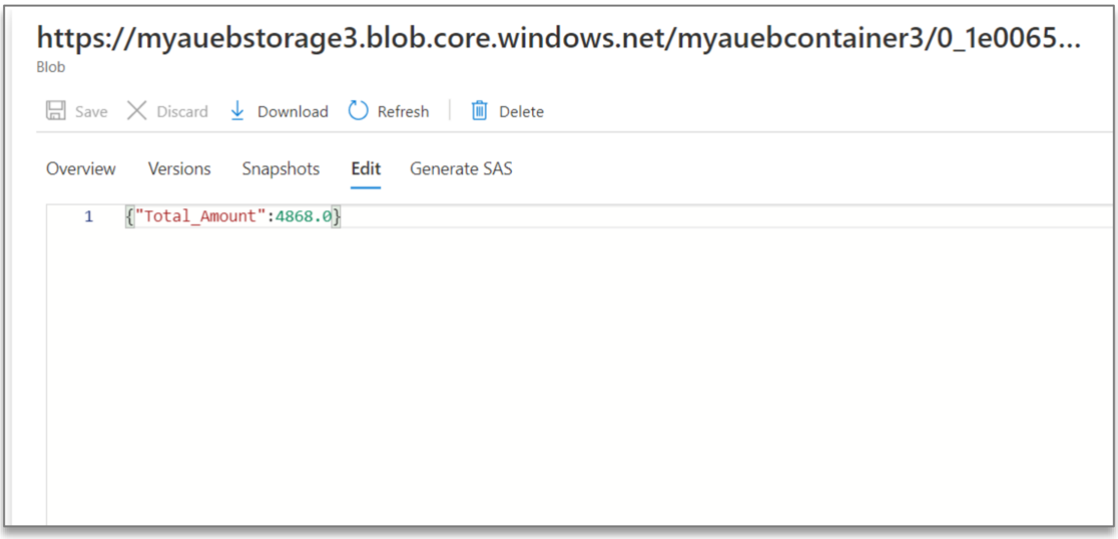
```sql
select SUM(Amount) as Total_Amount
into [Output3Zero]
from [Input3Zero]
timestamp by EventEnqueuedUtcTime
where Type = 1 AND ATMCode = 21
group by Type, ATMCode, TumblingWindow(hour, 1)
```

Showing 1 rows from 'Output3Zero'.

Total_Amount
30

✓ Success

Figure 29 - Query 2 Test Results



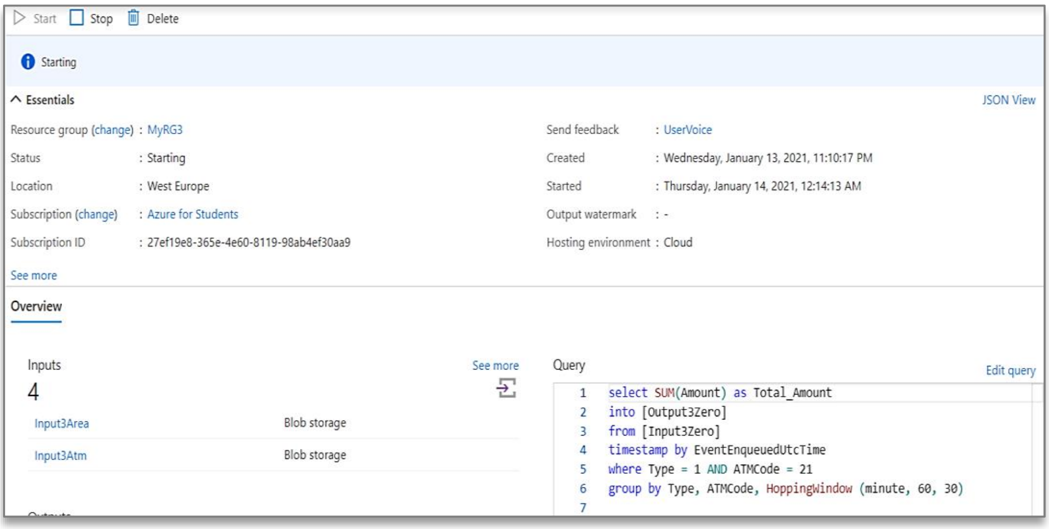
```json
{"Total_Amount":4868.0}
```

Figure 30 - Query 2 Output

Query 3: Show the total "Amount" of "Type = 1" transactions at "ATM Code = 21" of the last hour. Repeat once every 30 minutes (use a hopping window).

Query 3 – Code

```
select SUM(Amount) as Total_Amount
into [Output3Zero]
from [Input3Zero]
timestamp by EventEnqueuedUtcTime
where Type = 1 AND ATMCode = 21
group by Type, ATMCode, HoppingWindow (minute, 60, 30)
```



Resource group (change) : MyRG3
Status : Starting
Location : West Europe
Subscription (change) : Azure for Students
Subscription ID : 27ef19e8-365e-4e60-8119-98ab4ef30aa9

Send feedback : UserVoice
Created : Wednesday, January 13, 2021, 11:10:17 PM
Started : Thursday, January 14, 2021, 12:14:13 AM
Output watermark : -
Hosting environment : Cloud

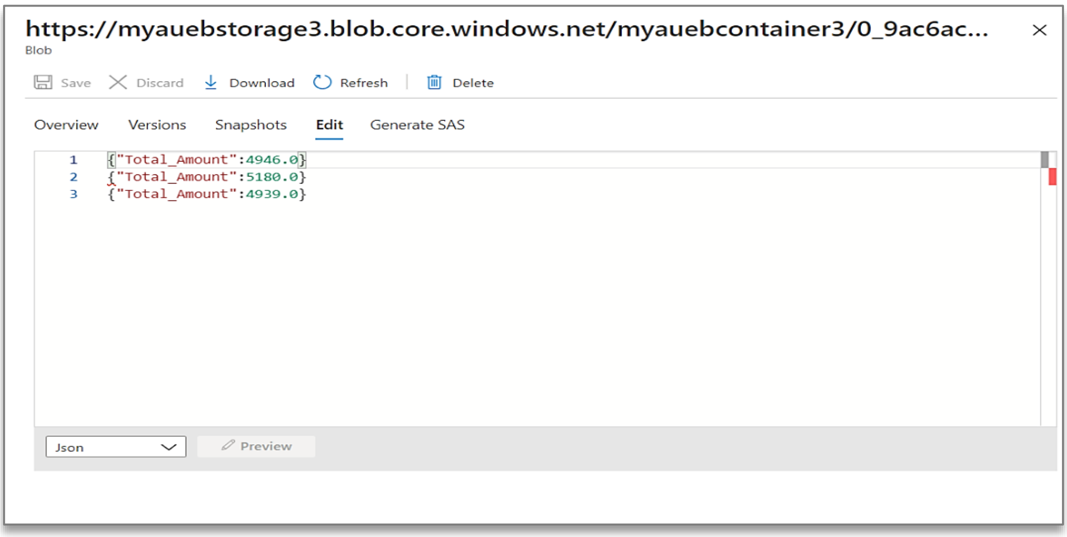
Inputs
4

Input3Area	Blob storage
Input3Atm	Blob storage

Query

```
select SUM(Amount) as Total_Amount
into [Output3Zero]
from [Input3Zero]
timestamp by EventEnqueuedUtcTime
where Type = 1 AND ATMCode = 21
group by Type, ATMCode, HoppingWindow (minute, 60, 30)
```

Figure 31 - Query 3 Test Results



https://myauebstorage3.blob.core.windows.net/myauebcontainer3/0_9ac6ac...
Blob

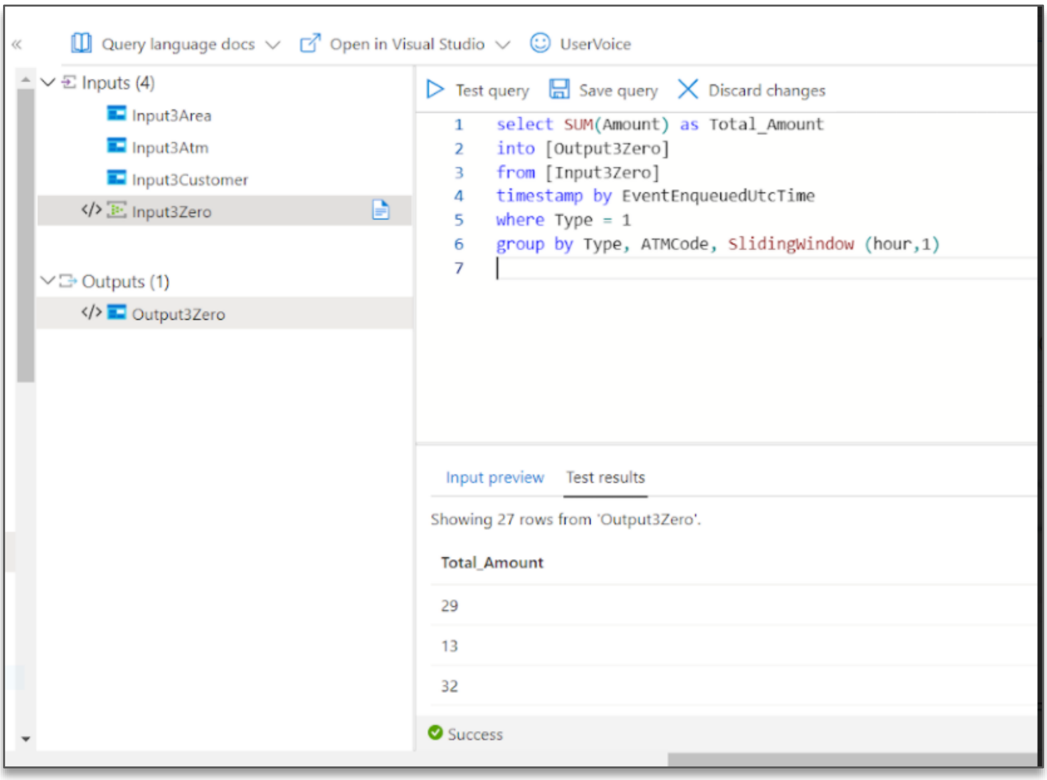
```
{"Total_Amount":4946.0}
{"Total_Amount":5180.0}
{"Total_Amount":4939.0}
```

Figure 32 - Query 3 Output

Query 4: Show the total "Amount" of "Type = 1" transactions per "ATM Code" of the last one hour (use a sliding window).

Query 4 – Code

```
select SUM(Amount) as Total_Amount
into [Output3Zero]
from [Input3Zero]
timestamp by EventEnqueuedUtcTime
where Type = 1
group by Type, ATMCode, SlidingWindow (hour,1)
```



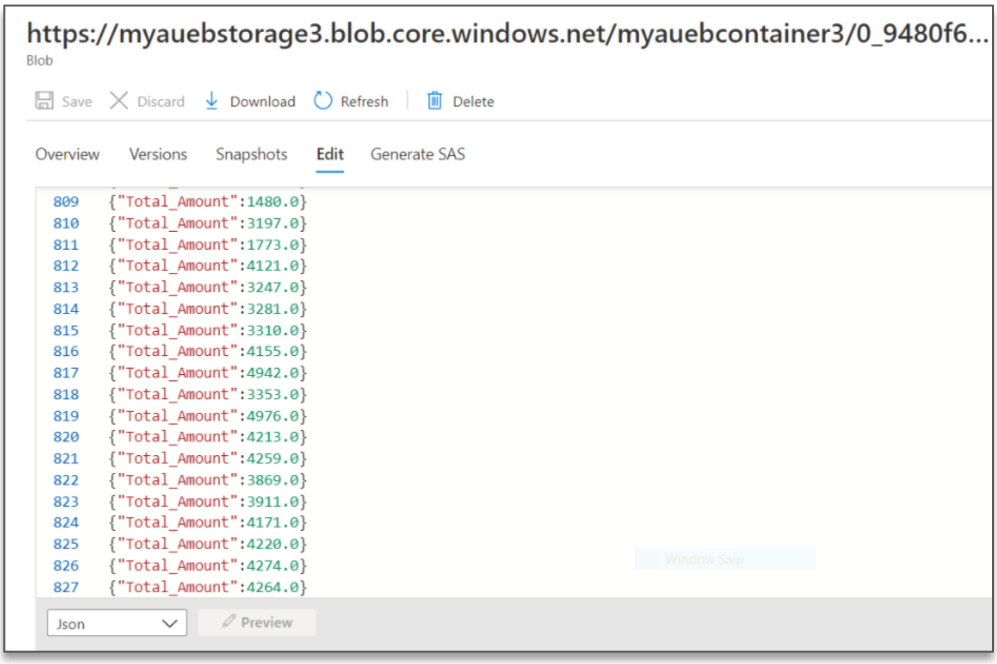
```
select SUM(Amount) as Total_Amount
into [Output3Zero]
from [Input3Zero]
timestamp by EventEnqueuedUtcTime
where Type = 1
group by Type, ATMCode, SlidingWindow (hour,1)
```

Showing 27 rows from 'Output3Zero'.

Total_Amount
29
13
32

Success

Figure 33 - Query 4 Test Results



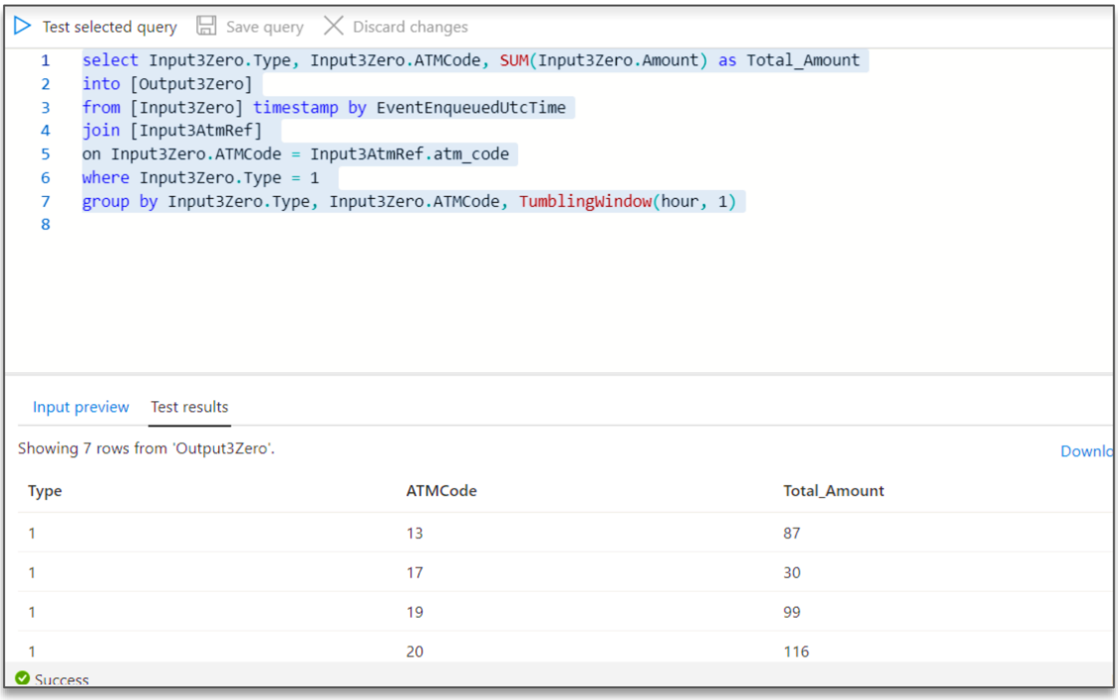
```
{"Total_Amount":1480.0}
{"Total_Amount":3197.0}
{"Total_Amount":1773.0}
{"Total_Amount":4121.0}
{"Total_Amount":3247.0}
{"Total_Amount":3281.0}
{"Total_Amount":3310.0}
{"Total_Amount":4155.0}
{"Total_Amount":4942.0}
{"Total_Amount":3353.0}
{"Total_Amount":4976.0}
{"Total_Amount":4213.0}
{"Total_Amount":4259.0}
{"Total_Amount":3869.0}
{"Total_Amount":3911.0}
{"Total_Amount":4171.0}
{"Total_Amount":4220.0}
{"Total_Amount":4274.0}
{"Total_Amount":4264.0}
```

Figure 34 - Query 4 Output

Query 5: Show the total “Amount” of “Type = 1” transactions per “Area Code” of the last hour. Repeat once every hour (use a tumbling window).

Query 5 – Code

```
select Input3Zero.Type, Input3Zero.ATMCode,
SUM(Input3Zero.Amount) as Total_Amount,
into [Output3Zero]
from [Input3Zero] timestamp by EventEnqueuedUtcTime
join [Input3AtmRef]
on Input3Zero.ATMCode = Input3AtmRef.atm_code
where Input3Zero.Type = 1
group by Input3Zero.Type, Input3Zero.ATMCode,
TumblingWindow(hour, 1)
```

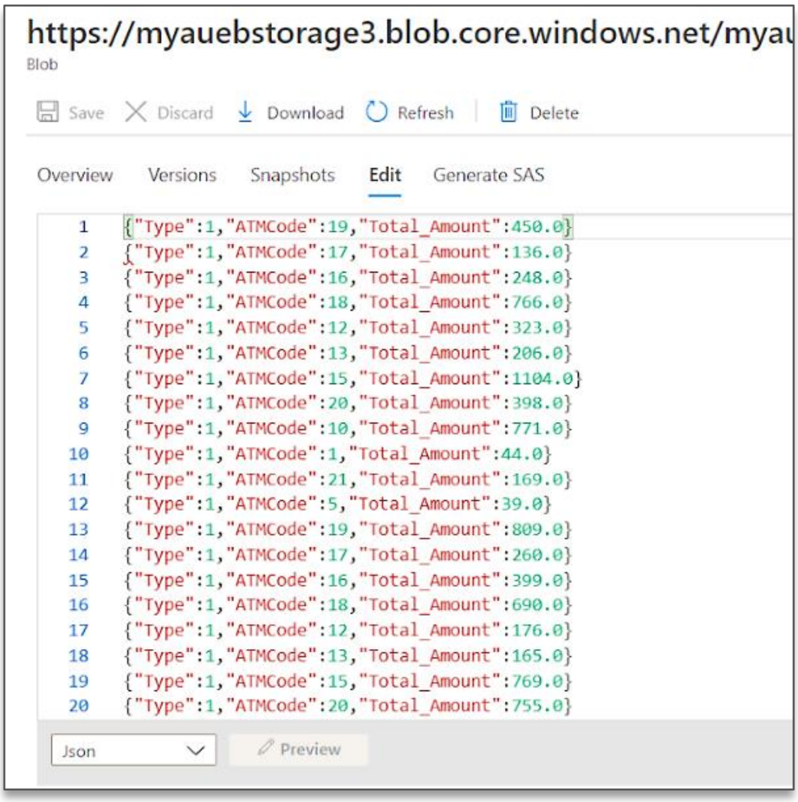


```sql
select Input3Zero.Type, Input3Zero.ATMCode, SUM(Input3Zero.Amount) as Total_Amount
into [Output3Zero]
from [Input3Zero] timestamp by EventEnqueuedUtcTime
join [Input3AtmRef]
on Input3Zero.ATMCode = Input3AtmRef.atm_code
where Input3Zero.Type = 1
group by Input3Zero.Type, Input3Zero.ATMCode, TumblingWindow(hour, 1)
```

Showing 7 rows from 'Output3Zero'.

Type	ATMCode	Total_Amount
1	13	87
1	17	30
1	19	99
1	20	116

Figure 35 - Query 5 Test Results



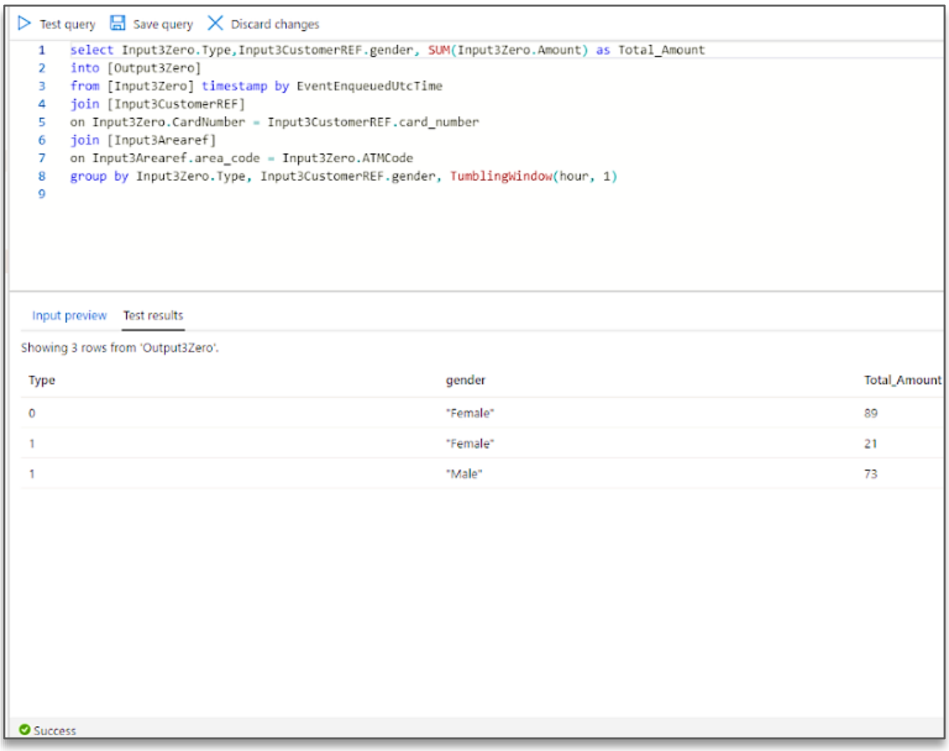
```json
{"Type":1,"ATMCode":19,"Total_Amount":450.0}
{"Type":1,"ATMCode":17,"Total_Amount":136.0}
{"Type":1,"ATMCode":16,"Total_Amount":248.0}
{"Type":1,"ATMCode":18,"Total_Amount":766.0}
{"Type":1,"ATMCode":12,"Total_Amount":323.0}
{"Type":1,"ATMCode":13,"Total_Amount":206.0}
{"Type":1,"ATMCode":15,"Total_Amount":1104.0}
{"Type":1,"ATMCode":20,"Total_Amount":398.0}
{"Type":1,"ATMCode":10,"Total_Amount":771.0}
{"Type":1,"ATMCode":1,"Total_Amount":44.0}
{"Type":1,"ATMCode":21,"Total_Amount":169.0}
{"Type":1,"ATMCode":5,"Total_Amount":39.0}
{"Type":1,"ATMCode":19,"Total_Amount":809.0}
{"Type":1,"ATMCode":17,"Total_Amount":260.0}
{"Type":1,"ATMCode":16,"Total_Amount":399.0}
{"Type":1,"ATMCode":18,"Total_Amount":690.0}
{"Type":1,"ATMCode":12,"Total_Amount":176.0}
{"Type":1,"ATMCode":13,"Total_Amount":165.0}
{"Type":1,"ATMCode":15,"Total_Amount":769.0}
{"Type":1,"ATMCode":20,"Total_Amount":755.0}
```

Figure 36 - Query 5 Output

Query 6: Show the total “Amount” per ATM’s “City” and Customer’s “Gender” of the last hour. Repeat once every hour (use a tumbling window).

Query 6 – Code

```
select Input3Zero.Type,Input3CustomerREF.gender,
SUM(Input3Zero.Amount) as Total_Amount
into [Output3Zero]
from [Input3Zero] timestamp by EventEnqueuedUtcTime
join [Input3CustomerREF]
on Input3Zero.CardNumber = Input3CustomerREF.card_number
join [Input3Arearef]
on Input3Arearef.area_code = Input3Zero.ATMCode
group by Input3Zero.Type, Input3CustomerREF.gender,
TumblingWindow(hour, 1)
```

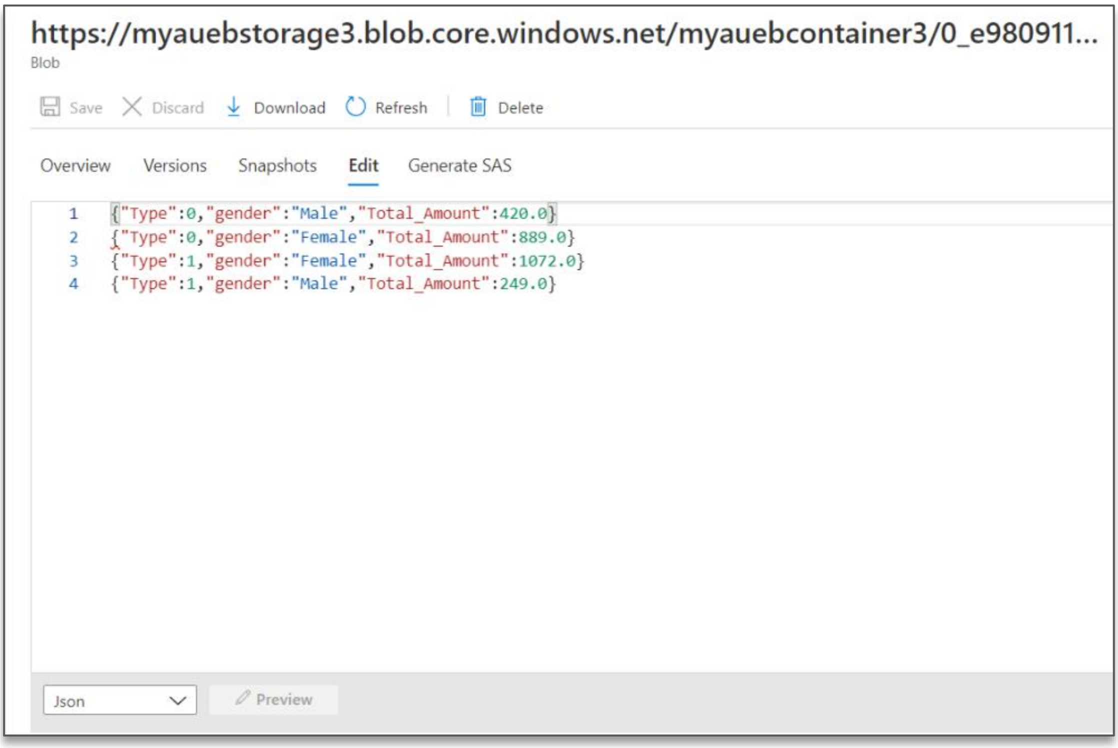


```
select Input3Zero.Type,Input3CustomerREF.gender, SUM(Input3Zero.Amount) as Total_Amount
into [Output3Zero]
from [Input3Zero] timestamp by EventEnqueuedUtcTime
join [Input3CustomerREF]
on Input3Zero.CardNumber = Input3CustomerREF.card_number
join [Input3Arearef]
on Input3Arearef.area_code = Input3Zero.ATMCode
group by Input3Zero.Type, Input3CustomerREF.gender, TumblingWindow(hour, 1)
```

Showing 3 rows from 'Output3Zero'.

Type	gender	Total_Amount
0	"Female"	89
1	"Female"	21
1	"Male"	73

Figure 37 - Query 6 Test Results



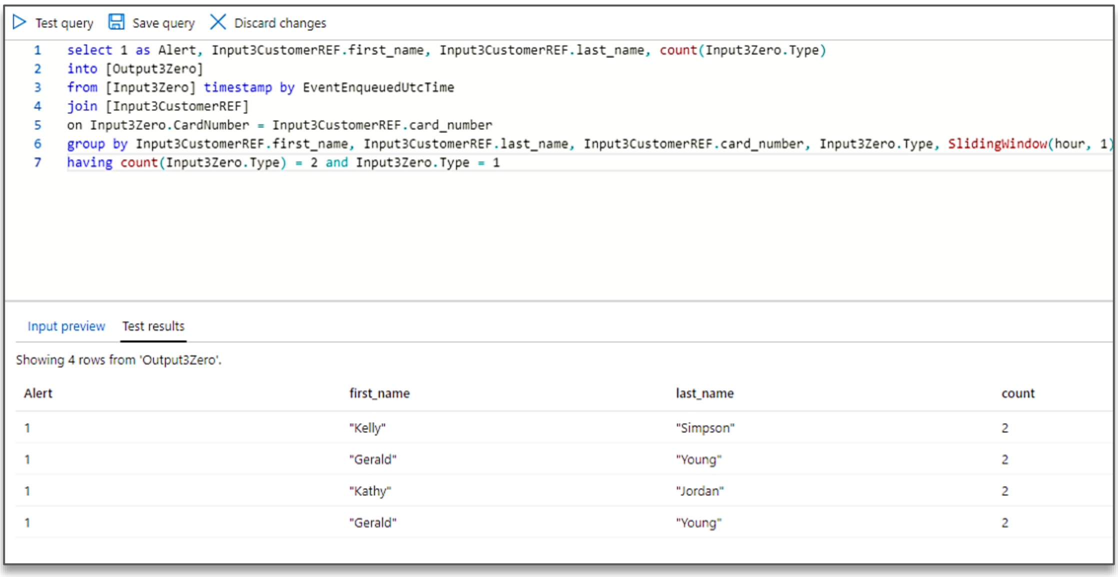
```
{"Type":0,"gender":"Male","Total_Amount":420.0}
{"Type":0,"gender":"Female","Total_Amount":889.0}
{"Type":1,"gender":"Female","Total_Amount":1072.0}
{"Type":1,"gender":"Male","Total_Amount":249.0}
```

Figure 38 - Query 6 Output

Query 7: Alert (Do a simple SELECT “1”) if a Customer has performed two transactions of “Type = 1” in a window of an hour (use a sliding window).

Query 7 – Code

```
select 1 as Alert, Input3CustomerREF.first_name,
Input3CustomerREF.last_name, count(Input3Zero.Type)
into [Output3Zero]
from [Input3Zero] timestamp by EventEnqueuedUtcTime
join [Input3CustomerREF]
on Input3Zero.CardNumber = Input3CustomerREF.card_number
group by Input3CustomerREF.first_name,
Input3CustomerREF.last_name,
Input3CustomerREF.card_number, Input3Zero.Type,
SlidingWindow(hour, 1)
having count(Input3Zero.Type) = 2 and Input3Zero.Type = 1
```

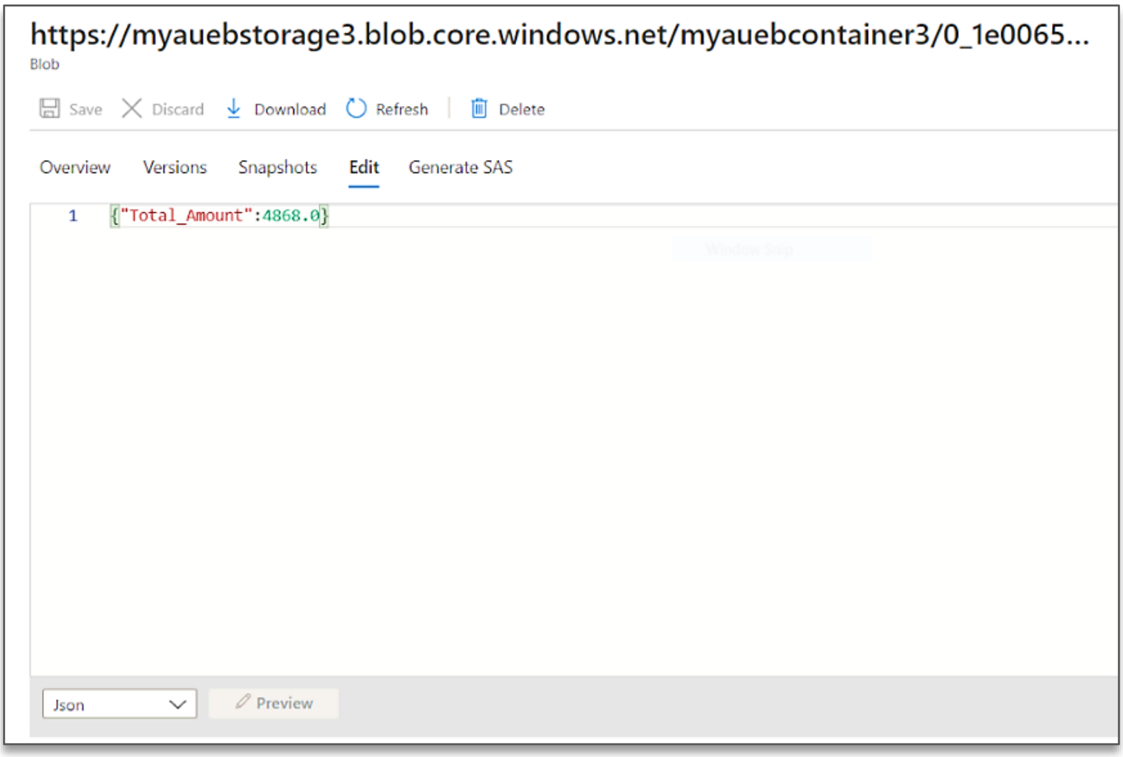


```
select 1 as Alert, Input3CustomerREF.first_name, Input3CustomerREF.last_name, count(Input3Zero.Type)
into [Output3Zero]
from [Input3Zero] timestamp by EventEnqueuedUtcTime
join [Input3CustomerREF]
on Input3Zero.CardNumber = Input3CustomerREF.card_number
group by Input3CustomerREF.first_name, Input3CustomerREF.last_name, Input3CustomerREF.card_number, Input3Zero.Type, SlidingWindow(hour, 1)
having count(Input3Zero.Type) = 2 and Input3Zero.Type = 1
```

Showing 4 rows from 'Output3Zero'.

Alert	first_name	last_name	count
1	"Kelly"	"Simpson"	2
1	"Gerald"	"Young"	2
1	"Kathy"	"Jordan"	2
1	"Gerald"	"Young"	2

Figure 39 - Query 7 Test Results


https://myauebstorage3.blob.core.windows.net/myauebcontainer3/0_1e0065...

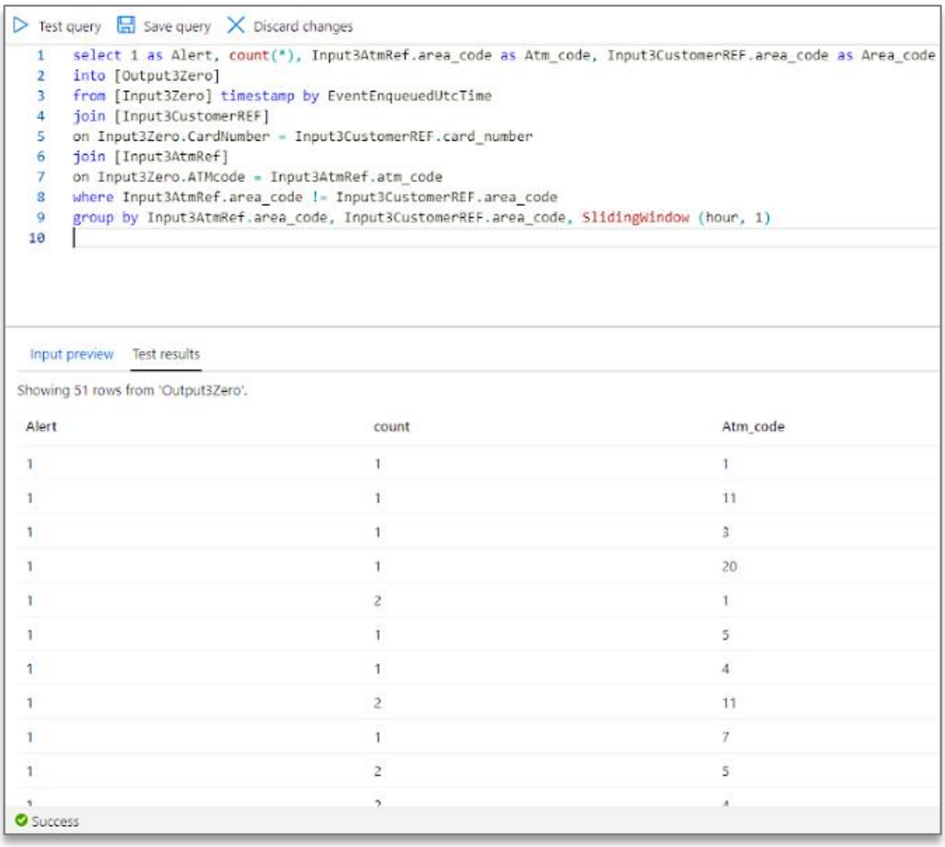
```
{"Total_Amount":4868.0}
```

Figure 40 - Query 7 Output

Query 8: Alert (Do a simple SELECT “1”) if the “Area Code” of the ATM of the transaction is not the same as the “Area Code” of the “Card Number” (Customer’s Area Code) - (use a sliding window)

Query 8 – Code

```
select 1 as Alert, count(*), Input3AtmRef.area_code as
Atm_code, Input3CustomerREF.area_code as Area_code
into [Output3Zero]
from [Input3Zero] timestamp by EventEnqueuedUtcTime
join [Input3CustomerREF]
on Input3Zero.CardNumber = Input3CustomerREF.card_number
join [Input3AtmRef]
on Input3Zero.ATMcode = Input3AtmRef.atm_code
where Input3AtmRef.area_code !=
Input3CustomerREF.area_code
group by Input3AtmRef.area_code,
Input3CustomerREF.area_code, SlidingWindow (hour, 1)
```



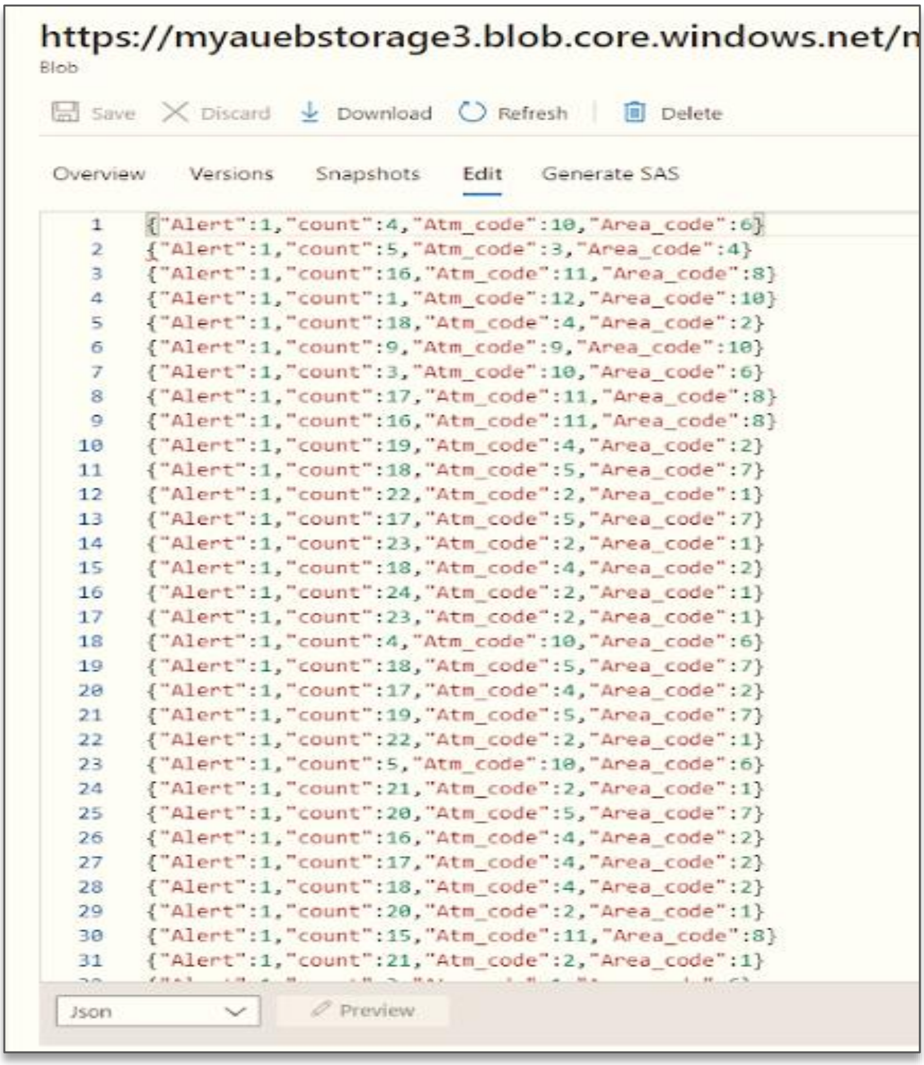
```
select 1 as Alert, count(*), Input3AtmRef.area_code as Atm_code, Input3CustomerREF.area_code as Area_code
into [Output3Zero]
from [Input3Zero] timestamp by EventEnqueuedUtcTime
join [Input3CustomerREF]
on Input3Zero.CardNumber = Input3CustomerREF.card_number
join [Input3AtmRef]
on Input3Zero.ATMcode = Input3AtmRef.atm_code
where Input3AtmRef.area_code != Input3CustomerREF.area_code
group by Input3AtmRef.area_code, Input3CustomerREF.area_code, SlidingWindow (hour, 1)
```

Showing 51 rows from 'Output3Zero'.

Alert	count	Atm_code
1	1	1
1	1	11
1	1	3
1	1	20
1	2	1
1	1	5
1	1	4
1	2	11
1	1	7
1	2	5
1	2	4

✓ Success

Figure 41- Query 8 Test Results



https://myauebstorage3.blob.core.windows.net/n

```
{"Alert":1,"count":4,"Atm_code":10,"Area_code":6}
{"Alert":1,"count":5,"Atm_code":3,"Area_code":4}
{"Alert":1,"count":16,"Atm_code":11,"Area_code":8}
{"Alert":1,"count":1,"Atm_code":12,"Area_code":10}
{"Alert":1,"count":18,"Atm_code":4,"Area_code":2}
{"Alert":1,"count":9,"Atm_code":9,"Area_code":10}
{"Alert":1,"count":3,"Atm_code":10,"Area_code":6}
{"Alert":1,"count":17,"Atm_code":11,"Area_code":8}
{"Alert":1,"count":16,"Atm_code":11,"Area_code":8}
{"Alert":1,"count":19,"Atm_code":4,"Area_code":2}
{"Alert":1,"count":18,"Atm_code":5,"Area_code":7}
{"Alert":1,"count":22,"Atm_code":2,"Area_code":1}
{"Alert":1,"count":17,"Atm_code":5,"Area_code":7}
{"Alert":1,"count":23,"Atm_code":2,"Area_code":1}
{"Alert":1,"count":18,"Atm_code":4,"Area_code":2}
{"Alert":1,"count":24,"Atm_code":2,"Area_code":1}
{"Alert":1,"count":23,"Atm_code":2,"Area_code":1}
{"Alert":1,"count":4,"Atm_code":10,"Area_code":6}
{"Alert":1,"count":18,"Atm_code":5,"Area_code":7}
{"Alert":1,"count":17,"Atm_code":4,"Area_code":2}
{"Alert":1,"count":19,"Atm_code":5,"Area_code":7}
{"Alert":1,"count":22,"Atm_code":2,"Area_code":1}
{"Alert":1,"count":5,"Atm_code":10,"Area_code":6}
{"Alert":1,"count":21,"Atm_code":2,"Area_code":1}
{"Alert":1,"count":20,"Atm_code":5,"Area_code":7}
{"Alert":1,"count":16,"Atm_code":4,"Area_code":2}
{"Alert":1,"count":17,"Atm_code":4,"Area_code":2}
{"Alert":1,"count":18,"Atm_code":4,"Area_code":2}
{"Alert":1,"count":20,"Atm_code":2,"Area_code":1}
{"Alert":1,"count":15,"Atm_code":11,"Area_code":8}
{"Alert":1,"count":21,"Atm_code":2,"Area_code":1}
```

Figure 42 - Query 8 Output

References

- Azure Stream Analytics documentation
  https://docs.microsoft.com/en-us/azure/stream-analytics/