ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ
ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS

MODERN DATA MANAGEMENT & BUSINESS INTELLIGENCE

ASSIGNMENT 3

By Giotopoulou Panagiota: p2822005 & Vlassi Georgia: p2822001
Professor: Chantziantoniou Damianos

MSc in Business Analytics

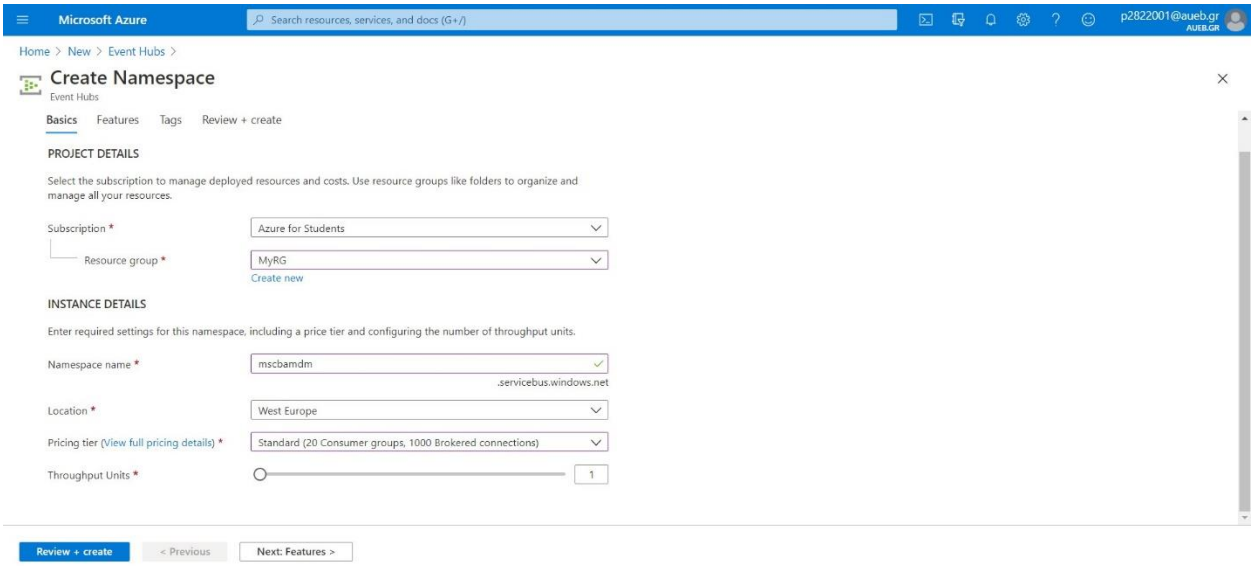
MSc in Business Analytics | Part Time 2020
Department of Management Science & Technology

Athens, Greece
January 15th 2021

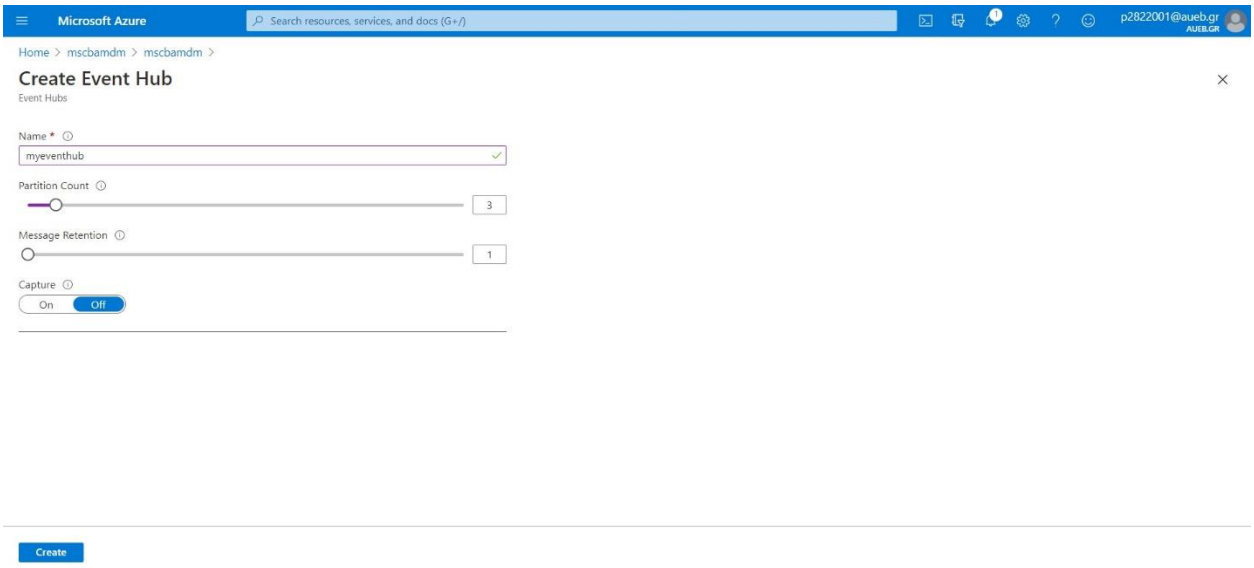
CONTENTS

EVENT HUB SETUP PROCESS

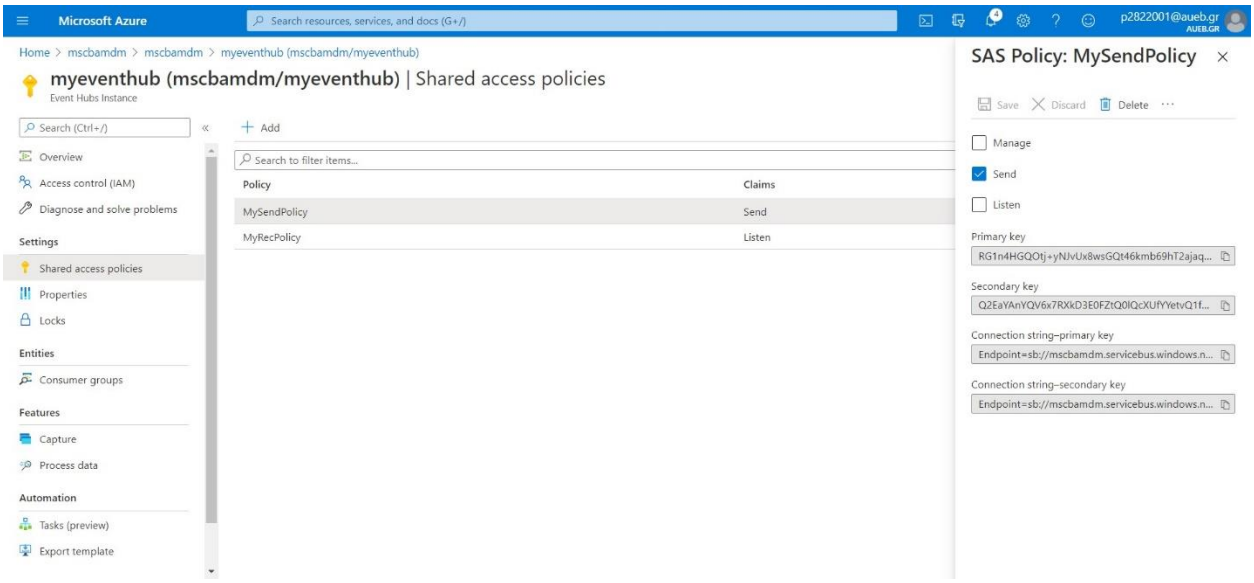
Creation of Namespace

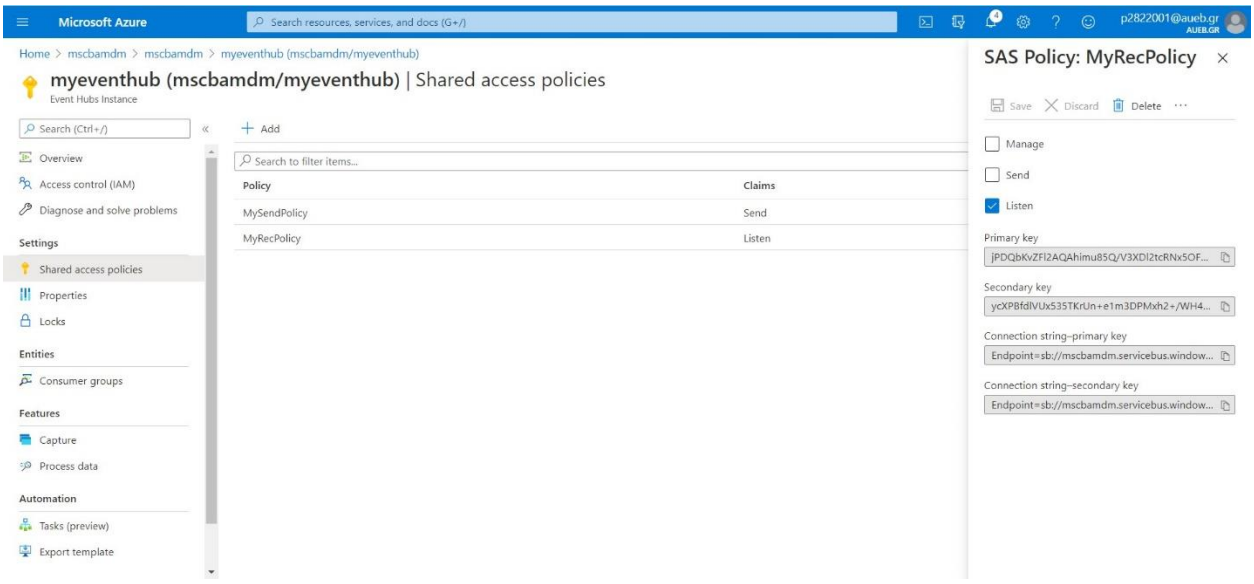


Creation of Event Hub

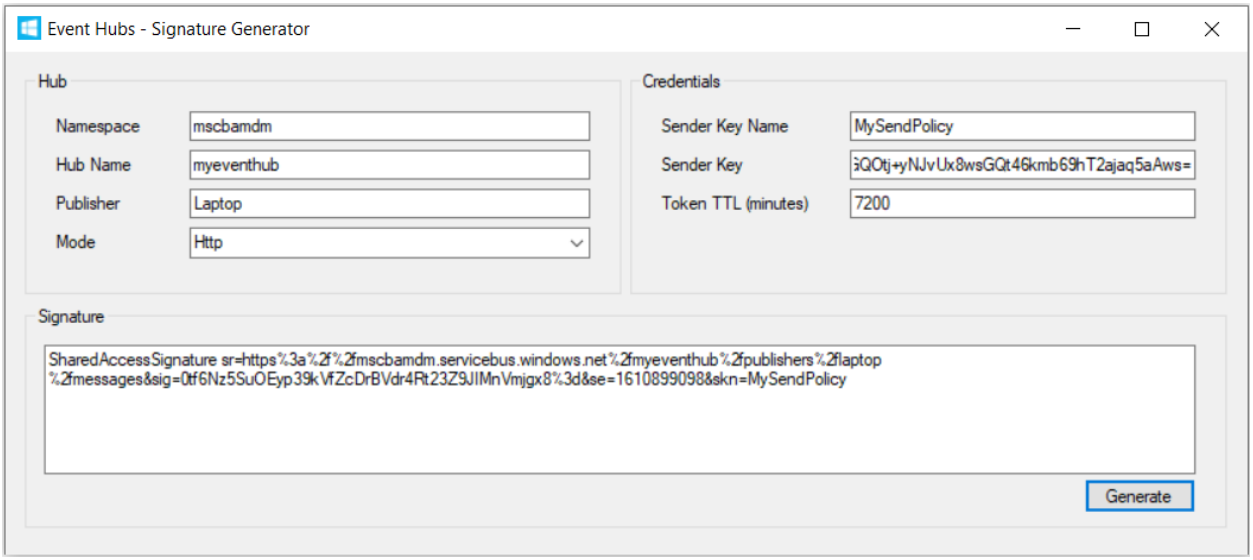


Setting Policies of Event Hub





Setting Signature



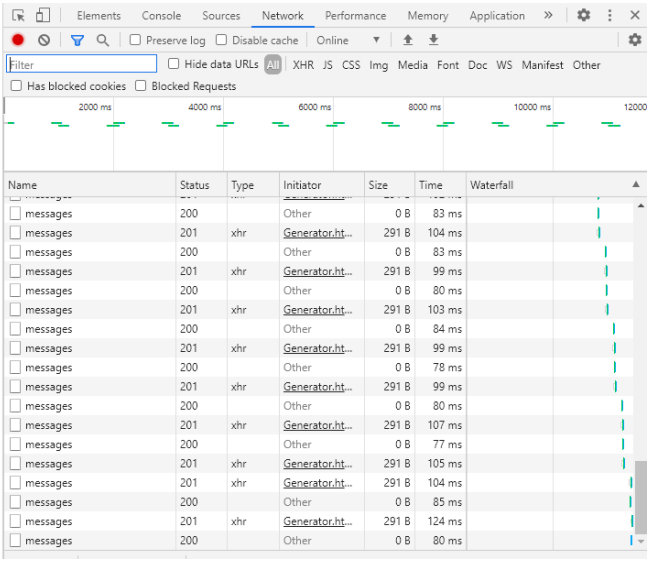
Edit configuration of Genarator.html

```
//Use the signature generator: https://github.com/sandrinodimattia/RedDog/releases
var sas = "SharedAccessSignature sr=https%3a%2f%2fmscbamdm.servicebus.windows.net%2fmyeventhub%2fpublishers%2flaptop%2fmessages&sig=0tf6Nz5SuOEyp39kVfZcDrBVd
r4Rt23Z9JIMnVmjgx8%3d&se=1610899098&skn=MySendPolicy";
var serviceNamespace = "mscbamdm";
var hubName = "myeventhub";
var deviceName = "Laptop";
```

Send data with Genarator.html

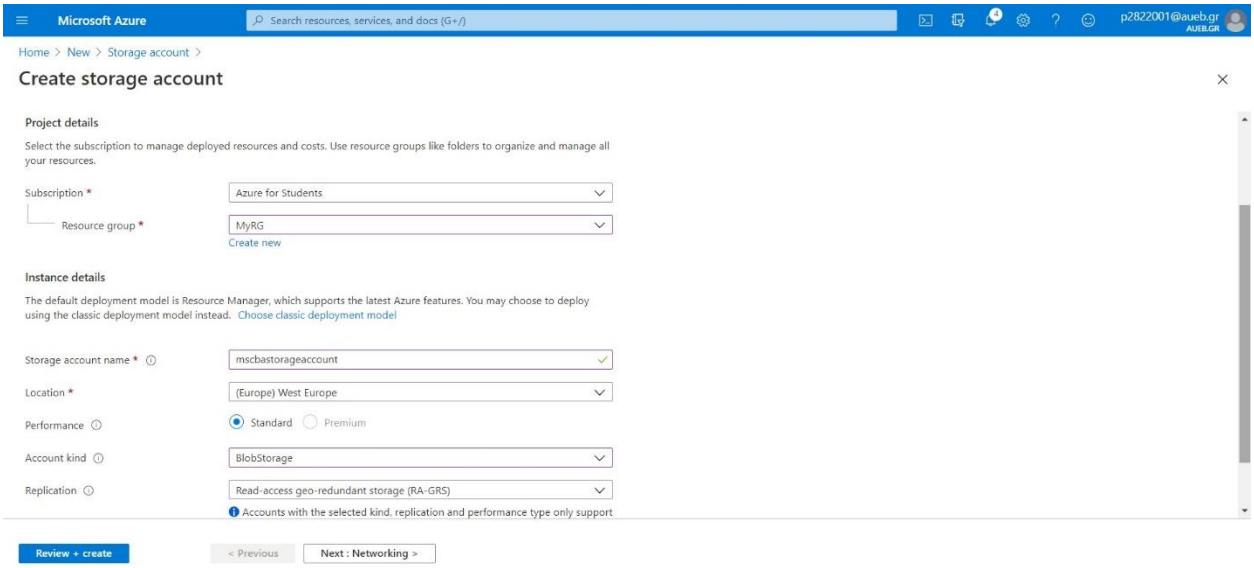
Send Data

Sent: { "ATMCode": 16 , "CardNumber": 3583257214000023 , "Type": 0 , "Amount": 32 }

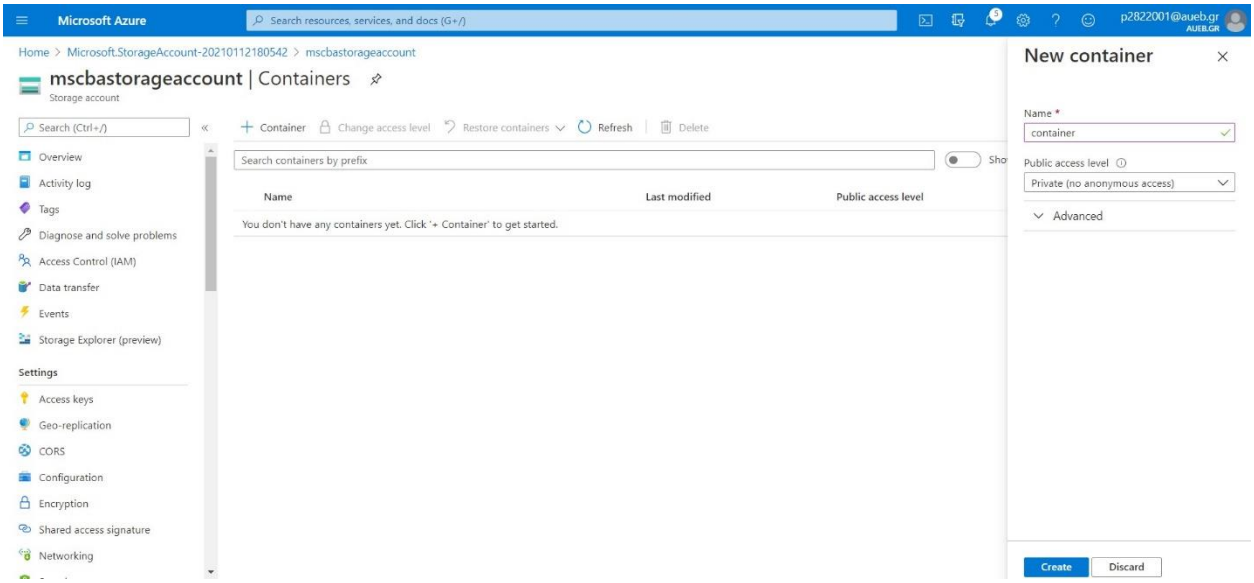


STORAGE ACCOUNT SETUP PROCESS

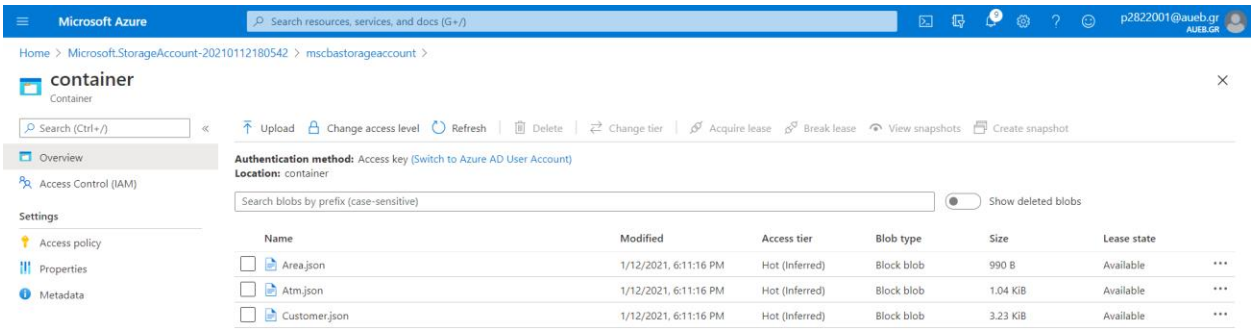
Creation of Blob Storage Account



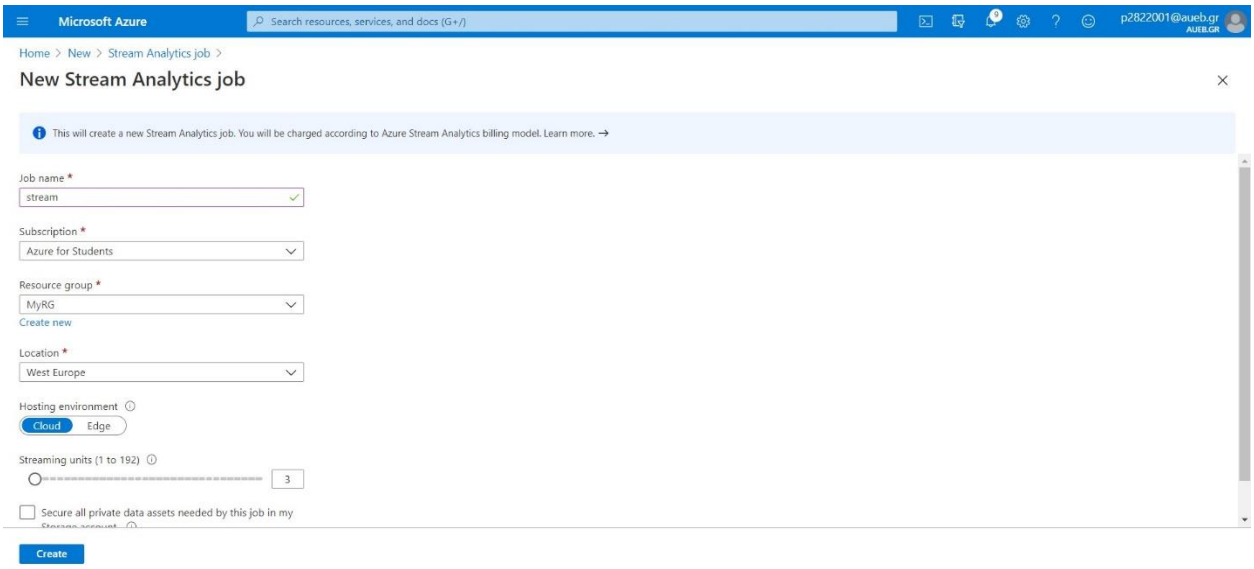
Creation of Container at Blob Storage Account



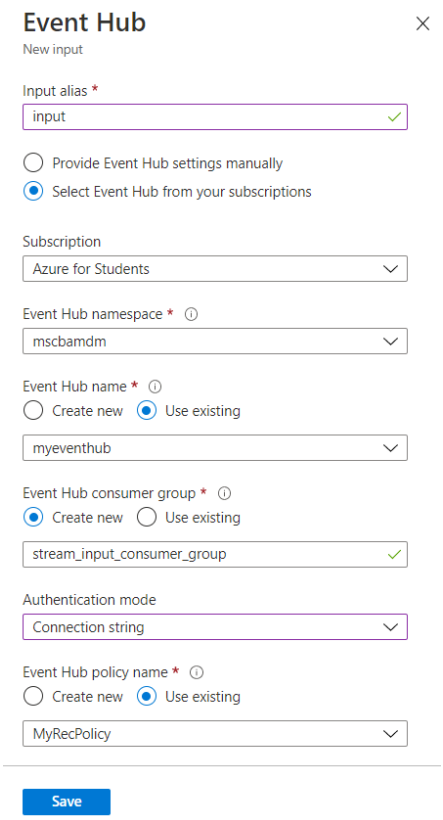
Upload reference data at Container



Creation of Stream Analytics Job



Add Inputs at Stream



Blob storage/ADLS Gen2

New input

Input alias *
area

Provide Blob storage/ADLS Gen2 settings manually
Select Blob storage/ADLS Gen2 from your subscriptions

Subscription
Azure for Students

Storage account *
mscbastorageaccount

Container *
Create new Use existing
container

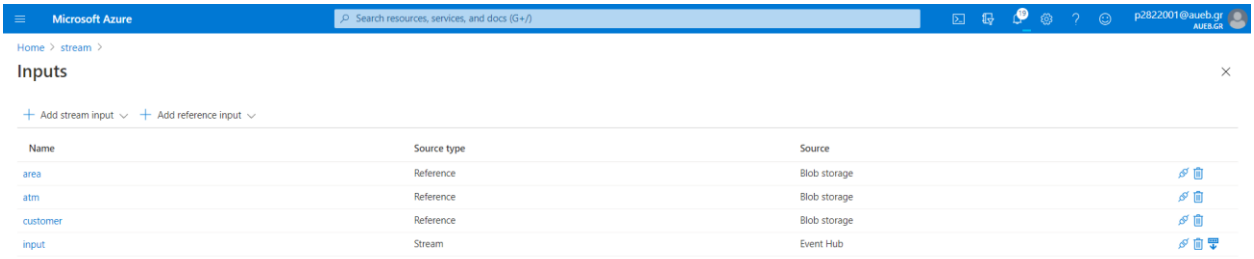
Storage account key
••••••••••••••••••••••••••

Path pattern * ⓘ
Area.json

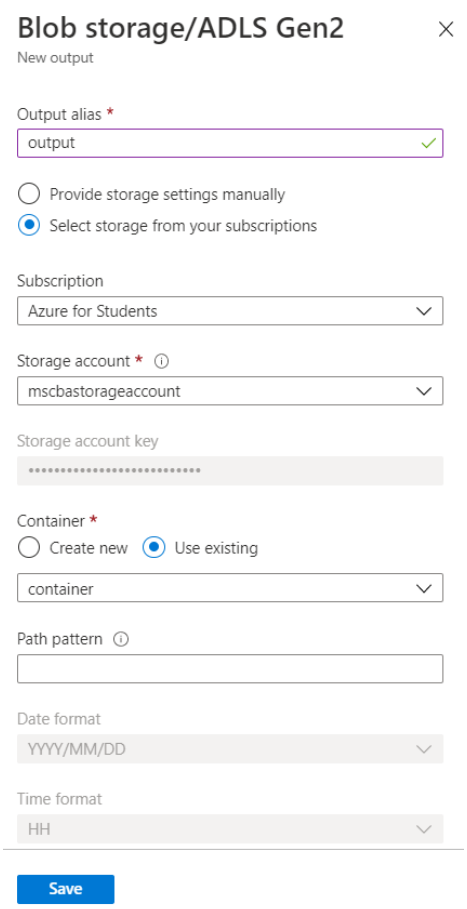
Date format
YYYY/MM/DD

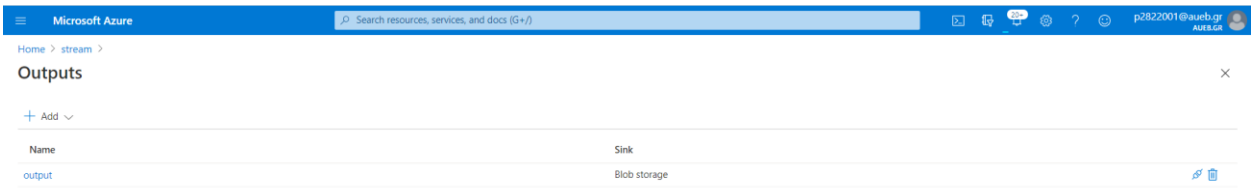
Time format
HH

Save



Add Output at Stream



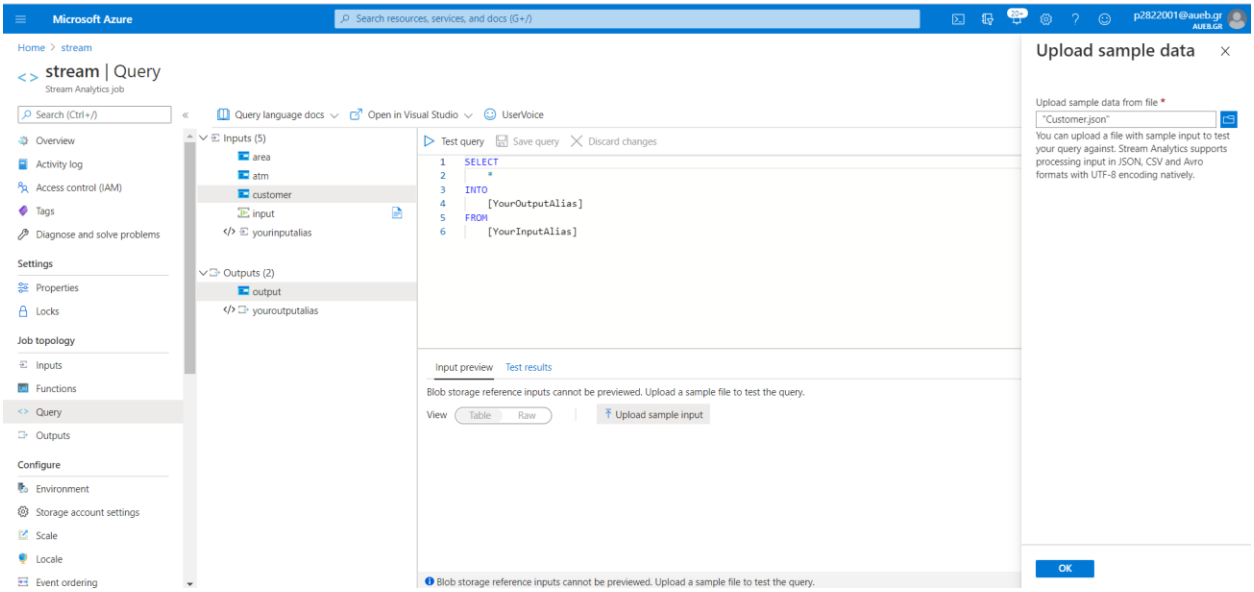


Sample data from input

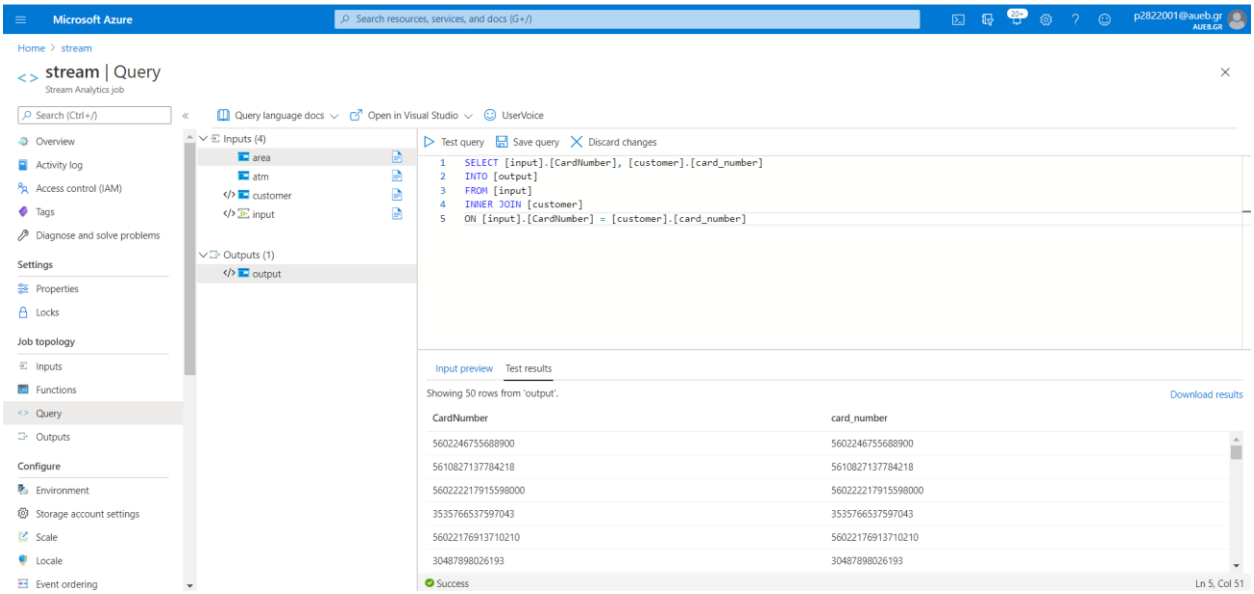
streamdemo-input.json | Generator.html | Customer.json


```
[
    {
        "ATMCode":15,
        "CardNumber":201634601435467,
        "Type":1,
        "Amount":11,
        "EventProcessedUtcTime":"2021-01-10T17:01:03.0743048Z",
        "PartitionId":0,
        "EventEnqueuedUtcTime":"2021-01-10T16:57:29.0800000Z"
    },
    {
        "ATMCode":19,
        "CardNumber":5200253312538103,
        "Type":1,
        "Amount":11,
        "EventProcessedUtcTime":"2021-01-10T17:01:03.0743048Z",
        "PartitionId":0,
        "EventEnqueuedUtcTime":"2021-01-10T16:57:29.0960000Z"
    },
    {
        "ATMCode":13,
        "CardNumber":50383945269330136,
        "Type":0,
        "Amount":42,
        "EventProcessedUtcTime":"2021-01-10T17:01:03.0743048Z",
        "PartitionId":0,
        "EventEnqueuedUtcTime":"2021-01-10T16:57:29.0960000Z"
    },
    {
        "ATMCode":12,
        "CardNumber":5893112367133403000,
        "Type":1,
        "Amount":15,
        "EventProcessedUtcTime":"2021-01-10T17:01:03.0743048Z",
        "PartitionId":0,
        "EventEnqueuedUtcTime":"2021-01-10T16:57:29.0960000Z"
    },
    {
        "ATMCode":21,
        "CardNumber":5610827137784218,
        "Type":0,
        "Amount":43,
        "EventProcessedUtcTime":"2021-01-10T17:01:03.0743048Z",
        "PartitionId":0,
        "EventEnqueuedUtcTime":"2021-01-10T16:57:29.1270000Z"
    },
```

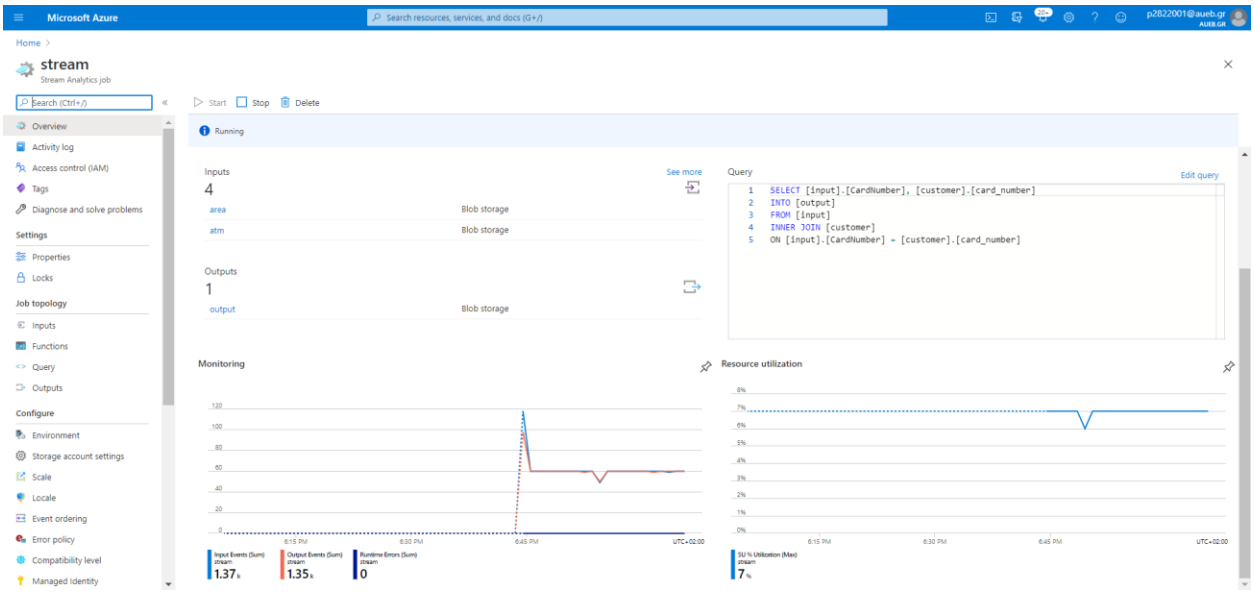
Upload test data at Stream



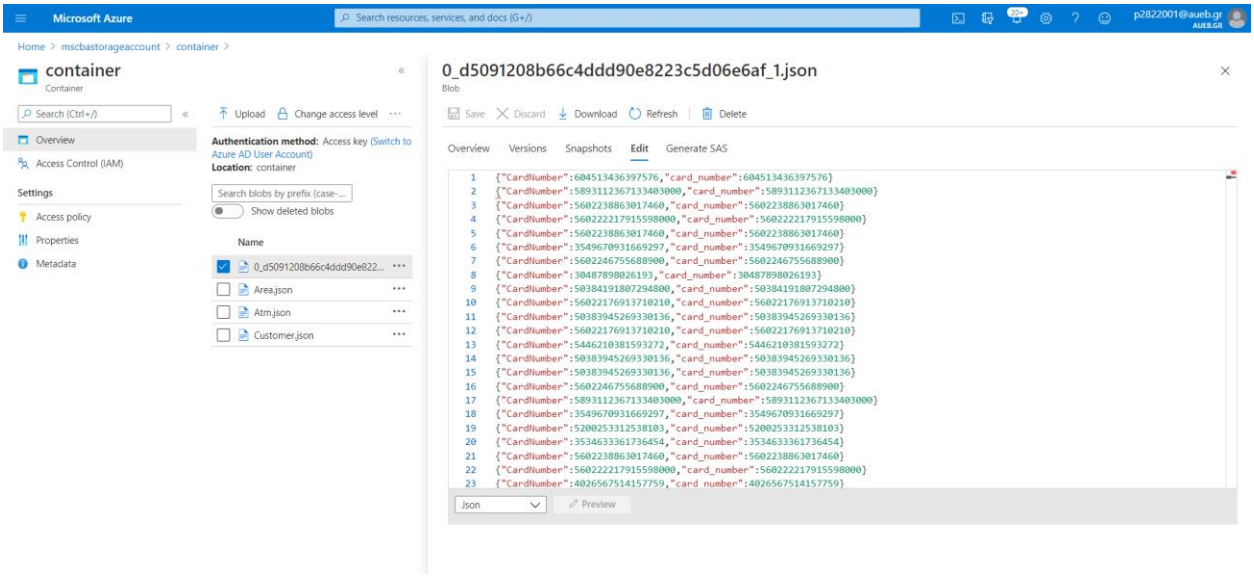
Test query



Start stream



Blob output of stream

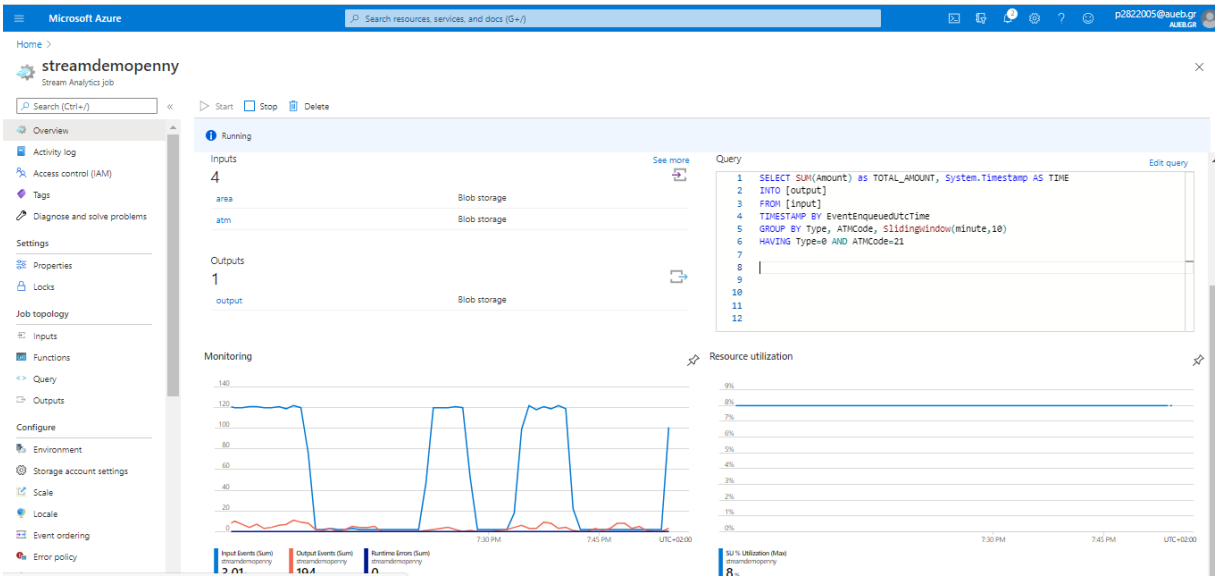


QUERIES

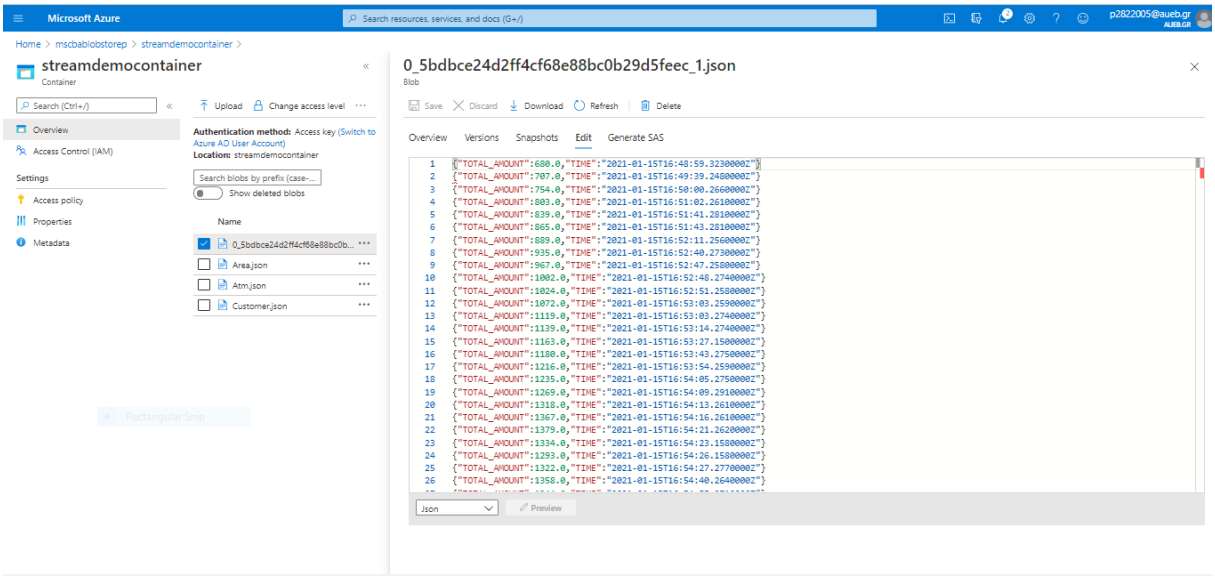
1. SHOW THE TOTAL "AMOUNT" OF "TYPE = 0" TRANSACTIONS AT "ATM CODE = 21" OF THE LAST 10 MINUTES. REPEAT AS NEW EVENTS KEEP FLOWING IN (USE A SLIDING WINDOW).

```
SELECT SUM(Amount) as TOTAL_AMOUNT, System.Timestamp AS TIME
INTO [output]
FROM [input]
TIMESTAMP BY EventEnqueuedUtcTime
GROUP BY Type, ATMCode, SlidingWindow(minute,10)
HAVING Type=0 AND ATMCode=21
```

Input



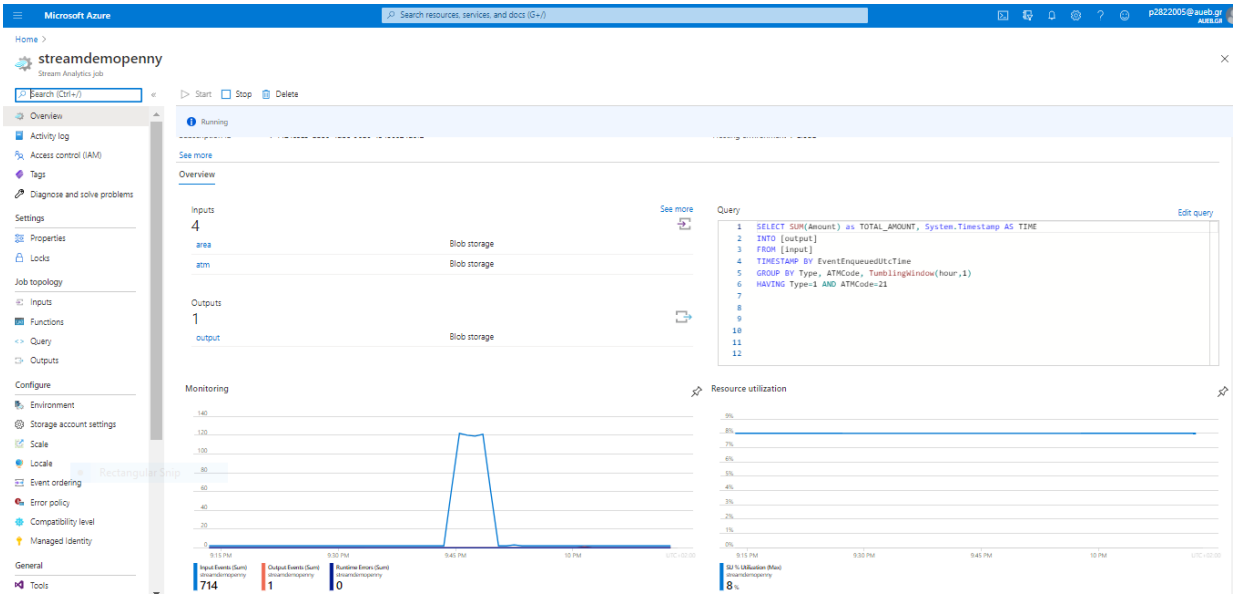
Output



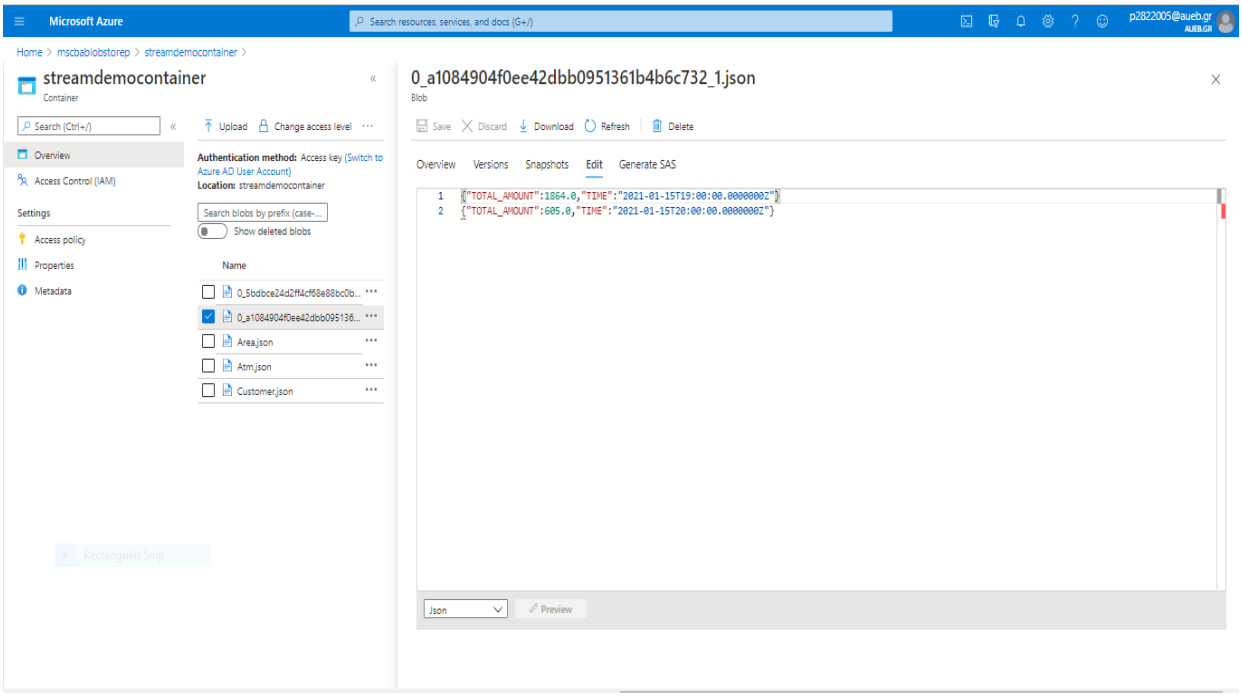
2. SHOW THE TOTAL "AMOUNT" OF "TYPE = 1" TRANSACTIONS AT "ATM CODE = 21" OF THE LAST HOUR. REPEAT ONCE EVERY HOUR (USE A TUMBLING WINDOW).

```
SELECT SUM(Amount) as TOTAL_AMOUNT, System.Timestamp AS TIME
INTO [output]
FROM [input]
TIMESTAMP BY EventEnqueuedUtcTime
GROUP BY Type, ATMCode, TumblingWindow(hour,1)
HAVING Type=1 AND ATMCode=21
```

Input



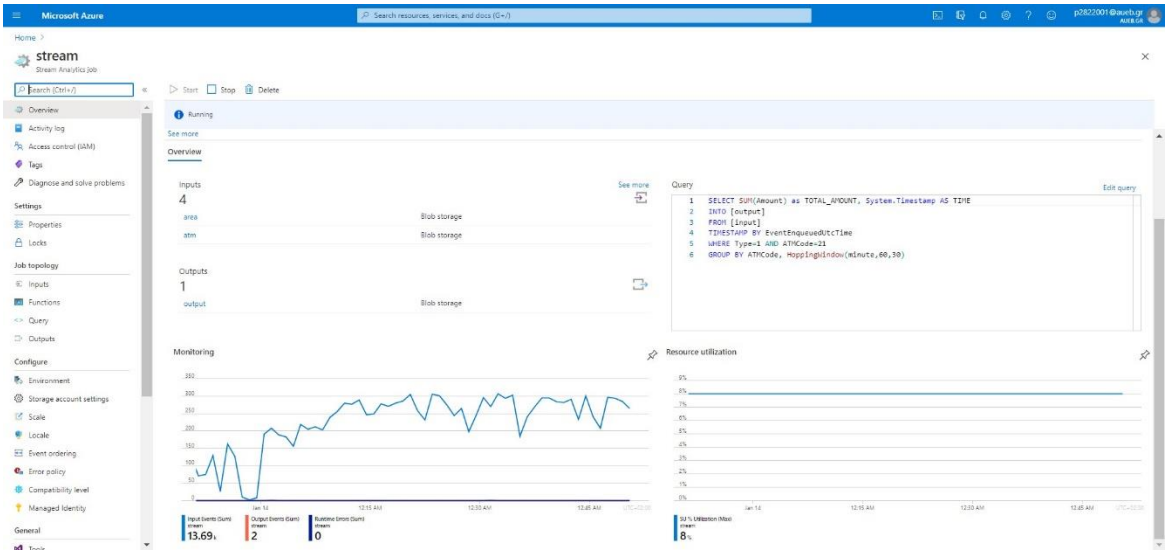
Output



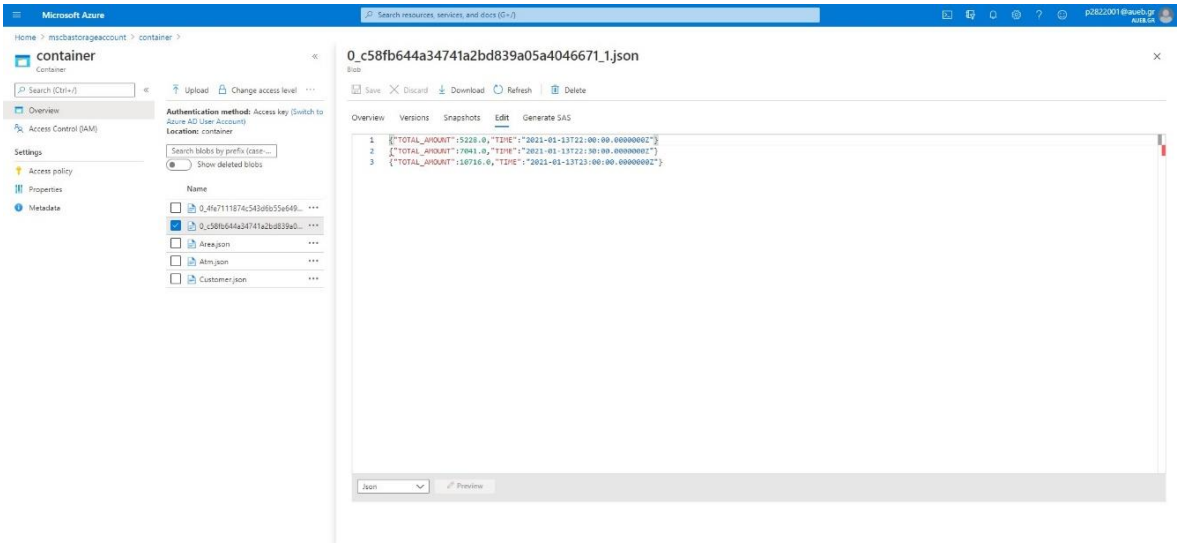
3. SHOW THE TOTAL "AMOUNT" OF "TYPE = 1" TRANSACTIONS AT "ATM CODE = 21" OF THE LAST HOUR.REPEAT ONCE EVERY 30 MINUTES (USE A HOPPING WINDOW).

```
SELECT SUM(Amount) as TOTAL_AMOUNT, System.Timestamp AS TIME
INTO [output]
FROM [input]
TIMESTAMP BY EventEnqueuedUtcTime
WHERE Type=1 AND ATMCode=21
GROUP BY ATMCode, HoppingWindow(minute,60,30)
```

Input



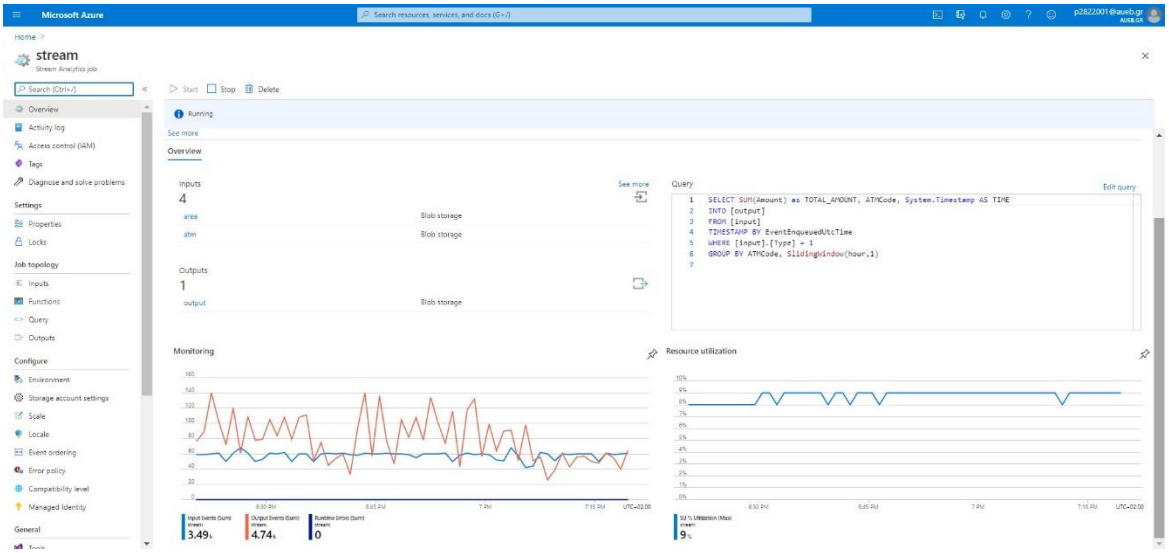
Output



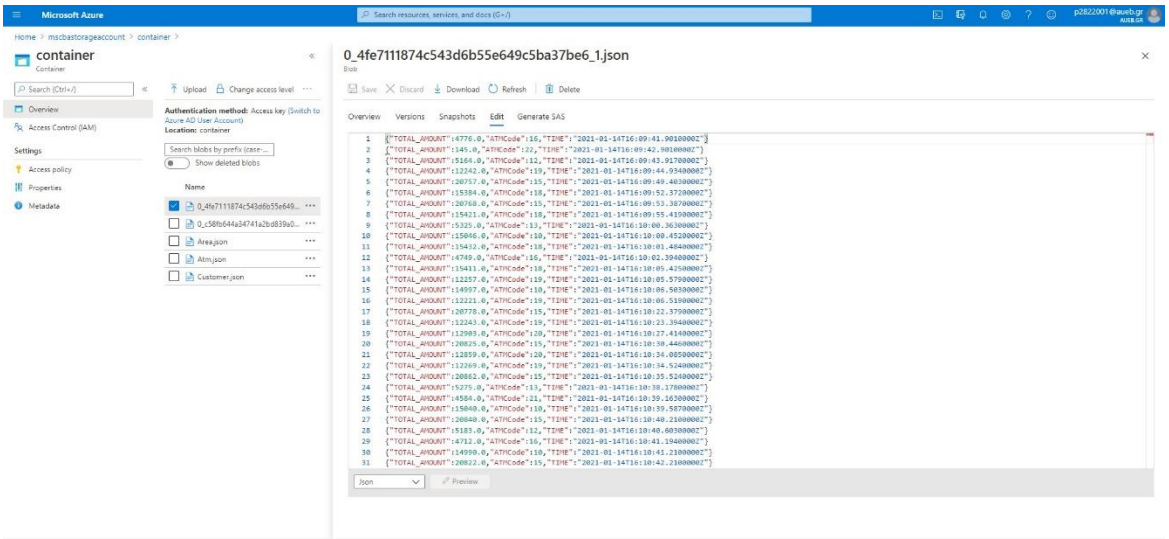
4. SHOW THE TOTAL "AMOUNT" OF "TYPE = 1" TRANSACTIONS PER "ATM CODE" OF THE LAST ONE HOUR (USE A SLIDING WINDOW).

```
SELECT SUM(Amount) as TOTAL_AMOUNT, ATMCode, System.Timestamp AS TIME
INTO [output]
FROM [input]
TIMESTAMP BY EventEnqueuedUtcTime
WHERE Type =1
GROUP BY ATMCode, SlidingWindow(minute,60)
```

Input



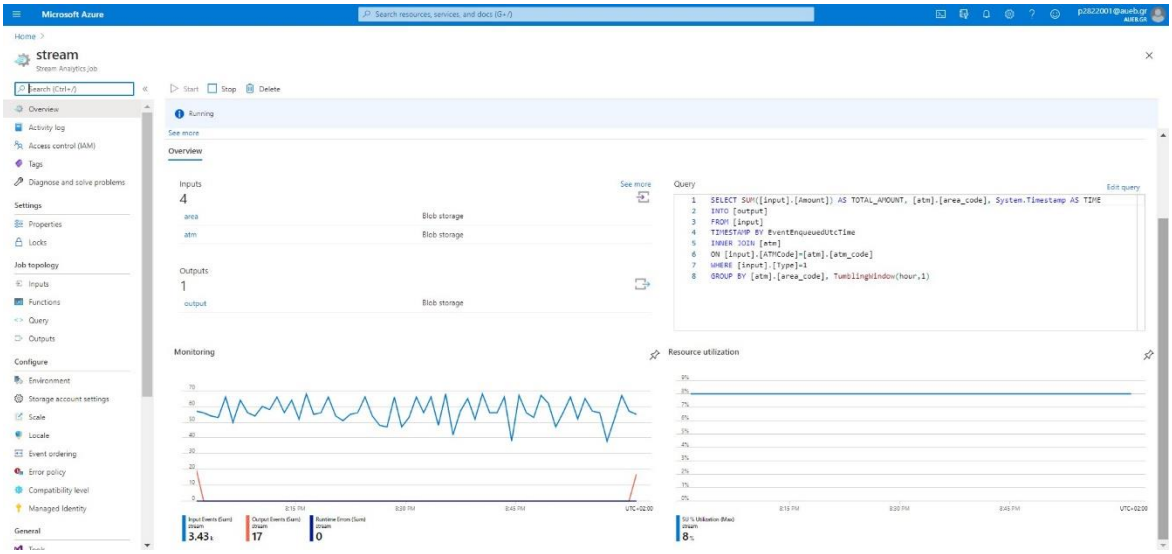
Output



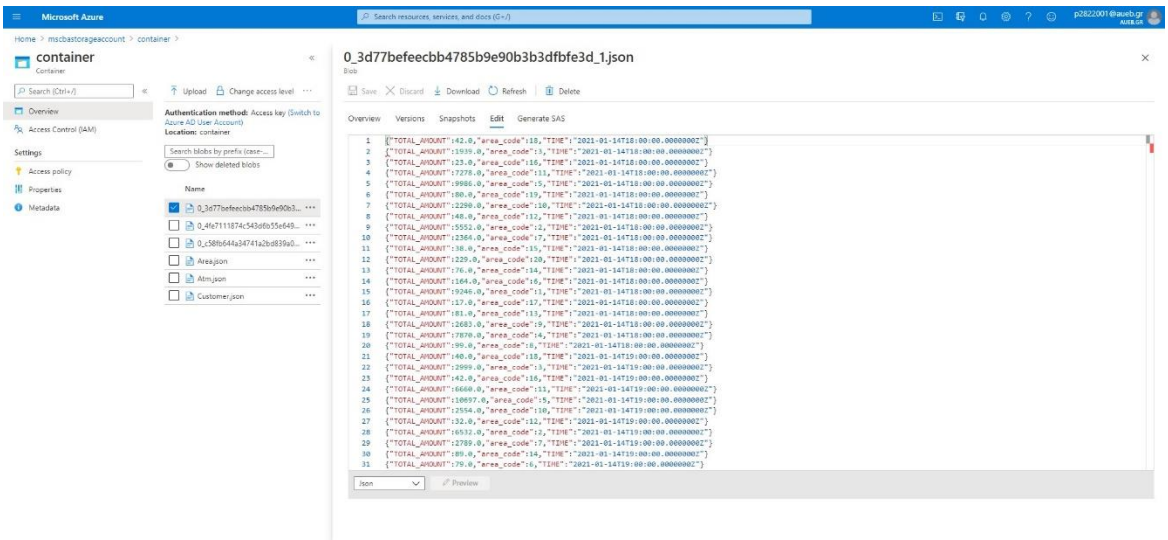
5. SHOW THE TOTAL "AMOUNT" OF "TYPE = 1" TRANSACTIONS PER "AREA CODE" OF THE LAST HOUR. REPEAT ONCE EVERY HOUR (USE A TUMBLING WINDOW).

```
SELECT SUM([input].[Amount]) AS TOTAL_AMOUNT, [atm].[area_code], System.Timestamp
AS TIME
INTO [output]
FROM [input]
TIMESTAMP BY EventEnqueuedUtcTime
INNER JOIN [atm]
ON [input].[ATMCode]=[atm].[atm_code]
WHERE [input].[Type]=1
GROUP BY [atm].[area_code], TumblingWindow(hour,1)
```

Input



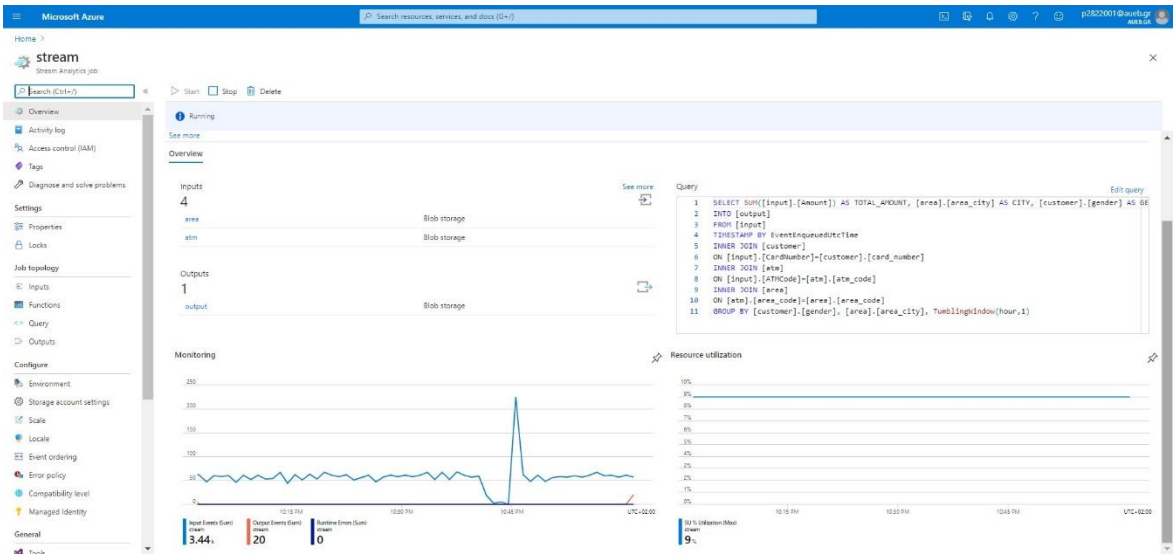
Output



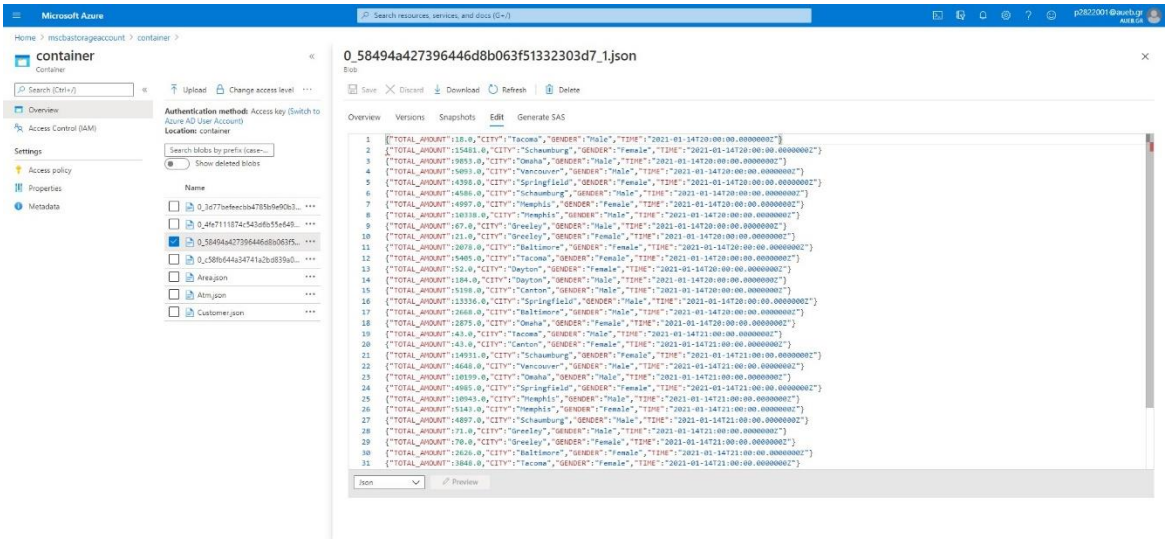
6. SHOW THE TOTAL "AMOUNT" PER ATM'S "CITY" AND CUSTOMER'S "GENDER" OF THE LAST HOUR. REPEAT ONCE EVERY HOUR (USE A TUMBLING WINDOW).

```
SELECT SUM([input].[Amount]) AS TOTAL_AMOUNT, [area].[area_city] AS CITY,
[customer].[gender] AS GENDER, System.Timestamp AS TIME
INTO [output]
FROM [input]
TIMESTAMP BY EventEnqueuedUtcTime
INNER JOIN [customer]
ON [input].[CardNumber]=[customer].[card_number]
INNER JOIN [atm]
ON [input].[ATMCode]=[atm].[atm_code]
INNER JOIN [area]
ON [atm].[area_code]=[area].[area_code]
GROUP BY [customer].[gender], [area].[area_city], TumblingWindow(hour,1)
```

Input



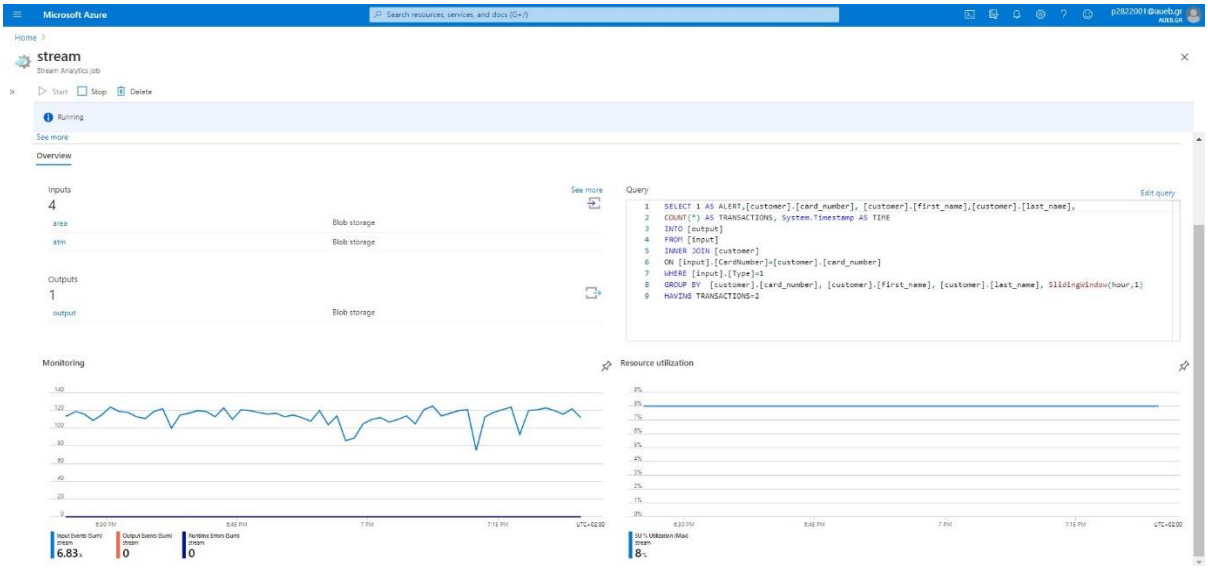
Output



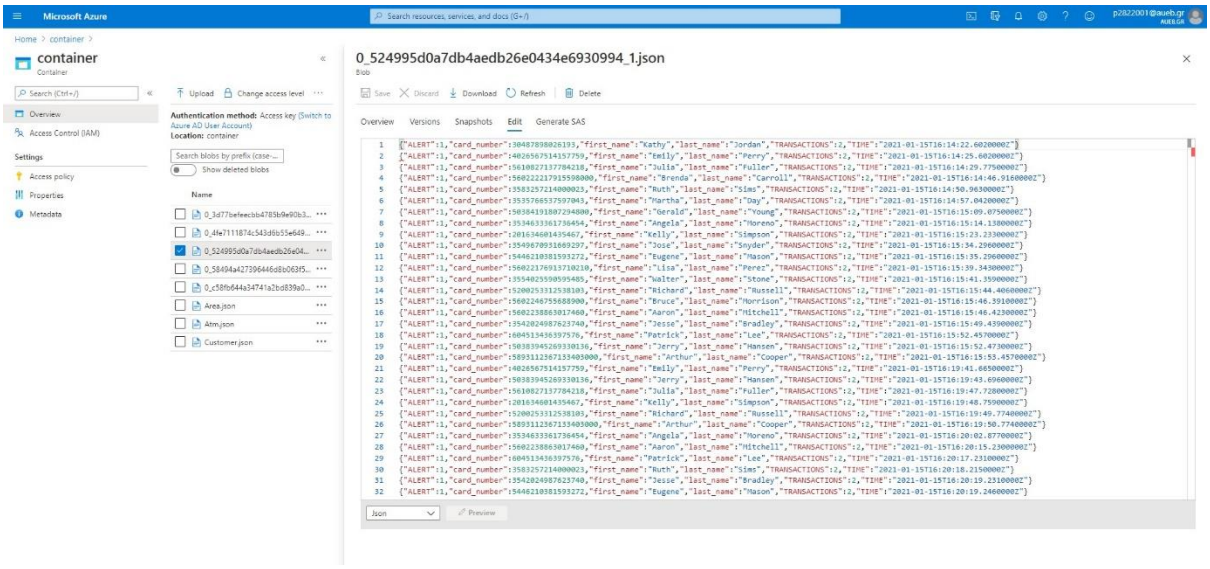
7. ALERT (DO A SIMPLE SELECT "1") IF A CUSTOMER HAS PERFORMED TWO TRANSACTIONS OF "TYPE = 1" IN A WINDOW OF AN HOUR (USE A SLIDING WINDOW).

```
SELECT 1 AS ALERT,[customer].[card_number],
[customer].[first_name],[customer].[last_name], COUNT(*) AS TRANSACTIONS,
System.Timestamp AS TIME
INTO [output]
FROM [input]
INNER JOIN [customer]
ON [input].[CardNumber]=[customer].[card_number]
WHERE [input].[Type]=1
GROUP BY  [customer].[card_number], [customer].[first_name],
[customer].[last_name],SlidingWindow(hour,1)
HAVING TRANSACTIONS=2
```

Input



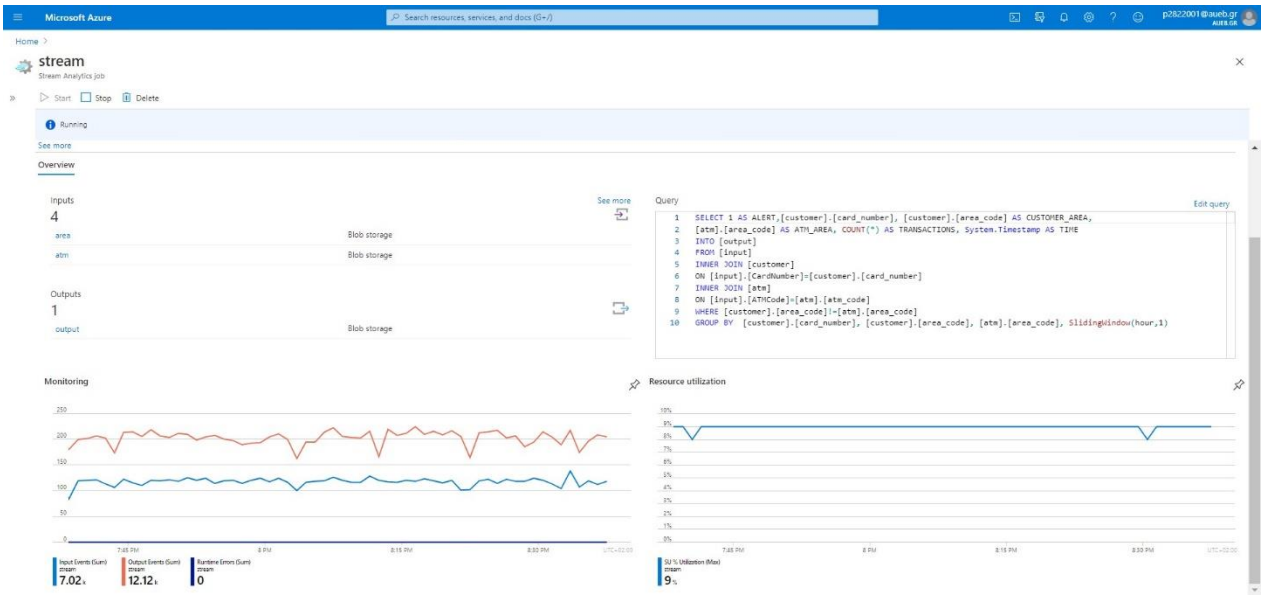
Output



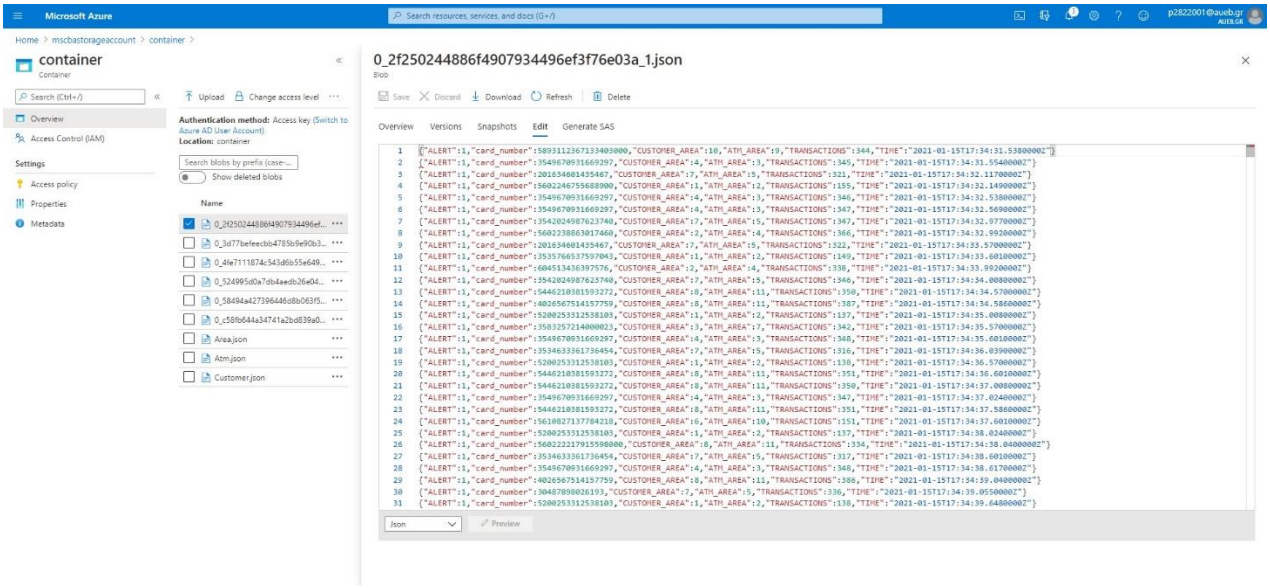
8. ALERT (DO A SIMPLE SELECT "1") IF THE "AREA CODE" OF THE ATM OF THE TRANSACTION IS NOT THE SAME AS THE "AREA CODE" OF THE "CARD NUMBER" (CUSTOMER'S AREA CODE) - (USE A SLIDING WINDOW)

```
SELECT 1 AS ALERT,[customer].[card_number], [customer].[area_code] AS
CUSTOMER_AREA,[atm].[area_code] AS ATM_AREA, COUNT(*) AS TRANSACTIONS,
System.Timestamp AS TIME
INTO [output]
FROM [input]
INNER JOIN [customer]
ON [input].[CardNumber]=[customer].[card_number]
INNER JOIN [atm]
ON [input].[ATMCode]=[atm].[atm_code]
WHERE [customer].[area_code]!=[atm].[area_code]
GROUP BY  [customer].[card_number], [customer].[area_code], [atm].[area_code],
SlidingWindow(hour,1)
```

Input



Output


Microsoft Azure
Search resources, services, and docs (G+/)
p2822001@aueb.gr
AUEB GK
Home > mscbastorageaccount > container >
container
Container
Search (Ctrl+/)
Upload
Change access level
Overview
Access Control (IAM)
Settings
Access policy
Properties
Metadata
Authentication method: Access key (Switch to Azure AD User Account)
Location: container
Search blobs by prefix (case-...
Show deleted blobs
Name
0_2f250244886f4907934496ef...
0_3d77befeecbb4785b9e90b3...
0_4fe7111874c543d6b55e649...
0_524995d0a7db4aedb26e04...
0_58494a427396446d8b063f5...
0_c58fb644a34741a2bd839a0...
Area.json
Atm.json
Customer.json
0_2f250244886f4907934496ef3f76e03a_1.json
Blob
Save
Discard
Download
Refresh
Delete
Overview
Versions
Snapshots
Edit
Generate SAS
Json
Preview