Assignment #3 - (Teamwork)

Personal Ethics & Academic Integrity Statement

Student Name: Abdallah Mohamed Mahmoud Mohamed Ragab Student ID: 300327288

Student Name: Hosam Mahmoud Ibrahim Mahmoud Student ID: 300327269

Student Name: Sondos Mohammed Hussein Ali Student ID: 300327219

Student name: Esraa Ahmed Abdelhakam Abo wadaa Student ID: 300327225

By typing in my name and student ID on this form and submitting it electronically, I am attesting to the fact that I have reviewed not only my work but the work of my team member, in its entirety.

I attest to the fact that my work in this project adheres to the fraud policies as outlined in the Academic Regulations in the University's Graduate Studies Calendar. I further attest that I have knowledge of and have respected the "Beware of Plagiarism" brochure for the university. To the best of my knowledge, I also believe that each of my group colleagues has also met the aforementioned requirements and regulations. I understand that if my group assignment is submitted without a completed copy of this Personal Work Statement from each group member, it will be interpreted by the school that the missing student(s) name is confirmation of the nonparticipation of the aforementioned student(s) in the required work.

We, by typing in our names and student IDs on this form and submitting it electronically,

- warrant that the work submitted herein is our own group members' work and not the work of others

acknowledge that we have read and understood the University Regulations on Academic Misconduct

acknowledge that it is a breach of University Regulations to give or receive unauthorized and/or unacknowledged assistance on a graded piece of work

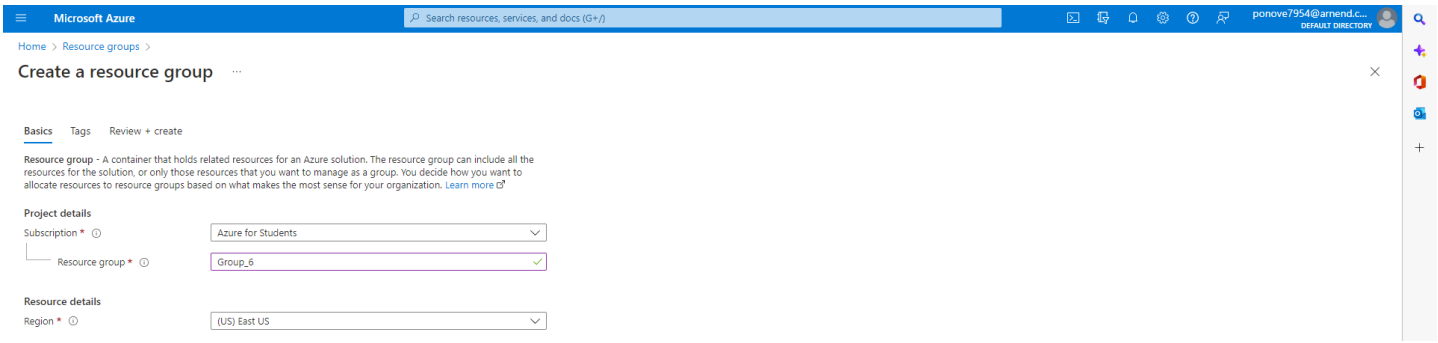
- # Part 1 – Event Hubs Analytics

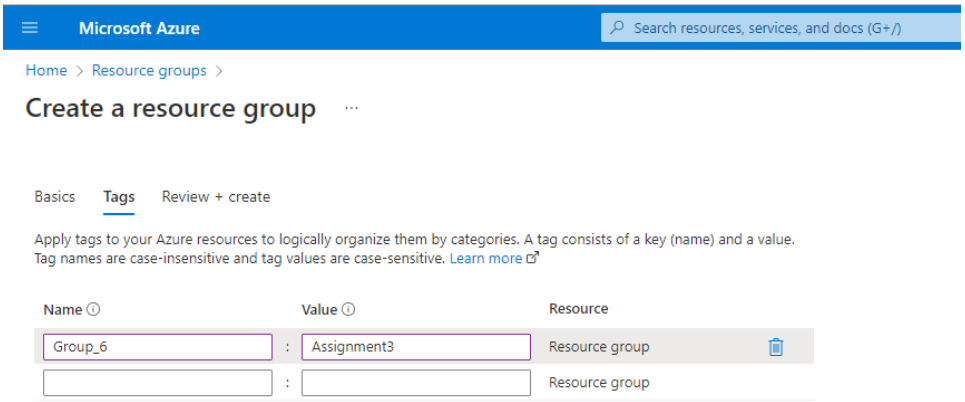
A. Question a)

1- **First, we created a student account on Microsoft Azure.**

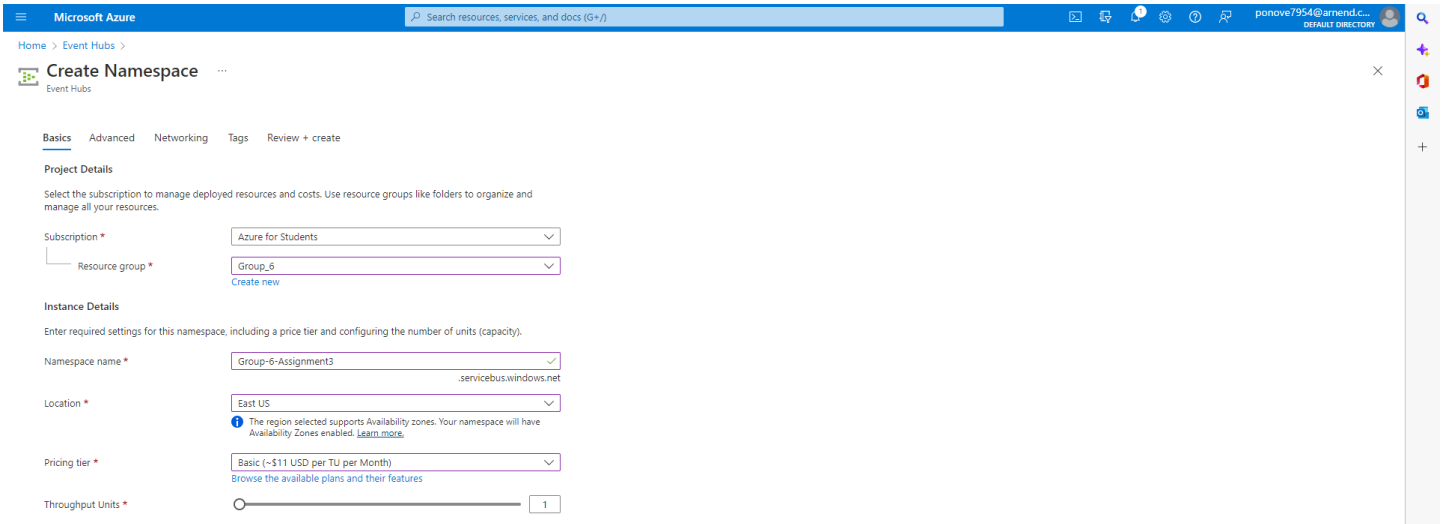
2- **After that, we installed Visual Studio Community Edition (version 16.0)**

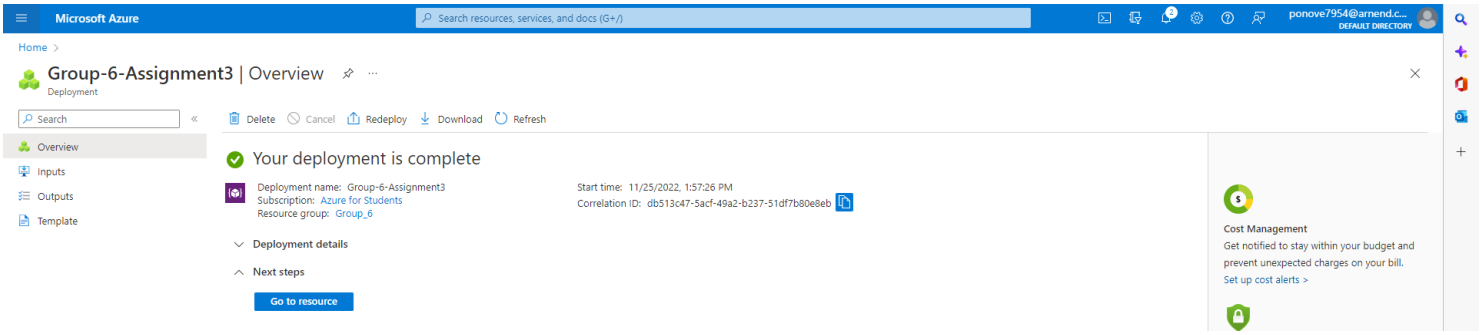
3- **Creating the resource group:**



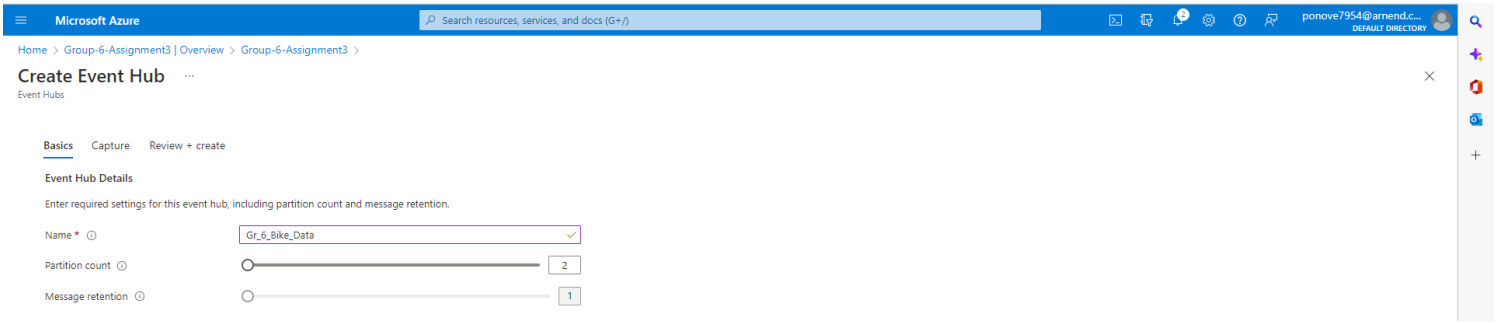


4- **Creating Namespace:**





5- Creating Event Hub:



6- Modifying EventHubsSender application:

```
namespace Sender
{
    0 references
    class Program
    {
        private static string eventHubName = "gr_6_bike_data";
        private static string connectionString = "Endpoint=sb://group-6-assignment3.servicebus.windows.net/;SharedAccessKeyNam
        private static string cachedEventSource = @"D:\Group6_assignment3\bike_data.json";
        private static int numOfBatchesEvents = 25;
        private static int numBatchSize = 20;
```

```
// Add artificial 10 seconds delay to the events
System.Threading.Thread.Sleep(10 * 1000);
```

7- Modifying EventModel application:

- **BikeData.cs**

```
// Convert using https://json2csharp.com/

[JsonProperty("Trip ID")]
0 references
public string TripID { get; set; }
0 references
public string Duration { get; set; }

[JsonProperty("Start Date")]
0 references
public string StartDate { get; set; }

[JsonProperty("Start Station")]
0 references
public string StartStation { get; set; }

[JsonProperty("Start Terminal")]
0 references
public string StartTerminal { get; set; }

[JsonProperty("End Date")]
0 references
public string EndDate { get; set; }

[JsonProperty("End Station")]
0 references
public string EndStation { get; set; }

[JsonProperty("End Terminal")]
0 references
public string EndTerminal { get; set; }

[JsonProperty("Bike #")]
0 references
public string Bike { get; set; }

[JsonProperty("Subscriber Type")]
0 references
public string SubscriberType { get; set; }

[JsonProperty("Zip Code")]
0 references
public string ZipCode { get; set; }

1 reference
public DateTime ProcessDate { get; set; }
```

- **BikeRentalEvents.cs**

```
public class BikeRentalEvents
{
    private int marker = 0;

    private List<BikeData> bikeDataCollection = new List<BikeData>();
    4 references
    public string FileName
    {
        get;
        set;
    }

    1 reference
    public void Initialize(string file)
    {
        FileName = file;

        if (!File.Exists(FileName))
        {
            throw new Exception(String.Format("File {0} does not exist", FileName));
        }
    }

    1 reference
    public List<BikeData> GetBatch(int size = 20)
    {
        var bikeDataCollection = JsonConvert.DeserializeObject<List<BikeData>>(File.ReadAllText(FileName));
        int length = bikeDataCollection.Count;

        var list = new List<BikeData>();

        int end = ((marker + size) > length) ? length : marker + size;

        for (var i = marker; i < end; i++)
        {
            list.Add(bikeDataCollection[i]);
        }
        marker = end;

        return list;
    }
}
```

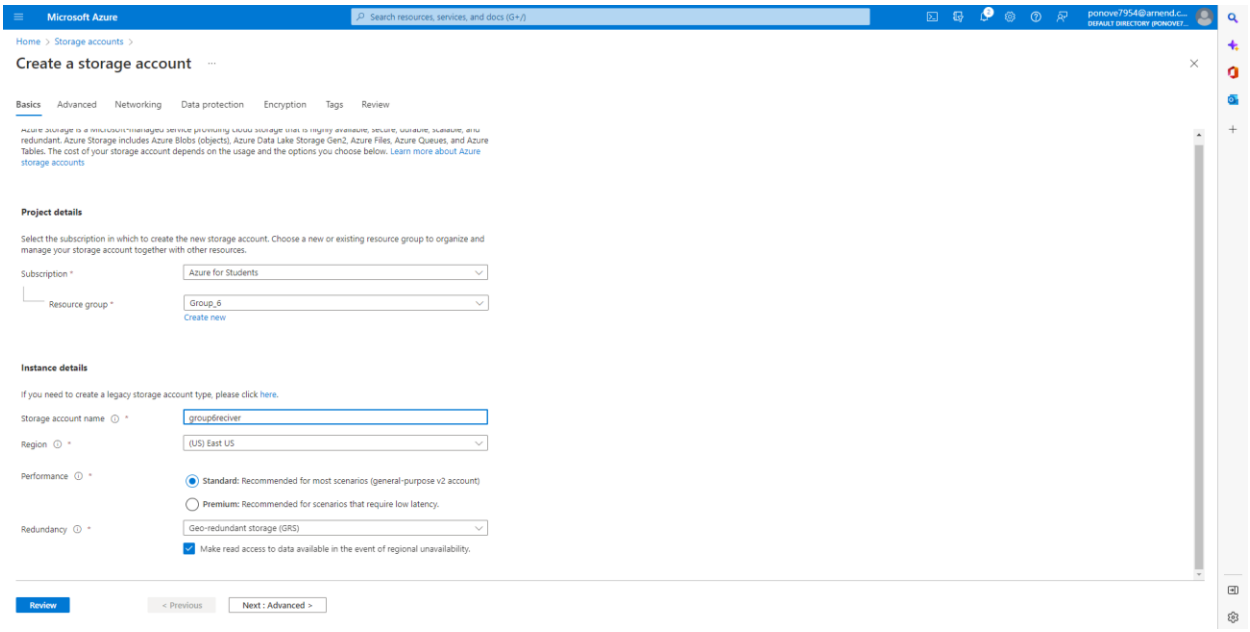
8- EventHubsSender application Running (10-second intervals per batch):

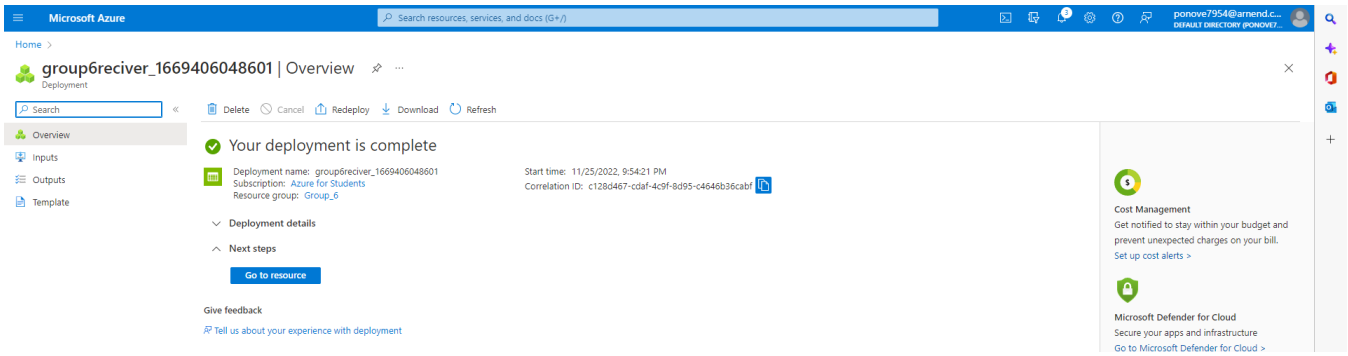
Select Microsoft Visual Studio Debug Console

```
A batch of 20 events has been published.
A batch of 20 events has been published.
A batch of 20 events has been published.
A batch of 20 events has been published.
A batch of 20 events has been published.
A batch of 20 events has been published.
A batch of 20 events has been published.
A batch of 20 events has been published.
A batch of 20 events has been published.
A batch of 20 events has been published.
A batch of 20 events has been published.
A batch of 20 events has been published.
A batch of 20 events has been published.
A batch of 20 events has been published.
A batch of 20 events has been published.
A batch of 20 events has been published.
A batch of 20 events has been published.
A batch of 20 events has been published.
A batch of 20 events has been published.
A batch of 20 events has been published.
A batch of 20 events has been published.
A batch of 20 events has been published.
A batch of 20 events has been published.
A batch of 20 events has been published.
A batch of 20 events has been published.
Event publishing complete
```

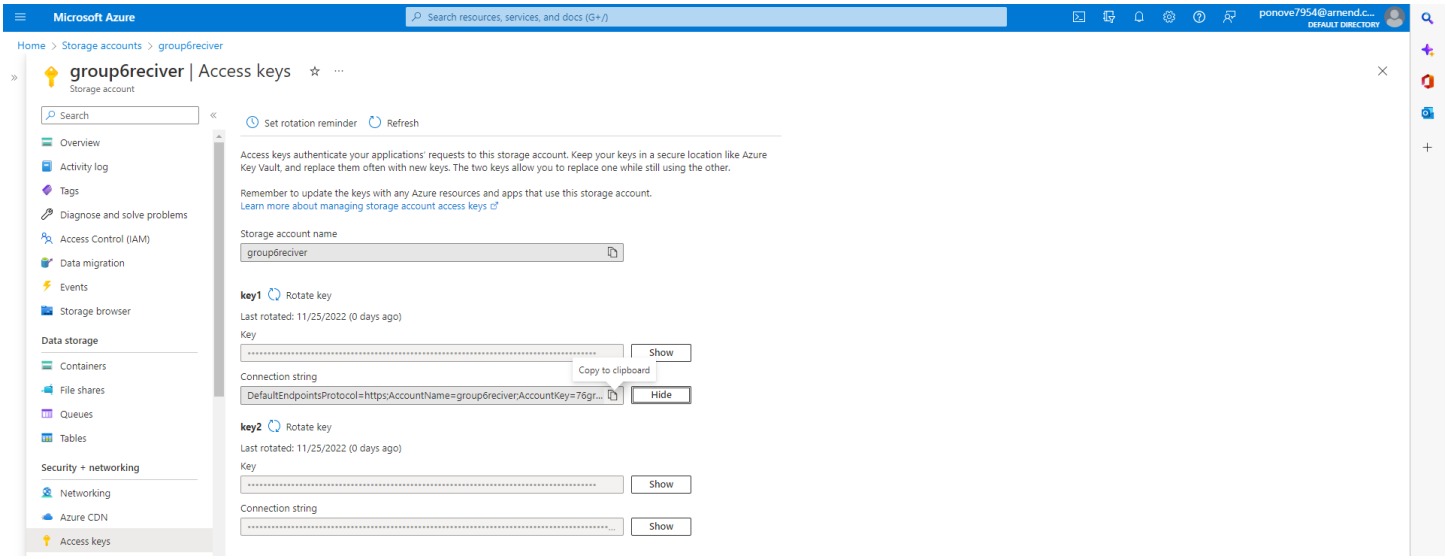
→ 25

9- Creating a Storage account:

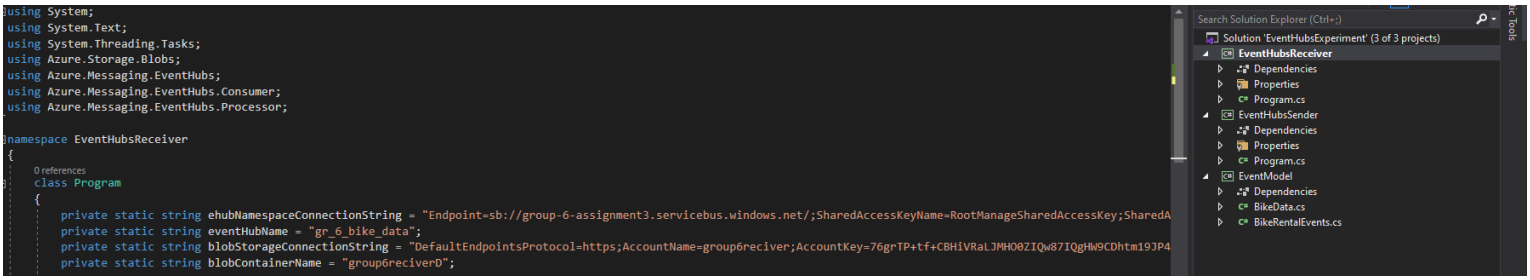




10- Getting the Access key:



11- Modifying EventHubsReceiver application:

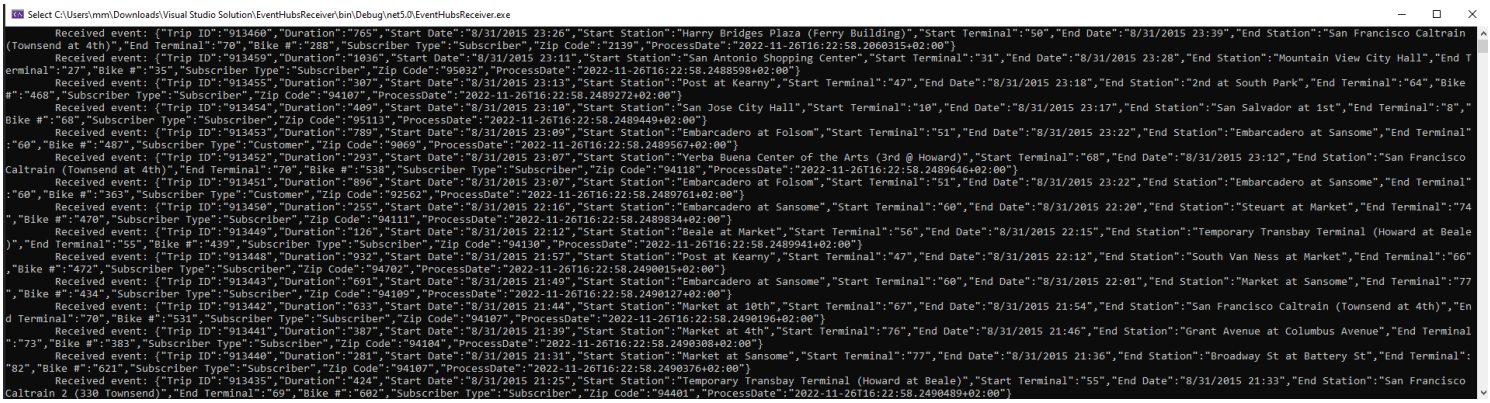


```
using System;
using System.Text;
using System.Threading.Tasks;
using Azure.Storage.Blobs;
using Azure.Messaging.EventHubs;
using Azure.Messaging.EventHubs.Consumer;
using Azure.Messaging.EventHubs.Processor;

namespace EventHubsReceiver
{
    0 references
    class Program
    {
        private static string ehubNamespaceConnectionString = "Endpoint=sb://group-6-assignment3.servicebus.windows.net/;SharedAccessKeyName=RootManageSharedAccessKey;SharedA
        private static string eventHubName = "gr_6_bike_data";
        private static string blobStorageConnectionString = "DefaultEndpointsProtocol=https;AccountName=group6reciver;AccountKey=76grTP+tf+CBHiVRaLJMHO0ZIQw87IQgHW9CDhtm19JP4
        private static string blobContainerName = "group6reciverD";
```

Search Solution Explorer (Ctrl+;)

- Solution 'EventHubsExperiment' (3 of 3 projects)
  - EventHubsReceiver
    - Dependencies
    - Properties
    - Program.cs
  - EventHubsSender
    - Dependencies
    - Properties
    - Program.cs
  - EventModel
    - Dependencies
    - BikeData.cs
    - BikeRentalEvents.cs

12- Running EventHubsReceiver application:

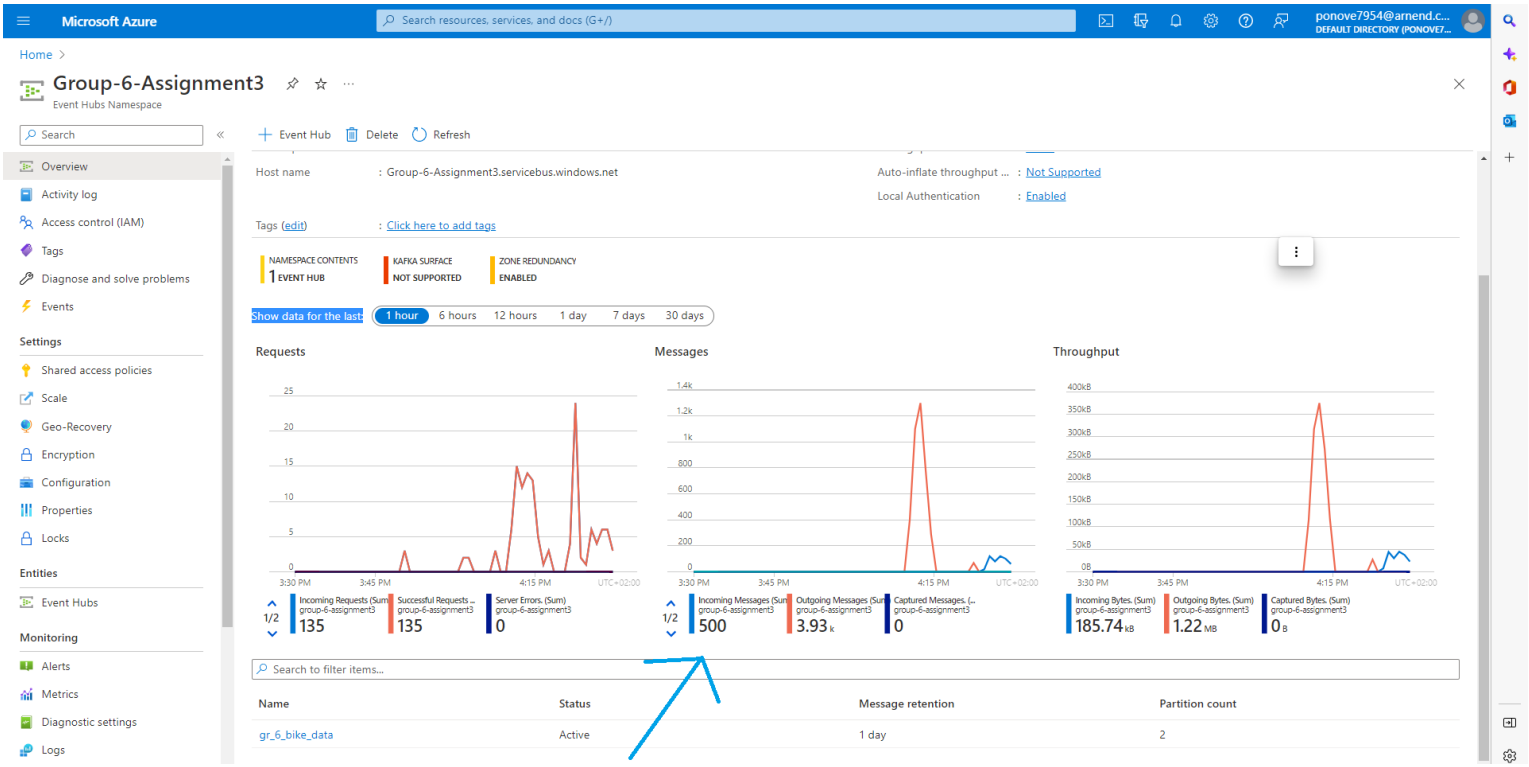


Select C:\Users\mm\Downloads\Visual Studio Solution\EventHubsReceiver\bin\Debug\net5.0\EventHubsReceiver.exe

```
        Received event: {"Trip ID":"913460","Duration":"765","Start Date":"8/31/2015 23:26","Start Station":"Harry Bridges Plaza (Ferry Building)","Start Terminal":"50","End Date":"8/31/2015 23:39","End Station":"San Francisco Caltrain (Townsend at 4th)","End Terminal":"70","Bike #":"288","Subscriber Type":"Subscriber","Zip Code":"2139","ProcessDate":"2022-11-26T16:22:58.2060315+02:00"}
        Received event: {"Trip ID":"913459","Duration":"1036","Start Date":"8/31/2015 23:11","Start Station":"San Antonio Shopping Center","Start Terminal":"31","End Date":"8/31/2015 23:28","End Station":"Mountain View City Hall","End Terminal":"27","Bike #":"35","Subscriber Type":"Subscriber","Zip Code":"95032","ProcessDate":"2022-11-26T16:22:58.2488598+02:00"}
        Received event: {"Trip ID":"913455","Duration":"307","Start Date":"8/31/2015 23:13","Start Station":"Post at Kearny","Start Terminal":"47","End Date":"8/31/2015 23:18","End Station":"2nd at South Park","End Terminal":"64","Bike #":"468","Subscriber Type":"Subscriber","Zip Code":"94107","ProcessDate":"2022-11-26T16:22:58.2489272+02:00"}
        Received event: {"Trip ID":"913454","Duration":"409","Start Date":"8/31/2015 23:10","Start Station":"San Jose City Hall","Start Terminal":"10","End Date":"8/31/2015 23:17","End Station":"San Salvador at 1st","End Terminal":"8","Bike #":"68","Subscriber Type":"Subscriber","Zip Code":"95113","ProcessDate":"2022-11-26T16:22:58.2489449+02:00"}
        Received event: {"Trip ID":"913453","Duration":"789","Start Date":"8/31/2015 23:09","Start Station":"Embarcadero at Folsom","Start Terminal":"51","End Date":"8/31/2015 23:22","End Station":"Embarcadero at Sansome","End Terminal":"60","Bike #":"487","Subscriber Type":"Customer","Zip Code":"9069","ProcessDate":"2022-11-26T16:22:58.2489567+02:00"}
        Received event: {"Trip ID":"913452","Duration":"293","Start Date":"8/31/2015 23:07","Start Station":"Yerba Buena Center of the Arts (3rd @ Howard)","Start Terminal":"68","End Date":"8/31/2015 23:12","End Station":"San Francisco Caltrain (Townsend at 4th)","End Terminal":"70","Bike #":"538","Subscriber Type":"Subscriber","Zip Code":"94118","ProcessDate":"2022-11-26T16:22:58.2489646+02:00"}
        Received event: {"Trip ID":"913451","Duration":"896","Start Date":"8/31/2015 23:07","Start Station":"Embarcadero at Folsom","Start Terminal":"51","End Date":"8/31/2015 23:22","End Station":"Embarcadero at Sansome","End Terminal":"60","Bike #":"363","Subscriber Type":"Customer","Zip Code":"92562","ProcessDate":"2022-11-26T16:22:58.2489761+02:00"}
        Received event: {"Trip ID":"913450","Duration":"255","Start Date":"8/31/2015 22:16","Start Station":"Embarcadero at Sansome","Start Terminal":"60","End Date":"8/31/2015 22:20","End Station":"Steuart at Market","End Terminal":"74","Bike #":"470","Subscriber Type":"Subscriber","Zip Code":"94111","ProcessDate":"2022-11-26T16:22:58.2489834+02:00"}
        Received event: {"Trip ID":"913449","Duration":"126","Start Date":"8/31/2015 22:12","Start Station":"Beale at Market","Start Terminal":"56","End Date":"8/31/2015 22:15","End Station":"Temporary Transbay Terminal (Howard at Beale)","End Terminal":"55","Bike #":"439","Subscriber Type":"Subscriber","Zip Code":"94130","ProcessDate":"2022-11-26T16:22:58.2489941+02:00"}
        Received event: {"Trip ID":"913448","Duration":"932","Start Date":"8/31/2015 21:57","Start Station":"Post at Kearny","Start Terminal":"47","End Date":"8/31/2015 22:12","End Station":"South Van Ness at Market","End Terminal":"66","Bike #":"472","Subscriber Type":"Subscriber","Zip Code":"94702","ProcessDate":"2022-11-26T16:22:58.2490015+02:00"}
        Received event: {"Trip ID":"913443","Duration":"691","Start Date":"8/31/2015 21:49","Start Station":"Embarcadero at Sansome","Start Terminal":"60","End Date":"8/31/2015 22:01","End Station":"Market at Sansome","End Terminal":"77","Bike #":"434","Subscriber Type":"Subscriber","Zip Code":"94109","ProcessDate":"2022-11-26T16:22:58.2490127+02:00"}
        Received event: {"Trip ID":"913442","Duration":"633","Start Date":"8/31/2015 21:44","Start Station":"Market at 10th","Start Terminal":"67","End Date":"8/31/2015 21:54","End Station":"San Francisco Caltrain (Townsend at 4th)","End Terminal":"70","Bike #":"531","Subscriber Type":"Subscriber","Zip Code":"94107","ProcessDate":"2022-11-26T16:22:58.2490196+02:00"}
        Received event: {"Trip ID":"913441","Duration":"387","Start Date":"8/31/2015 21:39","Start Station":"Market at 4th","Start Terminal":"76","End Date":"8/31/2015 21:46","End Station":"Grant Avenue at Columbus Avenue","End Terminal":"73","Bike #":"383","Subscriber Type":"Subscriber","Zip Code":"94104","ProcessDate":"2022-11-26T16:22:58.2490308+02:00"}
        Received event: {"Trip ID":"913440","Duration":"281","Start Date":"8/31/2015 21:31","Start Station":"Market at Sansome","Start Terminal":"77","End Date":"8/31/2015 21:36","End Station":"Broadway St at Battery St","End Terminal":"82","Bike #":"621","Subscriber Type":"Subscriber","Zip Code":"94107","ProcessDate":"2022-11-26T16:22:58.2490376+02:00"}
        Received event: {"Trip ID":"913435","Duration":"424","Start Date":"8/31/2015 21:25","Start Station":"Temporary Transbay Terminal (Howard at Beale)","Start Terminal":"55","End Date":"8/31/2015 21:33","End Station":"San Francisco Caltrain 2 (330 Townsend)","End Terminal":"69","Bike #":"602","Subscriber Type":"Subscriber","Zip Code":"94401","ProcessDate":"2022-11-26T16:22:58.2490489+02:00"}
```

B. Question b)

1- 500 messages incoming (25 x 20).



Microsoft Azure — Search resources, services, and docs (G+/) — ponove7954@arnend.c... DEFAULT DIRECTORY (PONOVE7...

Home >

Group-6-Assignment3

Event Hubs Namespace

Search

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

Settings

- Shared access policies
- Scale
- Geo-Recovery
- Encryption
- Configuration
- Properties
- Locks

Entities

- Event Hubs

Monitoring

- Alerts
- Metrics
- Diagnostic settings
- Logs

+ Event Hub | Delete | Refresh

Host name : Group-6-Assignment3.servicebus.windows.net

Auto-inflate throughput ... : Not Supported

Local Authentication : Enabled

Tags (edit) : Click here to add tags

NAMESPACE CONTENTS: 1 EVENT HUB | KAFKA SURFACE: NOT SUPPORTED | ZONE REDUNDANCY: ENABLED

Show data for the last: 1 hour | 6 hours | 12 hours | 1 day | 7 days | 30 days

Requests

1/2 | Incoming Requests (Sum) group-6-assignment3: 135 | Successful Requests ... group-6-assignment3: 135 | Server Errors. (Sum) group-6-assignment3: 0

Messages

1/2 | Incoming Messages (Sum) group-6-assignment3: 500 | Outgoing Messages (Sum) group-6-assignment3: 3.93 k | Captured Messages. (... group-6-assignment3: 0

Throughput

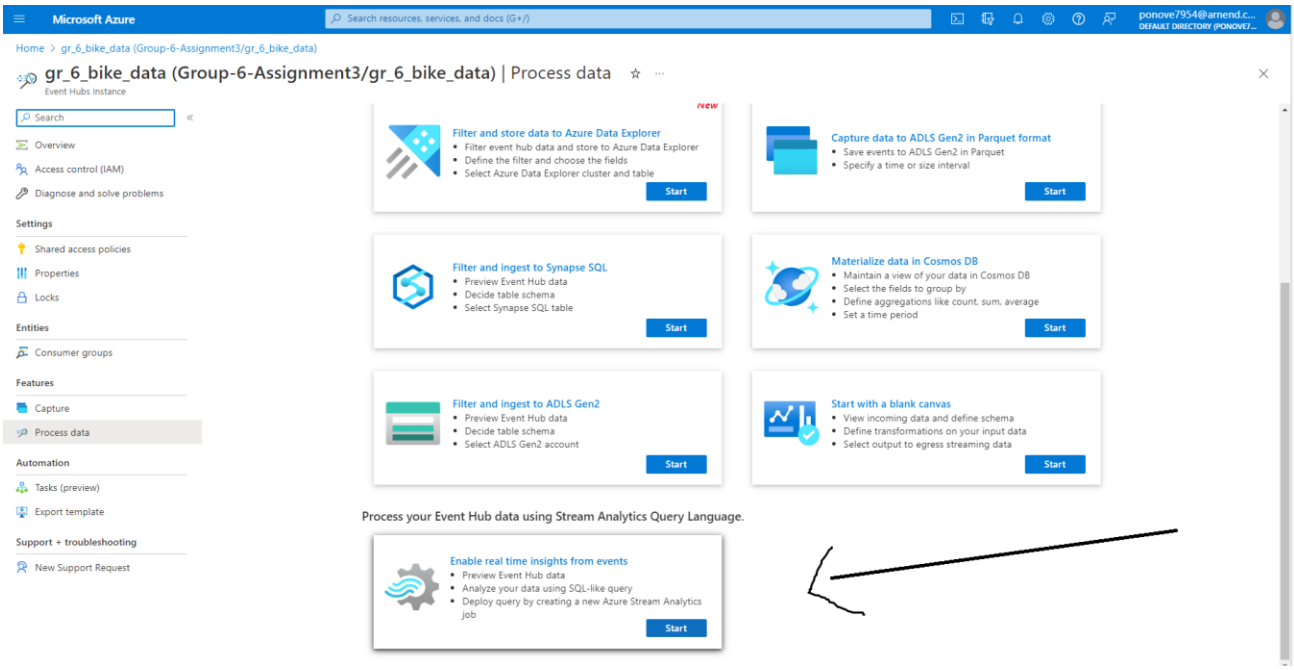
Incoming Bytes. (Sum) group-6-assignment3: 185.74 kB | Outgoing Bytes. (Sum) group-6-assignment3: 1.22 MB | Captured Bytes. (Sum) group-6-assignment3: 0 B

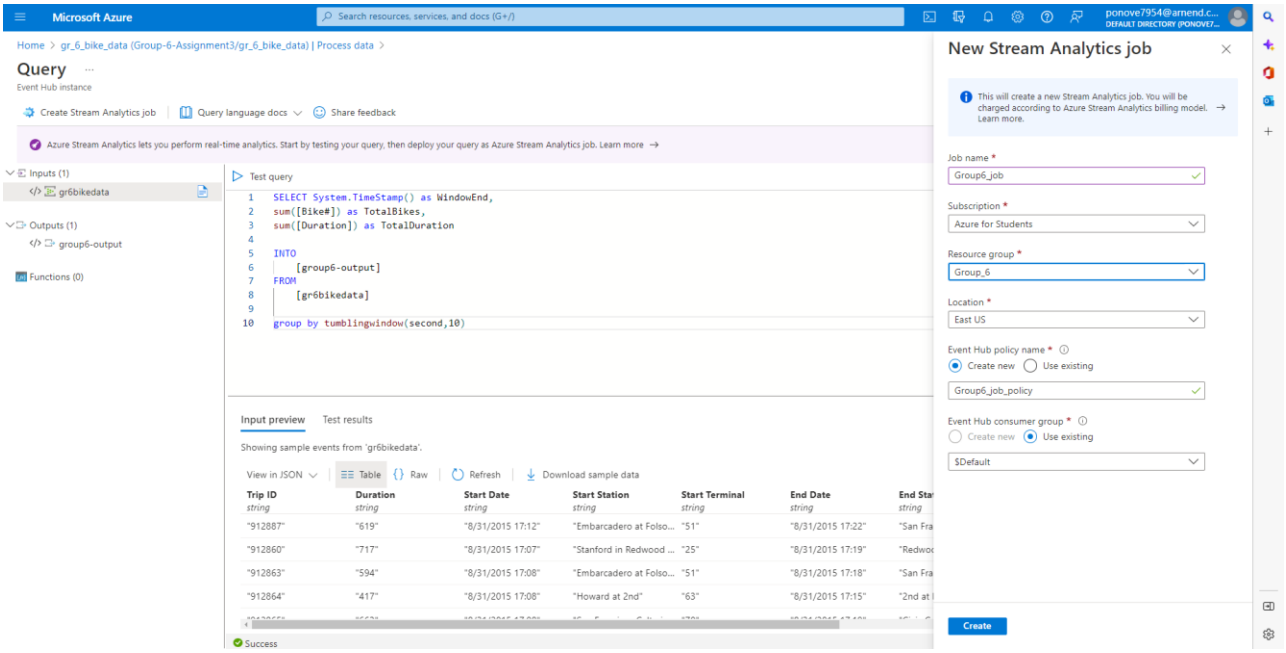
Search to filter items...

Name	Status	Message retention	Partition count
gr_6_bike_data	Active	1 day	2

C. Question c)

1- Creating stream analytics job:

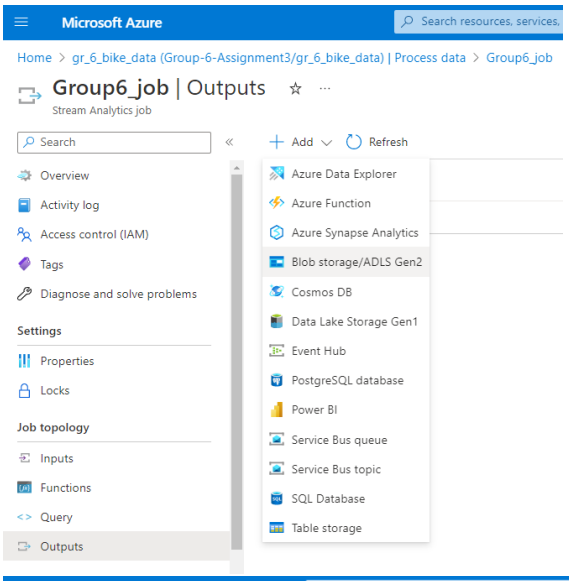


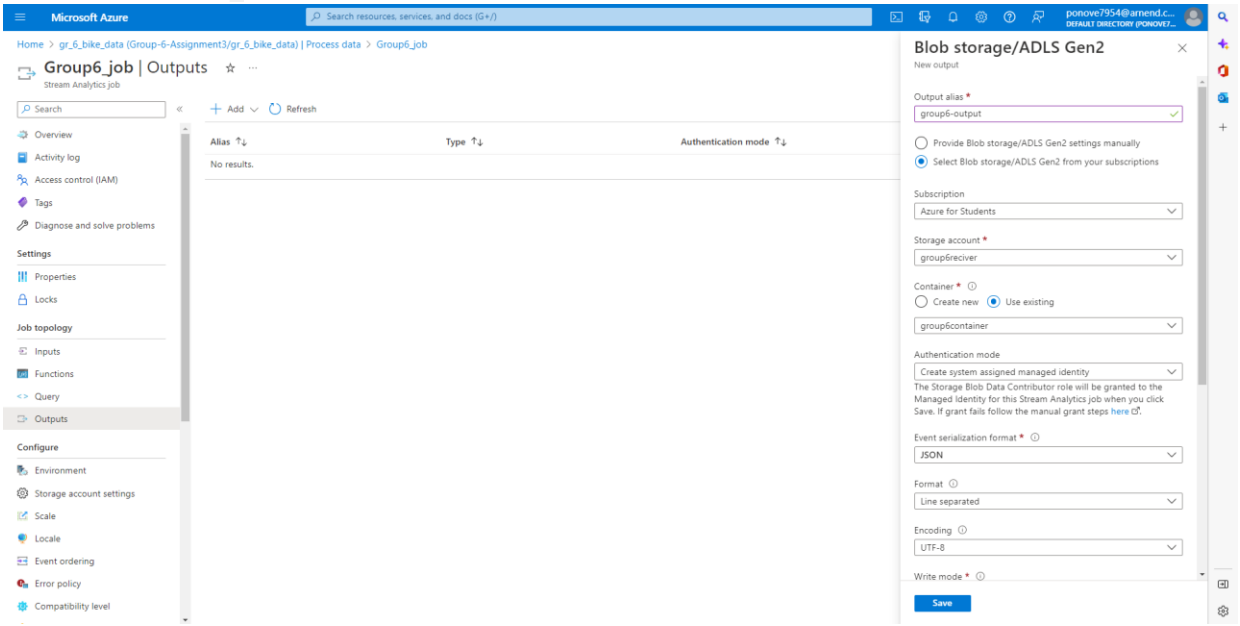


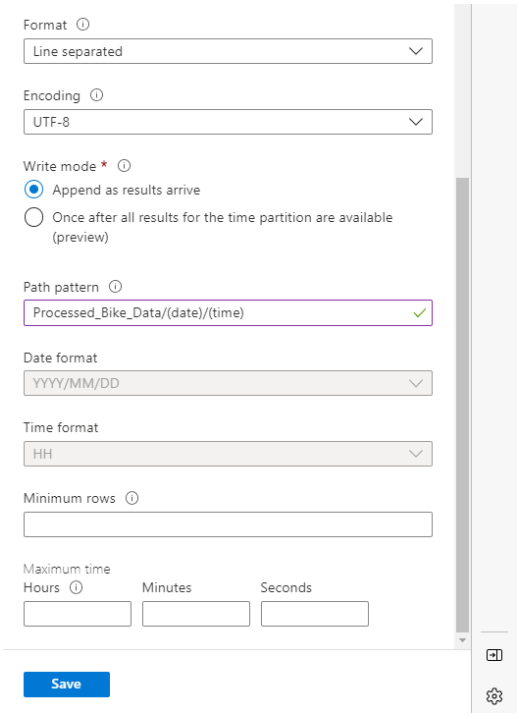
2- Query:

```
SELECT System.TimeStamp() as WindowEnd,
sum([Bike#]) as TotalBikes,
sum([Duration]) as TotalDuration

INTO
    [group6-output]
FROM
    [gr6bikedata]

group by tumblingwindow(second,10)
```

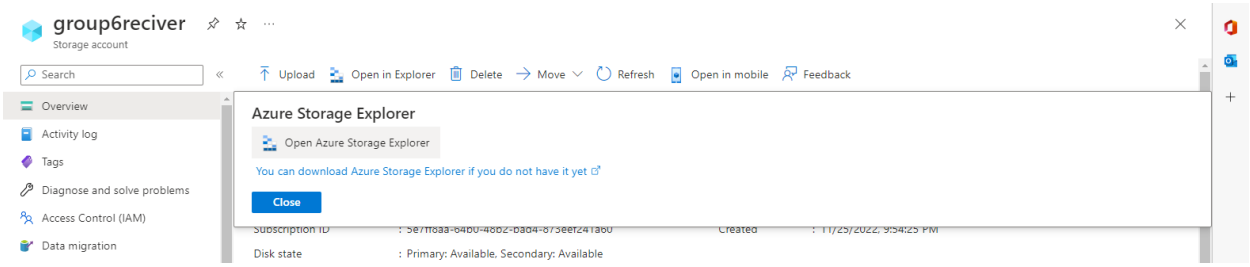
3- Creating outputs:

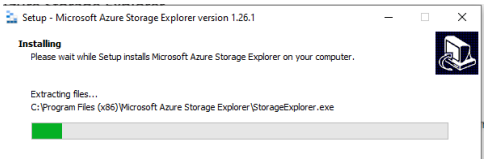


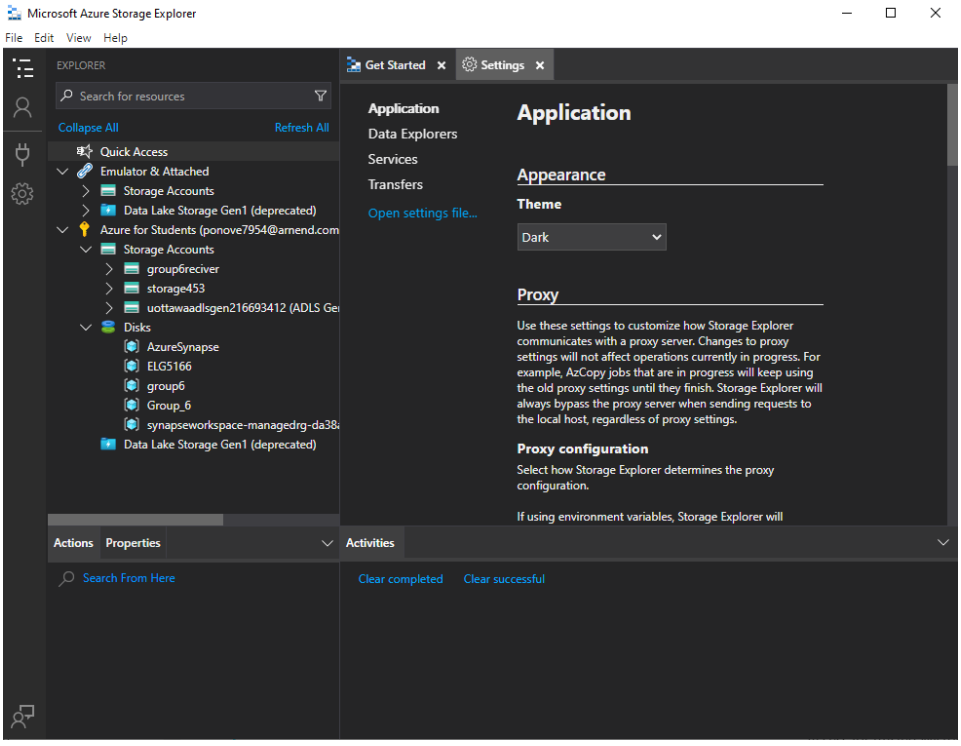




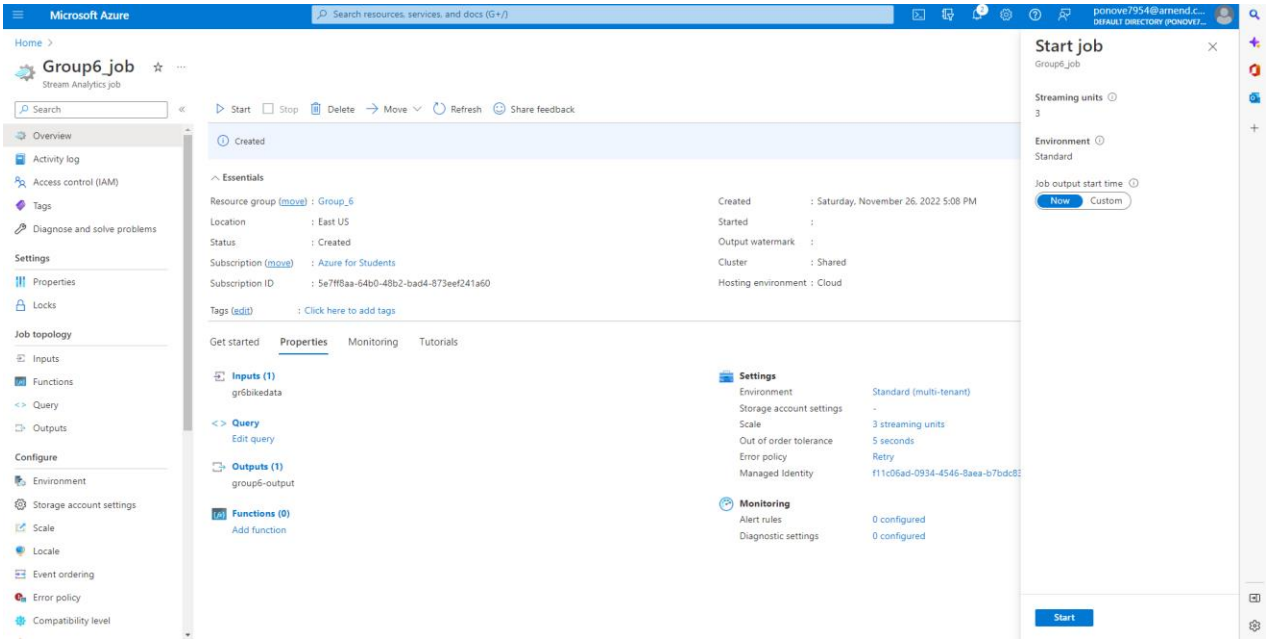
4- Downloading and installing Azure Storage Explorer:

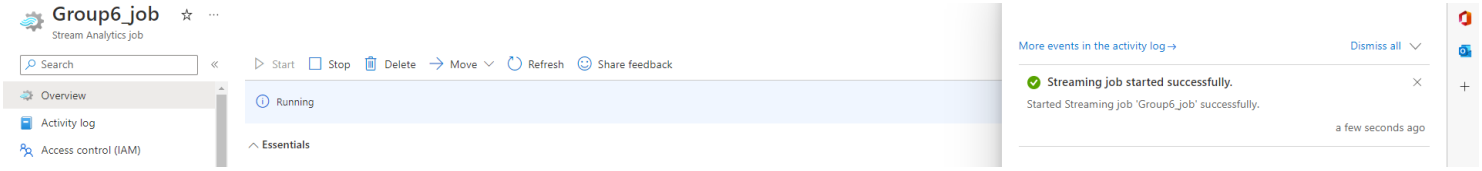




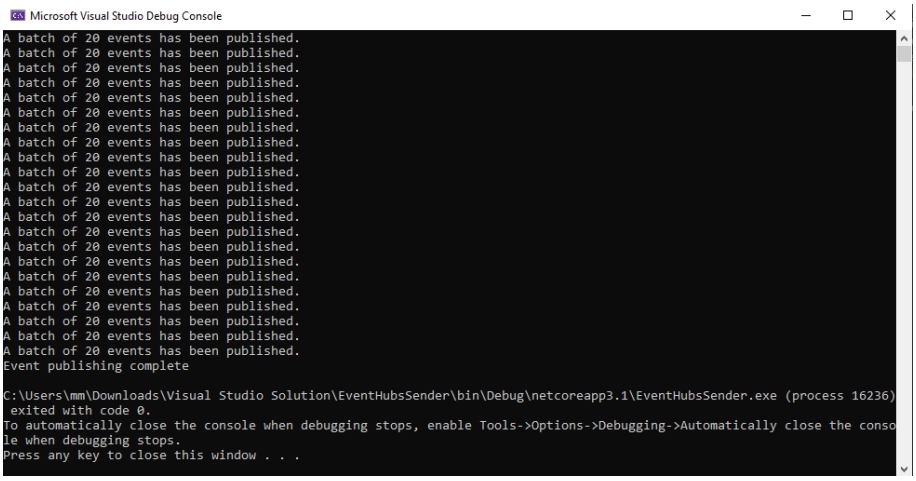


5- Starting the streaming job:

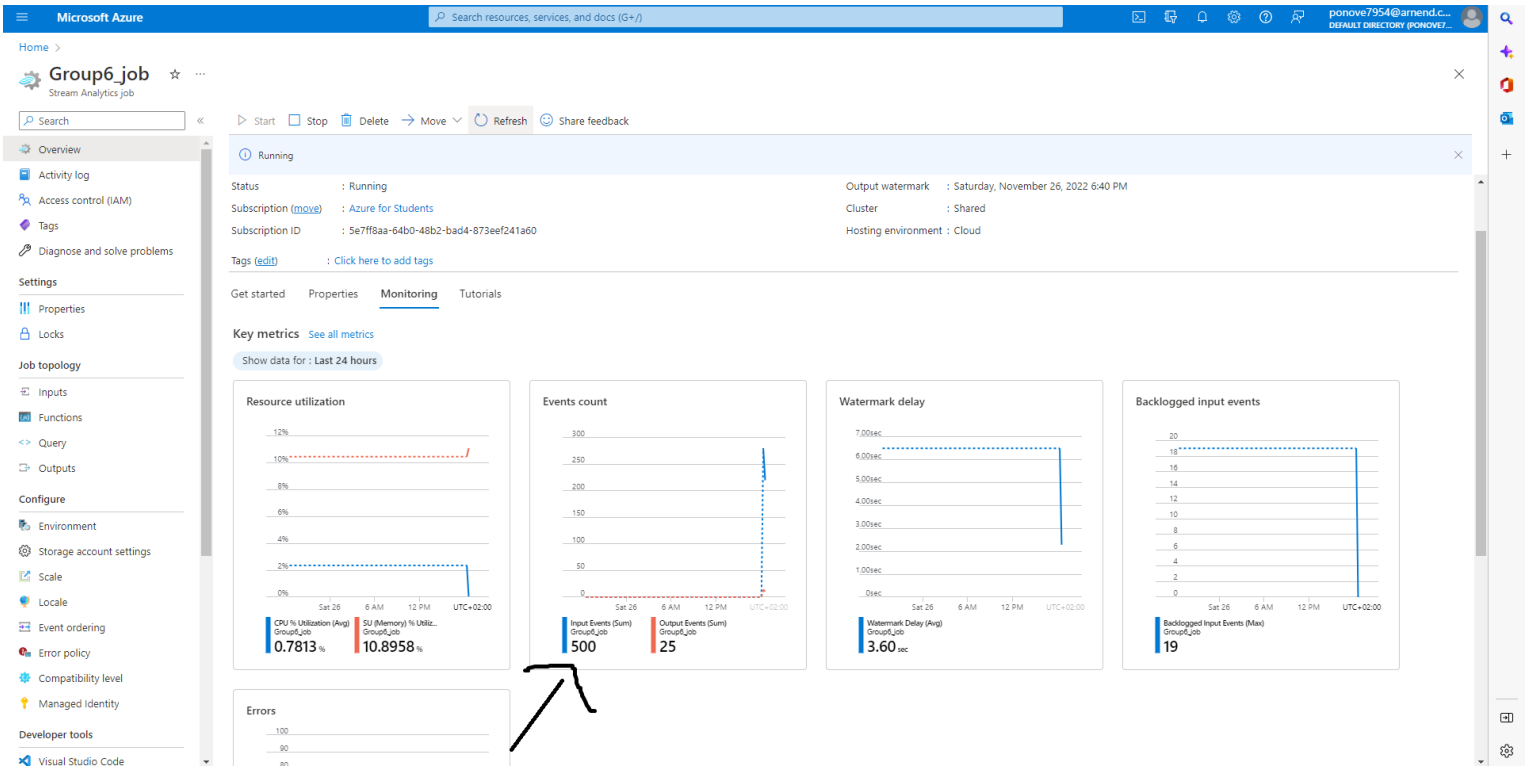




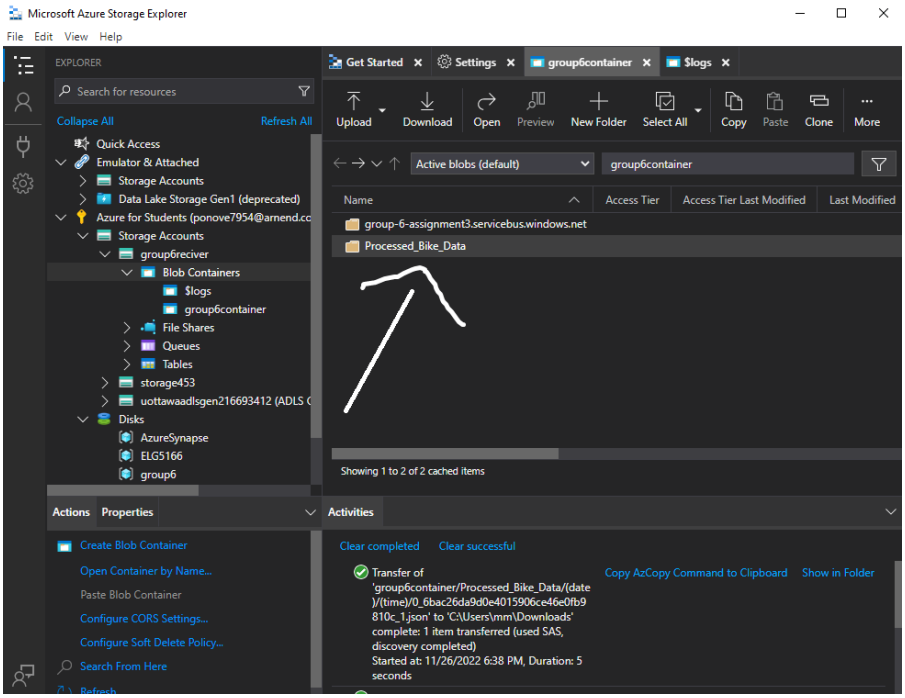
6- Starting the EventHubsSender application:

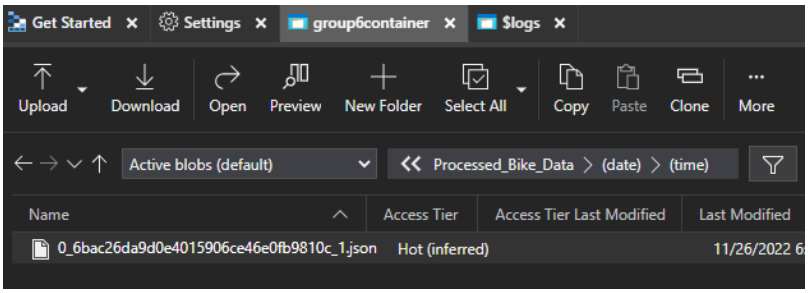


7- The Result of the streaming job:



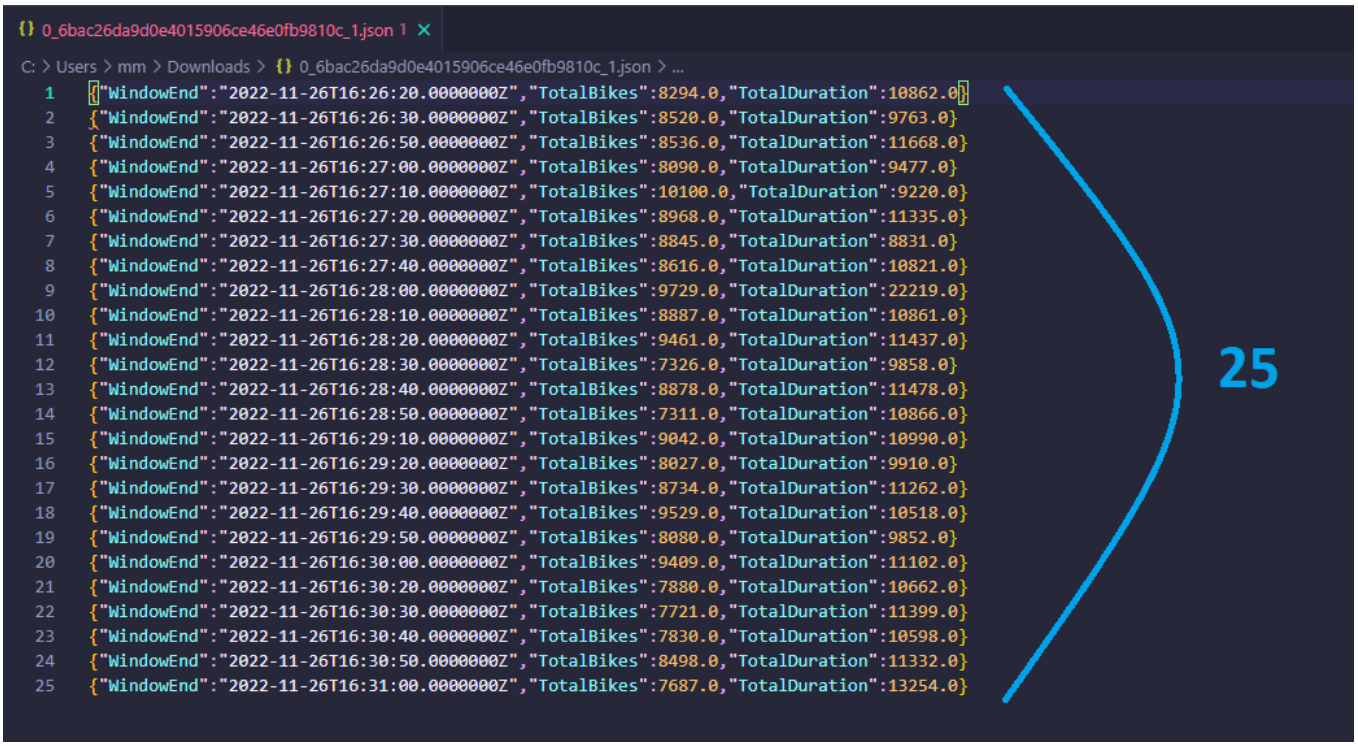
8- Getting the summary of the event received:





9- The Summary of the events received:

- **Each record is the summarized information of each batch (and there are 25 batches).**



10- Finally, we can run the Receiver again.

- ## Part 2 - Azure Synapse Analytics

A) Top 20 zip codes for bike up

Query:

```
select top(20) ZipCode, COUNT(ZipCode) AS ZipCodeCount
from trip_data
where ZipCode != 'nil'
GROUP BY ZipCode
ORDER BY COUNT(ZipCode) DESC
```


```
select top(20) ZipCode, COUNT(ZipCode) AS ZipCodeCount
from trip_data
where ZipCode != 'nil'
GROUP BY ZipCode
ORDER BY COUNT(ZipCode) DESC
```


ZipCode	ZipCodeCount
94111	10960
94102	10150
94109	6413
95112	4829
94403	4199
94611	4088
94158	4071
94117	4070
94501	4034
94602	3850
94110	3703
94114	3575
95110	3528
94010	3472
94610	3317
94040	3312

B) Monthly duration aggregate across the rental subscriber types, ordered in descending order of the busiest months (use a meaningful measure for the aggregate)

Query:

```
SELECT
       sum(duration) as duration,
       SubscriberType,
      STR(MONTH(StartDate)) as MonthDate
from trip_data
GROUP by SubscriberType, STR(MONTH(StartDate))
order by  duration DESC
```

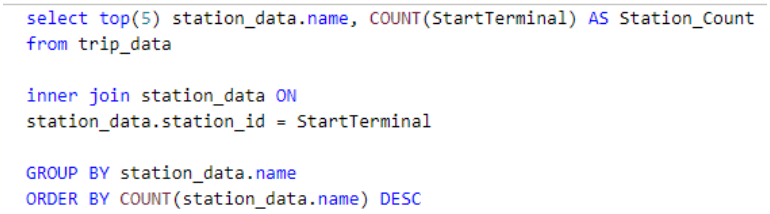

```
SELECT
       sum(duration) as duration,
       SubscriberType,
      STR(MONTH(StartDate)) as MonthDate
from trip_data

GROUP by SubscriberType, STR(MONTH(StartDate))
order by  duration DESC
```


duration	SubscriberType	MonthDate
27183543	Customer	12
18157584	Customer	7
17889840	Customer	6
17707291	Subscriber	10
16980462	Customer	9
16592087	Subscriber	6
16559867	Subscriber	5
16297208	Subscriber	3
16255508	Subscriber	4
16179559	Subscriber	9
15825478	Subscriber	7
15696847	Subscriber	8
15693808	Customer	10
15392019	Customer	8
15024766	Customer	5
14533223	Subscriber	1

C) What are the top 5 busiest terminals for bike pickup?

Query:

```
select top(5) station_data.name, COUNT(StartTerminal) AS Station_Count
from trip_data

inner join station_data ON
station_data.station_id = StartTerminal

GROUP BY station_data.name
ORDER BY COUNT(station_data.name) DESC
```

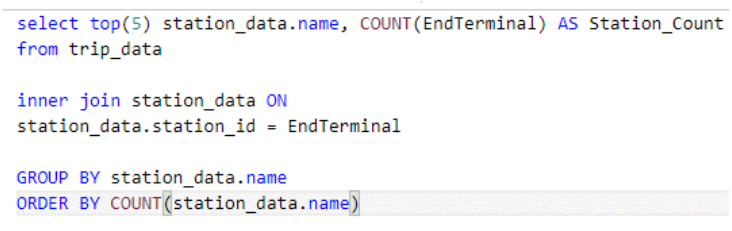


name	Station_Count
San Francisco Caltrain (Townsend...	26304
San Francisco Caltrain 2 (330 Tow...	21758
Harry Bridges Plaza (Ferry Buildin...	17255
Temporary Transbay Terminal (H...	14436
Embarcadero at Sansome	14158

D) Which 5 terminal has the least drop-offs?

Query:

```
select top(5) station_data.name, COUNT(EndTerminal) AS Station_Count
from trip_data

inner join station_data ON
station_data.station_id = EndTerminal

GROUP BY station_data.name
ORDER BY COUNT(station_data.name)
```



name	Station_Count
Redwood City Public Library	98
Franklin at Maple	100
Mezes Park	145
San Mateo County Center	187
Redwood City Medical Center	230

E) Produce the monthly summary of bike rentals

Query:

```
WITH Monthly_most_StartStation AS (

    -- Select get the most busiest start Station in every month
SELECT
  STR(MONTH(StartDate)) + '/' +  STR(YEAR(StartDate)) as datee,
  station_data.name,
  COUNT(*) AS cnt,
  COUNT(*) OVER (PARTITION BY  STR(MONTH(StartDate)) + '/' +  STR(YEAR(StartDate))) AS cat_cnt,
  ROW_NUMBER() OVER (PARTITION BY   STR(MONTH(StartDate)) + '/' +  STR(YEAR(StartDate)) ORDER B
Y COUNT(*) DESC) AS rn

FROM trip_data
inner join station_data ON
station_data.station_id = StartTerminal
GROUP BY
   STR(MONTH(StartDate)) + '/' +  STR(YEAR(StartDate)),
  station_data.name

),
    -- Select get the most busiest end Station in every month
Monthly_most_EndStation AS (
SELECT
  STR(MONTH(StartDate)) + '/' +  STR(YEAR(StartDate)) as datee,
  station_data.name,
  COUNT(*) AS cnt,
  COUNT(*) OVER (PARTITION BY  STR(MONTH(StartDate)) + '/' +  STR(YEAR(StartDate))) AS cat_cnt,
  ROW_NUMBER() OVER (PARTITION BY   STR(MONTH(StartDate)) + '/' +  STR(YEAR(StartDate)) ORDER B
Y COUNT(*) DESC) AS rn

FROM trip_data
inner join station_data ON
station_data.station_id = EndTerminal
GROUP BY
   STR(MONTH(StartDate)) + '/' +  STR(YEAR(StartDate)),
  station_data.name
)
 SELECT
      STR(MONTH(StartDate)) + '/' +  STR(YEAR(StartDate))  as MonthDate,
      sum(duration) as duration,
      COUNT(TripID) as trip_counts,
      CAST(sum(duration) as DECIMAL ) / (select sum(duration) from trip_data )    as duration_A
vg,
      CAST(SUM(case when SubscriberType = 'Customer' then 1 else 0 end) as DECIMAL ) / COUNT(Tr
ipID)    as customer_avg,
      CAST(SUM(case when SubscriberType = 'Subscriber' then 1 else 0 end) as DECIMAL ) / COUNT(
TripID)    as Subscriber_avg,
      Monthly_most_StartStation.name as busiest_Startstation,
      Monthly_most_EndStation.name as busiest_Endstation
```

```
from trip_data

inner join Monthly_most_StartStation
on Monthly_most_StartStation.datee =STR(MONTH(StartDate)) + '/' +  STR(YEAR(StartDate)) and Mon
thly_most_StartStation.cat_cnt > 1 AND Monthly_most_StartStation.rn = 1

inner join Monthly_most_EndStation
on Monthly_most_EndStation.datee =STR(MONTH(StartDate)) + '/' +  STR(YEAR(StartDate)) and Month
ly_most_EndStation.cat_cnt > 1 AND Monthly_most_EndStation.rn = 1


GROUP by STR(MONTH(StartDate)) + '/' +  STR(YEAR(StartDate)), Monthly_most_StartStation.name ,M
onthly_most_EndStation.name
```

```
SELECT
    STR(MONTH(StartDate)) + '/' +  STR(YEAR(StartDate))  as MonthDate,
    sum(duration) as duration,
    COUNT(TripID) as trip_counts,
    CAST(sum(duration) as DECIMAL ) / (select sum(duration) from trip_data )    as duration_Avg,
    CAST(SUM(case when SubscriberType = 'Customer' then 1 else 0 end) as DECIMAL ) / COUNT(TripID)    as customer_avg,
    CAST(SUM(case when SubscriberType = 'Subscriber' then 1 else 0 end) as DECIMAL ) / COUNT(TripID)    as Subscriber_avg,
    Monthly_most_StartStation.name as busiest_Startstation,
    Monthly_most_EndStation.name as busiest_Endstation
from trip_data

inner join Monthly_most_StartStation
on Monthly_most_StartStation.datee =STR(MONTH(StartDate)) + '/' +  STR(YEAR(StartDate)) and Monthly_most_StartStation.cat_cnt > 1 AND Monthly_most_StartStation.rn = 1

inner join Monthly_most_EndStation
on Monthly_most_EndStation.datee =STR(MONTH(StartDate)) + '/' +  STR(YEAR(StartDate)) and Monthly_most_EndStation.cat_cnt > 1 AND Monthly_most_EndStation.rn = 1


GROUP by STR(MONTH(StartDate)) + '/' +  STR(YEAR(StartDate)), Monthly_most_StartStation.name ,Monthly_most_EndStation.name
```

MonthDate	duration	trip_counts	duration_Avg	customer_avg	Subscriber_avg	busiest_Startstation	busiest_Endstation
1/ 2015	25611358	27840	0.06913495158255023661	0.09956896551	0.90043103448	San Francisco Caltrain (Townsend at 4th)	San Francisco Caltrain (Townsend at 4th)
2/ 2015	25633016	26401	0.06919341489329599531	0.10276125904	0.89723874095	San Francisco Caltrain (Townsend at 4th)	San Francisco Caltrain (Townsend at 4th)
3/ 2015	29892301	31626	0.08069087091461600827	0.12249415038	0.87750584961	San Francisco Caltrain (Townsend at 4th)	San Francisco Caltrain (Townsend at 4th)
4/ 2015	28031940	31363	0.07566903772400328322	0.10601664381	0.89398335618	San Francisco Caltrain (Townsend at 4th)	San Francisco Caltrain (Townsend at 4th)
5/ 2015	31584633	29540	0.08525912890708951971	0.13524035206	0.86475964793	San Francisco Caltrain (Townsend at 4th)	San Francisco Caltrain (Townsend at 4th)
6/ 2015	34481927	31907	0.09308004494014068807	0.12658664242	0.87341335757	San Francisco Caltrain (Townsend at 4th)	San Francisco Caltrain (Townsend at 4th)
7/ 2015	33983062	32476	0.09173341554152664645	0.14854046064	0.85145953935	San Francisco Caltrain 2 (330 Townsend)	San Francisco Caltrain (Townsend at 4th)
8/ 2015	31088866	31904	0.08392086220755620394	0.14405717151	0.85594282848	San Francisco Caltrain 2 (330 Townsend)	San Francisco Caltrain (Townsend at 4th)
9/ 2014	33160021	31682	0.08951170985589085434	0.13338804368	0.86661195631	San Francisco Caltrain (Townsend at 4th)	San Francisco Caltrain (Townsend at 4th)
10/ 2014	33401099	34220	0.09016247253148259946	0.12819988310	0.87180011689	San Francisco Caltrain (Townsend at 4th)	San Francisco Caltrain (Townsend at 4th)
11/ 2014	22454934	25516	0.06061454355053570821	0.11337984010	0.88662015989	San Francisco Caltrain (Townsend at 4th)	San Francisco Caltrain (Townsend at 4th)
12/ 2014	41131402	19677	0.11102954735131225635	0.11643035015	0.88356964984	San Francisco Caltrain (Townsend at 4th)	San Francisco Caltrain (Townsend at 4th)

- ## Part 3 - Definitions (40 points)

1. Please compare briefly, based on at least 3 criteria, the differences in architecture between Apache Spark Structured Streaming and Azure Event Hubs & Synapse Analytics. [1:5]

	Spark Structured Streaming	Azure Event Hubs	Synapse Analytics.
data protection	Doesn't provide security services but you can manage it manually.	detecting the unauthorized transfer and providing encryption at rest and in transit by providing a TLS connection.	It provides access control, authentication, network security, and threat protection to identify unusual access locations, SQL injection attacks, and authentication attacks.
Supported protocols	TCP	AMQP, Kafka, and HTTPS	HTTP/1.1, and HTTPS
Supported languages	Scala, Java, and Python	. NET Standard, Java, Python, JavaScript, Go, C	SQL, Python, .NET, Java, Scala and R

2. Describe briefly 3 benefits of Azure Synapse Analytics over Apache Spark. Illustrate them briefly with some use cases? [6]

Some benefits of Azure Synapse Analytics:

1-**Unmatched security:**

Ensure the safety of your data with the most cutting-edge security and privacy tools available, including dynamic data masking and column- and row-level security.

Data experts can spend more time gaining insights if single sign-on and Azure Active Directory integration are used.

2-Limitless scale:

With lightning-fast delivery, big data analytics platforms and data warehouses can extract insights from all of your data.

Independent of the overall restriction for your storage account, Storage Analytics has a 20 TB limit on the volume of data that can be saved.

3-Machine Learning support:

Machine learning models can be built and saved in ONNX format, which is used with the native PREDICT command and stored in the Azure Synapse data storage.

4- Integration with Data Lake:

Files are read into the Data Lake in Parquet format from Azure Synapse, which results in a 13x performance improvement for Polybase execution.

5- Better BI & Data Visualization:

The reporting and analysis of important indicators are entertaining and simple to use thanks to the smooth and native connection with Power BI.

Sharing with appropriate stakeholders across business streams is now even simpler as a result of this.

Azure Synapse Analytics use cases:

By combining insights from various data sources, warehouses, and analytics solutions, Synapse Analytics helps businesses across sectors to use their data much more safely, accurately, productively, and efficiently.

1-Manufacturing:

Prevent Unplanned Downtime: Accurate forecasting of potential equipment failure to cut maintenance costs, prevent expensive downtime, and improve operational effectiveness.

2-Retail:

Create a strong supply chain by centralizing data from connected sales channels and generating insights in real-time to better serve customers.

3-Healthcare:

Automate Care Operations: Give patients quick access to the medical data they need to get the proper treatment quickly. Consolidating data from several health IT systems will speed up regular tasks and free up your care providers to concentrate on raising the standard of patient care.

3. What are the 5 characteristics of Azure Data Lake Storage that distinguish it from other Distributed Dataset Storage infrastructures such as Hadoop? [7:11]

1. ADLS is a fully managed, more flexible system.
2. The Data Lake is highly secure. Microsoft uses some of the most advanced security technology available.
3. Limitless Storage for a highly varied range of data, it stores data of any size, shape, and speed, and does all processing and analytics types across platforms and languages.
4. Optimized driver: The ABFS driver has been specifically designed for big data analytics.
5. ADLS serves as the batch processing layer's primary storage backbone.
6. Files and folders, similar to a local file system, with Azure Active Directory (AAD) controlling access to these file resources.
7. Enabling Hadoop for the Cloud: Because of the perceived limitations of cloud infrastructure, many enterprises implementing Hadoop choose an on-premises implementation. Azure Data Lake should bring the cloud's Hadoop limitations more in line with traditional installations.

- **References: -**

[1] Robb, Drew. 'Azure Synapse vs. Databricks: Data Platform Comparison'. EWEEK, 20 July 2022, https://www.eweek.com/big-data-and-analytics/azure-synapse-vs-databricks/.

[2] msmbaldwin. Azure Security Baseline for Event Hubs. https://learn.microsoft.com/en-us/security/benchmark/azure/baselines/event-hubs-security-baseline. Accessed 27 Nov. 2022.

[3] spelluru. Azure Event Hubs - Exchange Events Using Different Protocols - Azure Event Hubs. https://learn.microsoft.com/en-us/azure/event-hubs/event-hubs-exchange-events-different-protocols. Accessed 27 Nov. 2022.

[4] matt1883. Azure Synapse Analytics REST API Reference. https://learn.microsoft.com/en-us/rest/api/synapse/. Accessed 27 Nov. 2022.

[5] https://learn.microsoft.com/en-us/azure/synapse-analytics/sql/overview,-features

[6] "Azure Synapse Analytics Benefits & Use Cases for Industries." Rishabh Software, 2 Mar. 2022, https://www.rishabhsoft.com/blog/azure-synapse-analytics-use-cases-benefits.

[7] What Is Azure Data Lake? A Beginner's Guide to ADLS & Analytics. 9 Jan. 2022, https://cloudkeeda.com/azure-data-lake/.

[8] normesta. Azure Data Lake Storage Gen2 Introduction. https://learn.microsoft.com/en-us/azure/storage/blobs/data-lake-storage-introduction. Accessed 25 Nov. 2022.

[9] Nanua, Roshan. '6 Features of an Azure Data Lake to Boost Your Analytics'. Hitachi Solutions, 8 Mar. 2020, https://global.hitachi-solutions.com/blog/6-features-of-an-azure-data-lake-to-boost-your-analytics/.

[10] https://uottawa.brightspace.com/d2l/le/content/326283/viewContent/4735162/View. Accessed 25 Nov. 2022.

[11] Nanua, Roshan. '6 Features of an Azure Data Lake to Boost Your Analytics'. Hitachi Solutions, 8 Mar. 2020, https://global.hitachi-solutions.com/blog/6-features-of-an-azure-data-lake-to-boost-your-analytics/.