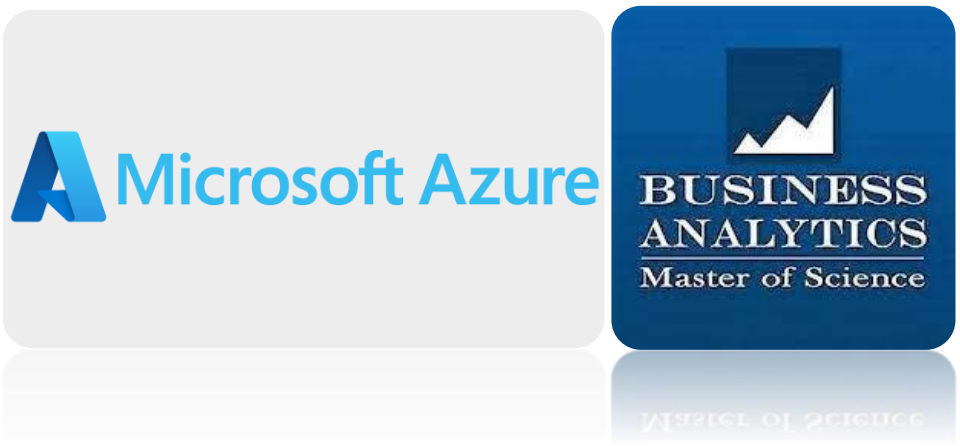


Assignment 3: Data Streams

Course: Data Management and Business Intelligence

Professors: D.Chatziantoniou, S.Safras

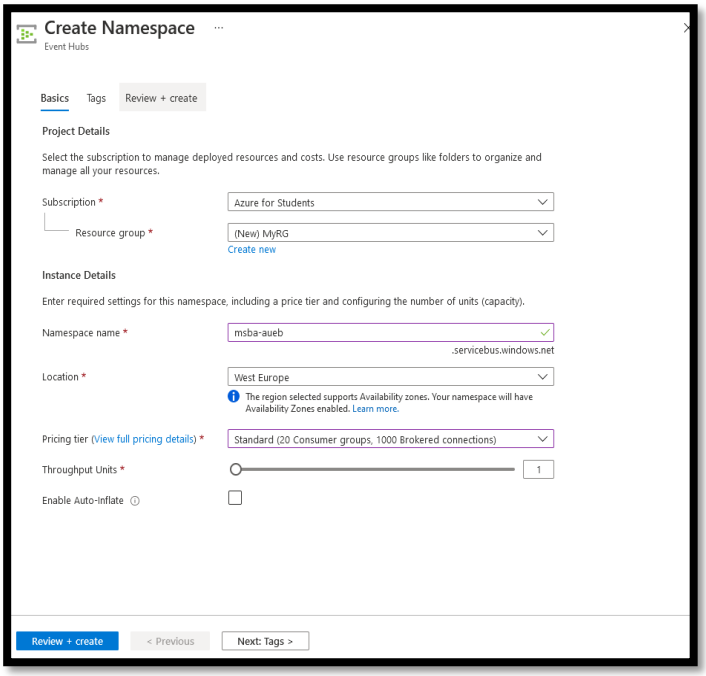
Students: Ch.Petrakogiannis (F2822112), I.Dimos (F2822102)

TABLE OF CONTENTS

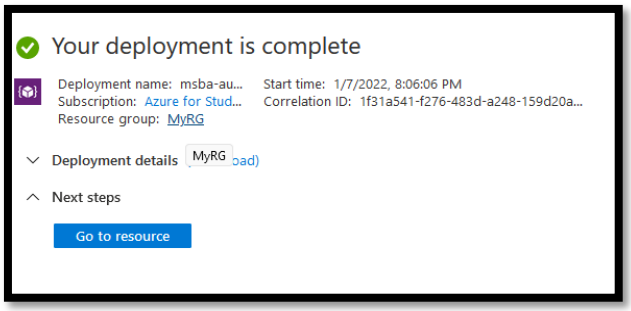
INTRODUCTION

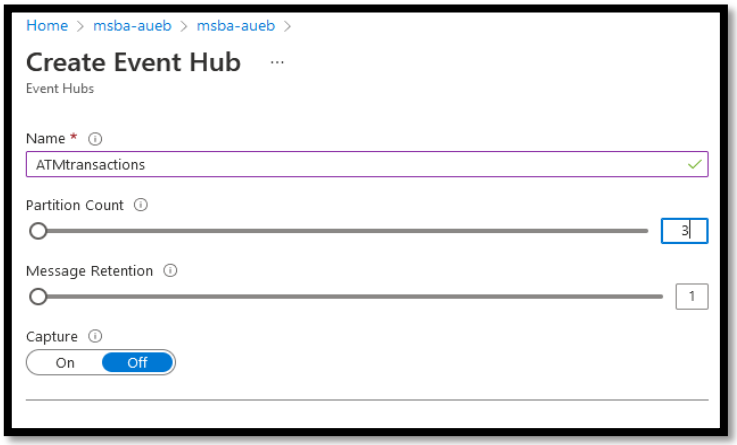
In this specific assignment we aim to work using Azure Stream Analytics in order for us to process a data stream of ATM transactions and give answers to some specified stream queries. For this reason, firstly we had to create a student account on Azure platform and then proceed with the rest of the appropriate steps, that we are going to present in the sections below, in order to succeed our goal.

EVENT HUB

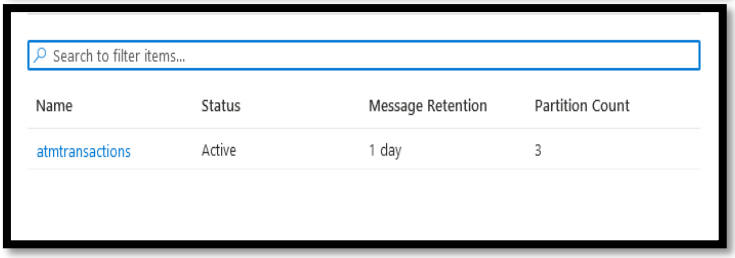


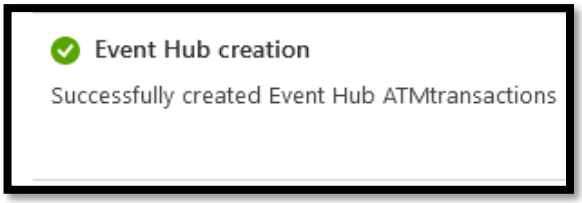
Comment: In first place we had to deploy the event hub feature using a new resource group through our students account.

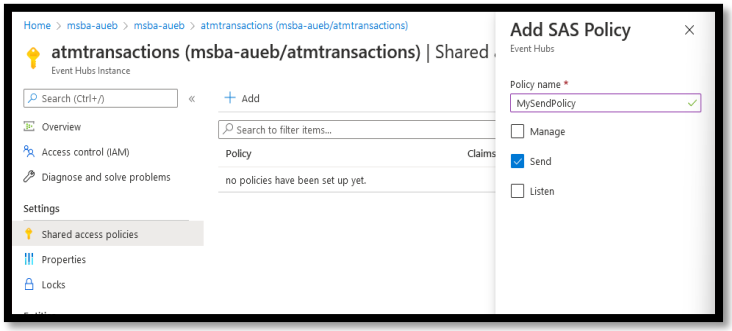




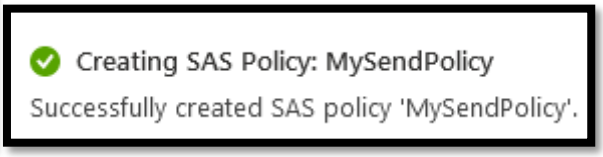
Comment: Then we had to proceed on the creation of a new event hub.

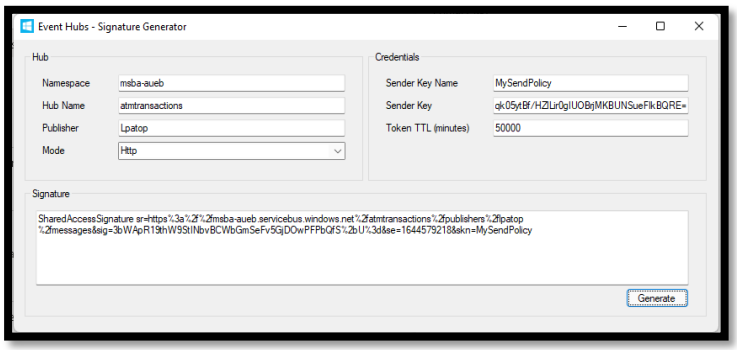






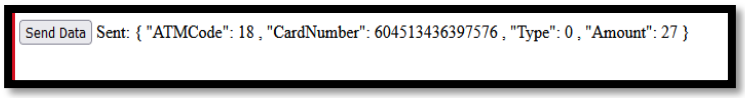
Comment: Later on, we had to set up a send policy on this new event hub that confers the right to send messages to an entity.



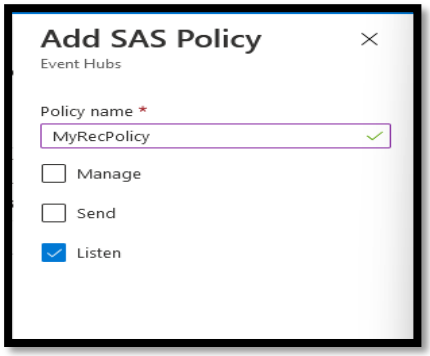


Comment: The next step had to do with the generator in which we updated the CONFIG variables and then we fed the event hub with the use of Generator.html in order to start the data stream through a browser.

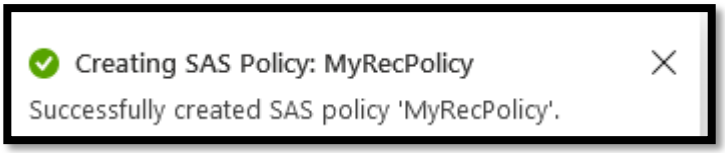
```
//Use the signature generator: https://github.com/sandrinodimattia/RedDog/releases
var sas = "SharedAccessSignature sr=https%3a%2f%2fmsba-aueb.servicebus.windows.net
%2fatmtransactions%2fpublishers%2flpatop%2fmessages&sig=3bWApR19thW9StINbvBCWbGmSeFv5GjDOwPFPbQfS
%2bU%3d&se=1644579218&skn=MySendPolicy";
var serviceNamespace = "msba-aueb";
var hubName = "atmtransactions";
var deviceName = "Lpatop";
```



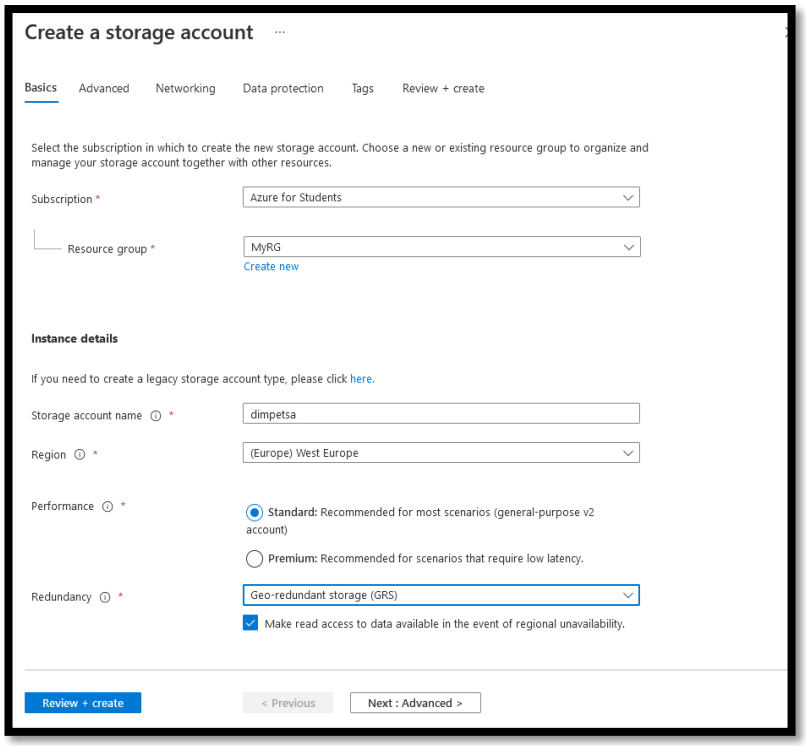
Comment: The stream has worked successfully!

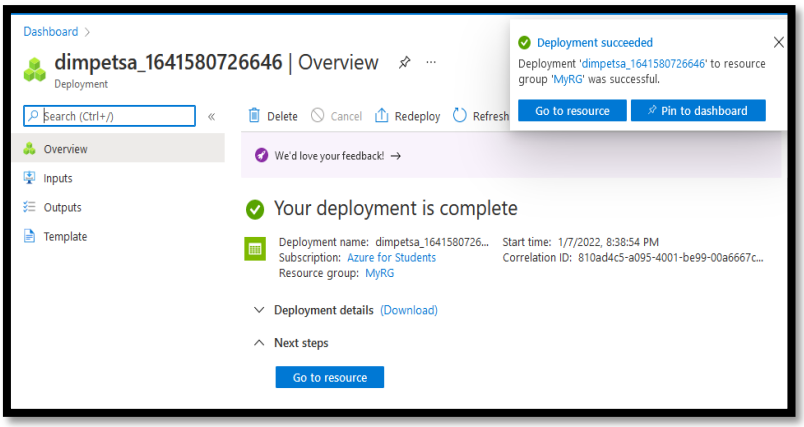


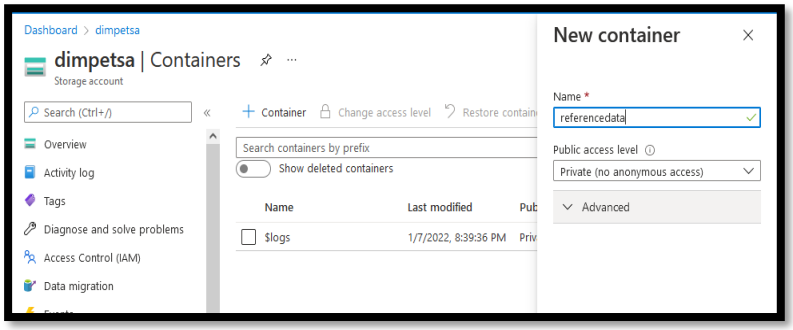
Comment: Next in queue was the creation of a receive policy in order for our event hub to obtain the data from the stream.



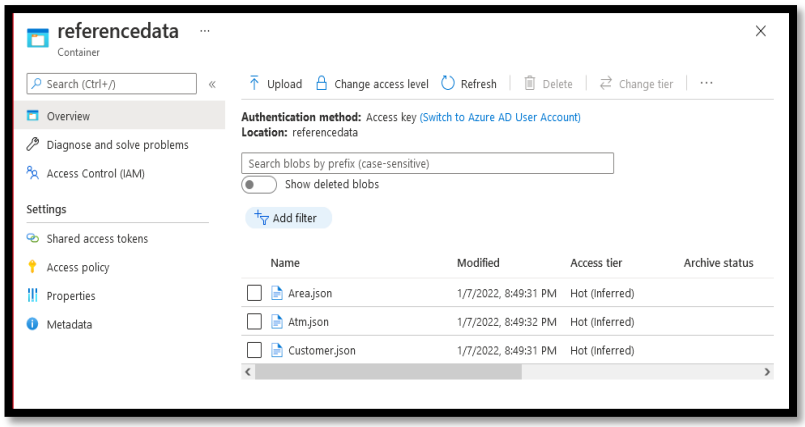
STORAGE ACCOUNT





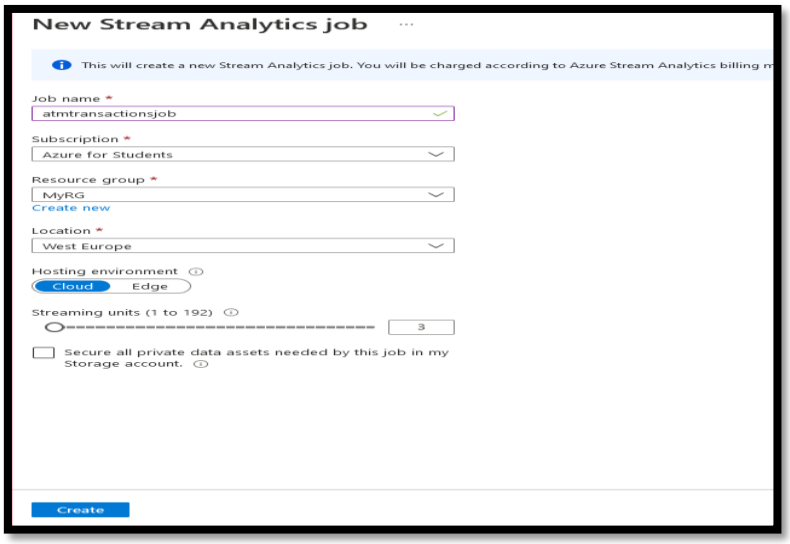


Comment: After the successful creation of the event hub and the data stream process, we had to create a storage account and use its container for uploading the reference data json files.

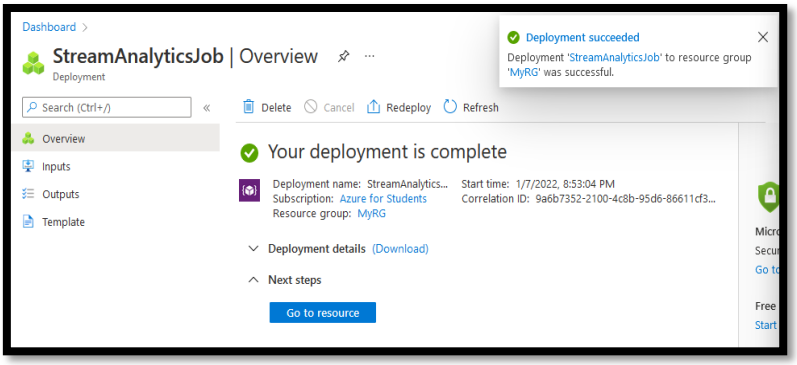


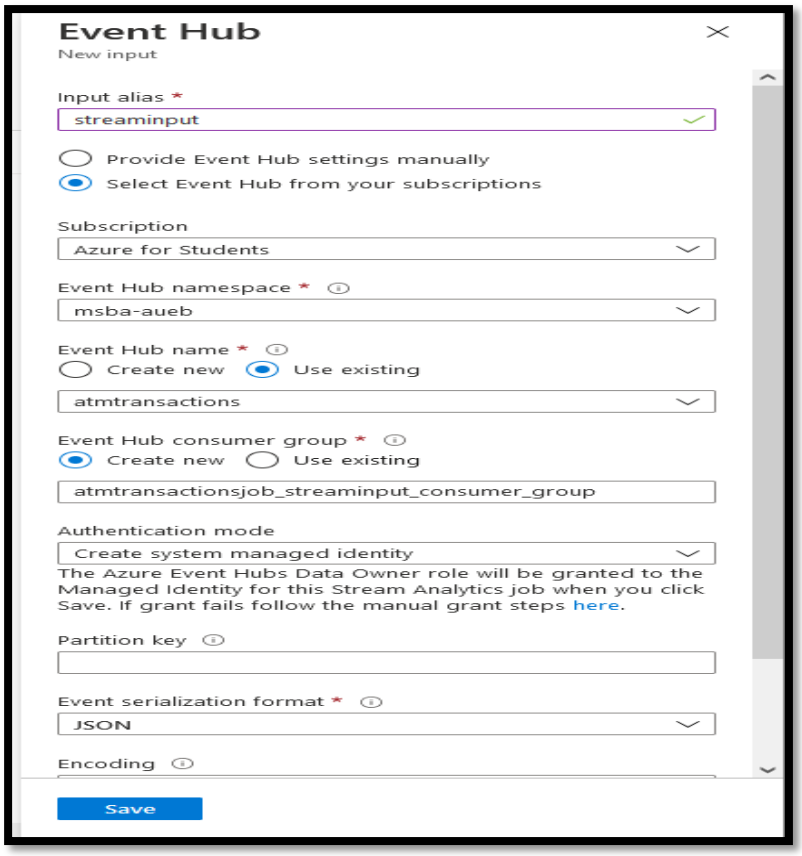
Comment: We have uploaded the files successfully!

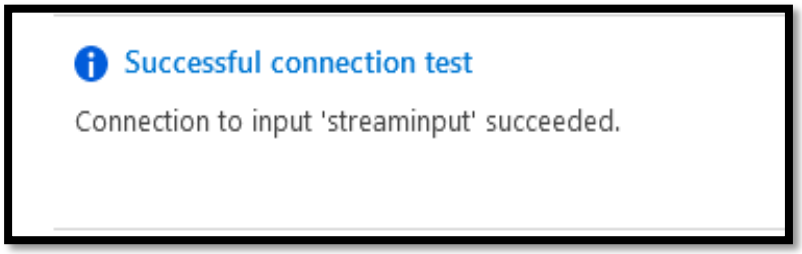
STREAM ANALYTICS JOB



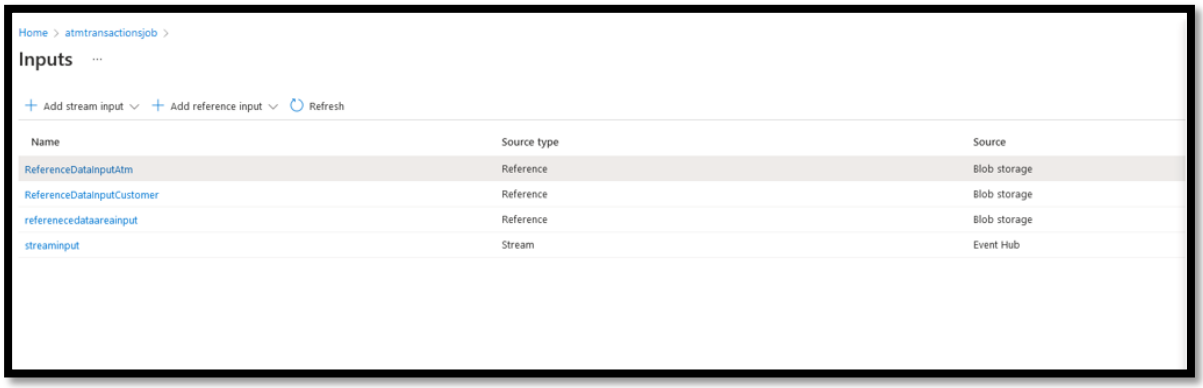
Comment: The next step for us in order to come closer to the end, was to set up a new stream analytics job and use the Event Hub + Reference Data Files as input.

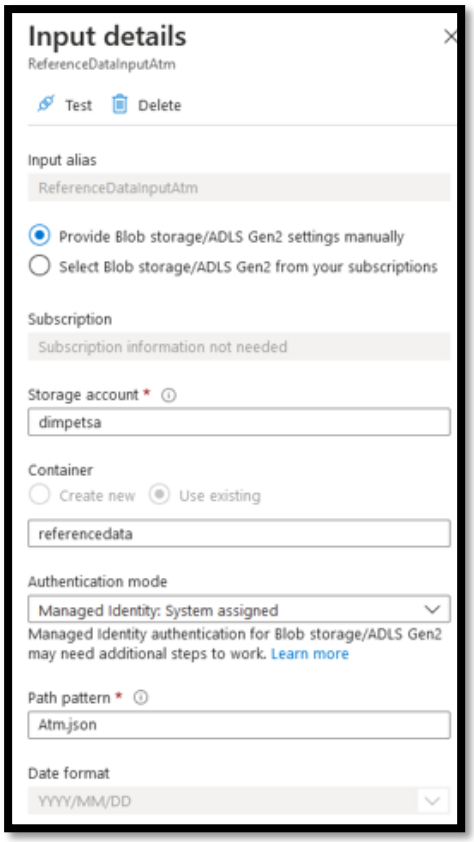


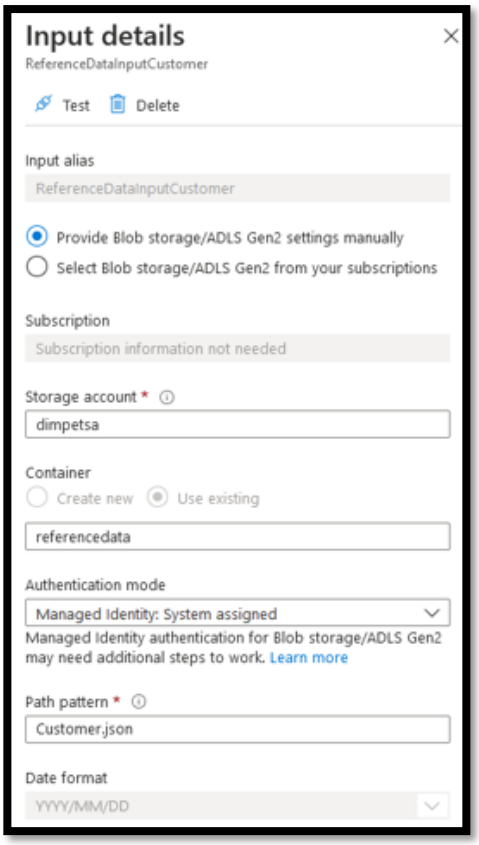


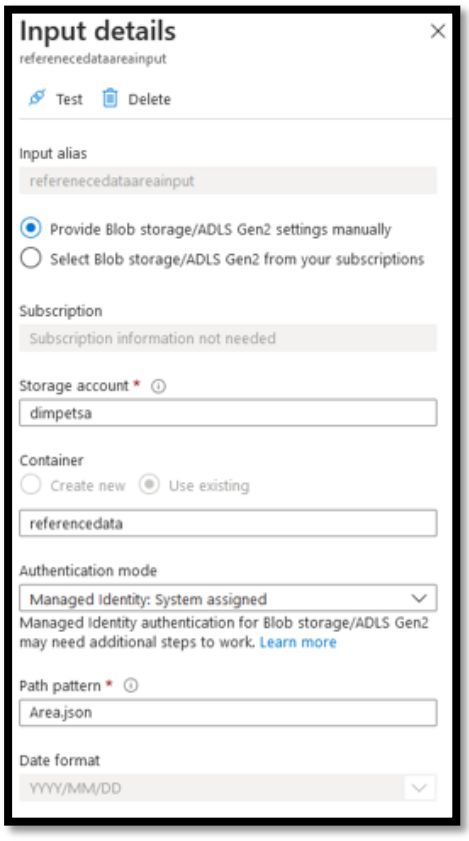


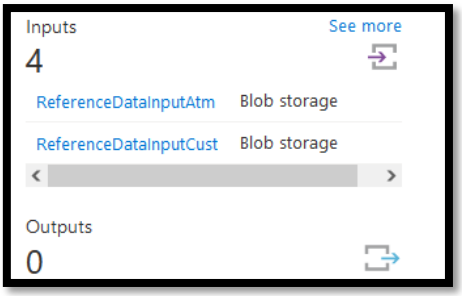
Comment: After the Event Hub input we had to upload the reference data and connect them with the stream analytics job.

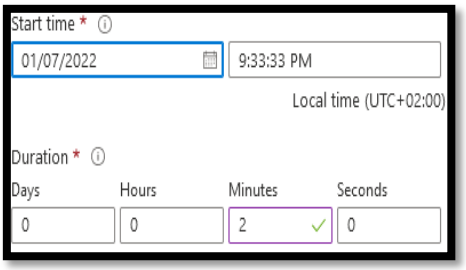


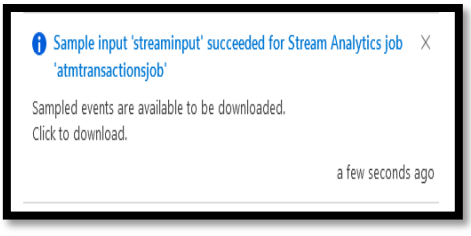






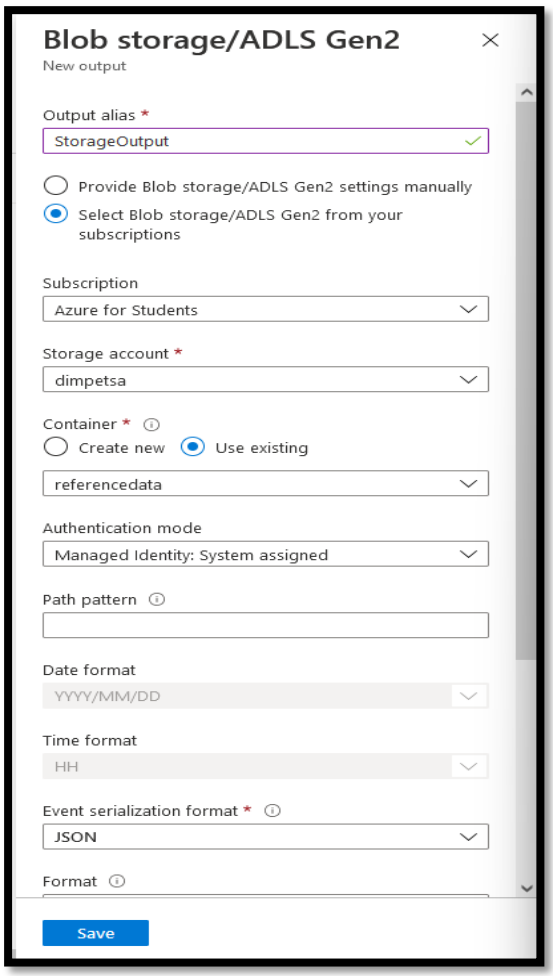




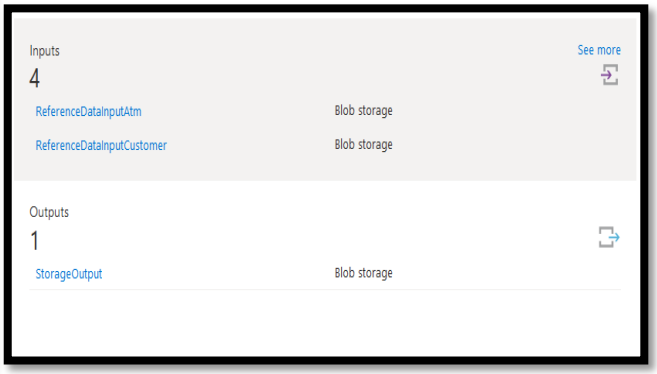


Comment: The 4 inputs have been loaded and we are able to download the sample!

BLOB STORAGE OUTPUT



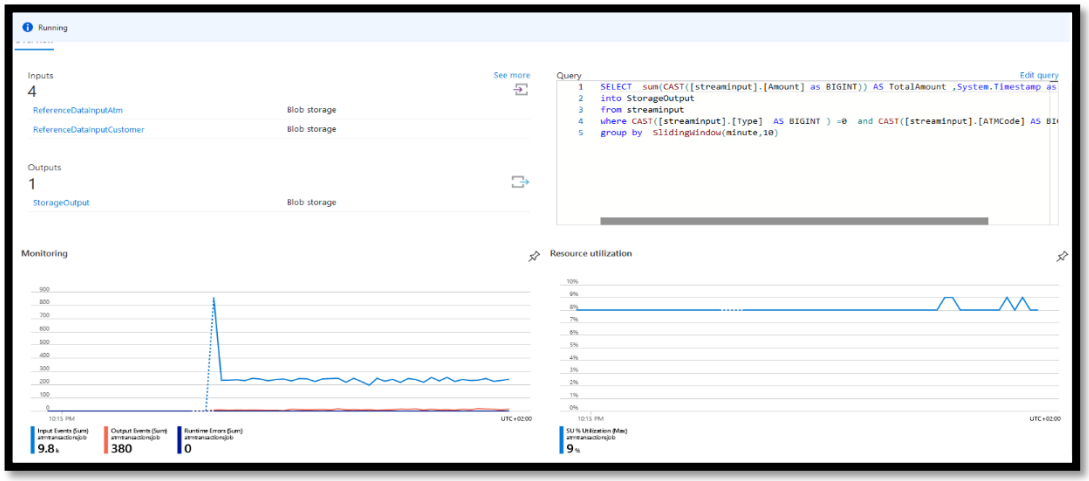
Comment: Last but not least, we had to create a blob storage output in which we would be able to save the results of our stream queries.

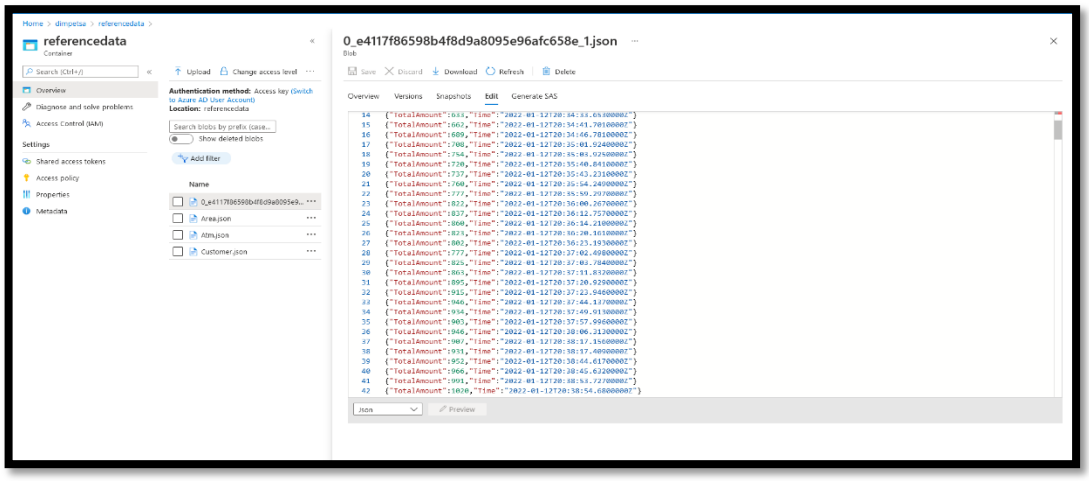


QUERIES

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).

Solution:

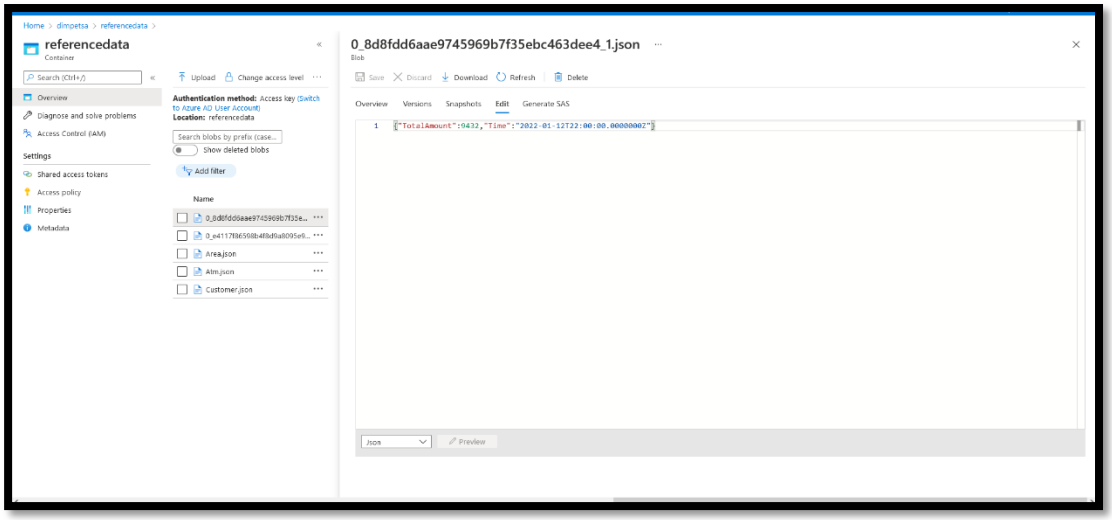




Comment: As we can see from the images above, the query has worked successfully ending up with results from both the sample file and the live data stream. In this specific query, but also in the next ones, we use the “CAST…AS BIGINT” function on a variety of columns that contain numeric data that can become extremely large. Doing that we ensure that the whole number will be visible to us.

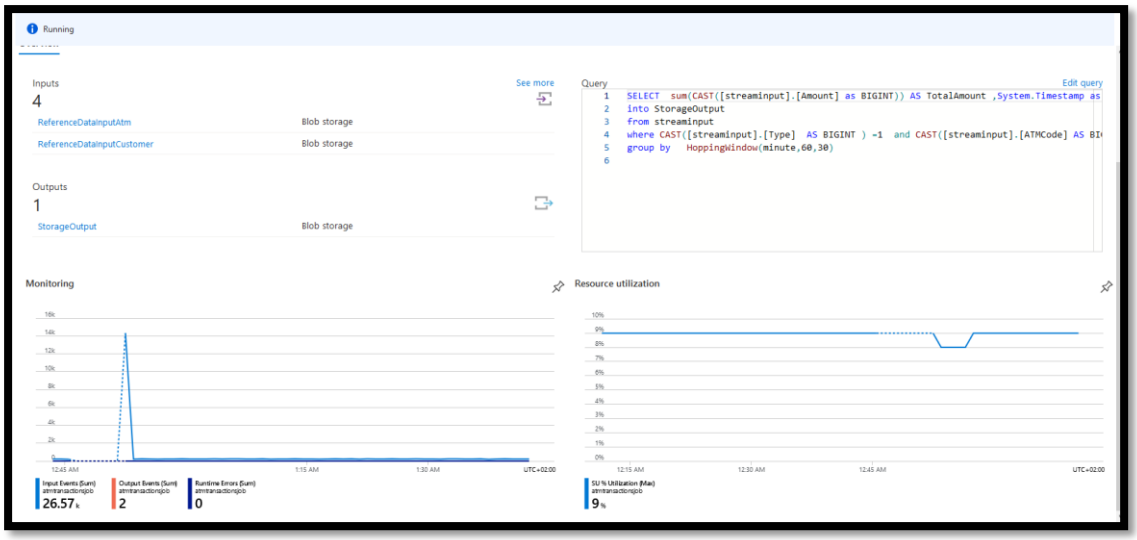
Query 2: Show the total “Amount” of “Type = 1” transactions at “ATM Code = 21” of the last hour. Repeat once every hour (use a tumbling window).

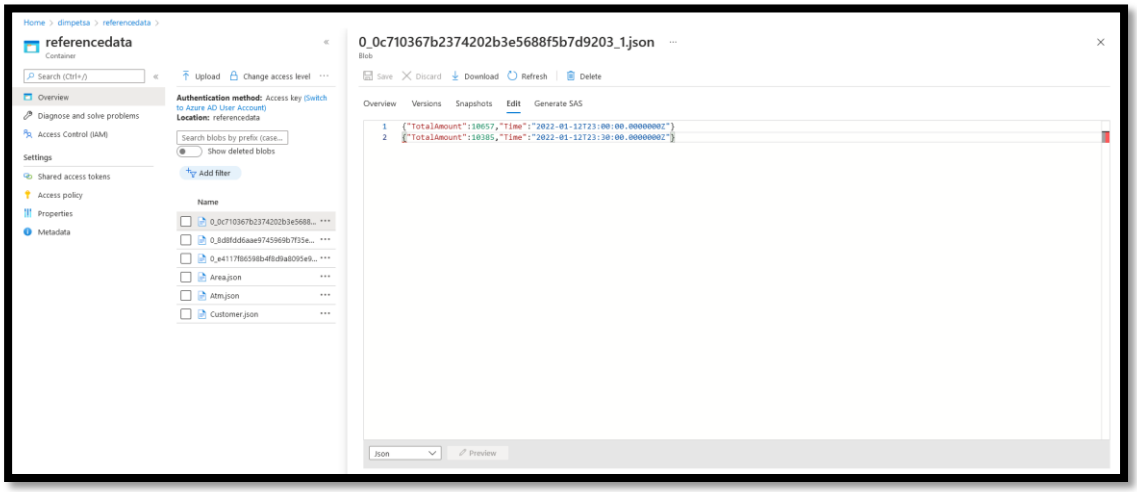
Solution:



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).

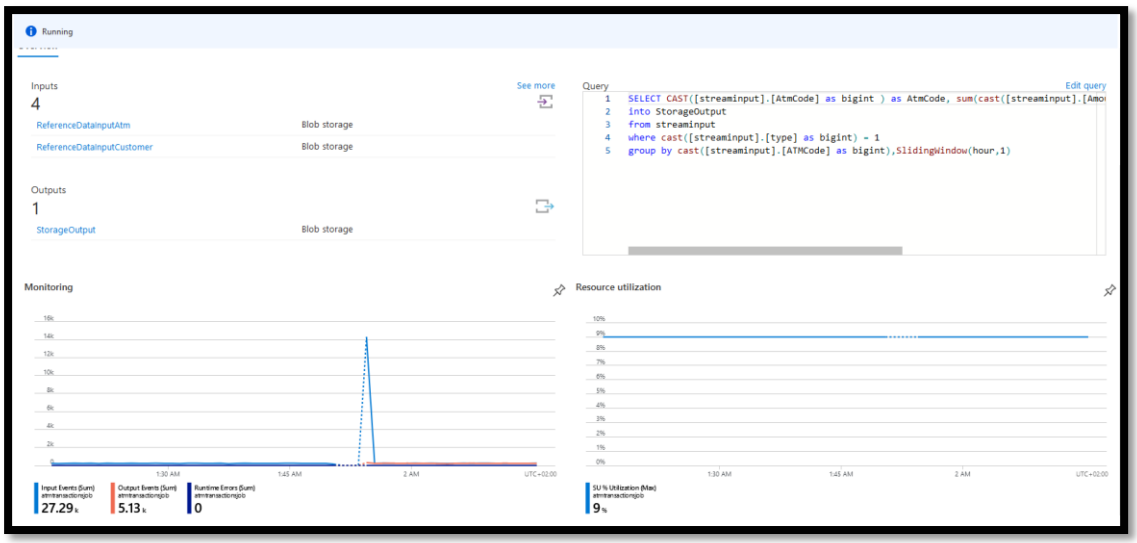
Solution:

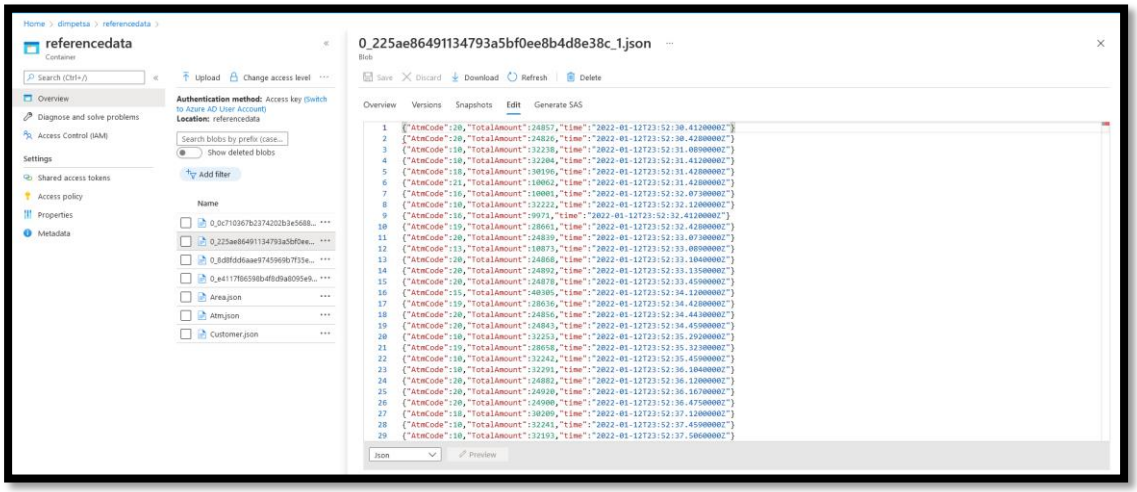




Query 4: Show the total “Amount” of “Type = 1” transactions per “ATM Code” of the last one hour (use a sliding window).

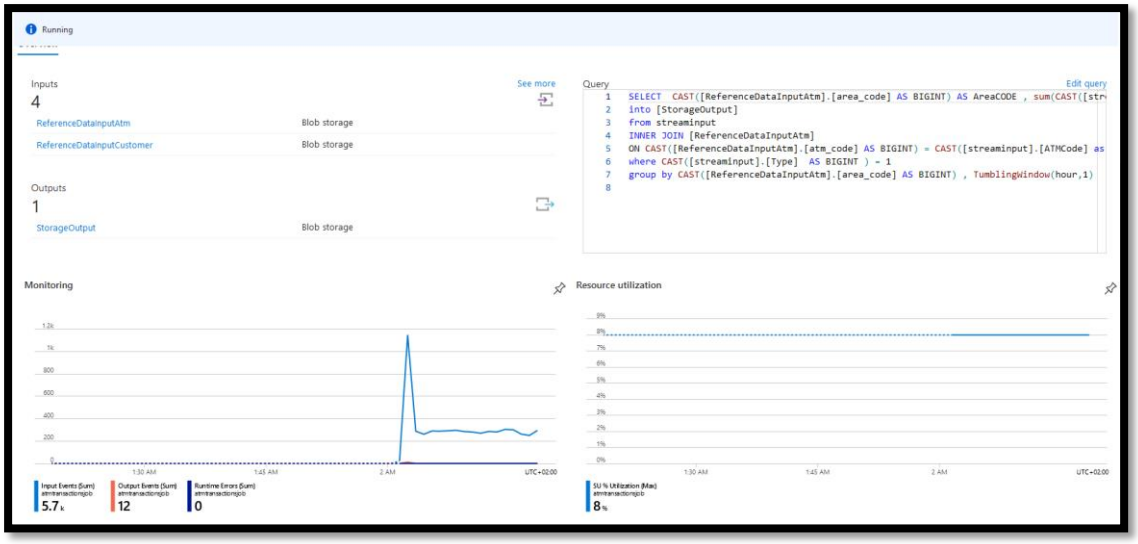
Solution:

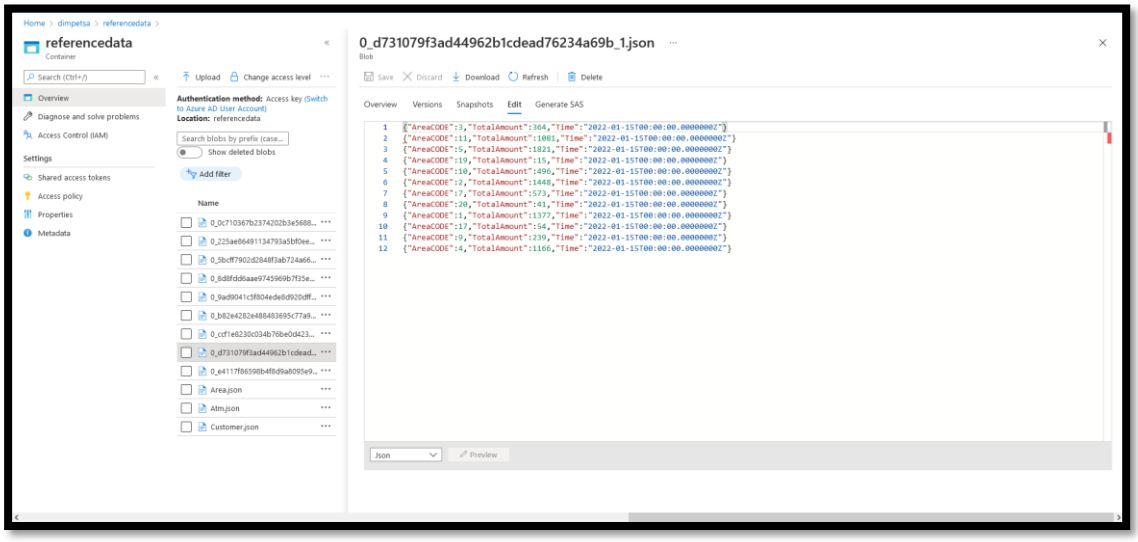




Query 5: Show the total “Amount” of “Type = 1” transactions per “Area Code” of the last hour. Repeat once every hour (use a tumbling window).

Solution:



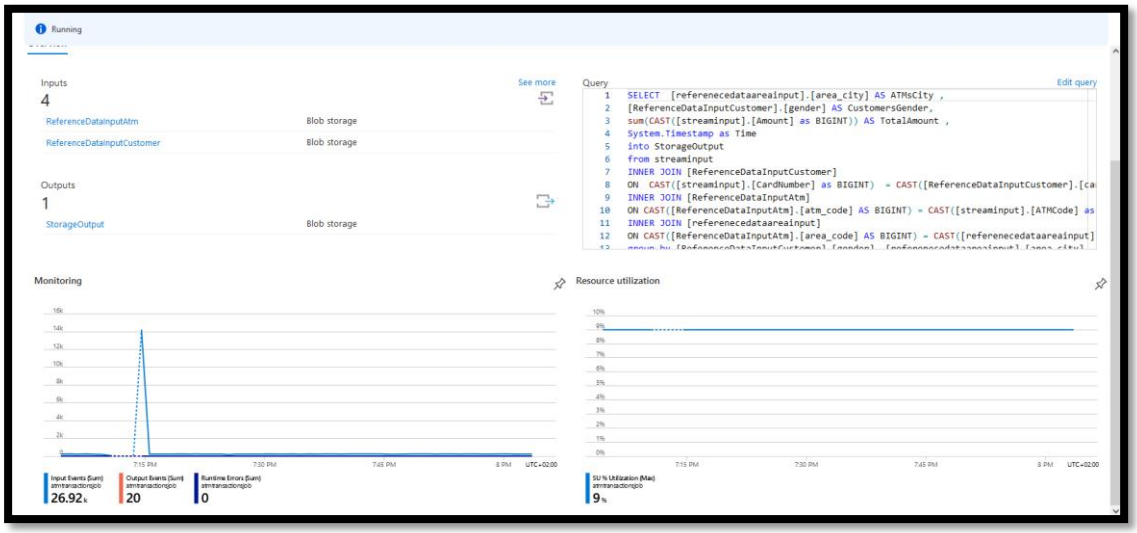


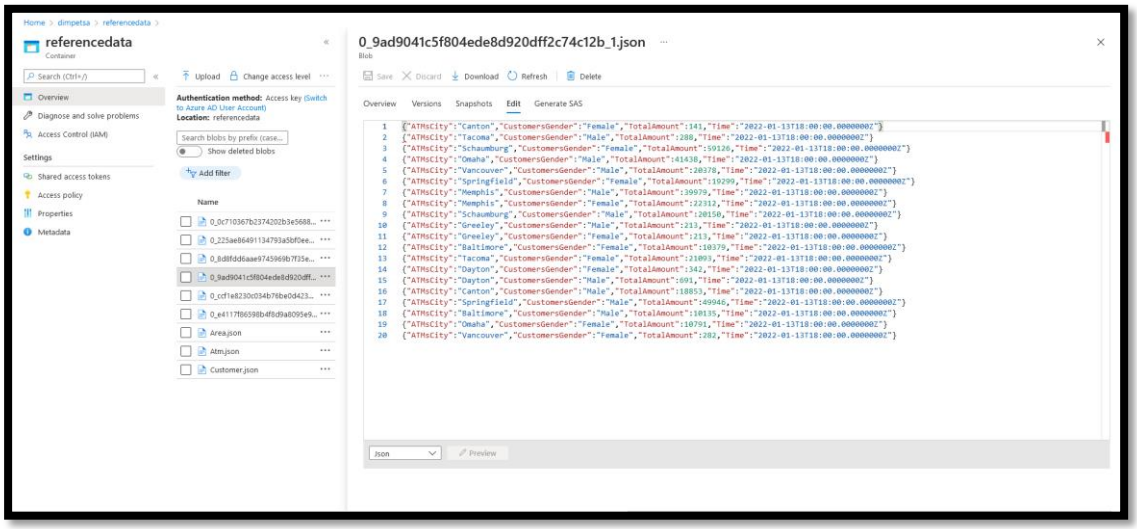
Query 6: Show the total “Amount” per ATM’s “City” and Customer’s “Gender” of the last hour. Repeat once every hour (use a tumbling window).

Solution:

```
SELECT  [referenecedataareainput].[area_city] AS ATMsCity ,
[ReferenceDataInputCustomer].[gender] AS CustomersGender,
sum(CAST([streaminput].[Amount] as BIGINT)) AS TotalAmount ,
System.Timestamp as Time
into StorageOutput
from streaminput
INNER JOIN [ReferenceDataInputCustomer]
ON  CAST([streaminput].[CardNumber] as BIGINT)  = CAST([ReferenceDataInputCustomer].[card_number] as BIGINT)
INNER JOIN [ReferenceDataInputAtm]
ON CAST([ReferenceDataInputAtm].[atm_code] AS BIGINT) = CAST([streaminput].[ATMCode] as BIGINT)
INNER JOIN [referenecedataareainput]
ON CAST([ReferenceDataInputAtm].[area_code] AS BIGINT) = CAST([referenecedataareainput].[area_code] as BIGINT)
group by [ReferenceDataInputCustomer].[gender] ,[referenecedataareainput].[area_city] , TumblingWindow(hour,1)
```

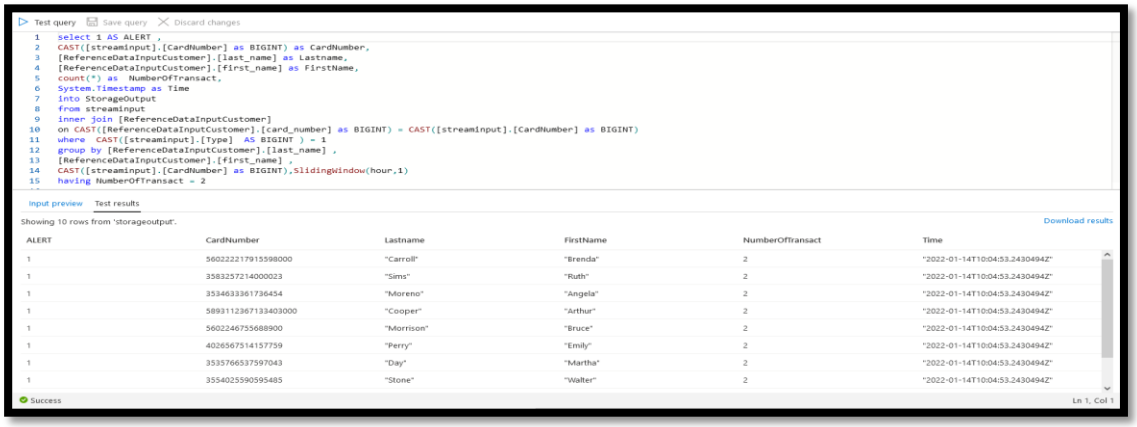
ATMsCity	CustomersGender	TotalAmount	Time
"Baltimore"	"Male"	57	"2022-01-13T19:00:00.0000000Z"
"Omaha"	"Male"	61	"2022-01-13T19:00:00.0000000Z"
"Schaumburg"	"Male"	44	"2022-01-13T19:00:00.0000000Z"
"Schaumburg"	"Female"	356	"2022-01-13T19:00:00.0000000Z"
"Memphis"	"Female"	66	"2022-01-13T19:00:00.0000000Z"
"Tacoma"	"Female"	65	"2022-01-13T19:00:00.0000000Z"
"Springfield"	"Male"	234	"2022-01-13T19:00:00.0000000Z"
"Vancouver"	"Male"	17	"2022-01-13T19:00:00.0000000Z"
"Tacoma"	"Male"	30	"2022-01-13T19:00:00.0000000Z"

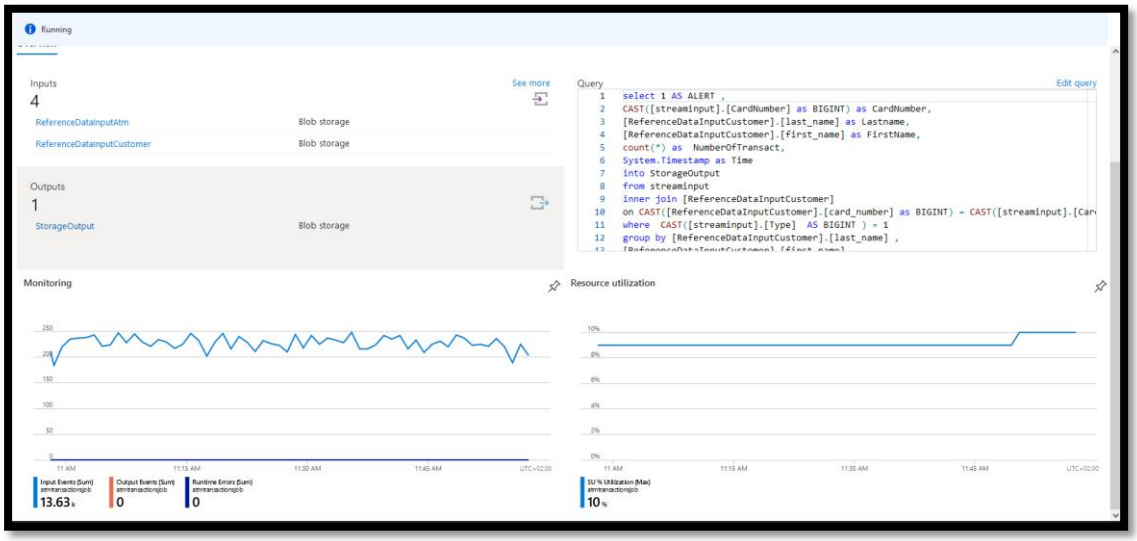


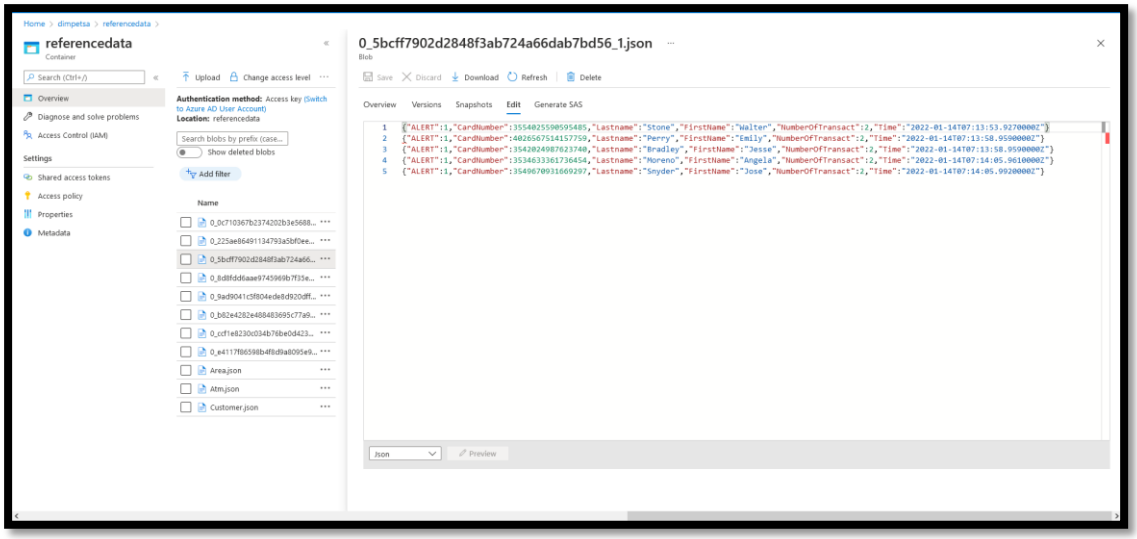


Query 7: Alert (Do a simple SELECT “1”) if a Customer has performed two transactions of “Type = 1” in a window of an hour (use a sliding window).

Solution: << The columns that contain information about the customer are not necessary but they are helpful for the viewer>>

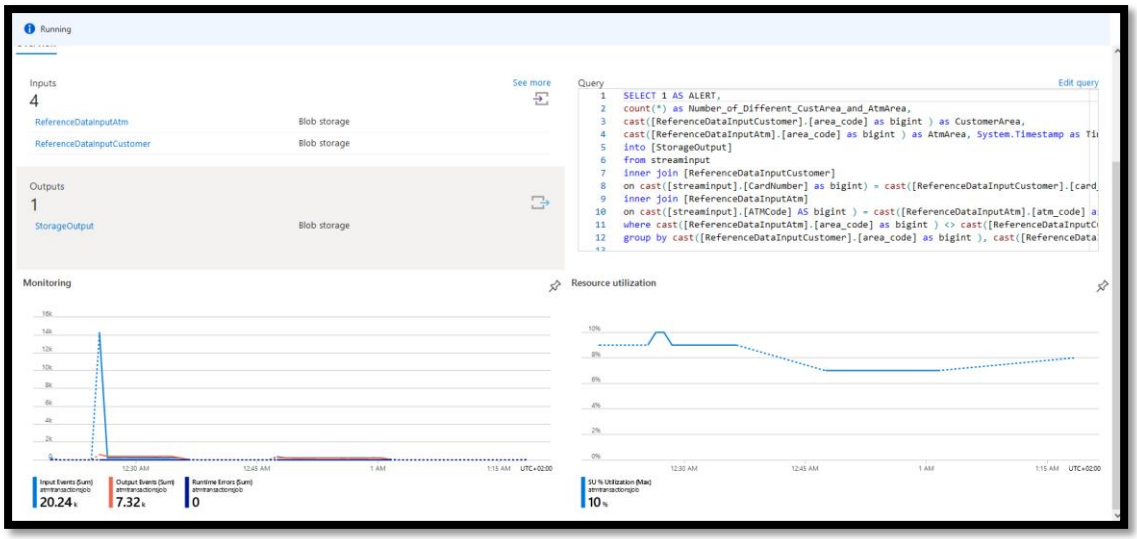


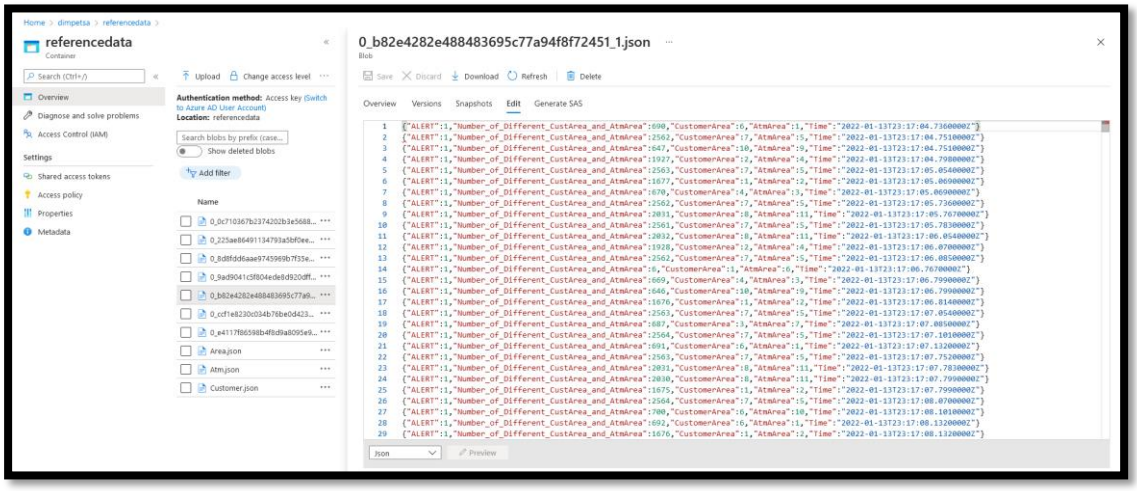




Query 8: Alert (Do a simple SELECT “1”) if the “Area Code” of the ATM of the transaction is not the same as the “Area Code” of the “Card Number” (Customer’s Area Code) - (use a sliding window)

Solution: << The columns that contain information about the customer are not necessary but they are helpful for the viewer>>






The End