ELG 5166 – Cloud Analytics

Assignment #3

Submitted to: Dr. Benjamin Eze

Submitted by: Group - 11

Name	ID	Email
Youssef Metwally	300267001	ymetw027@uottawa.ca
Hussien Khaled	300266718	hkhal023@uottawa.ca
Abdelrahman Basha	300266859	abash023@uottawa.ca
Khadija Hesham	300266888	khesh072@uottawa.ca

November 28, 2021

Personal Ethics & Academic Integrity Statement

Student name:	Youssef Metwally	Student ID:	300267001
Student Name:	Hussien Khaled	Student ID:	300266718
Student Name:	Abdelrahman Basha	Student ID:	300266859
Student Name:	Khadija Hesham	Student ID:	300266888

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 non-participation 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

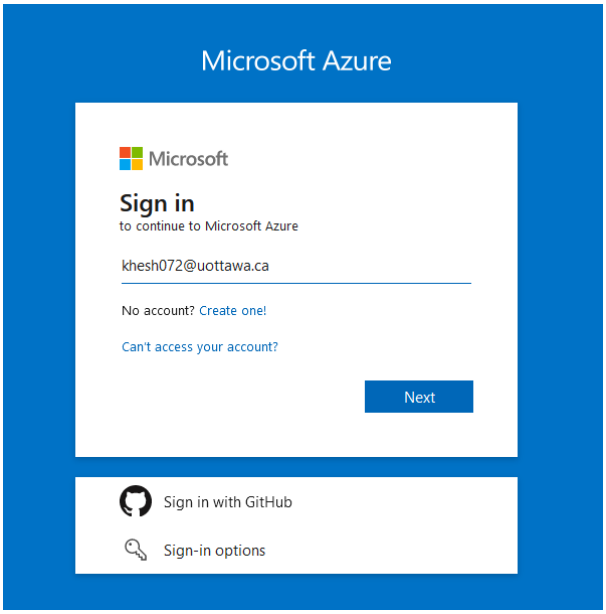
Part 1 - Event Hubs Analytics:

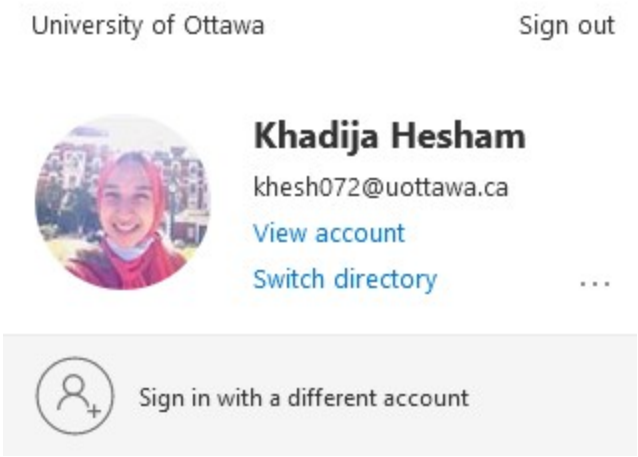
a) Top 20 zip codes where most bikes were rented from.

Part 1 – Event Hubs Analytics:

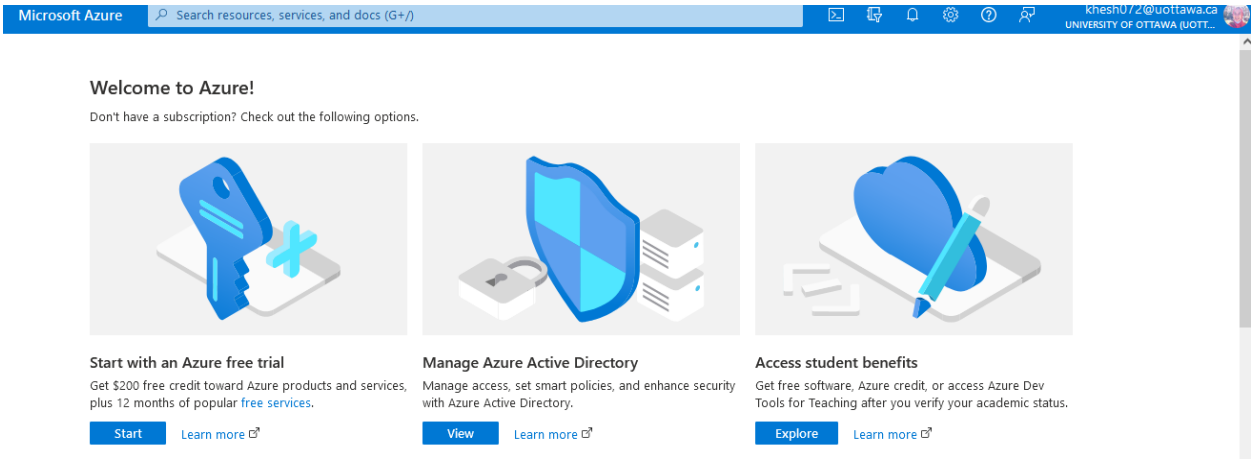
First of all, we set up our azure account.

I have signed in with my uOttawa account.

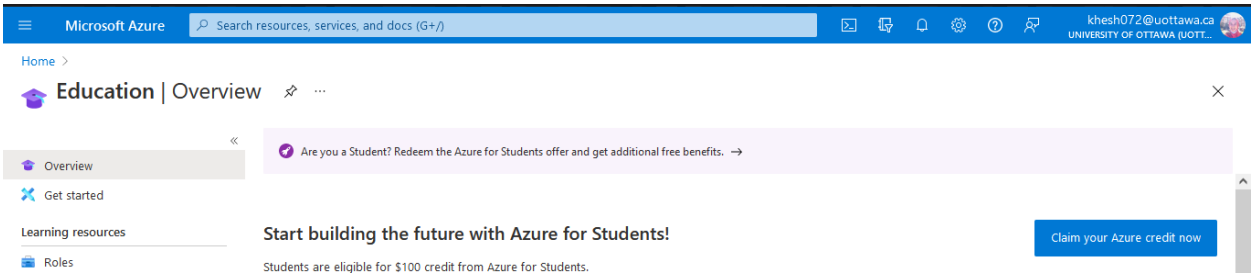


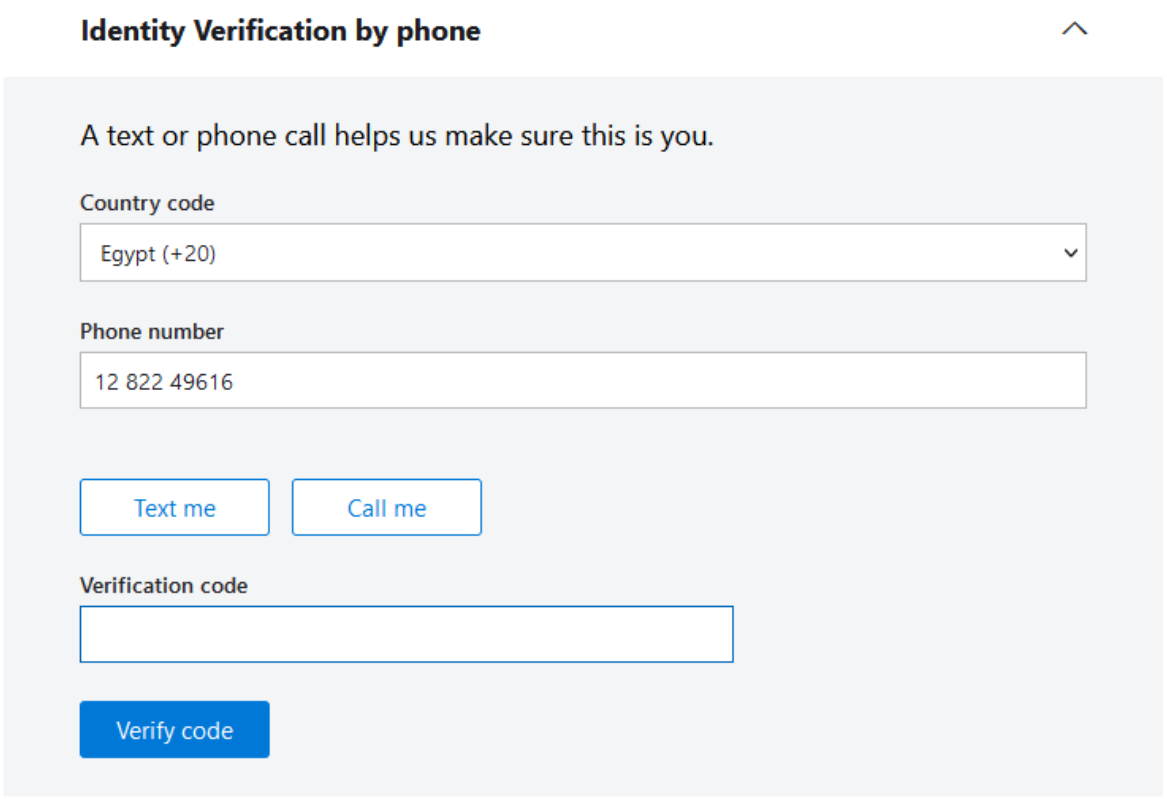


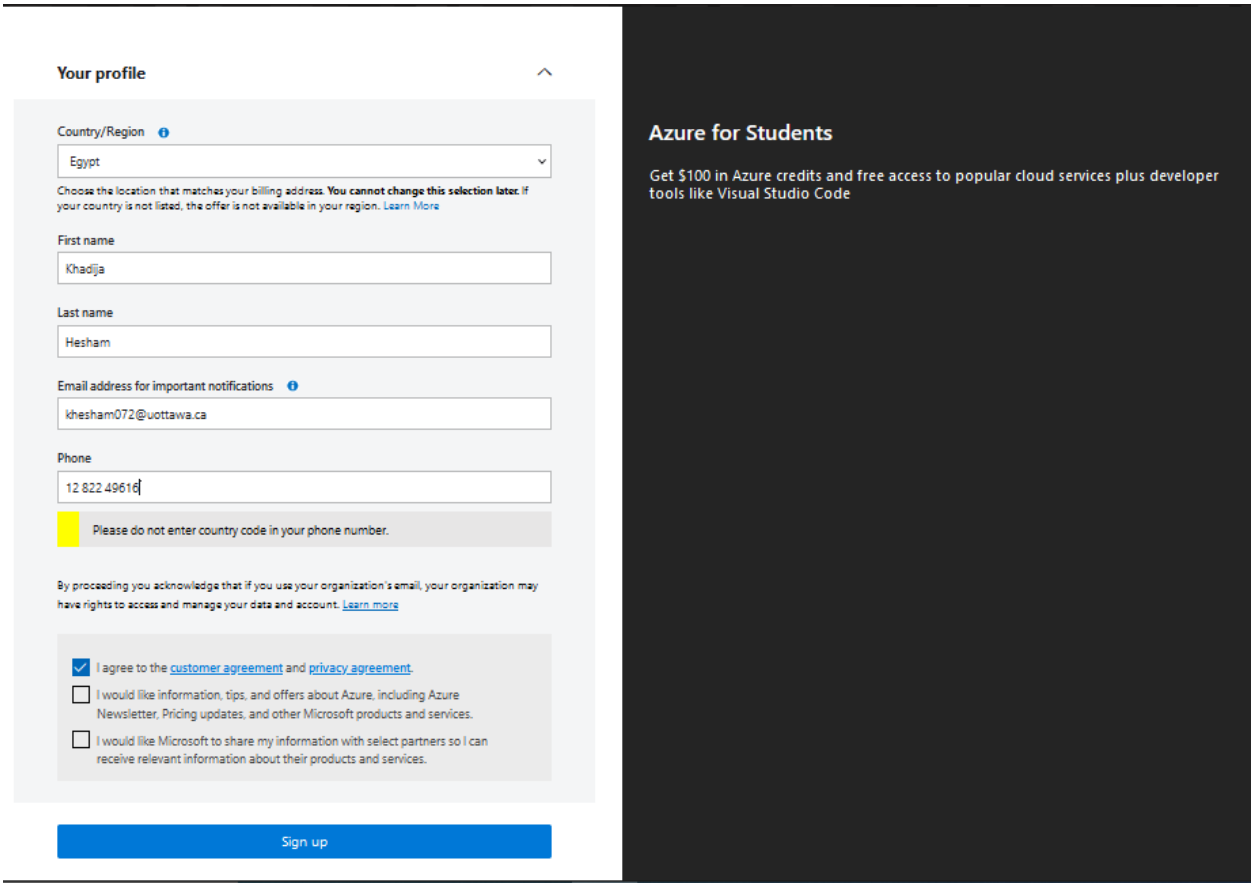
A screenshot from my azure.

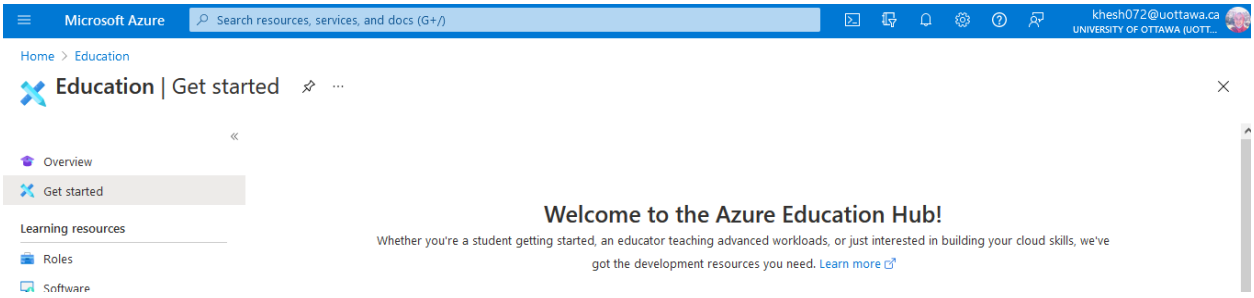


We click on the student tap to claim the students voucher.



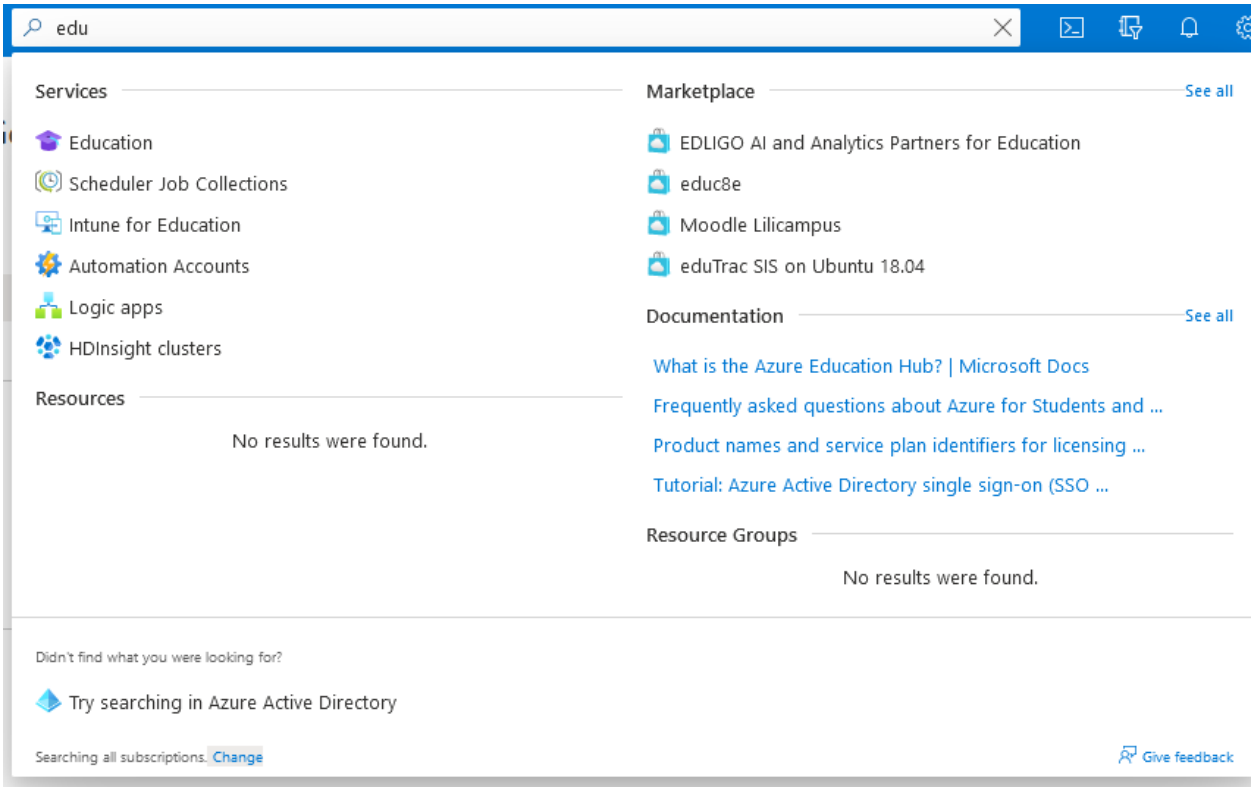

Identity Verification by phone
A text or phone call helps us make sure this is you.
Country code
Egypt (+20)
Phone number
12 822 49616
Text me
Call me
Verification code
Verify code


Your profile
Country/Region
Egypt
Choose the location that matches your billing address. You cannot change this selection later. If your country is not listed, the offer is not available in your region. Learn More
First name
Khadija
Last name
Hesham
Email address for important notifications
khesham072@uottawa.ca
Phone
12 822 49616
Please do not enter country code in your phone number.
By proceeding you acknowledge that if you use your organization's email, your organization may have rights to access and manage your data and account. Learn more
I agree to the customer agreement and privacy agreement.
I would like information, tips, and offers about Azure, including Azure Newsletter, Pricing updates, and other Microsoft products and services.
I would like Microsoft to share my information with select partners so I can receive relevant information about their products and services.
Sign up
Azure for Students
Get $100 in Azure credits and free access to popular cloud services plus developer tools like Visual Studio Code

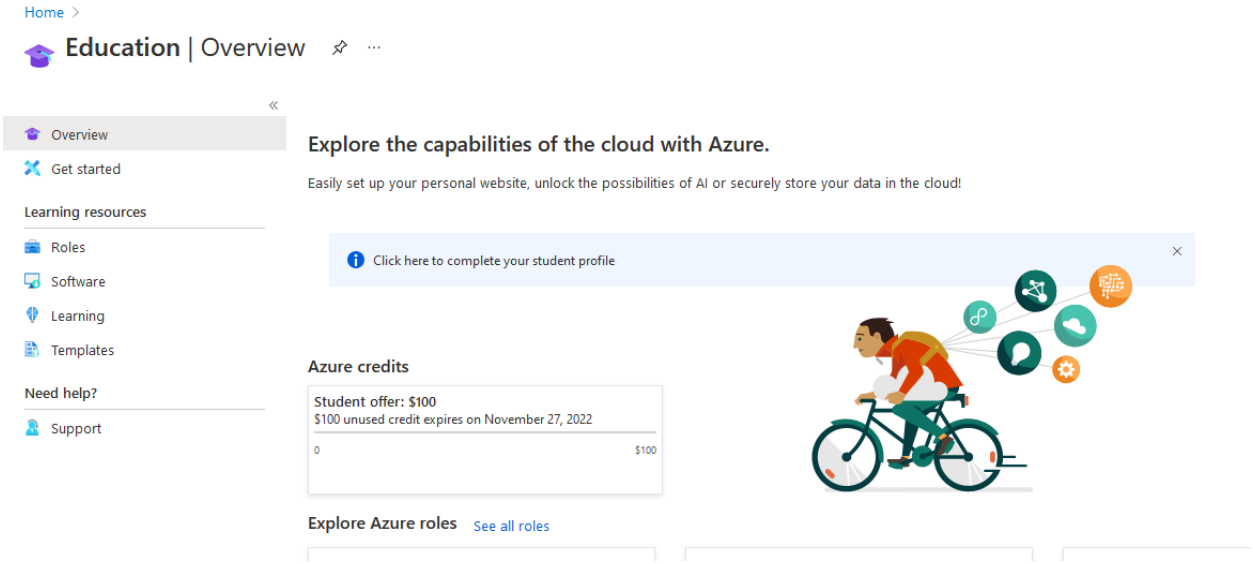


Install Visual Studio:

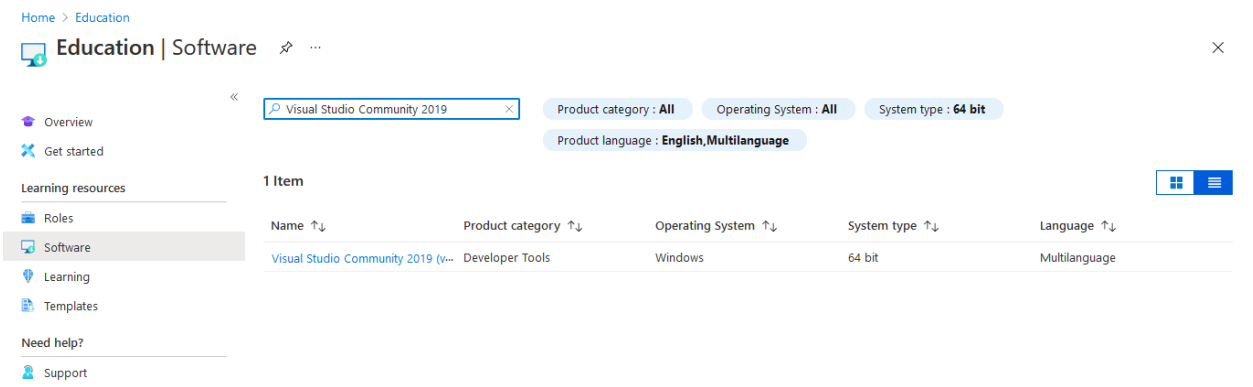
We choose the education services.



We got the 100$ voucher.

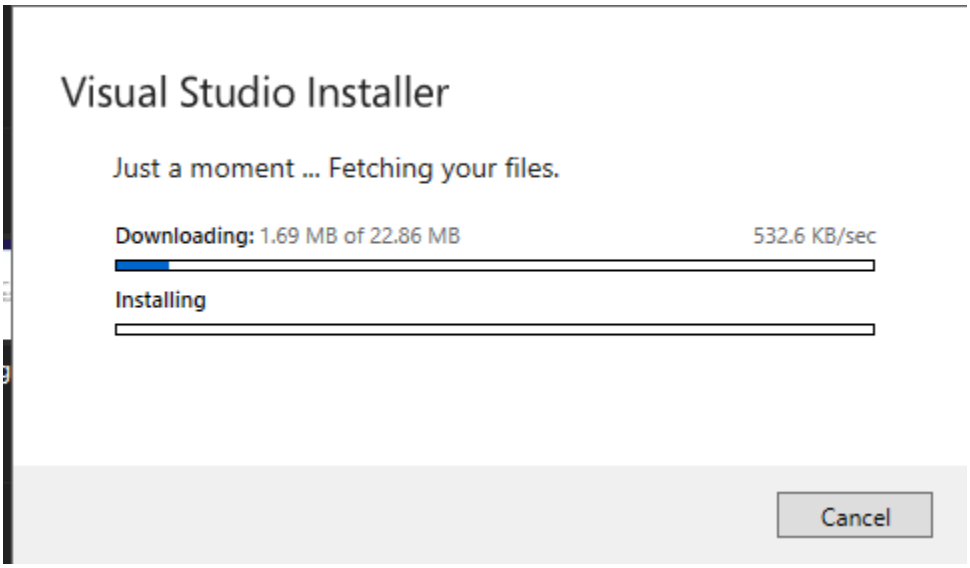


We search for Visual Studio Community 2019 in the software section in the menu at left.

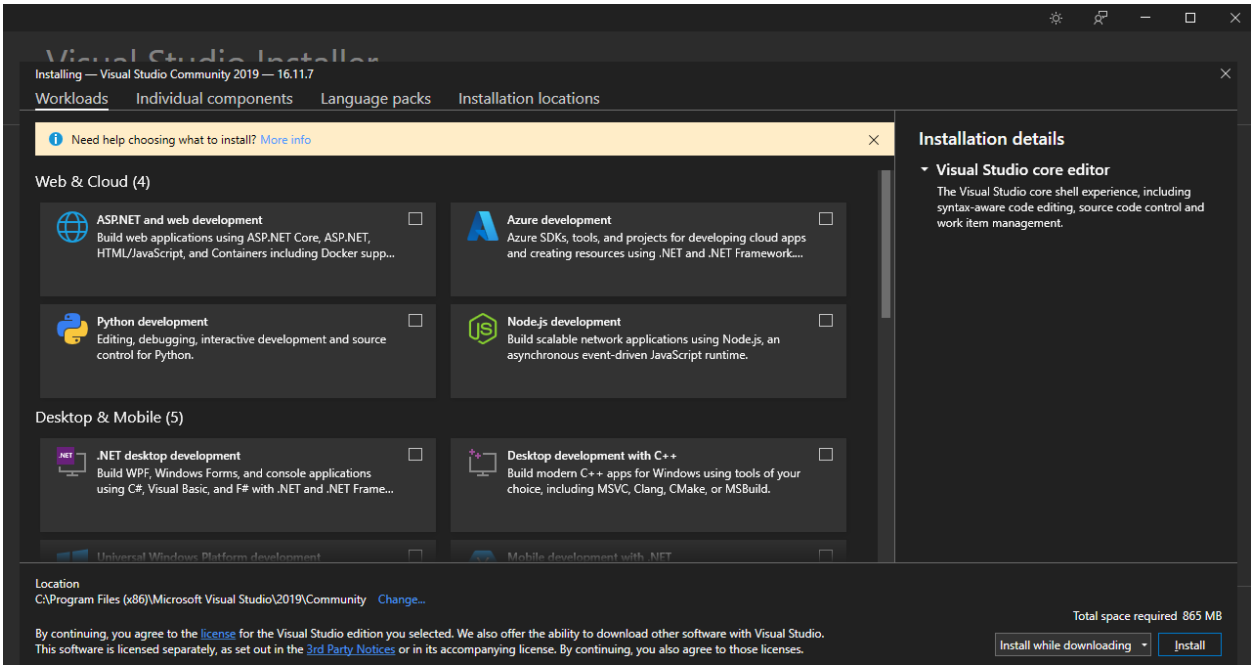


And choose to download it.

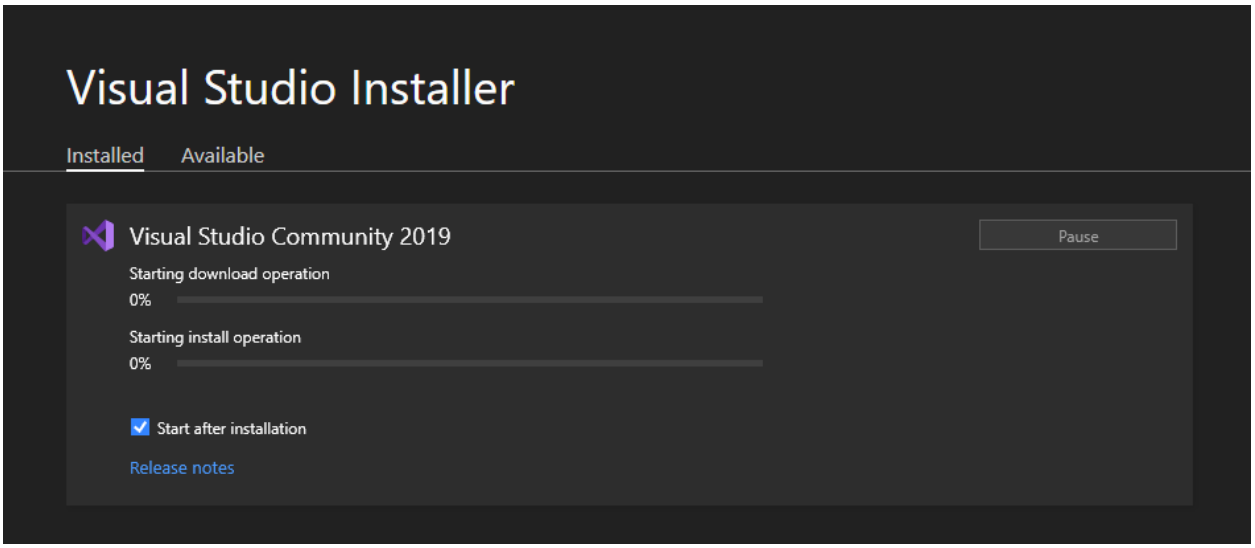
Download is started.



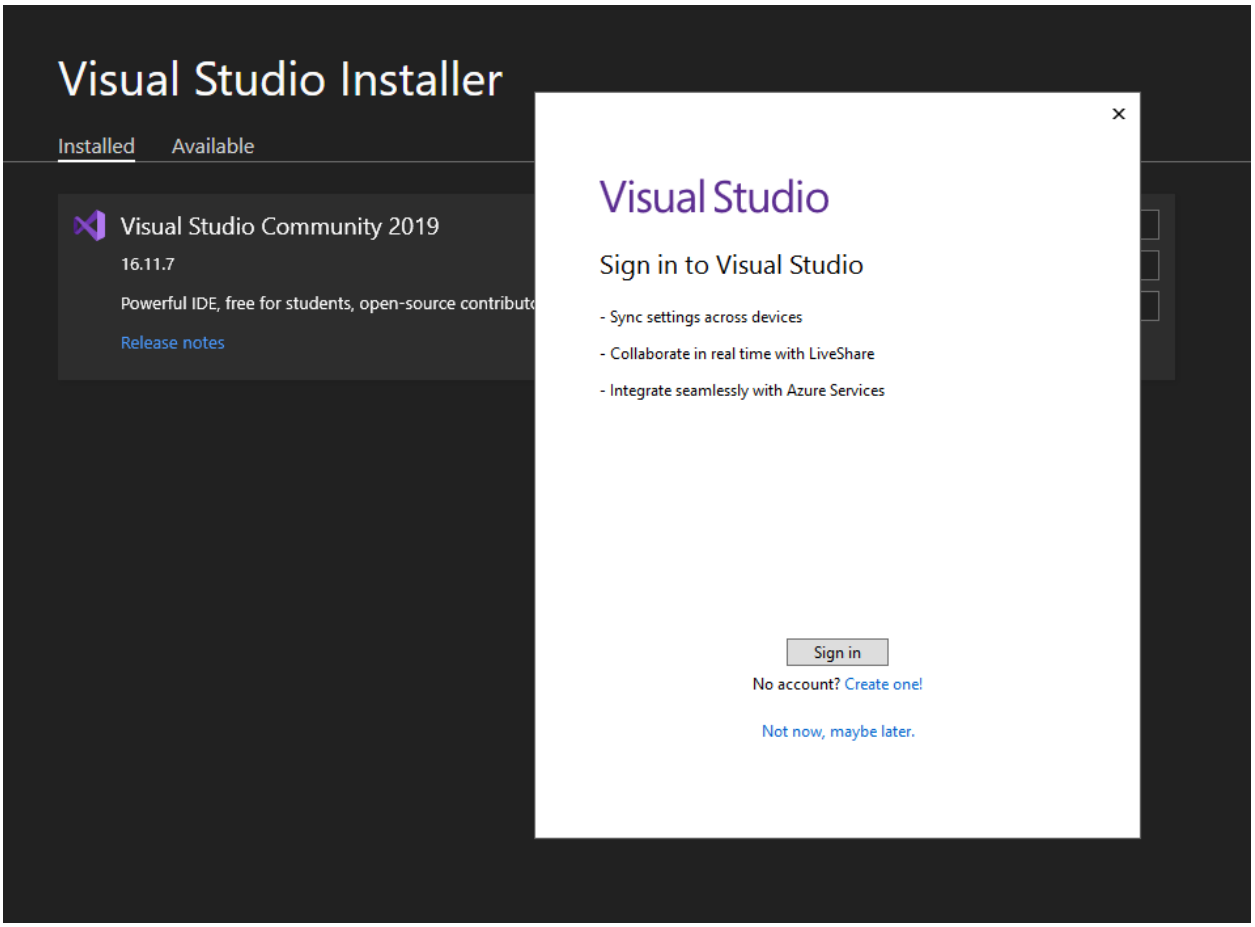
We start installing.



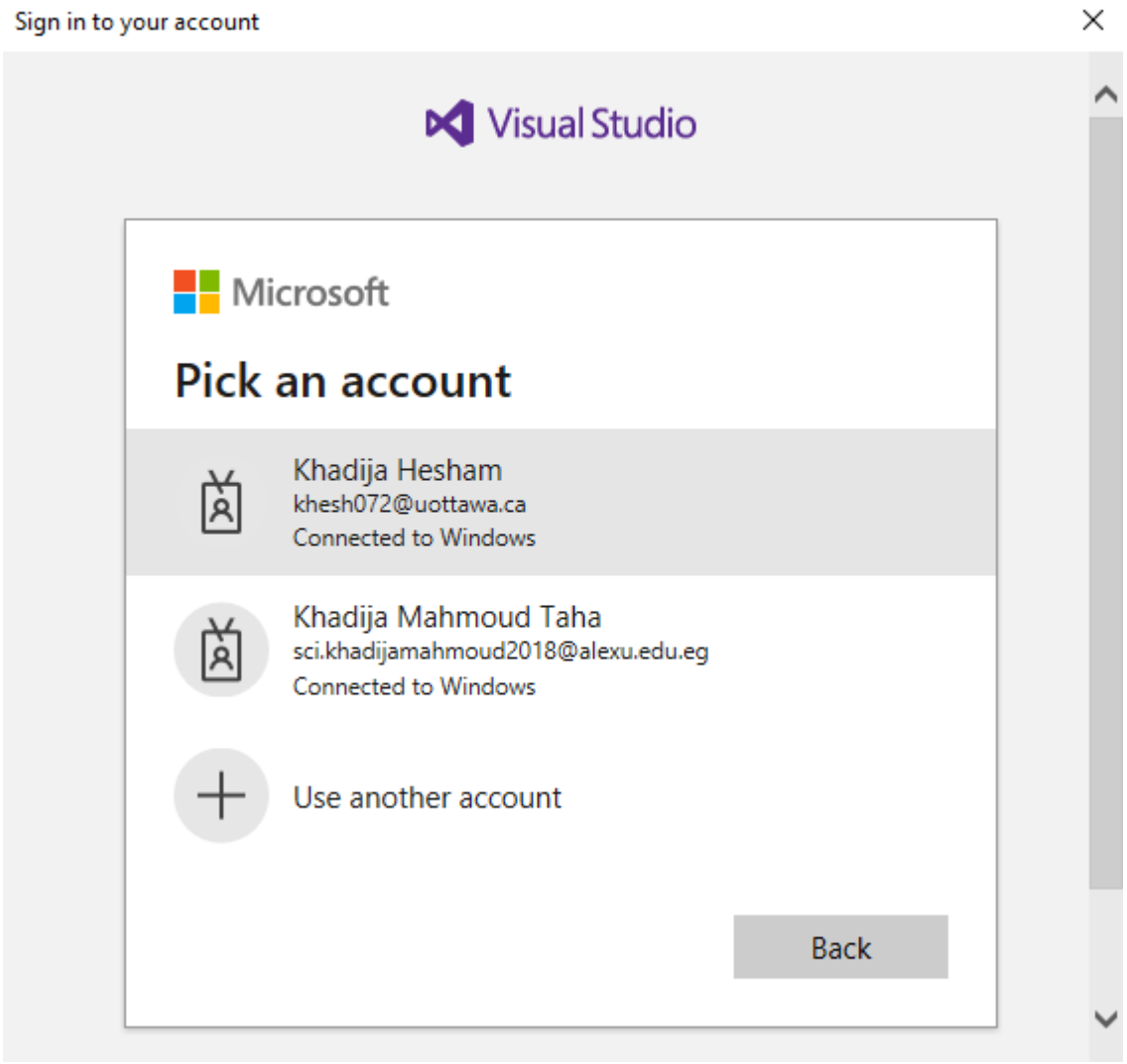
The installation is started.



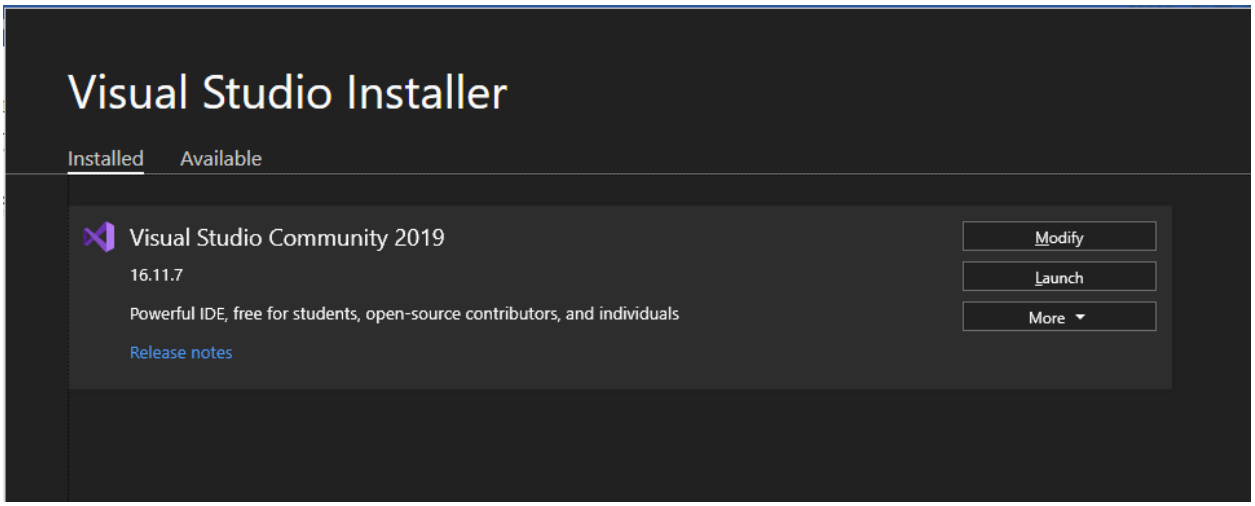
Installation is complete, and now we are signing in.



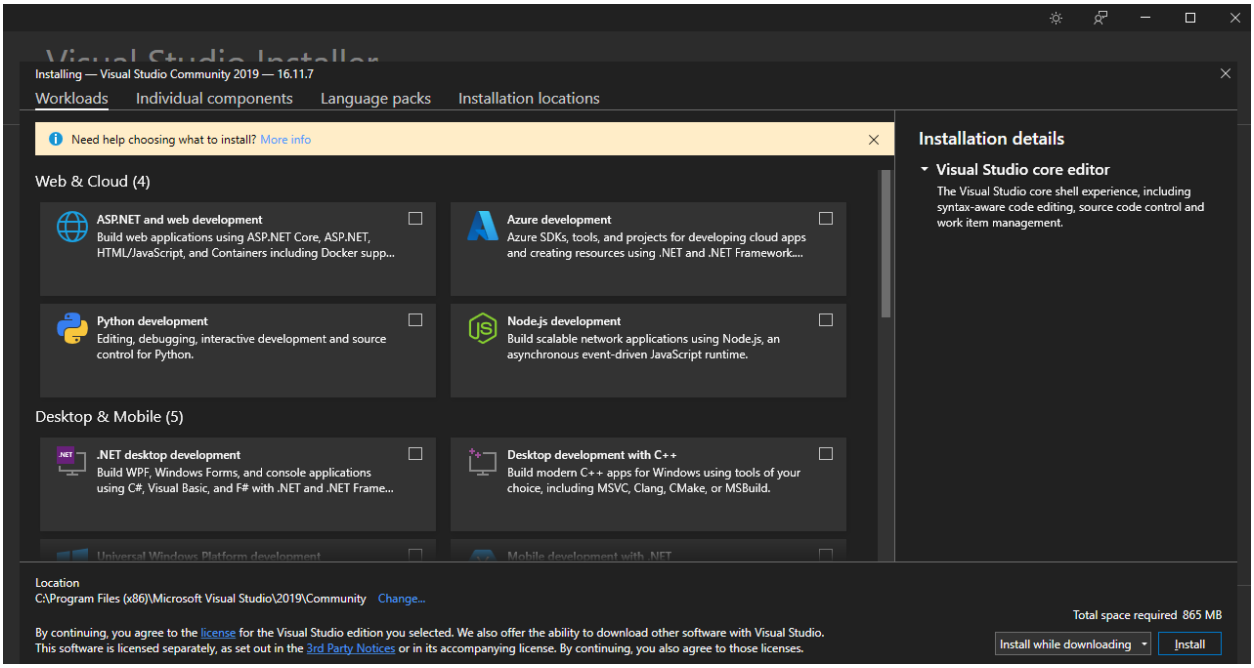
We choose the uOttawa account.

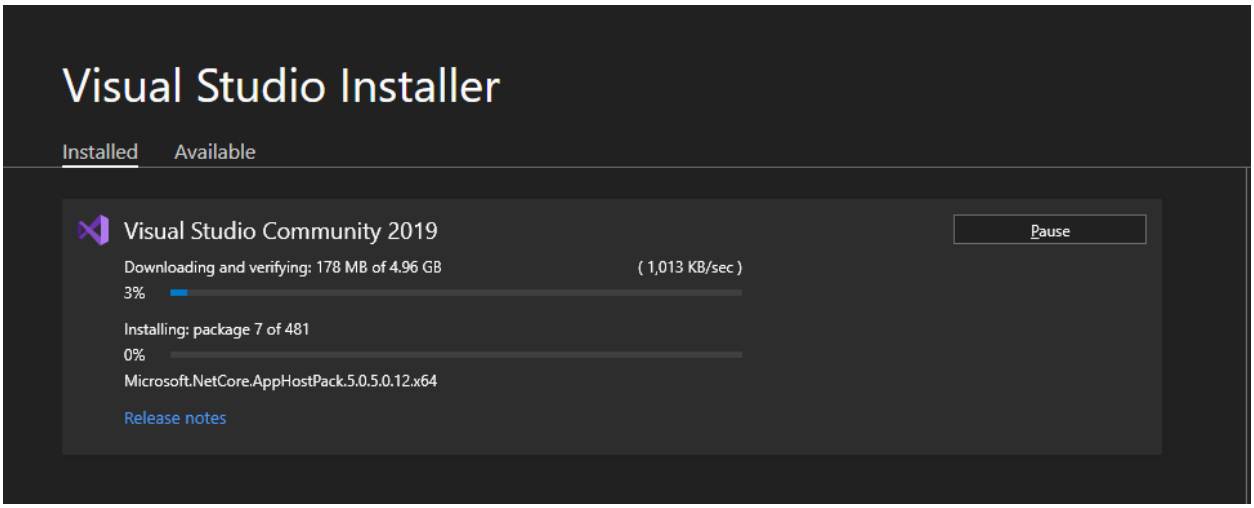


We open up again the installer application to download tools for assignments.

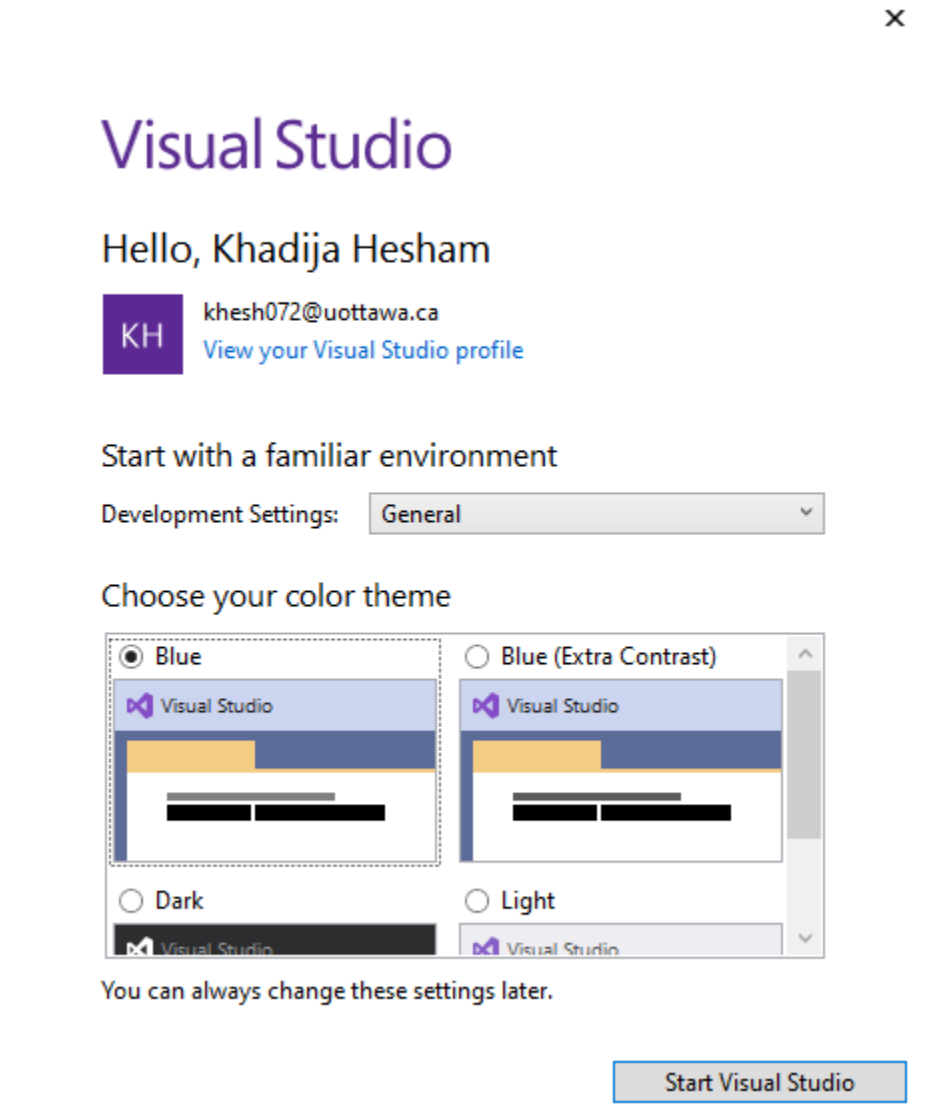


Then, we click on modify the program to install important workloads, we select some tools that you will use later such as Universal Windows Platform development, Azure Development, .NET desktop development, and Data storage and processing.

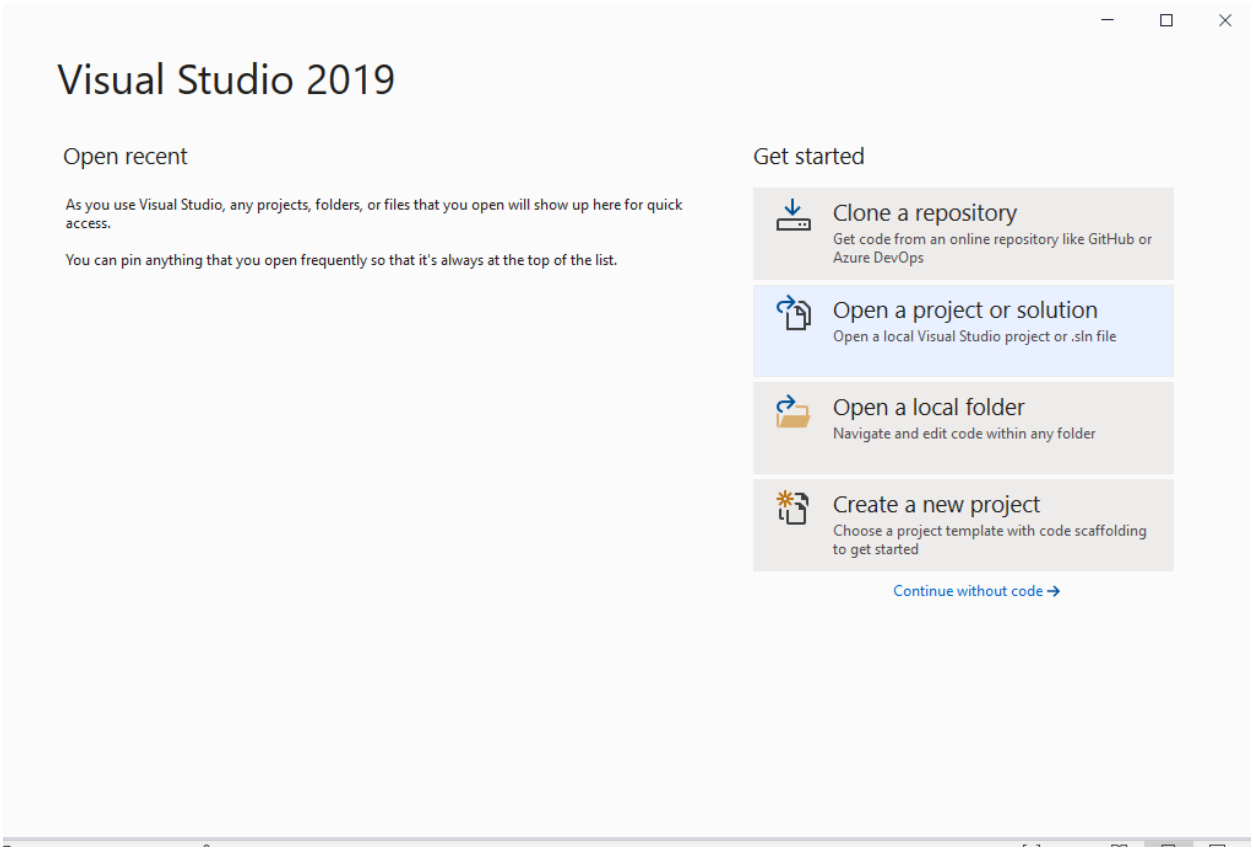




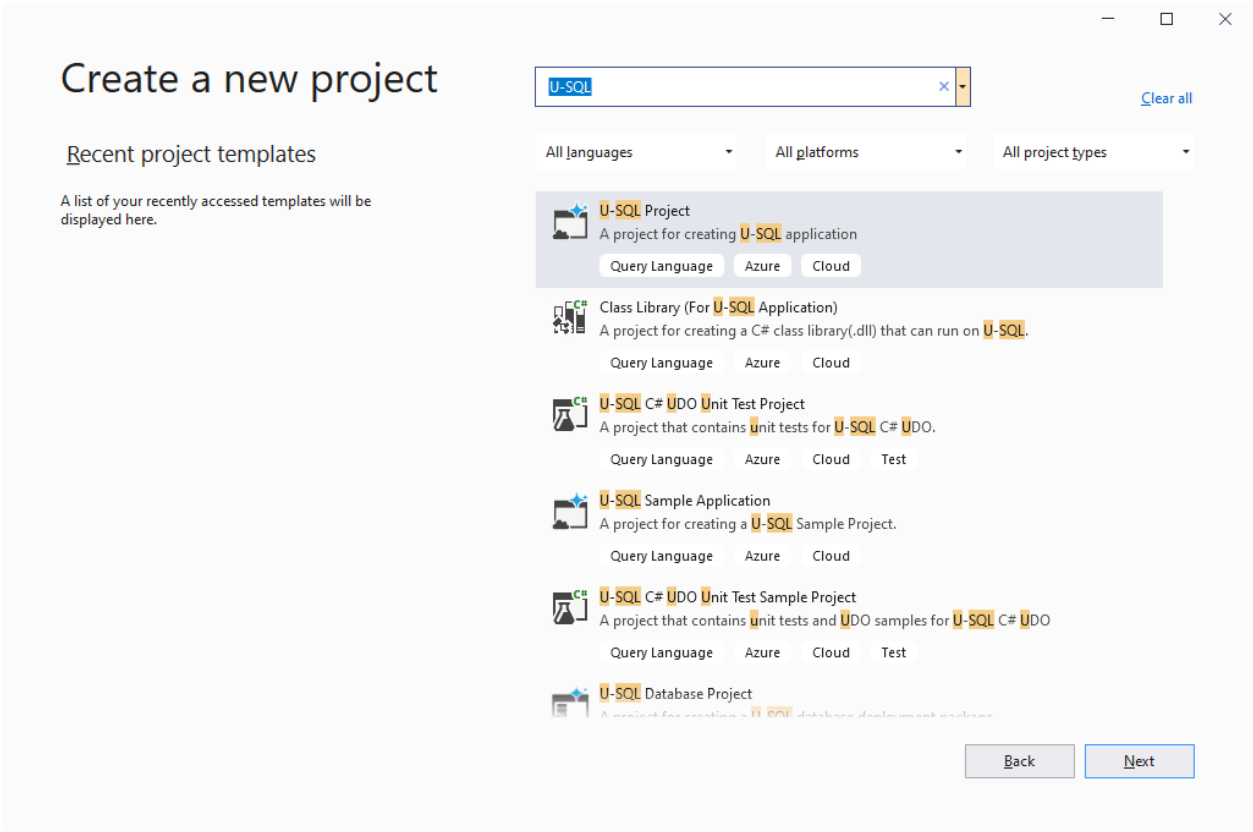
We click to start.



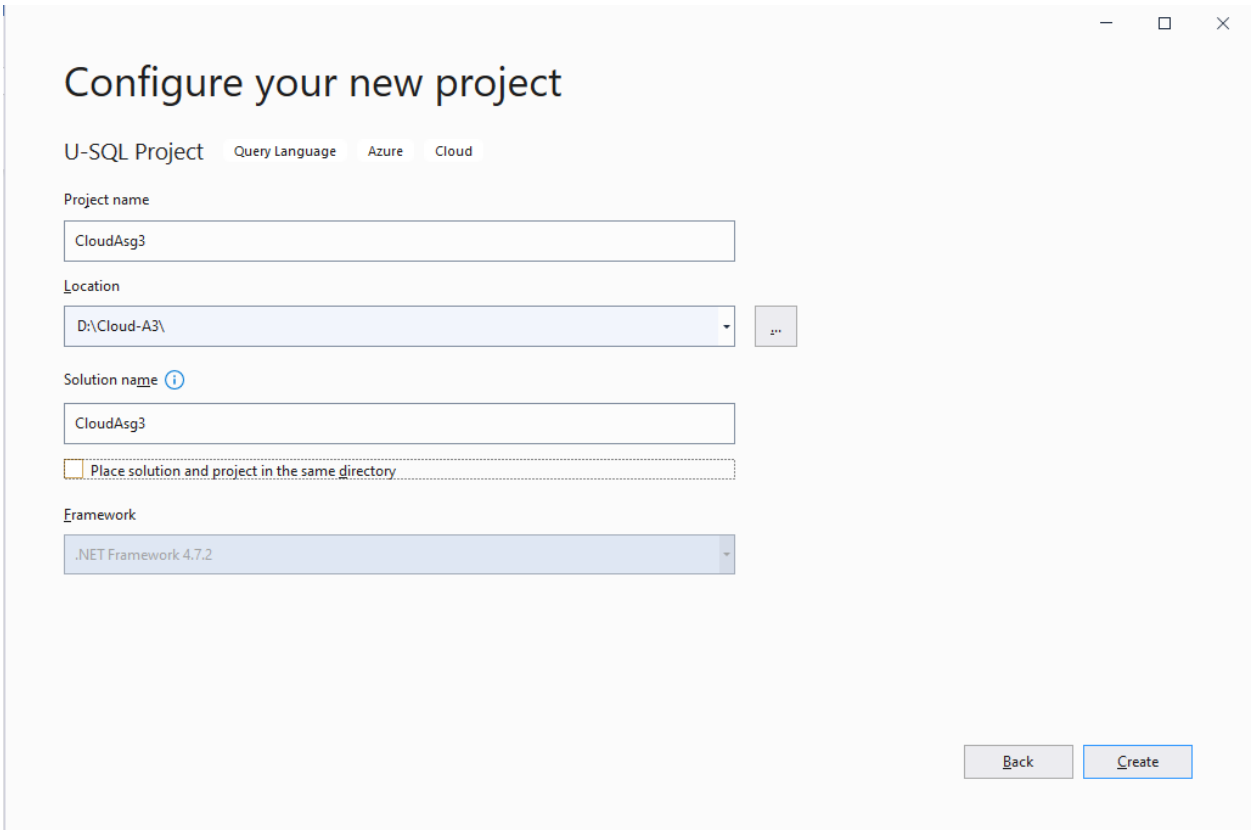
Then, we create a new project to use U-SQL.



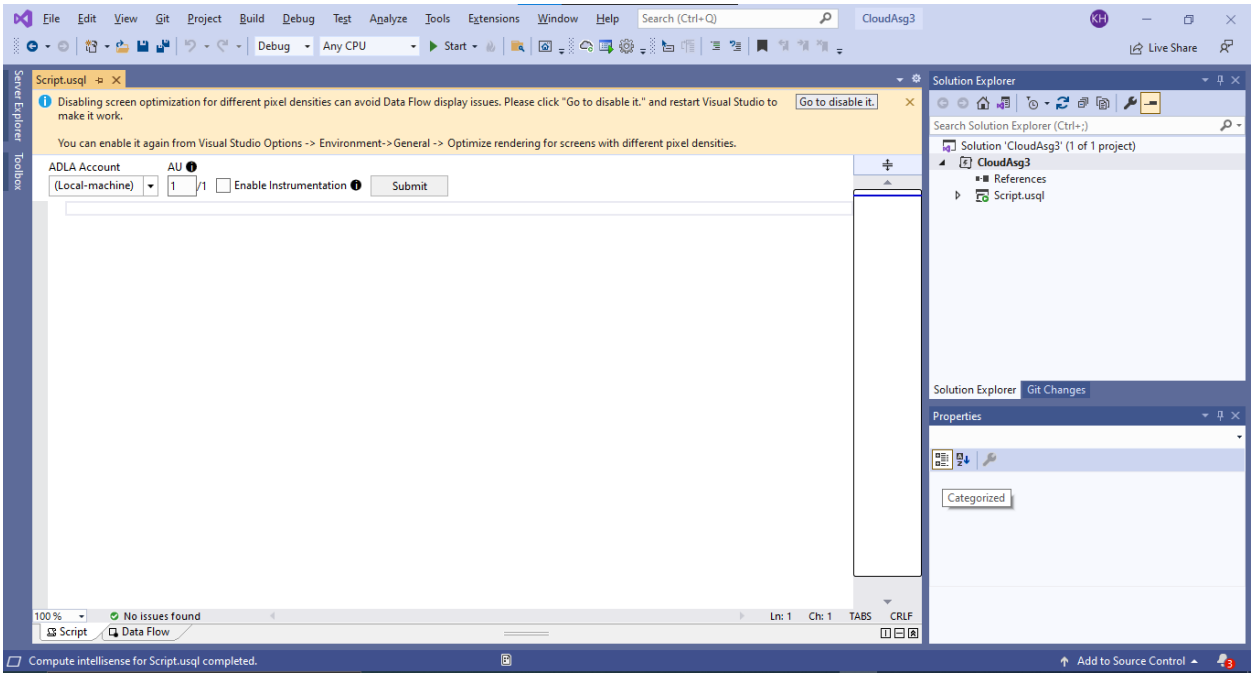
We search for U-SQL and then click next.



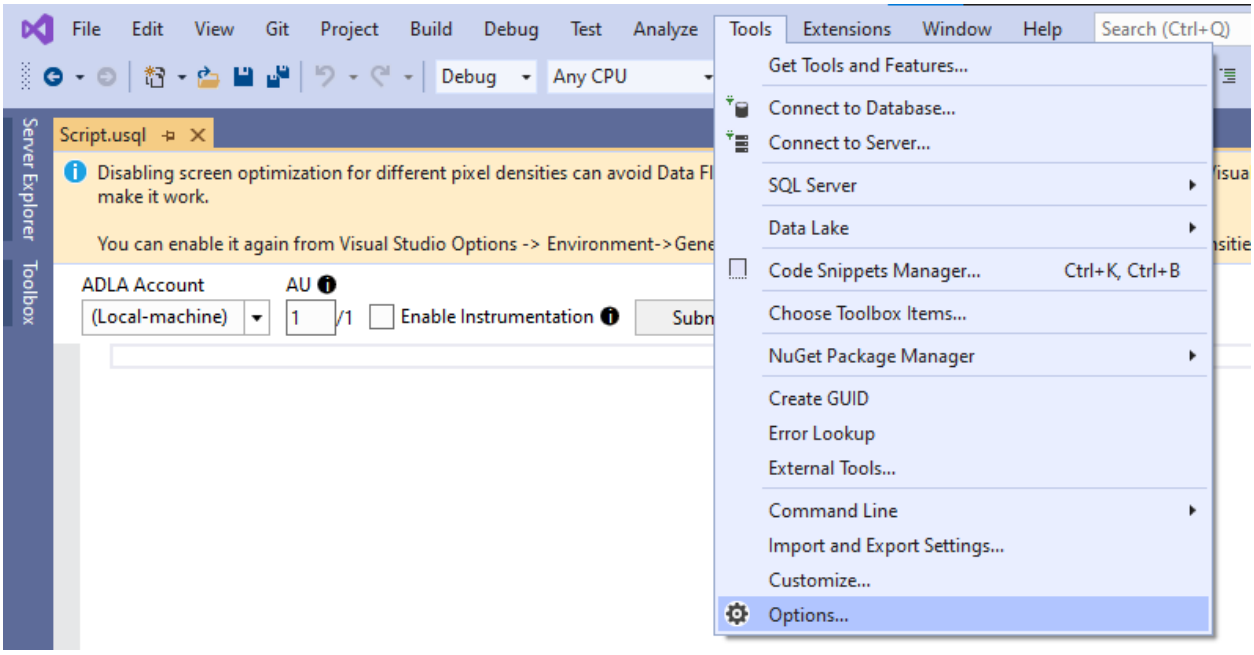
Specify the Visual Studio project name, location and solution, then we click create.



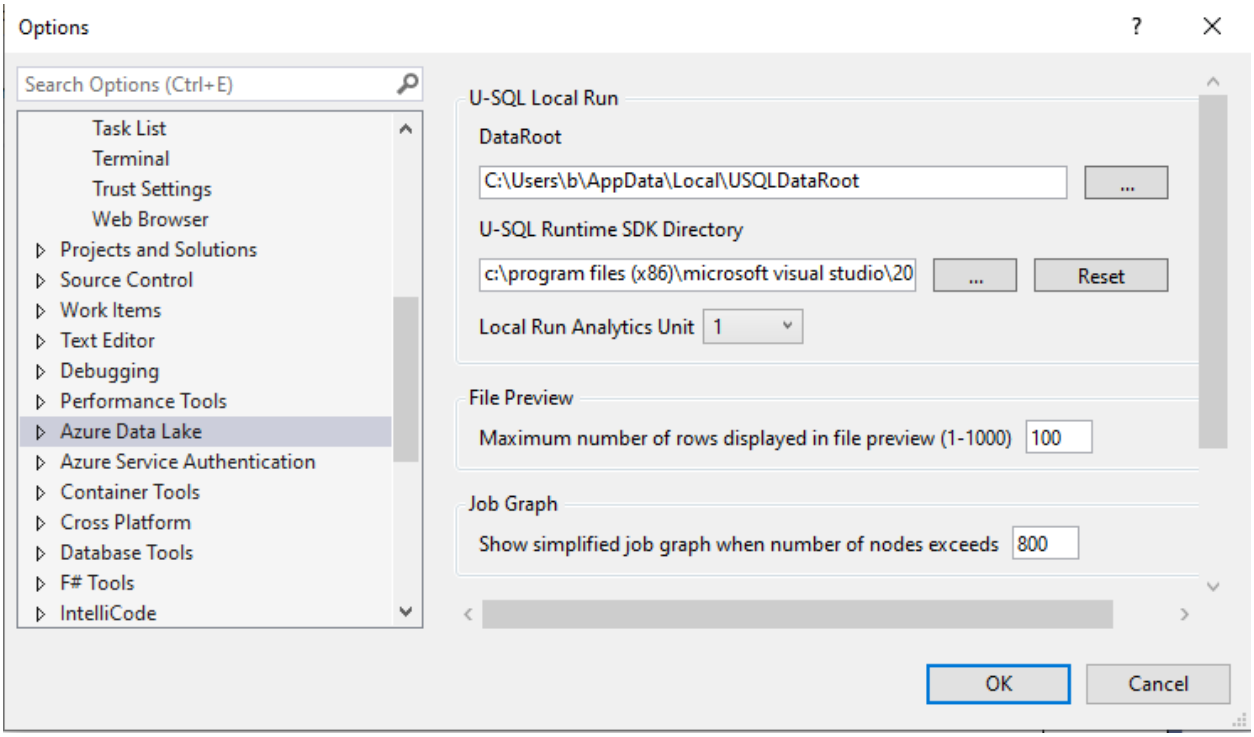
We wait until the project is created, after that we can see this screen.



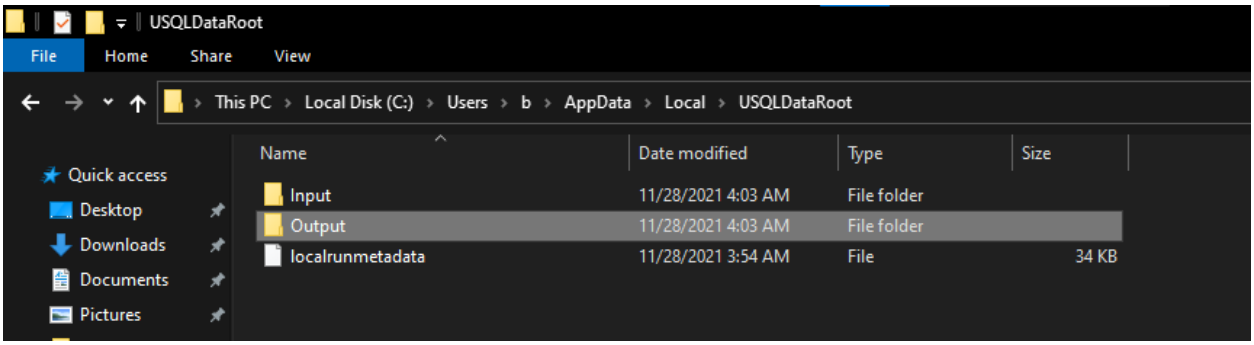
Before we write the script, we need to set the input and output folders path in Visual Studio.



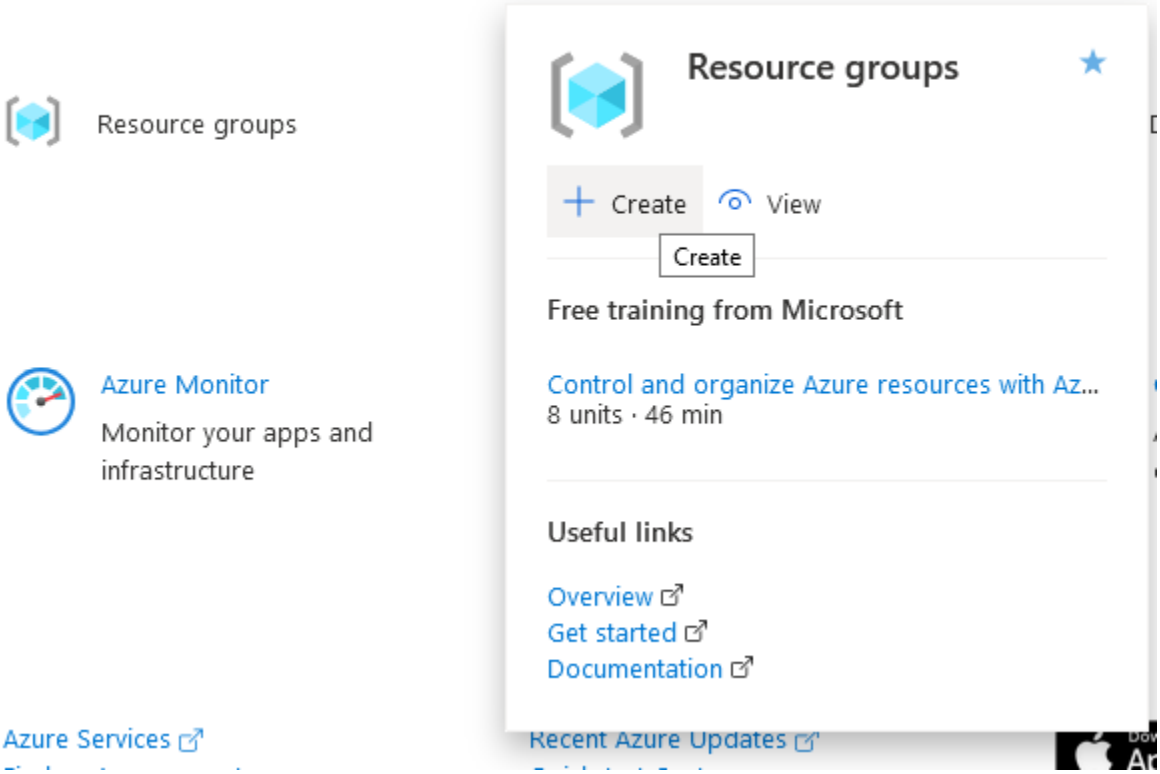
Now, browse to the data root directory and create two folders for input files and output files.

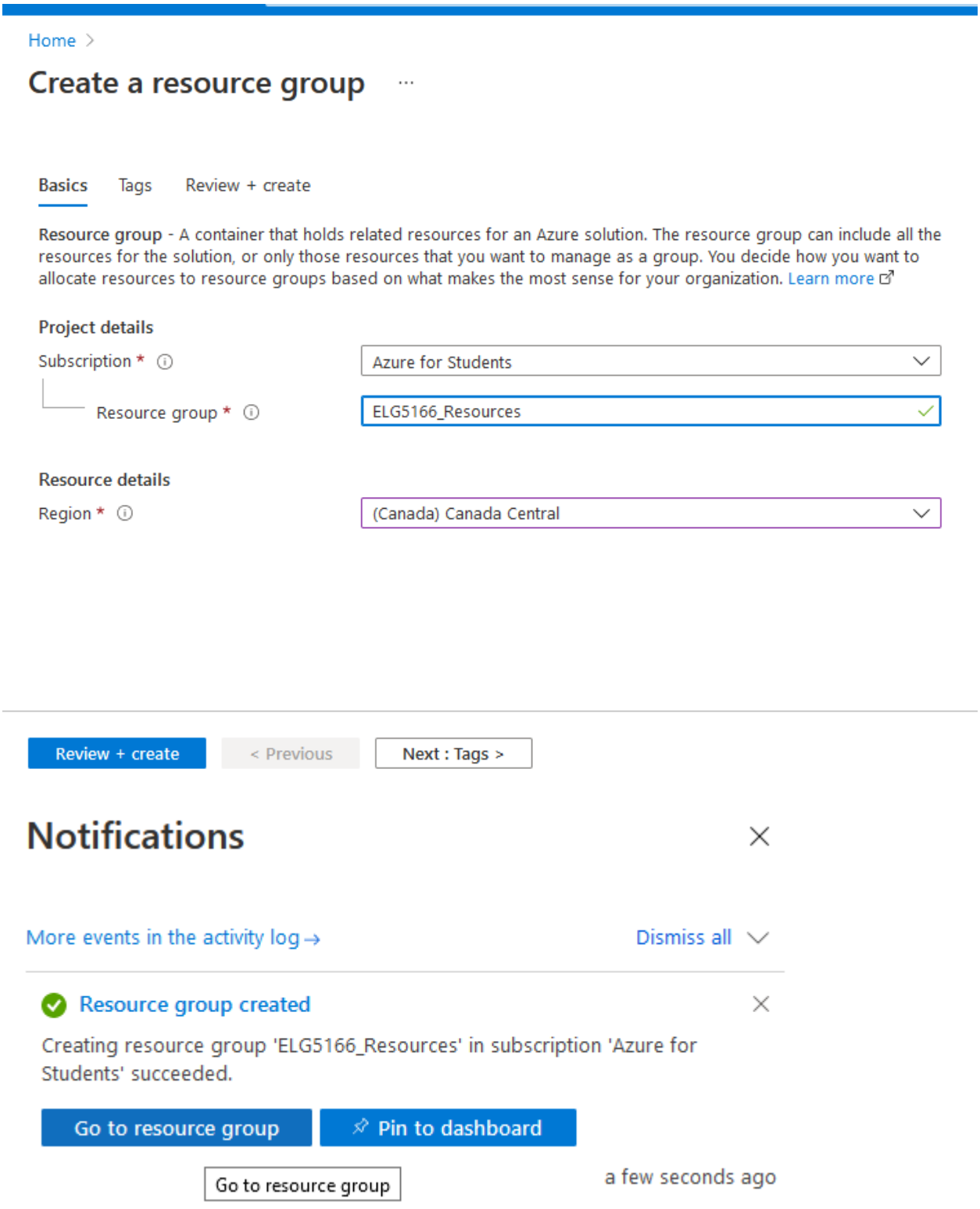


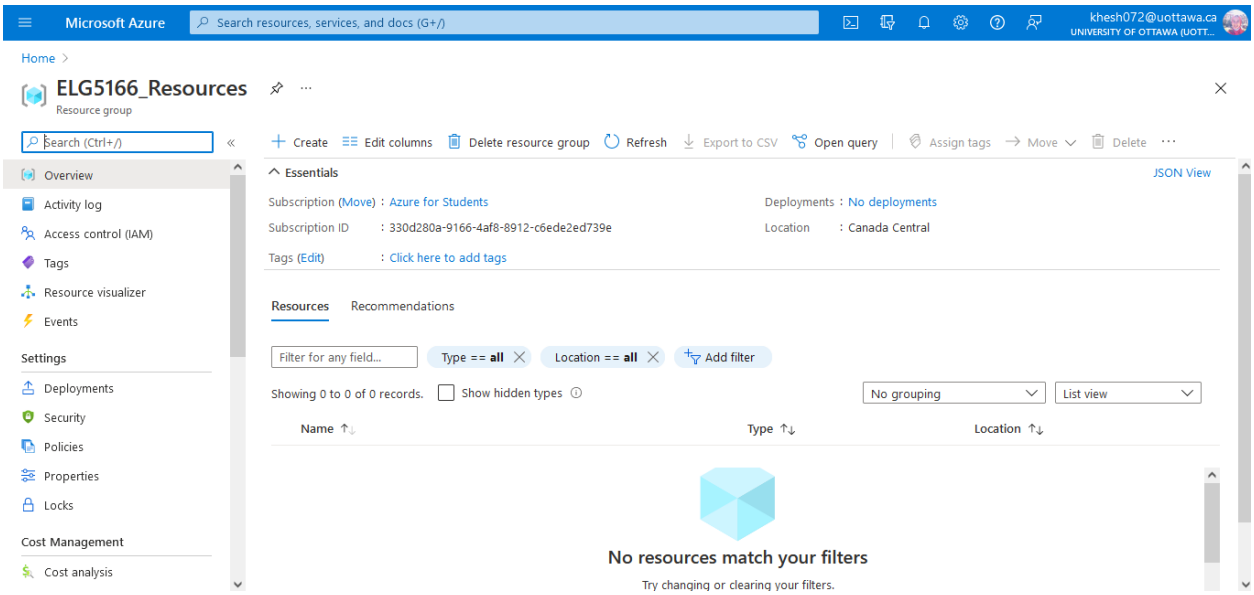
Now, browse to the data root directory and create two folders for input files and output files.



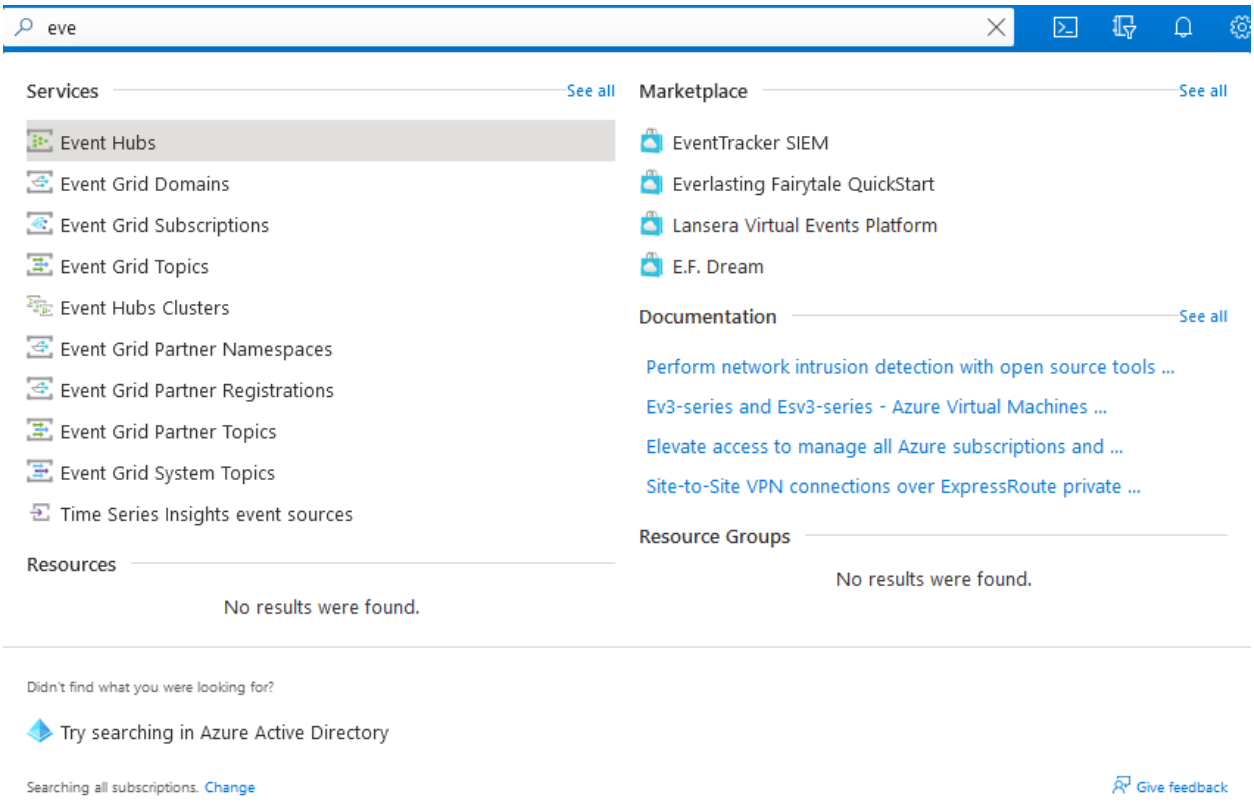
Create resource group



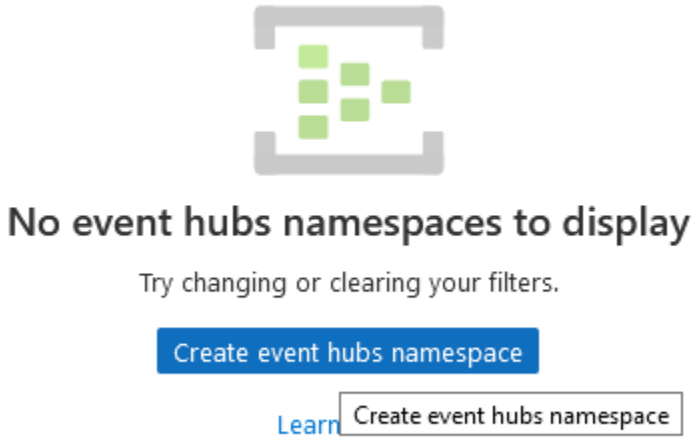

Home
Create a resource group
Basics
Tags
Review + create
Resource group - A container that holds related resources for an Azure solution. The resource group can include all the resources for the solution, or only those resources that you want to manage as a group. You decide how you want to allocate resources to resource groups based on what makes the most sense for your organization. Learn more
Project details
Subscription *
Azure for Students
Resource group *
ELG5166_Resources
Resource details
Region *
(Canada) Canada Central
Review + create
< Previous
Next : Tags >
Notifications
More events in the activity log →
Dismiss all
Resource group created
Creating resource group 'ELG5166_Resources' in subscription 'Azure for Students' succeeded.
Go to resource group
Pin to dashboard
Go to resource group
a few seconds ago



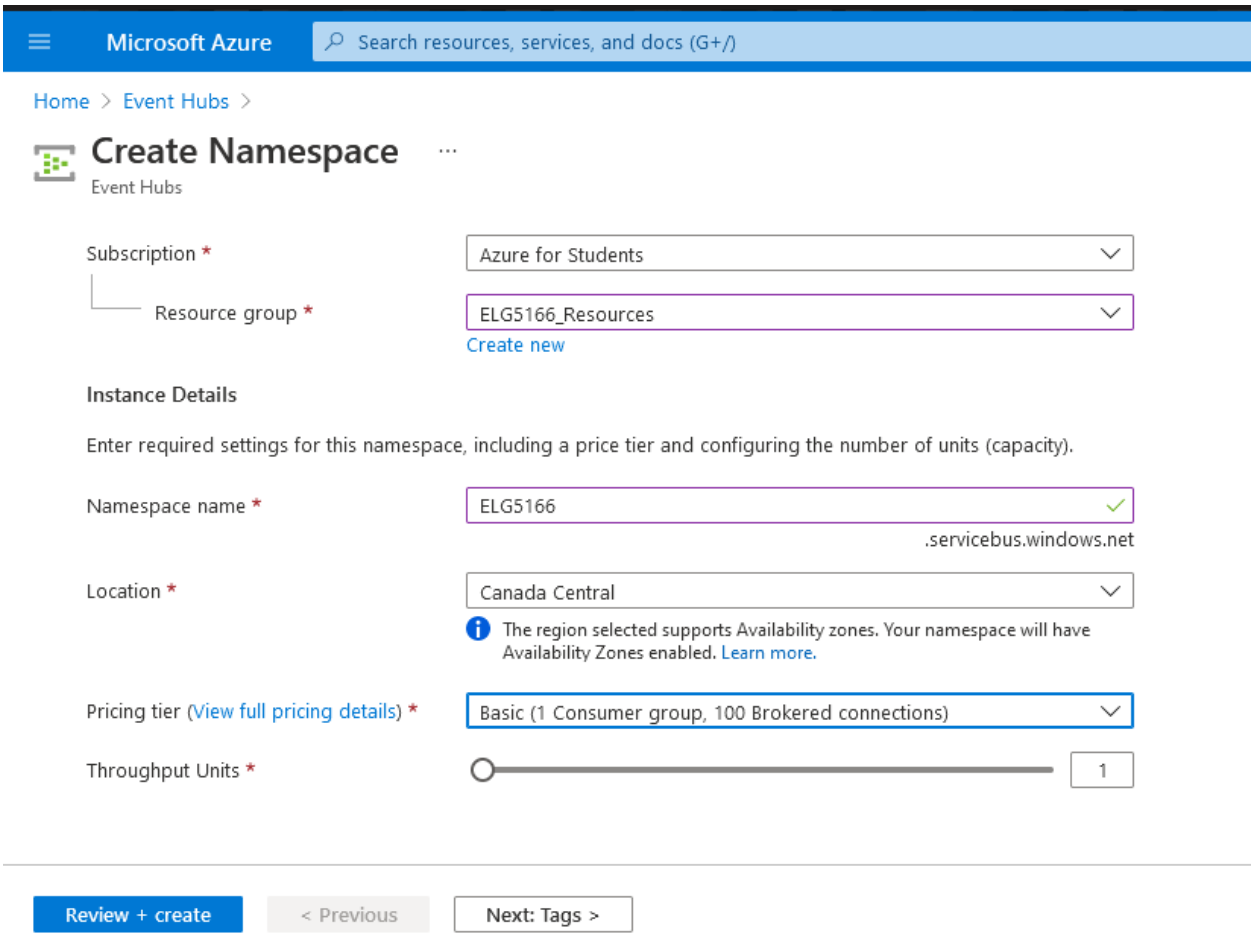
2 Creating the Event Hubs



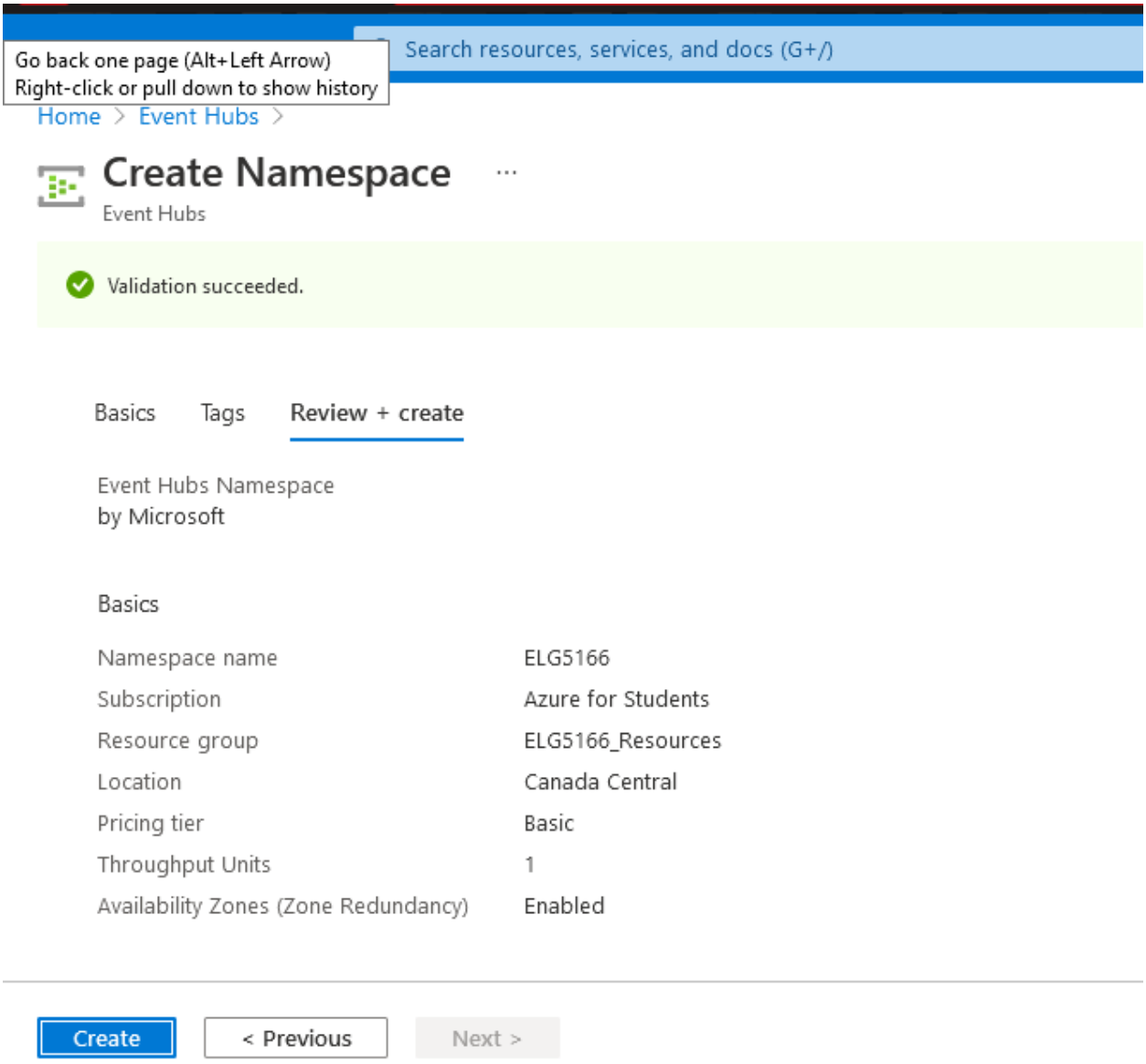
Click the “Create event hubs namespace” to create a namespace for your event hubs



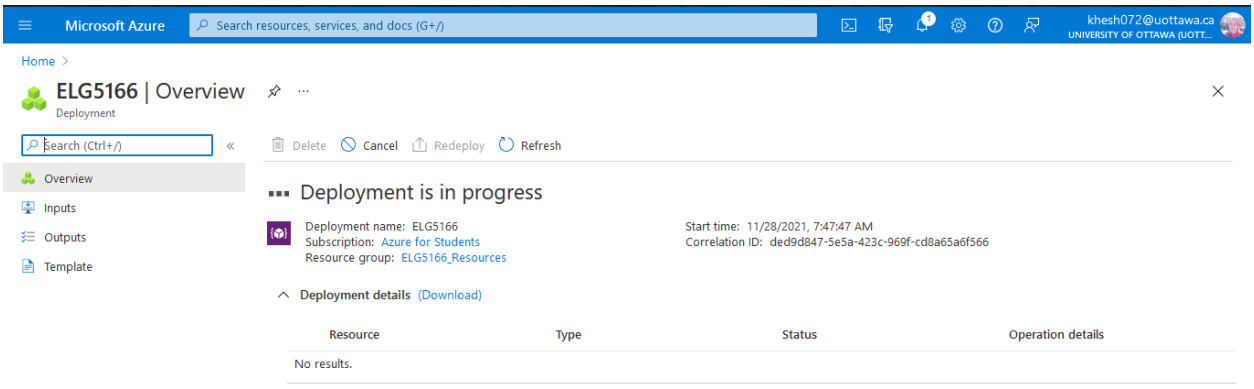
• Provide a name for the namespace, for example, the namespace is ELG5166.

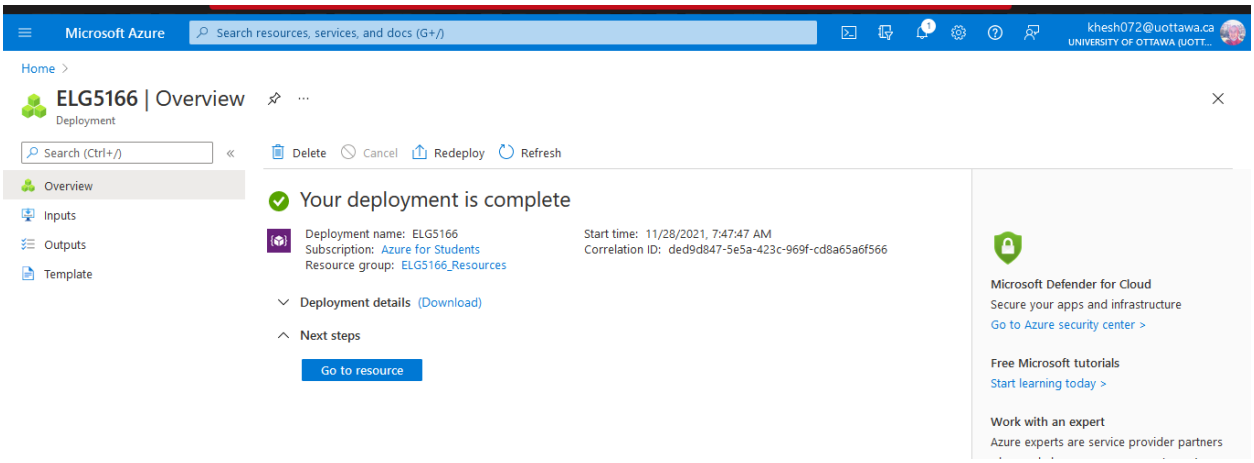


After validation, we click on create.



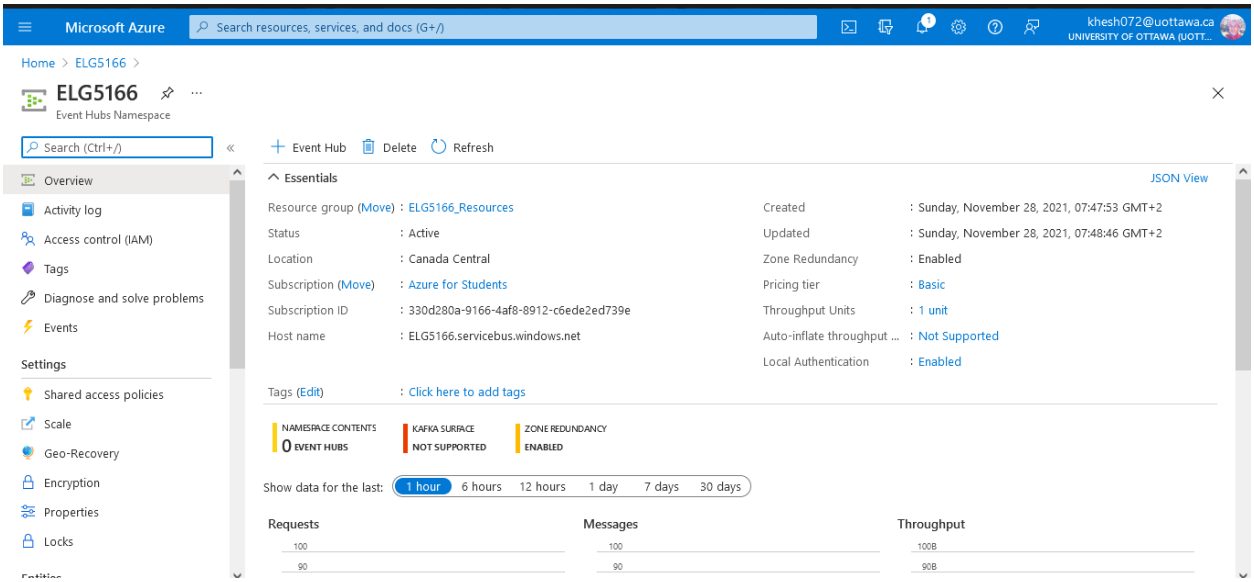
Deployment is in progress.



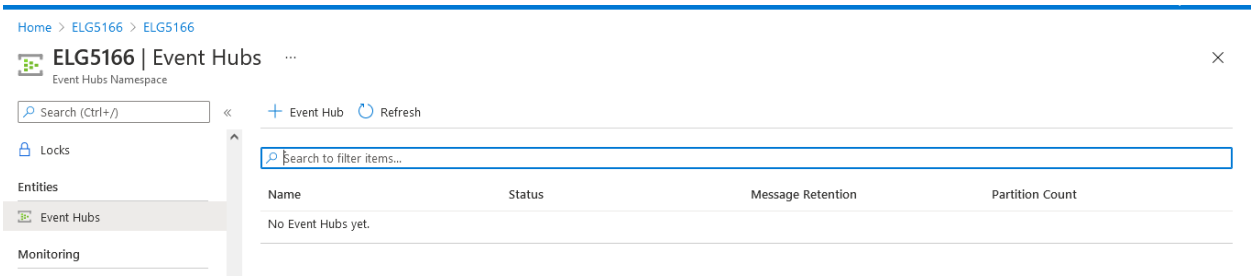


we have set up the Namespace for the Event Hubs, we can proceed to create the Event Hubs.

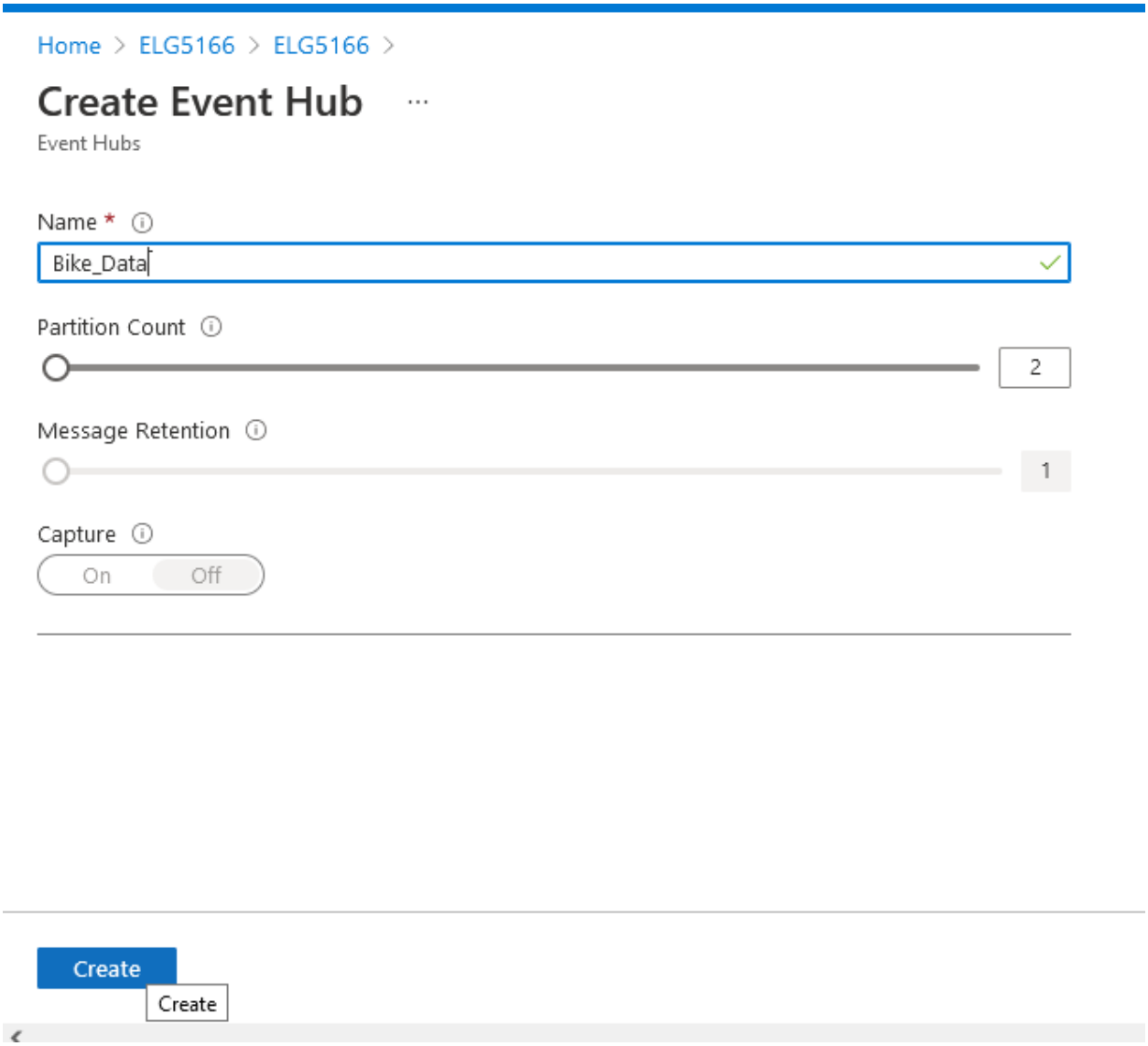
We click on go to resource.



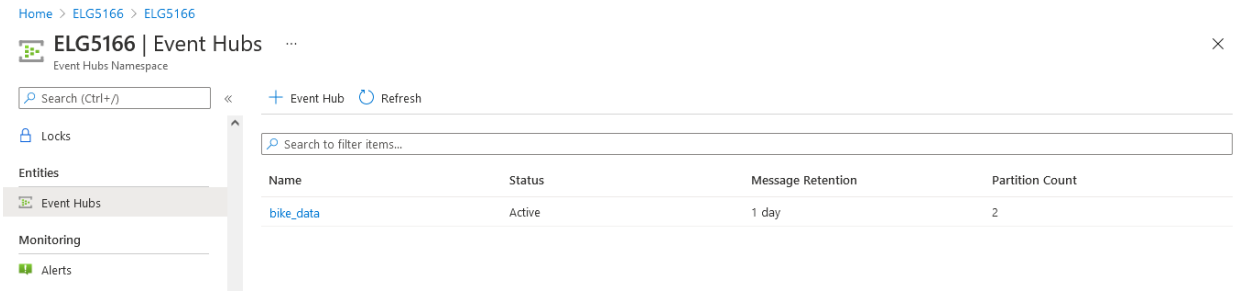
Choose event hubs then click on create event hub.



We type the name of the Event Hub. In this case, the name is Bike_Data.

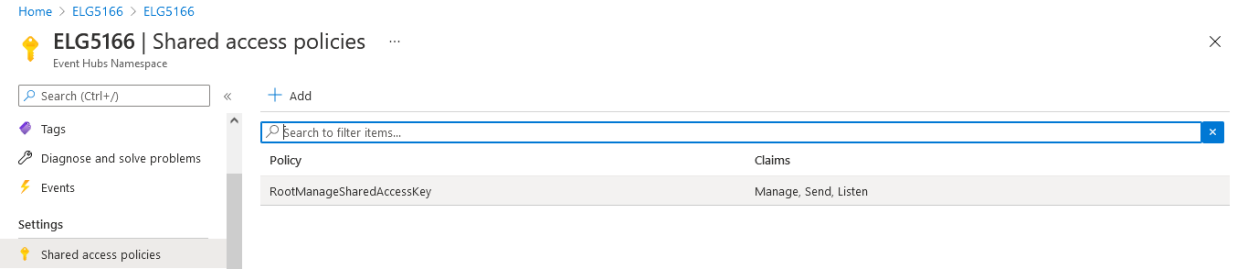


Event hub object is active.

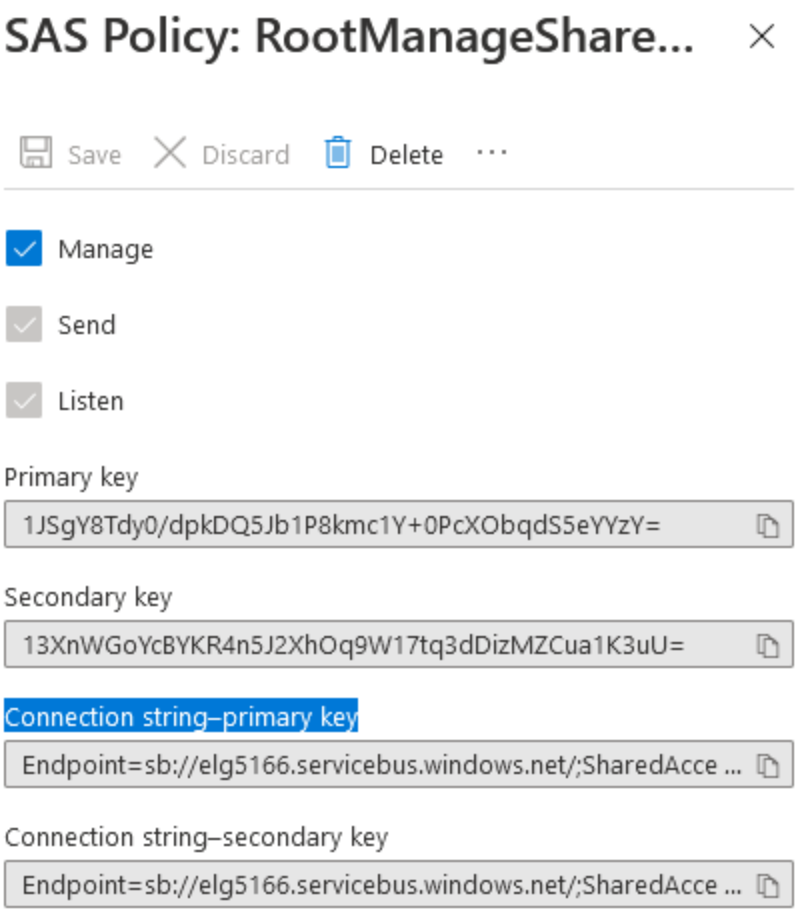


To create a connection string. This is done using the event hub namespace shared access policies. The shared access policies can be found under the Settings category of the left menu.

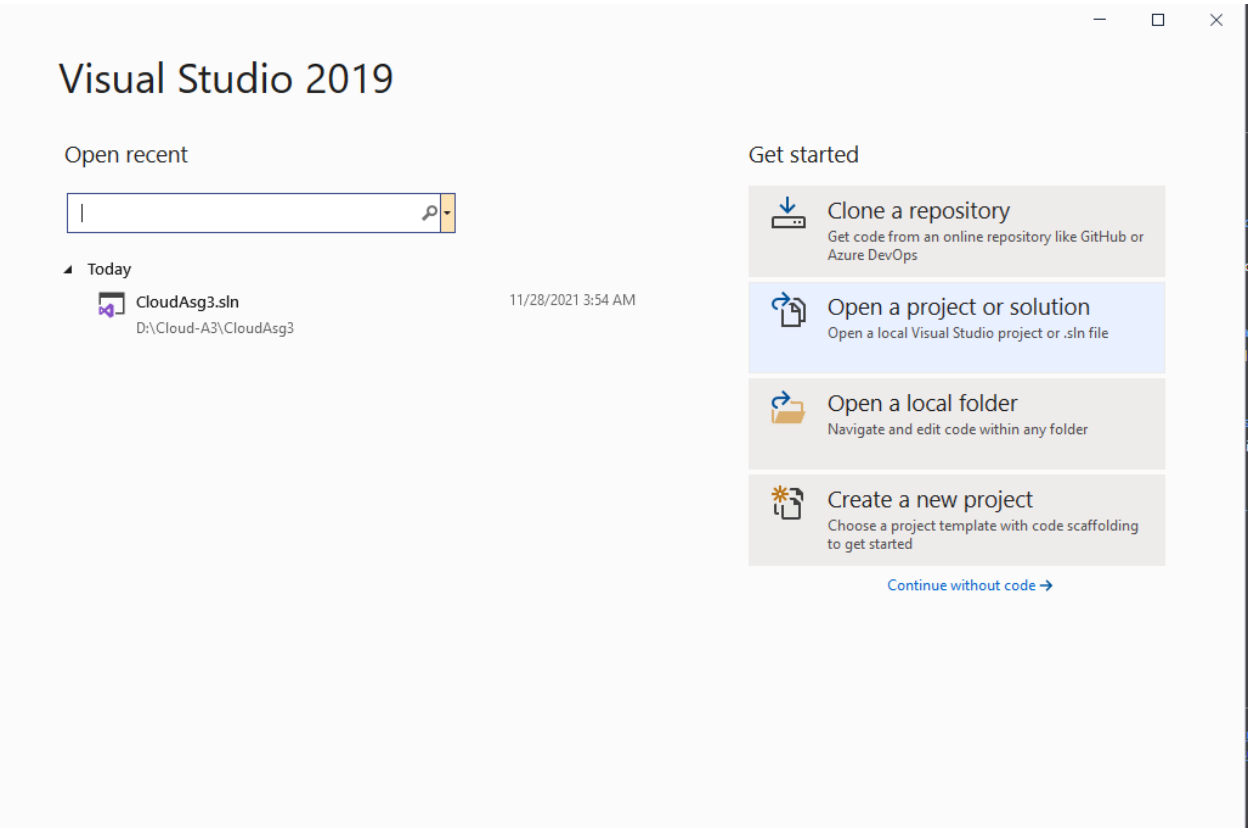
Click on the RootManageSharedAccessKey



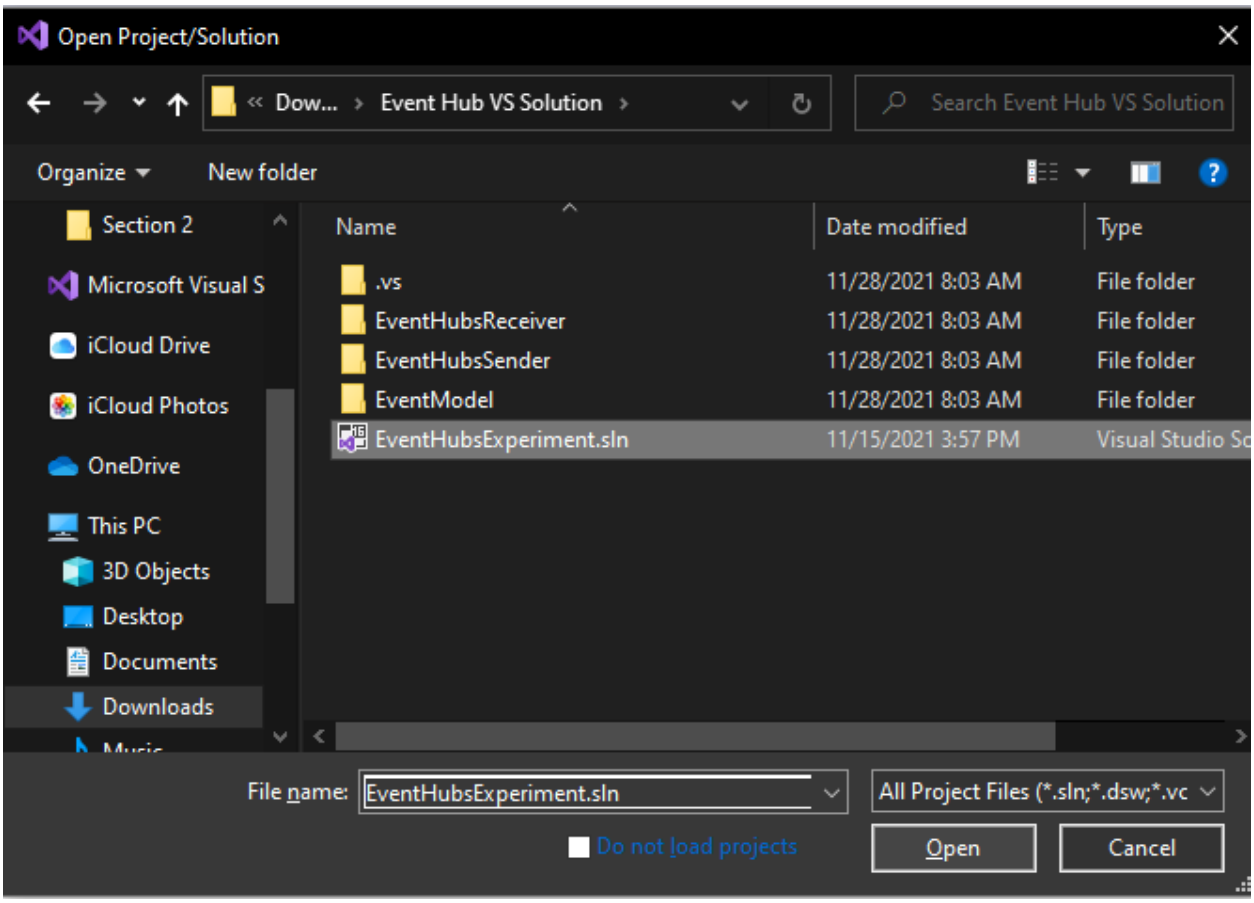
We will use the Connection String-Primary Key for creating the Event Hubs request generator.



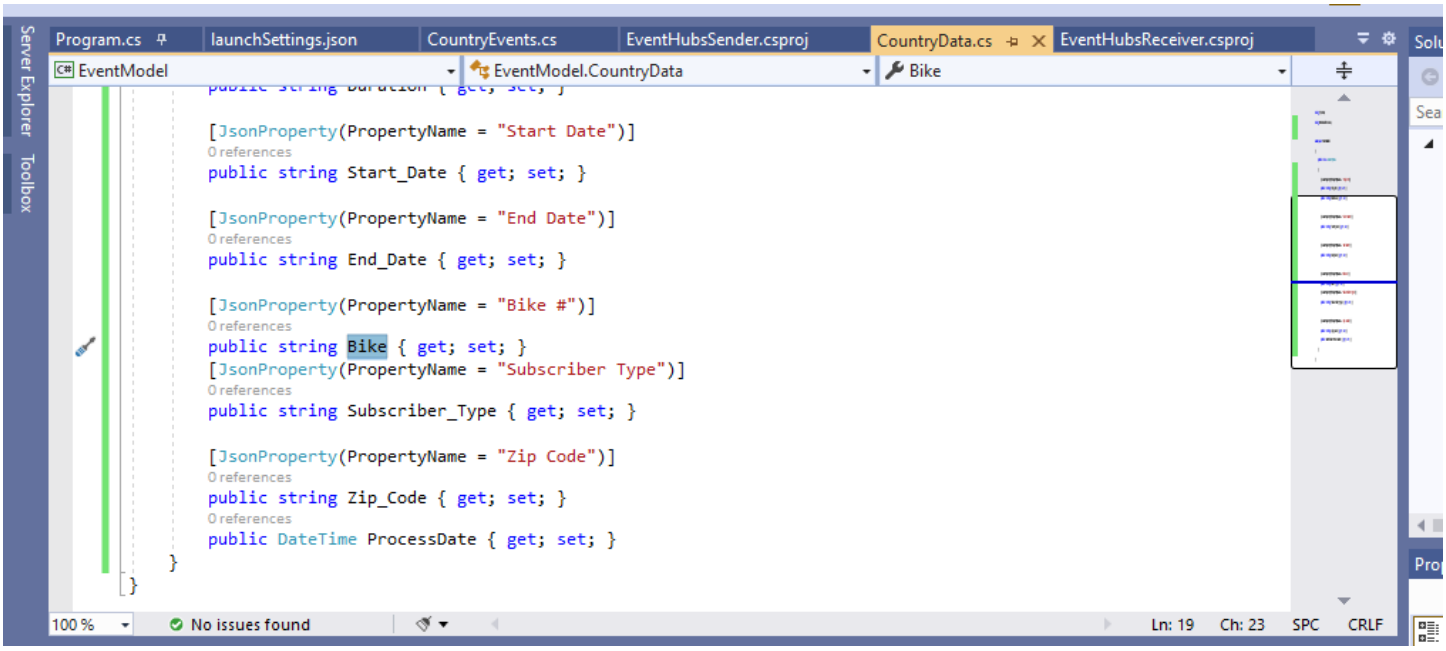
Open Visual Studio and choose to open a solution.



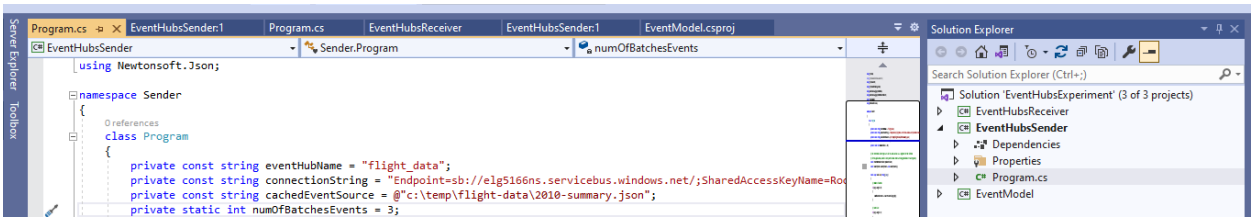
We choose the lab solution.

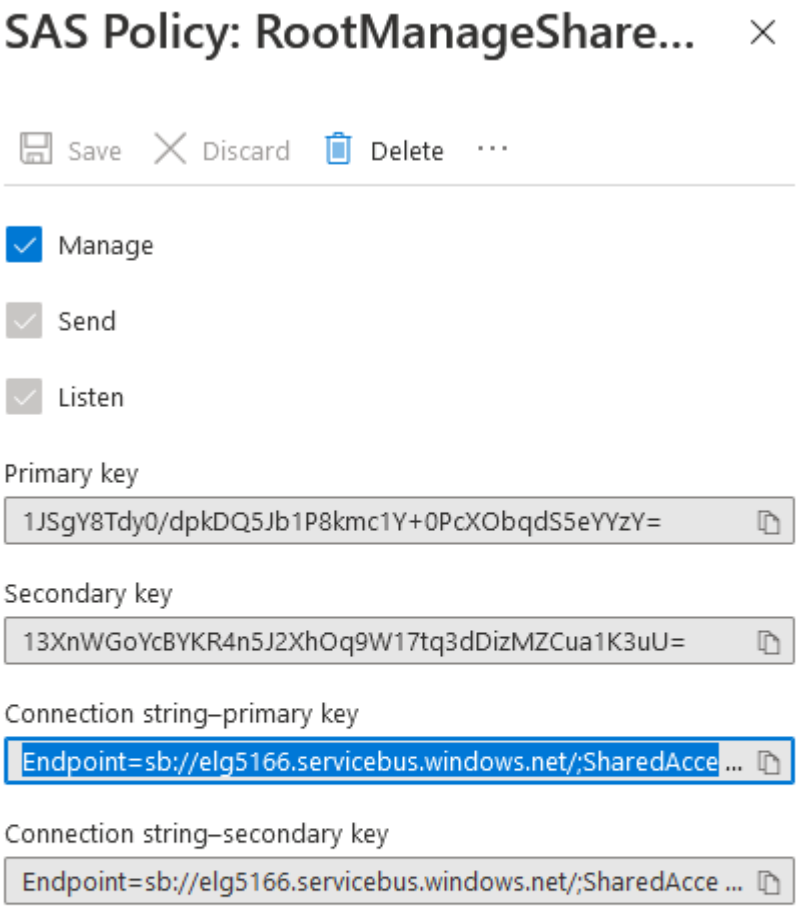


First of all, we edit the data schema with our bike data schema.



Go to event hub sender project and choose the program.cs file.

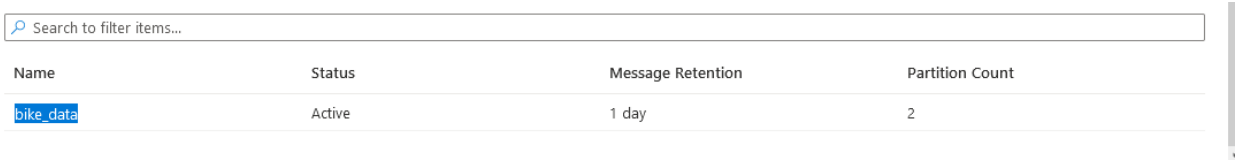




Copy the connection string primary key ad paste it into the code.

```
connectionString = "Endpoint=sb://elg5166.servicebus.windows.net/;SharedAccessKeyName=Roo
```

Edit the event hub name.

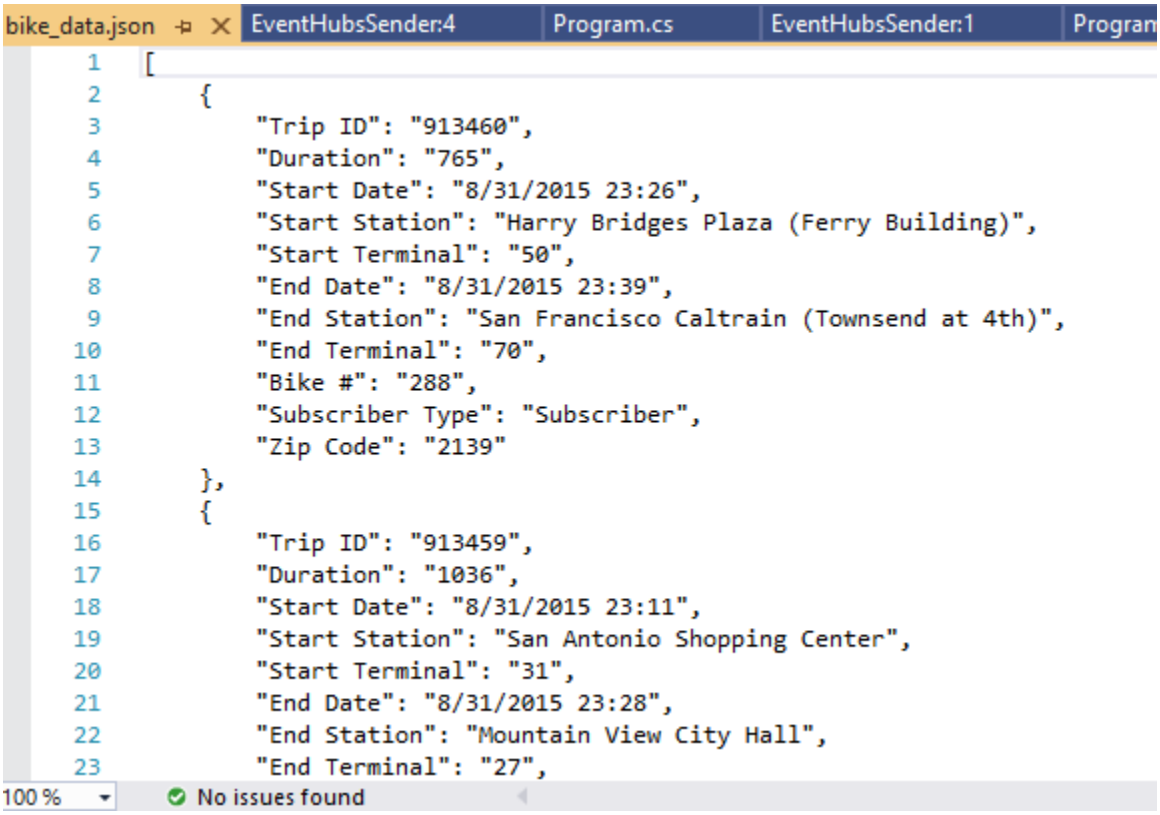



```
private const string eventHubName = "bike_data";
```

Edit the path of the data by writing our data path.

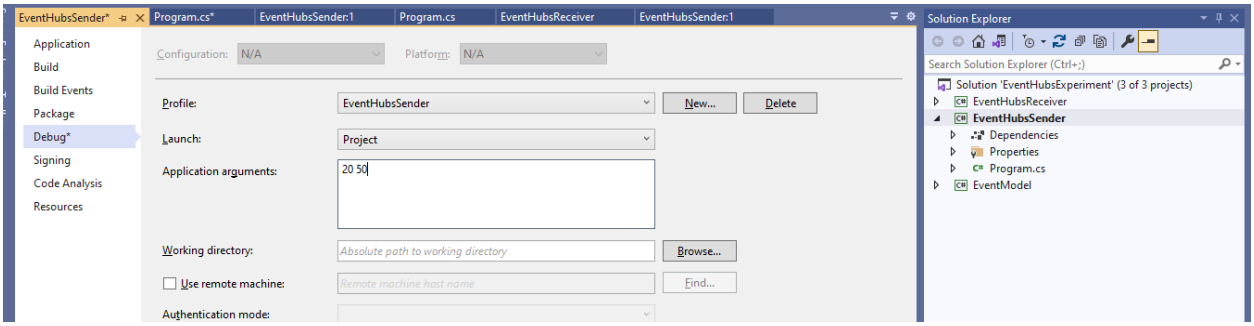
```
private const string cachedEventSource = @"D:\Cloud-A3\bike_data";
```

We can see our data.

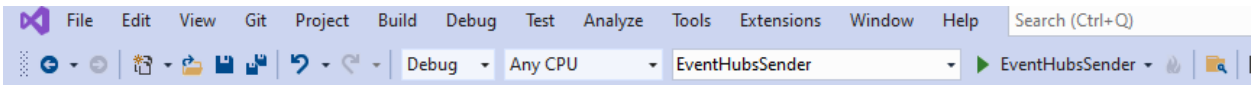


Broadcast bike rental events in batches of 20 trip entries and publish at least 50 events at 2-second intervals per batch.

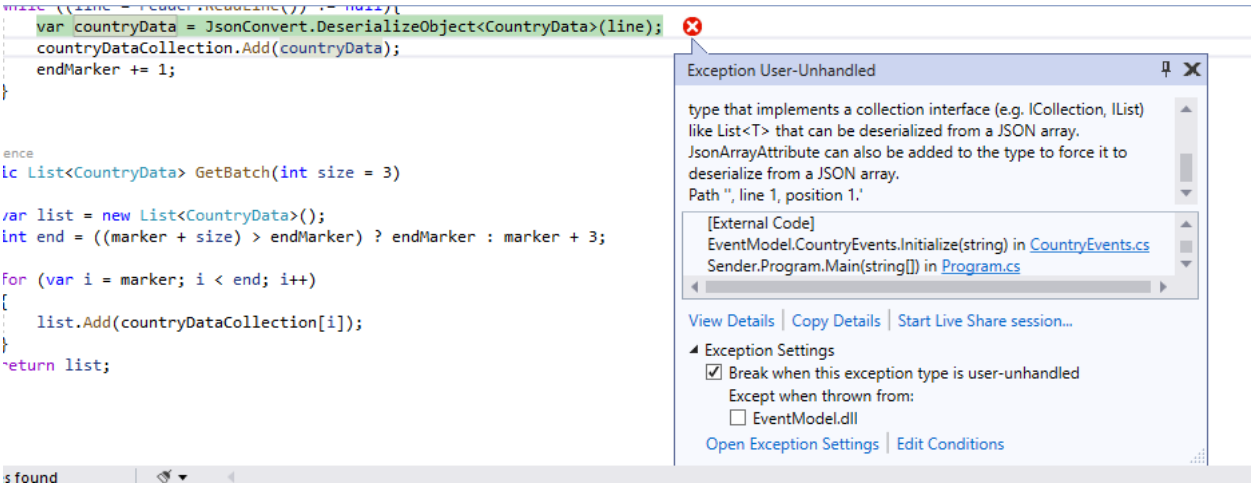
We highlight the event hub sender and right click on the properties, then we write a number of batches of 20 and let the number of events be 50.



Then we run the application



We got an error



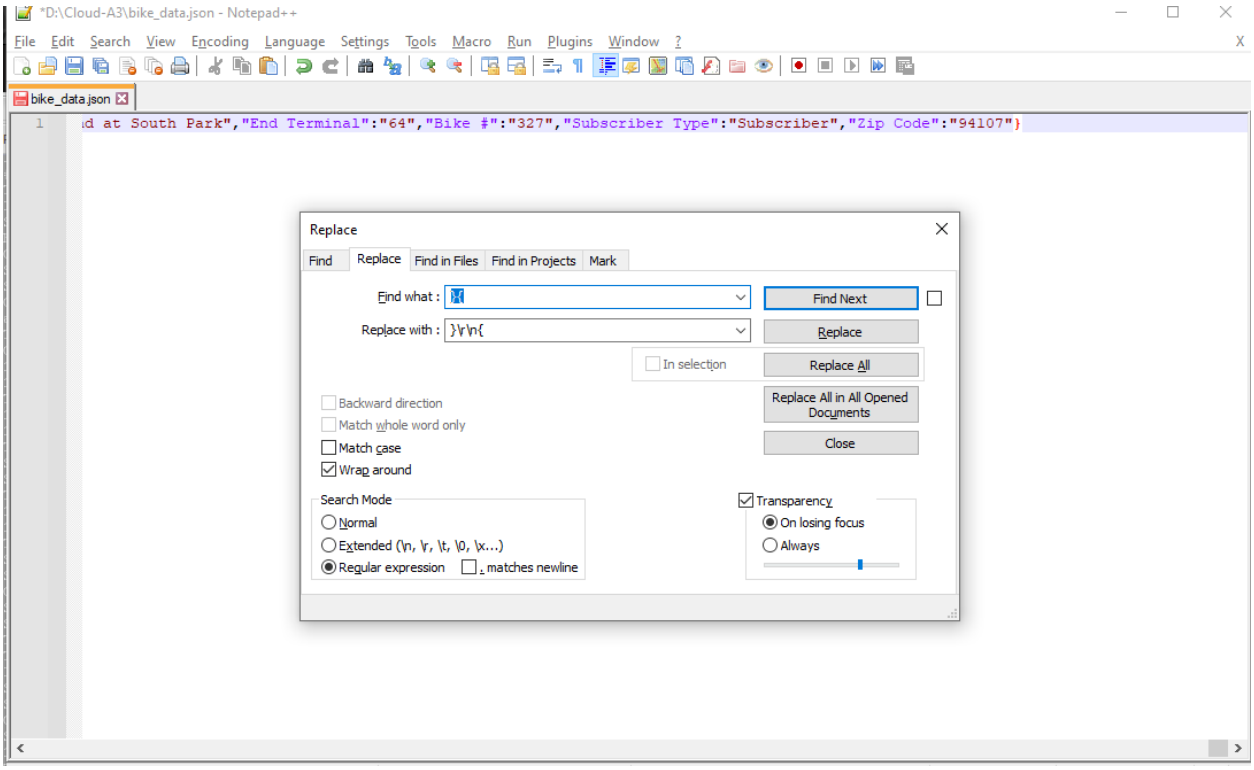
The data needs to be formatted in this way, to remove the “,” from each observation , remove the brackets “[“ in the beginning and “]” at last, remove empty json objects, and to reformat the data with removing the white space by the help of this website: http://jsonviewer.stack.hu/.

Data before formatting:

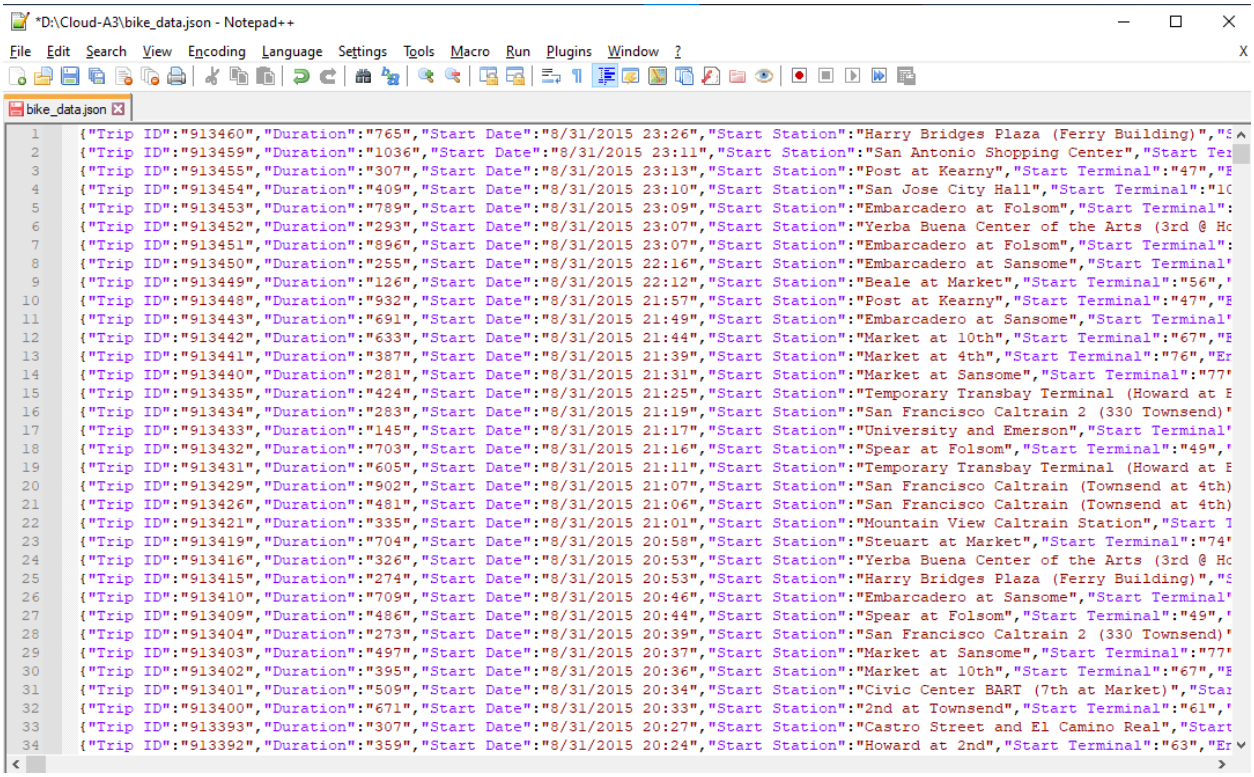

```
[
    {
        "Trip ID": "913460",
        "Duration": "765",
        "Start Date": "8/31/2015 23:26",
        "Start Station": "Harry Bridges Plaza (Ferry Building)",
        "Start Terminal": "50",
        "End Date": "8/31/2015 23:39",
        "End Station": "San Francisco Caltrain (Townsend at 4th)",
        "End Terminal": "70",
        "Bike #": "288",
        "Subscriber Type": "Subscriber",
        "Zip Code": "2139"
    },
    {
        "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"
```

```
{
    "Trip ID": "",
    "Duration": "",
    "Start Date": "",
    "Start Station": "",
    "Start Terminal": "",
    "End Date": "",
    "End Station": "",
    "End Terminal": "",
    "Bike #": "",
    "Subscriber Type": "",
    "Zip Code": ""
}
{
    "Trip ID": "",
    "Duration": "",
    "Start Date": ""
```

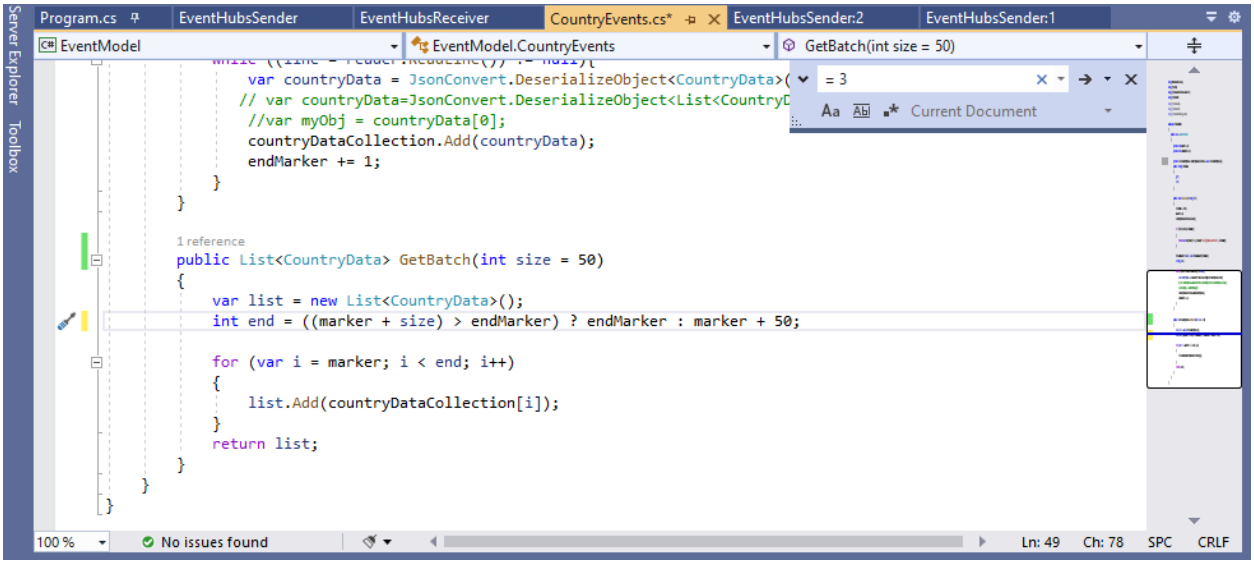

Formating:



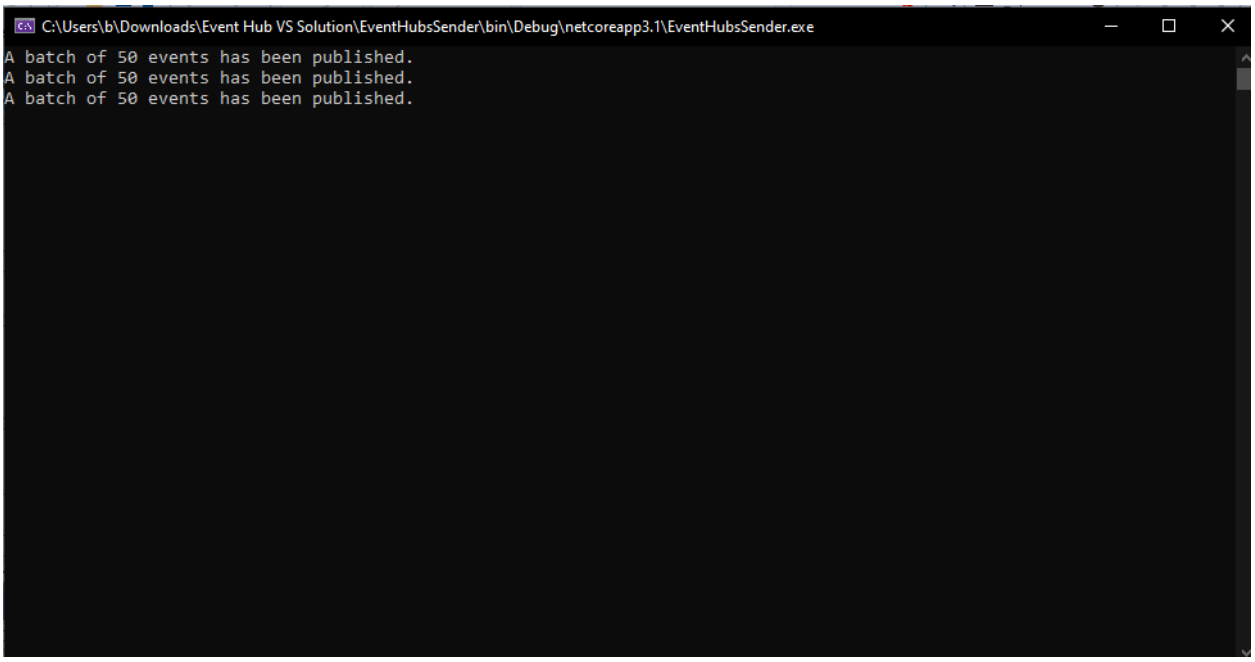
Data after reformatting:



We edit the batch size and set the number of events =50;

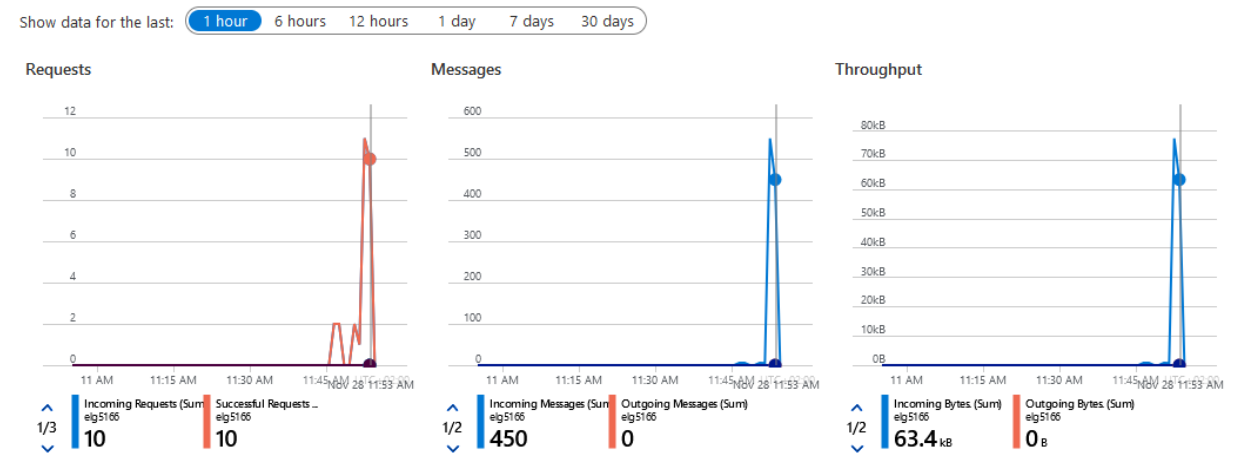


Then we run the code:



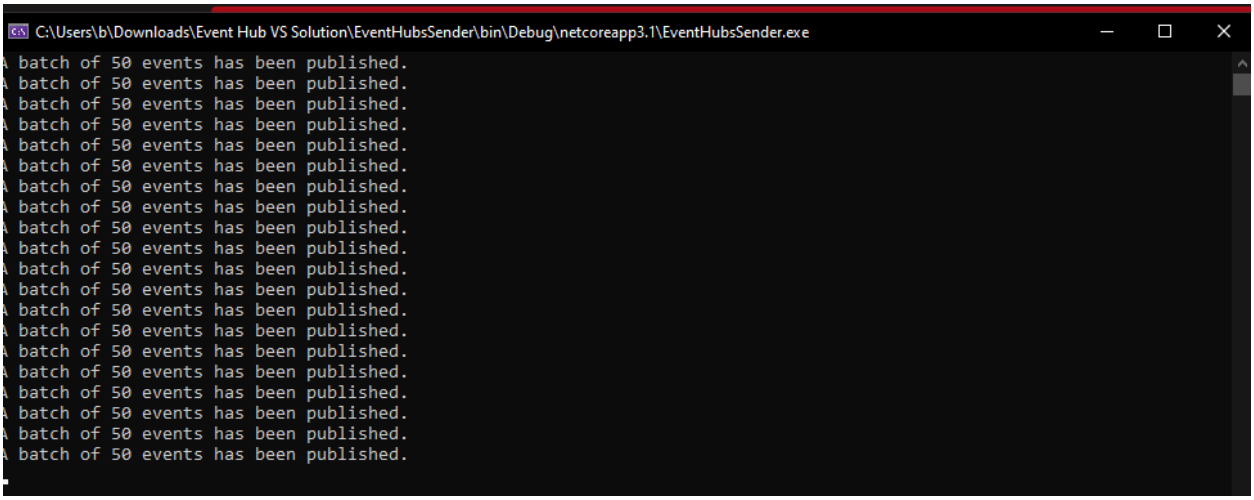
We look into azure, it seems that the connection has been successfully established.

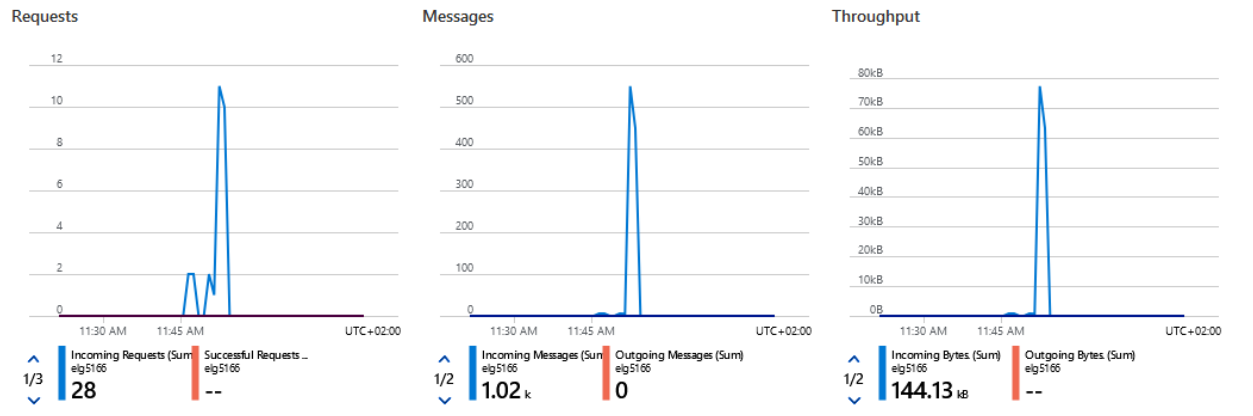
After 5 minutes:



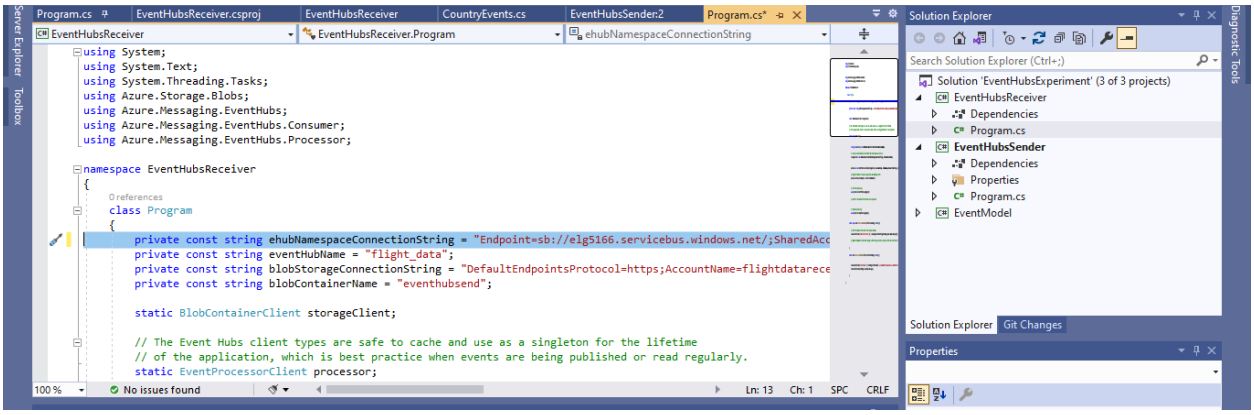
We wait until it's completed.

End of 20 batches:

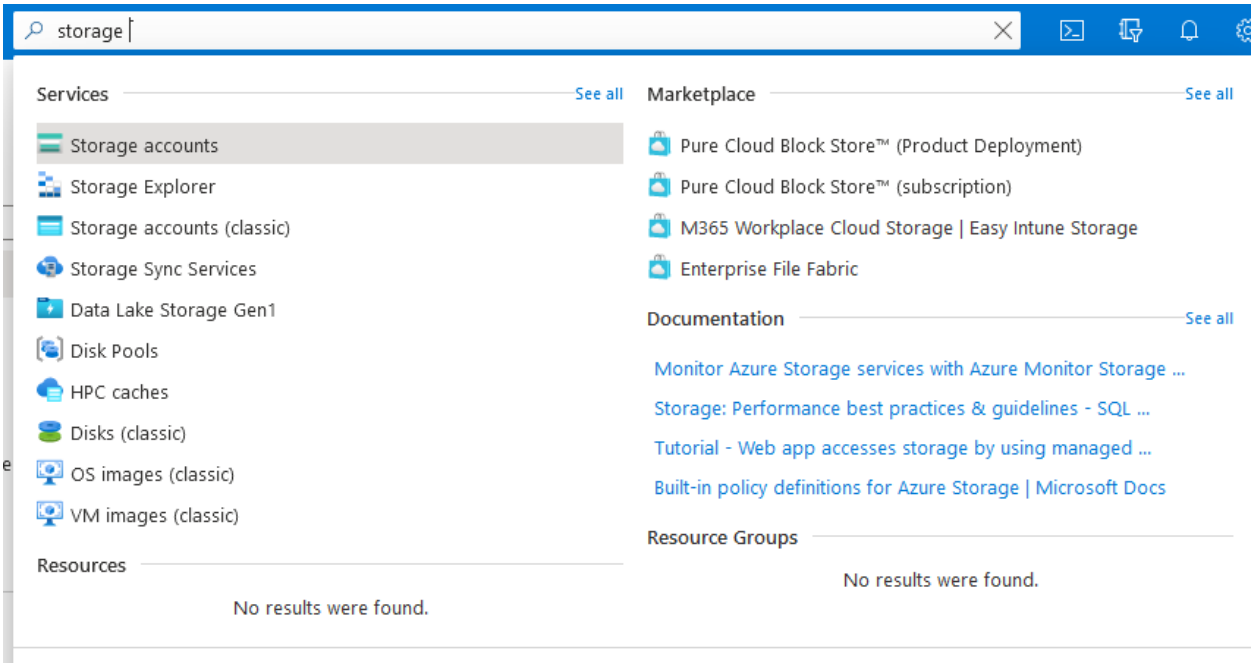




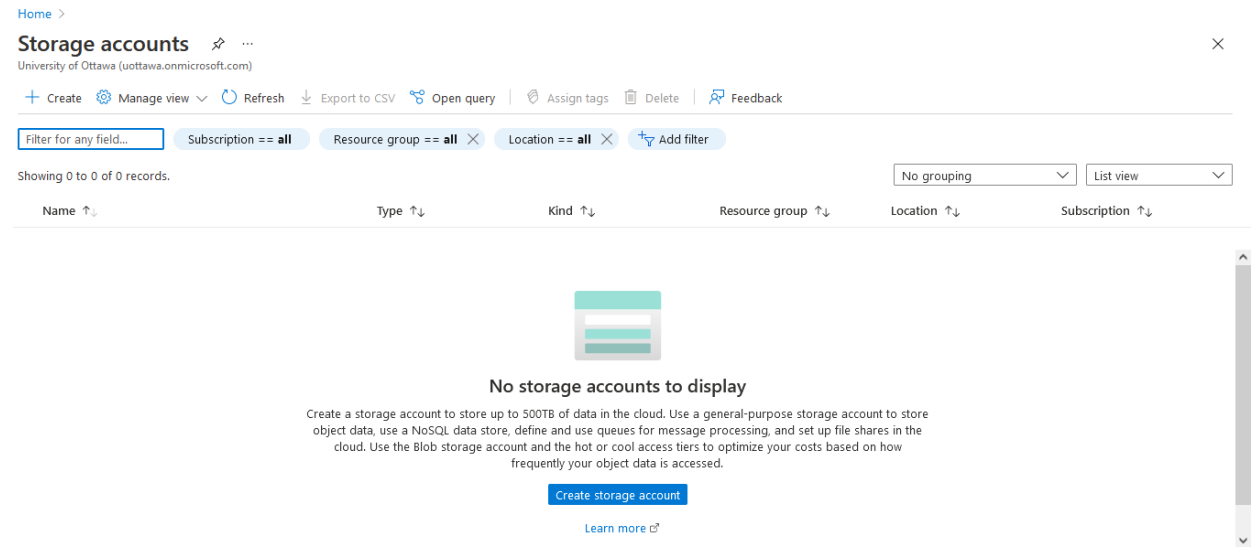
Then we copy the string primary key from the sender to the receiver.

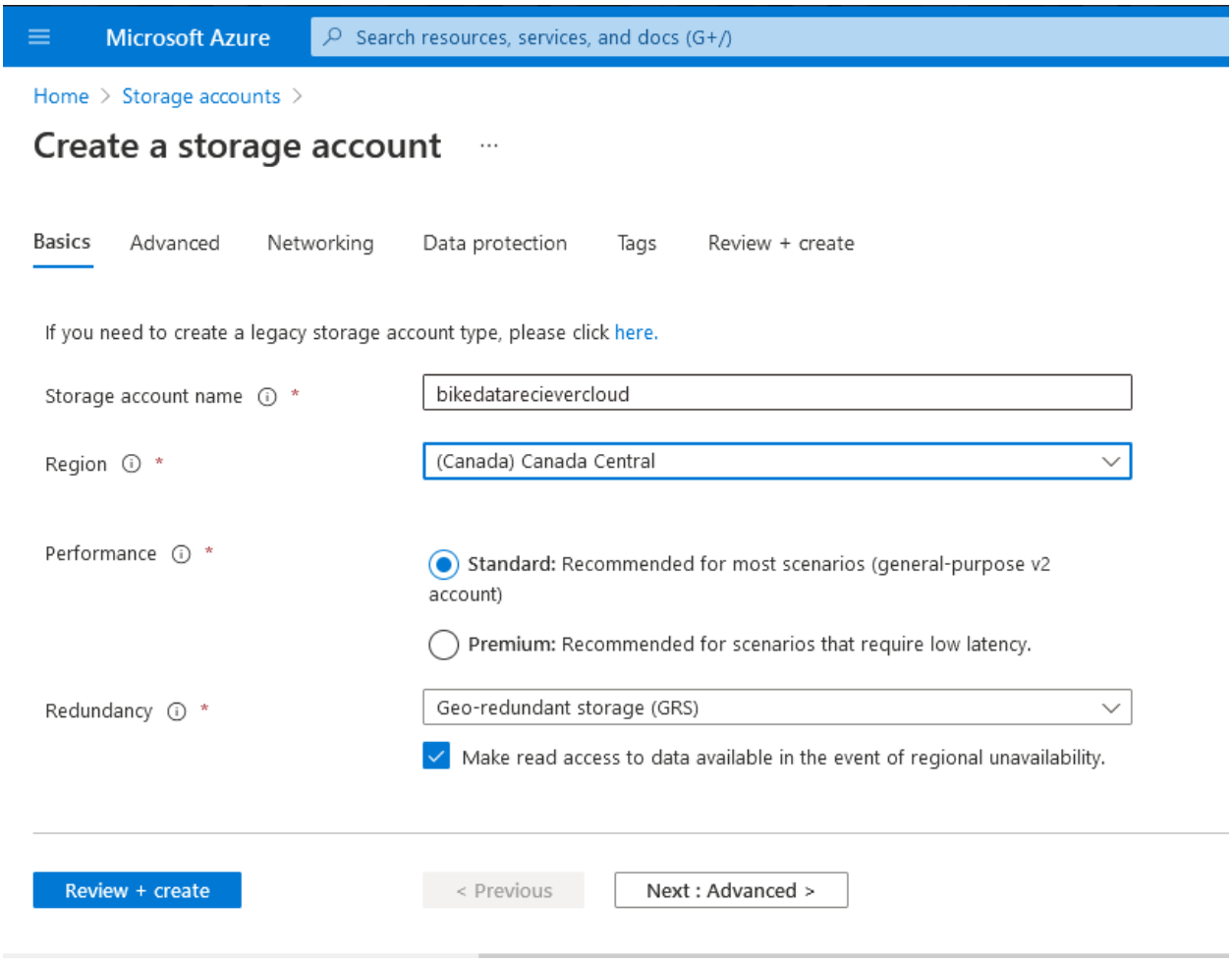


We search for storage account.

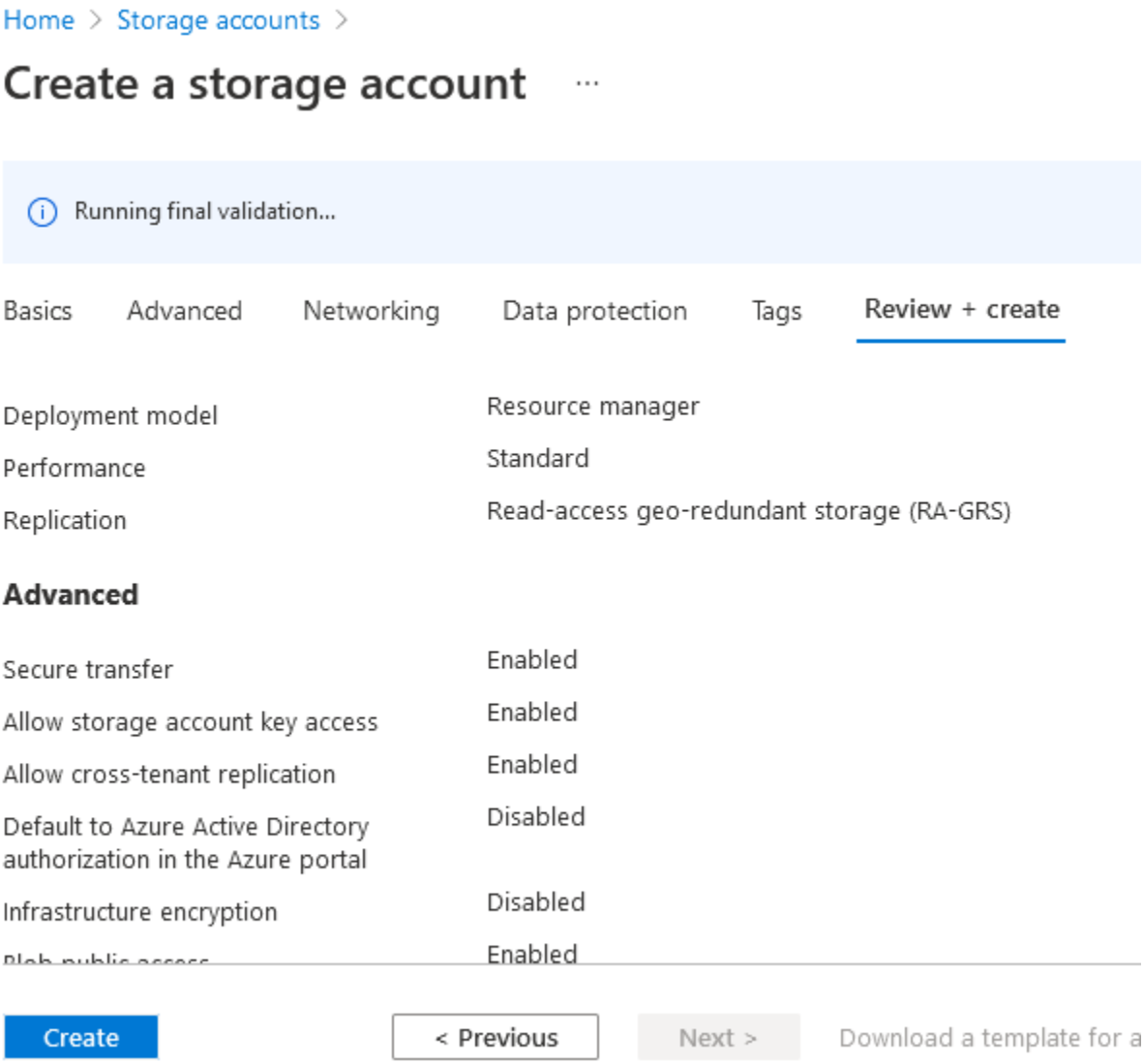


Create storage account.

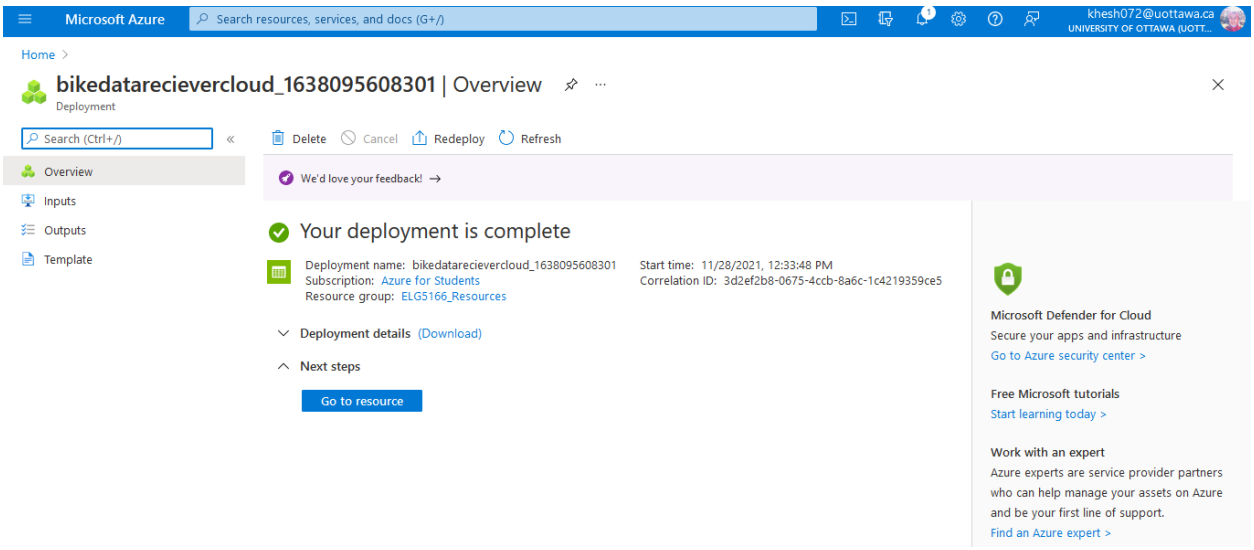




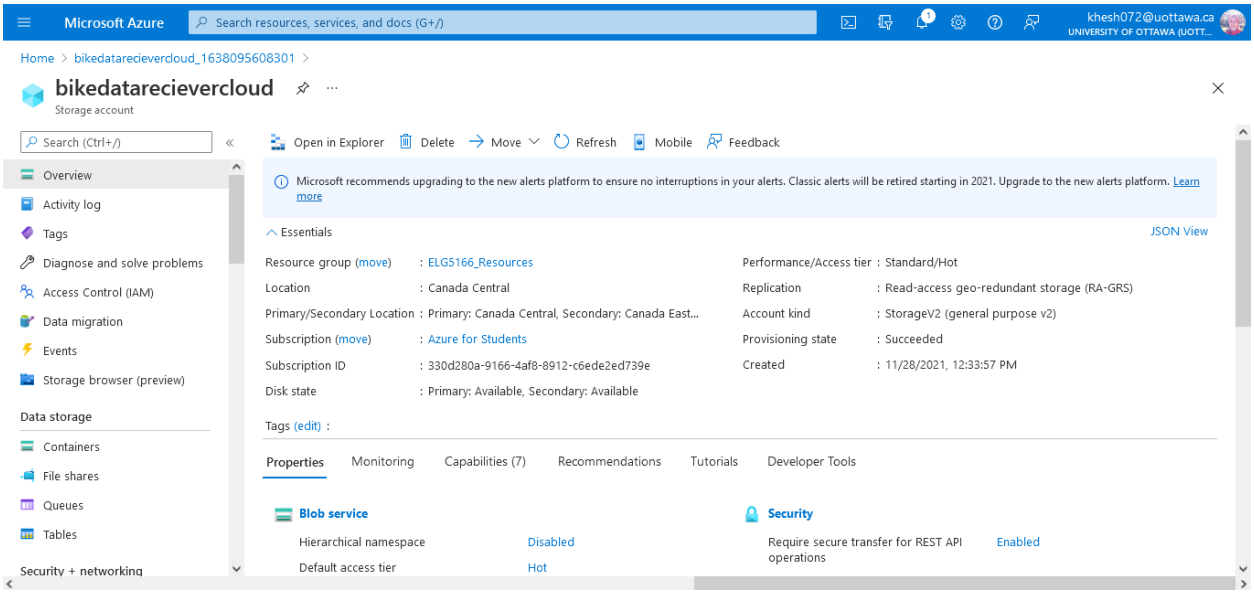
The we click create



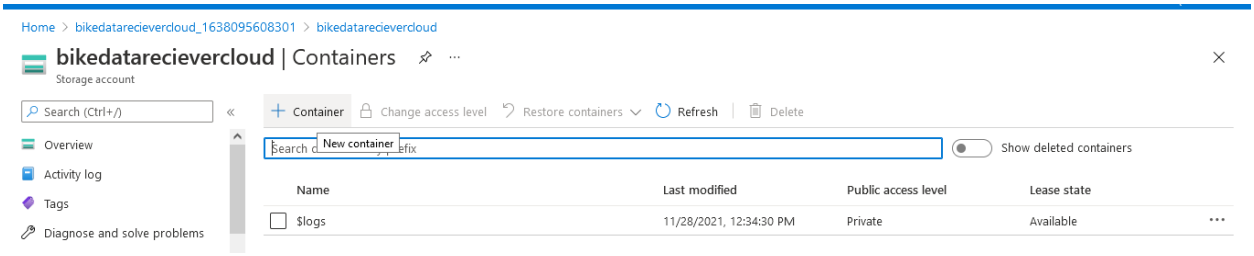
The deployment is done.



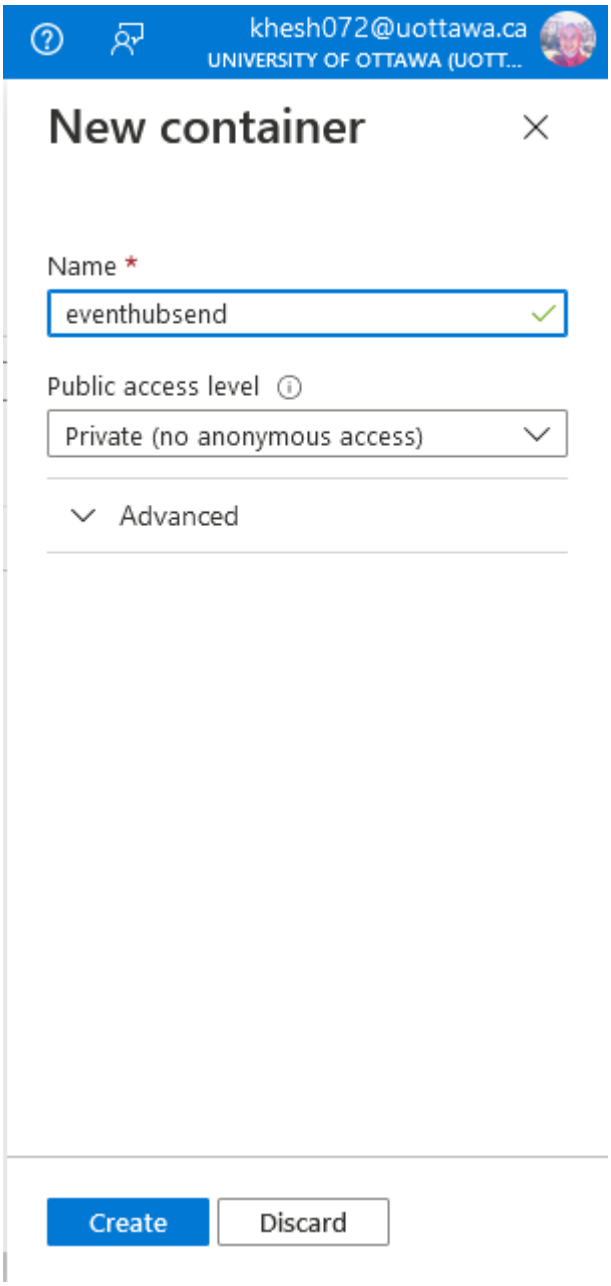
Once we go to the resource, the storage group appears.



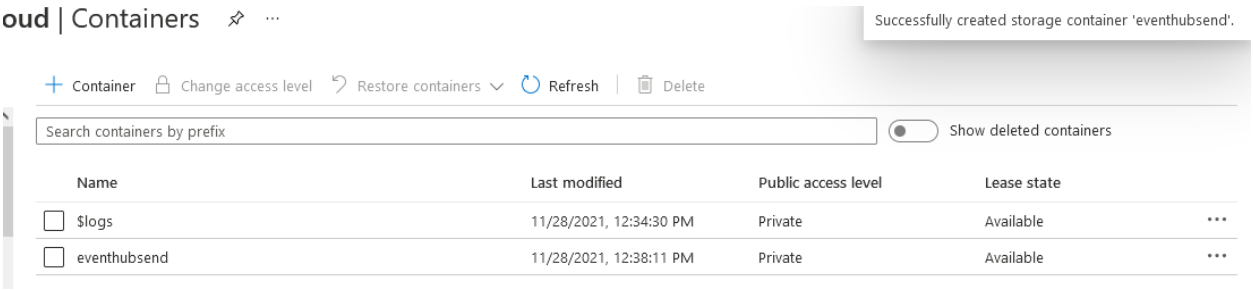
Then, the storage needs a container. We go to the containers to create a new one.



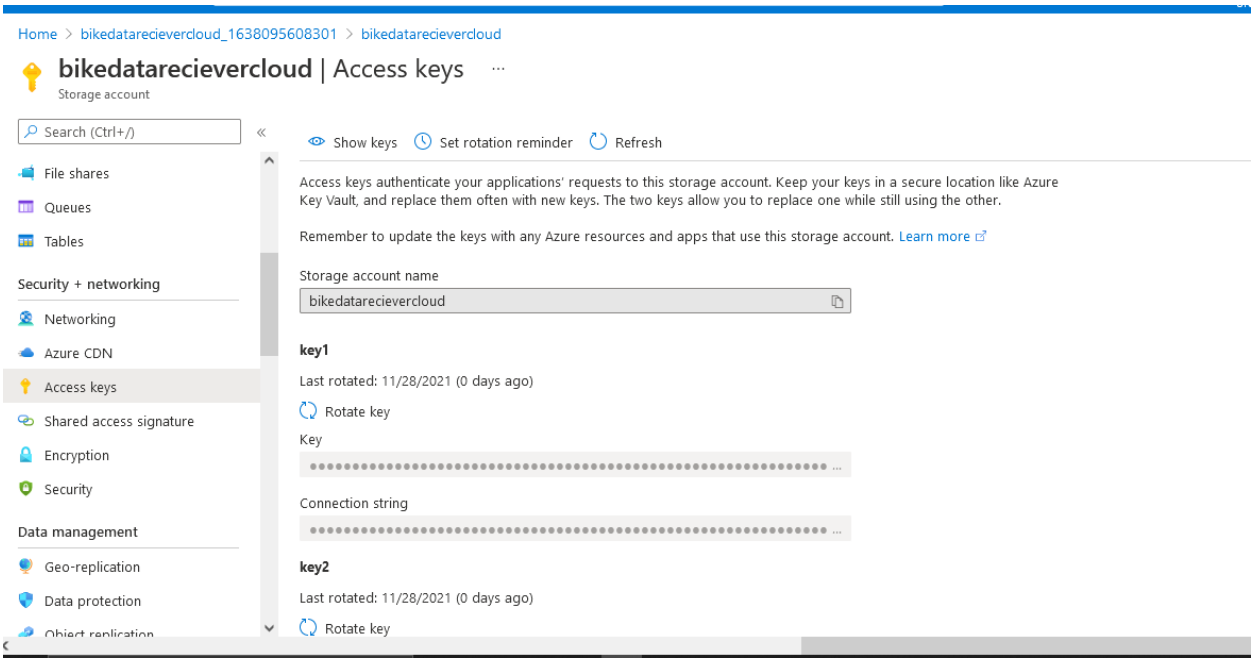
We specify a name then click create.



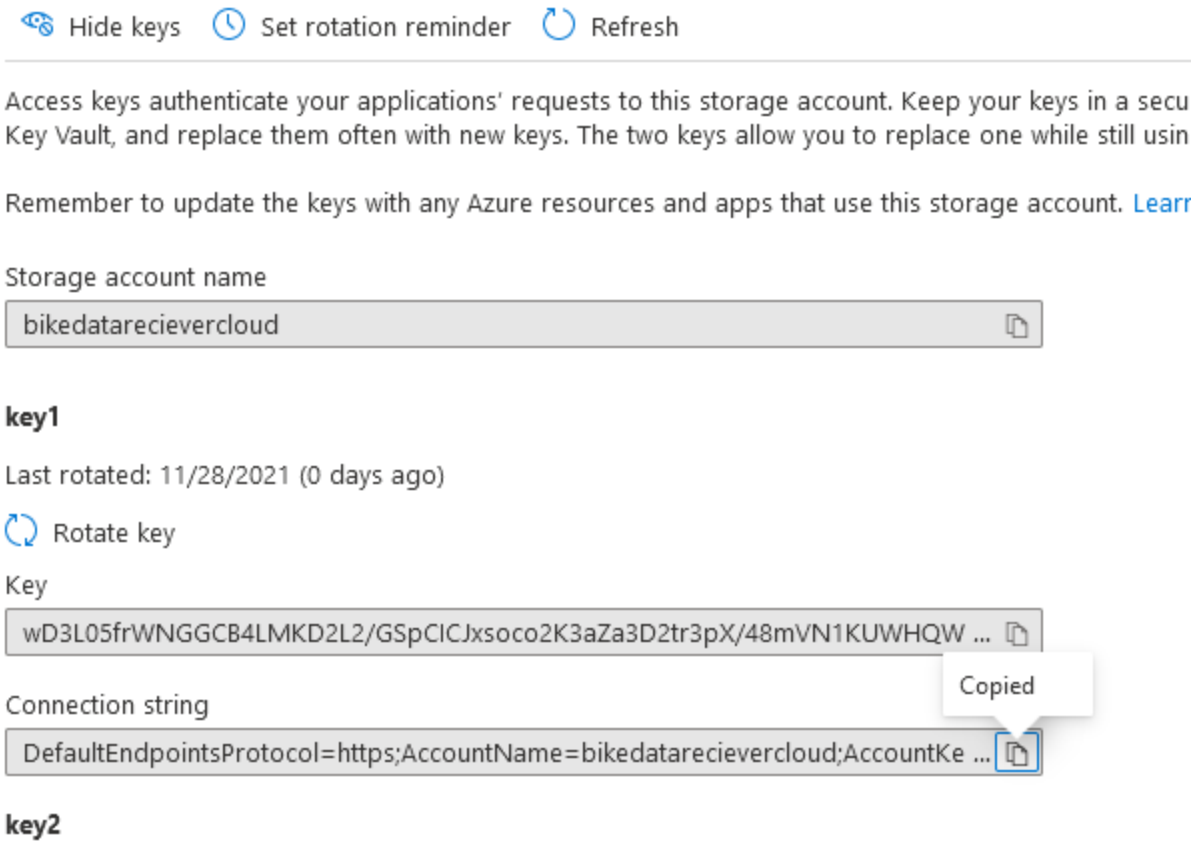
Successfully created.



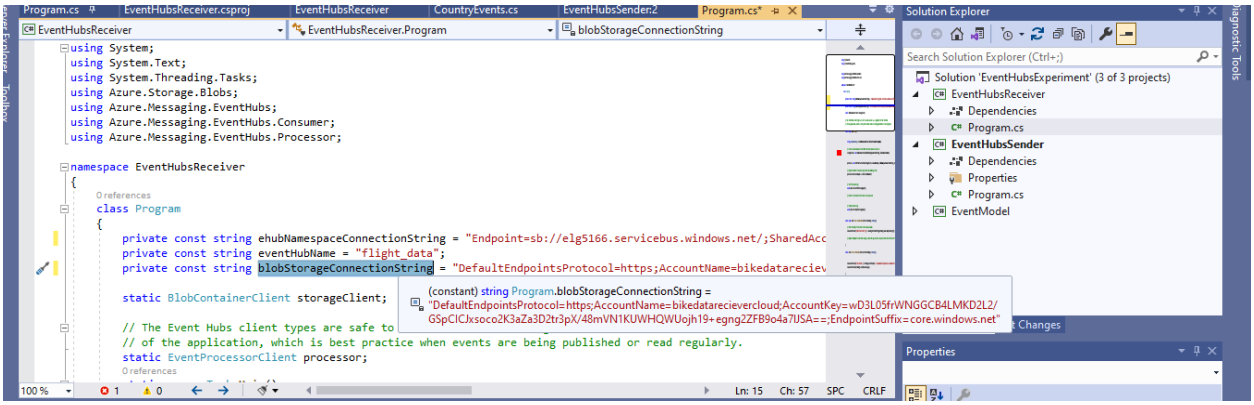
We need to connect things, so that we go back to the receiver to go to the access keys.



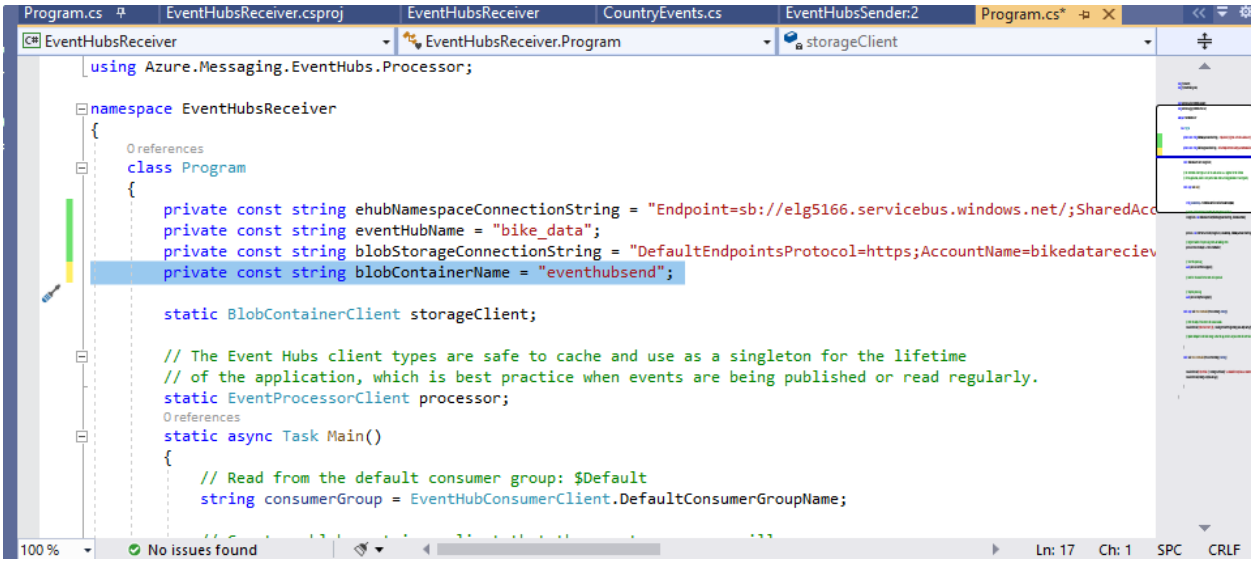
We need to copy the connection key of the first key.



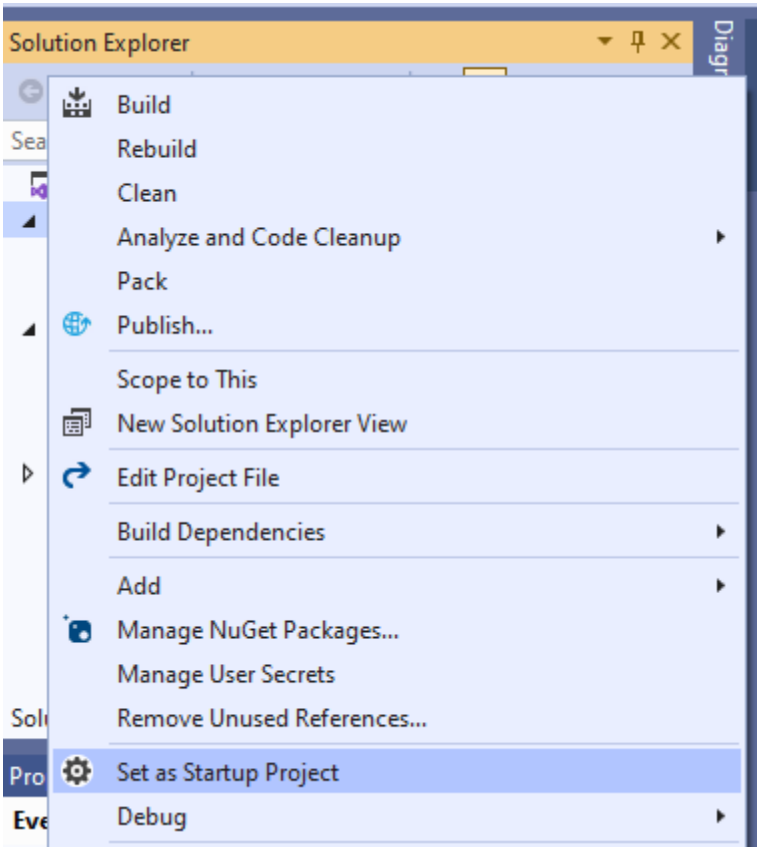
Then we go back to the receiver and paste the connection string to the blobStorageConnectionString



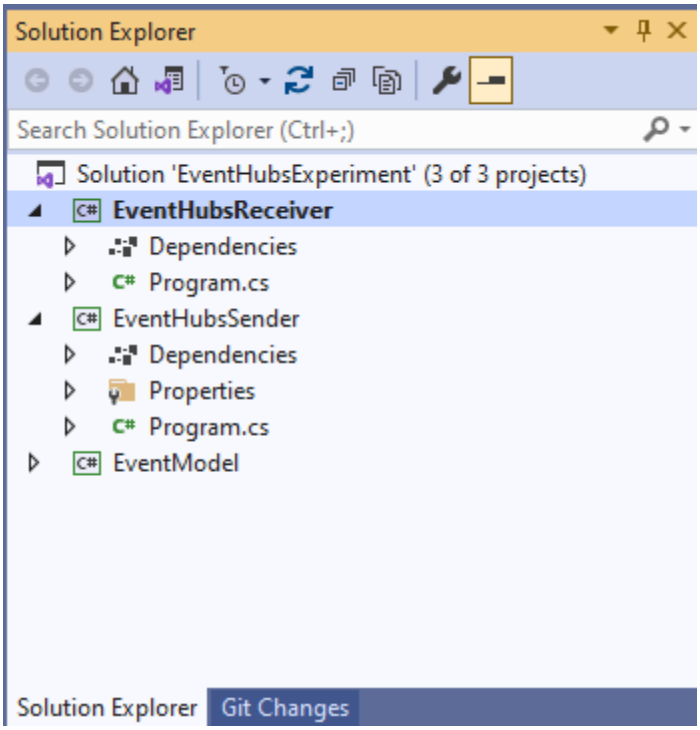
Set the name of the container as we have named it before and the name of the data.



Then we go to the receiver project and choose to set as a setup project.

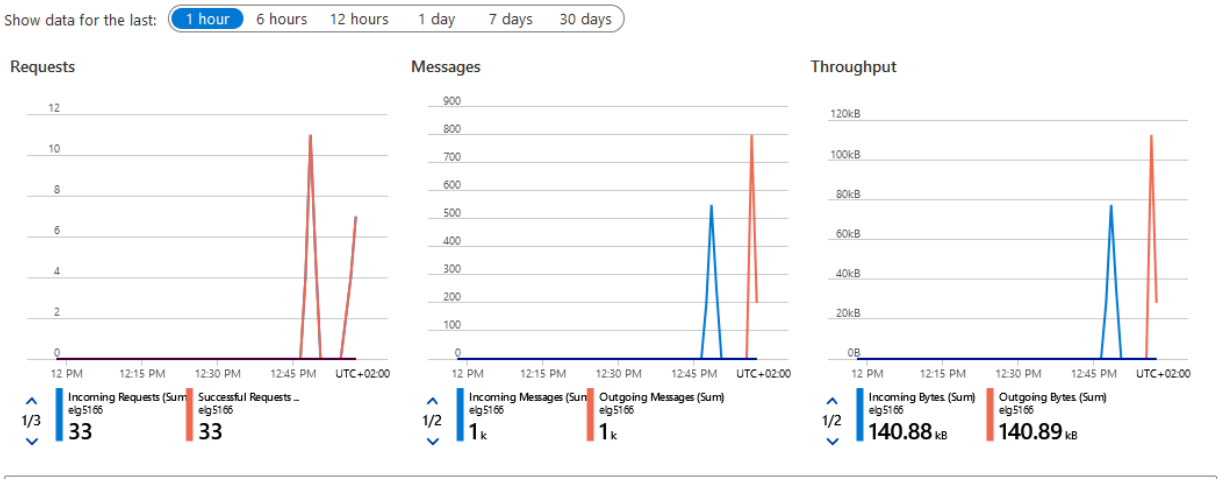


Then the project is being activated.



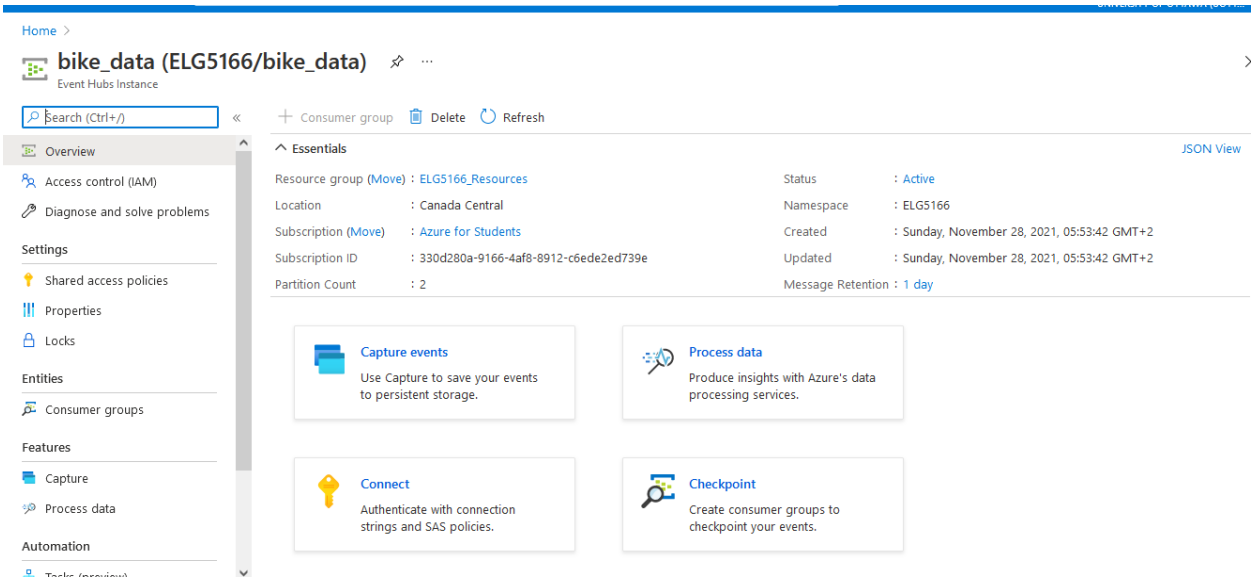
Receiving output:

Get back to the resource group, it seems that we consumed all the data we have.

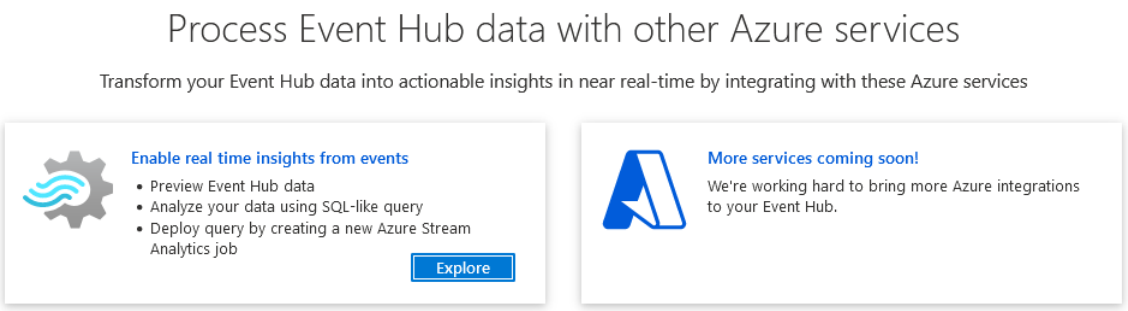


Stream Analytics:

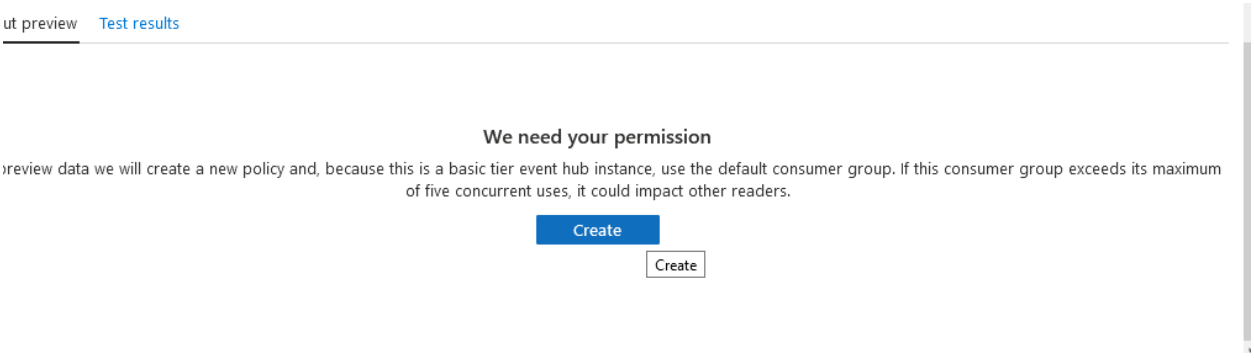
Go to the event-hub here then choose to process the data.



Then we click to explore the data.



Then we click on we need a permission.



Now we can see the data after running the default provided query.

Test query

```
SELECT
    *
INTO
    [OutputAlias]
FROM
    [bikedata]
```

Input preview | Test results

Showing events from 'bikedata'. This list of events might not be complete.

View in JSON | Table | Raw | Refresh | Download sample data

Trip ID	Duration	Start Date	End Date	Bike #	Subscriber Type	Zip Code
"913453"	"789"	"8/31/2015 23:09"	"8/31/2015 23:22"	"487"	"Customer"	"9069"
"913452"	"293"	"8/31/2015 23:07"	"8/31/2015 23:12"	"538"	"Subscriber"	"94118"
"913451"	"896"	"8/31/2015 23:07"	"8/31/2015 23:22"	"363"	"Customer"	"92562"
"913450"	"255"	"8/31/2015 22:16"	"8/31/2015 22:20"	"470"	"Subscriber"	"94111"

Now we can run the required query:

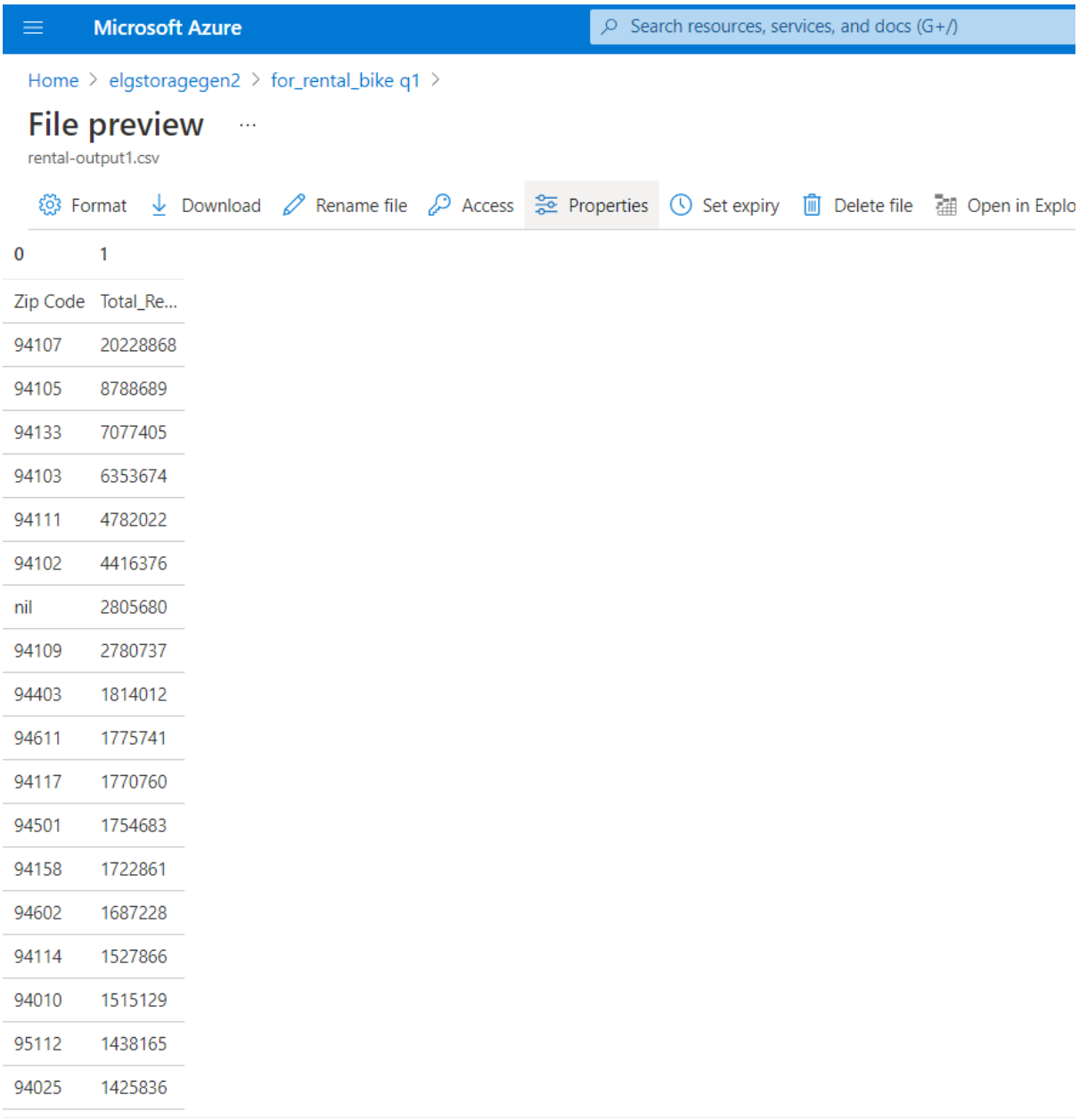
```
SELECT System.TimeStamp AS WindowEnd,End_Date,sum(Bike) As TotalBikes,sum(Duration) AS TotalDuration
INTO [OutputAlias]
FROM [bikedata]
GROUP BY TumblingWindow(Duration(second,2)),End_Date
```

Result:


```
[{"WindowEnd":"2021-11-28T13:17:46.0000000Z","End_Date":null,"TotalBikes":12228,"TotalDuration":23356.0}]
```

Part 2 - Azure Data Lake Analytics:

a) Top 20 zip codes where most bikes were rented from.



0	1
Zip Code	Total_Re...
94107	20228868
94105	8788689
94133	7077405
94103	6353674
94111	4782022
94102	4416376
nil	2805680
94109	2780737
94403	1814012
94611	1775741
94117	1770760
94501	1754683
94158	1722861
94602	1687228
94114	1527866
94010	1515129
95112	1438165
94025	1425836

b) Daily duration aggregate across the rental subscriber types.

File preview ...

rental-output2.csv

Format | Download | Rename file | Access | Properties | Set expiry | Delete file | Op

0	1	2
Subscriber ...	Daily_Date	Total_D...
Customer	2014-09-01	869202
Subscriber	2014-09-01	116866
Customer	2014-09-02	411520
Subscriber	2014-09-02	624576
Customer	2014-09-03	215913
Subscriber	2014-09-03	722325
Customer	2014-09-04	285461
Subscriber	2014-09-04	680215
Customer	2014-09-05	427192
Subscriber	2014-09-05	722445
Customer	2014-09-06	667018
Subscriber	2014-09-06	115238
Customer	2014-09-07	745473
Subscriber	2014-09-07	149557
Customer	2014-09-08	890772
Subscriber	2014-09-08	704323
Customer	2014-09-09	513122
Subscriber	2014-09-09	831411

c) What are the top busiest terminals for bike pickup.

File preview ...

rental-output3.csv

Format Download Rename file Access Properties

0	1
Start...	name
70	San Francisco Caltrain (Towns...
69	San Francisco Caltrain 2 (330 ...
50	Harry Bridges Plaza (Ferry Buil...
55	Temporary Transbay Terminal ...
60	Embarcadero at Sansome
61	2nd at Townsend
65	Townsend at 7th
74	Steuart at Market
67	Market at 10th
77	Market at Sansome
76	Market at 4th
39	Powell Street BART
64	2nd at South Park
56	Beale at Market
73	Grant Avenue at Columbus Av...
62	2nd at Folsom
72	Civic Center BART (7th at Mar...
57	5th at Howard

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

File preview ...

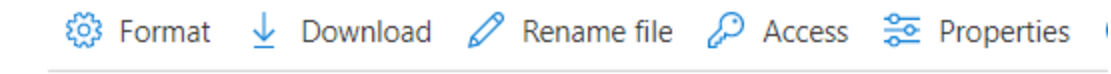
rental-output4.csv

Format | Download | Rename file | Access | Proper

0	1
End ...	name
24	Redwood City Public Libr...
21	Franklin at Maple
83	Mezes Park
23	San Mateo County Center
26	Redwood City Medical C...

e) What is the monthly summary of bike rentals (format - month/year ex. 06/2020).

We will show the monthly summary bike rentals assuming we want the trips count and duration of the bike rentals.



0	1	2	3
Month_y...	Total_bike...	Total_dur...	Trip_co...
2014-09	13310244	33160021	31682
2014-10	14386280	33401099	34220
2014-11	10752350	22454934	25516
2014-12	8382802	41131402	19677
2015-01	11756577	25611358	27840
2015-02	11139951	25633016	26401
2015-03	13444028	29892301	31626
2015-04	13313721	28031940	31363
2015-05	12574550	31584633	29540
2015-06	13447867	34481927	31907
2015-07	13718375	33983062	32476
2015-08	13564645	31088866	31904

Part 3 - Definitions: **1. Please compare briefly, based on at least 3 criteria, the differences in architecture between Apache Spark Structured Streaming and Azure Event Hubs & Stream Analytics?** Ref [1, 2]

	Spark Structured Streaming	Azure Event Hubs	Azure Stream Analytics
Definition	A stream processing engines that allow users to process huge amounts of data using complex algorithms expressed with high-level functions like map, reduce, join, and window to be then pushed to filesystems, databases, or even back to Event Hubs.	A streaming platform and event ingestion service.	A real-time analytics service designed to analyze and process fast moving streams of data from multiple sources simultaneously that can be used to get insights, build reports or trigger alerts and actions.
Working mechanism	• Works on the architecture of polling the data after some duration, based on your trigger interval, but, there is no concept of a batch. • Each event is represented by a row in an unbounded table, and event-time is a column value in the row. • It process the data based on event-time when the timestamp of the event is included in the data received which handles late coming data and gets more accurate results. • Is based on Dataframe and Dataset Spark APIs which generate logically and	• It ingests millions of events per second from any source to build dynamic data pipelines and immediately respond to business challenges. • It decouples event producers from event consumers using a partitioned consumer model.	• It ingests data from Azure Event Hubs (including Azure Event Hubs from Apache Kafka), Azure IoT Hub, or Azure Blob Storage. • The query, which is based on SQL query language, can be used to easily filter, sort, aggregate, and join streaming data over a period of time.

	physically optimized query plan automatically.		
Processing of the output	• It has a limited number of output sinks and with one sink only one operation can be performed and can't save the output at multiple external storages.	Allows multiple applications to process the stream concurrently with a controlled processing speed.	Can write query result into multiple Azure services, including Azure Cosmos DB, Azure Synapse Analytics, Azure SQL DB, storage.
Keeping relative state based on time	• Can maintain the intermediate state for partial aggregates for a long period of time such that late data can update aggregates of old windows correctly. For example, detect spike or get the maximum value all time.	Has limited time retention buffer that prevents it from doing a direct query on past event data.	Can't store a relative state regardless of how much time has passed.
Use case	Build end-to-end continuous applications using Apache Spark that combine: streaming, batch, and interactive queries.	• When there are multiple data sources and we need to feed them into one place for analysis. • Connect with Azure Stream Analytics to build an end-to-end serverless streaming solution. • Capture real-time data in an Azure Blob storage or Azure Data Lake Storage for long-term retention or micro-batch processing.	• Connect Azure event hubs with Azure Cosmos DB since Azure event hubs can't direct its output into Azure Cosmos DB, only a few Azure services can direct it. • Find patterns and relationships that can be used to trigger actions and initiate workflows such as creating alerts, feeding information to a reporting tool, or storing transformed data for later use. • Filter, sort,

			aggregate, and join streaming data over a period of time easily through simple language constructs and/or configurations.
Reusability	Can reuse the code and move its results to a different cloud provider with some work.	Can't take the code out of Azure and reuse it somewhere else, because it is a proprietary language and solution.	Can't take the code out of Azure and reuse it somewhere else, because it is a proprietary language and solution.
Programming Languages	Can use the Dataset/DataFrame API in Scala, Java, Python or R to express streaming aggregations, event-time windows, stream-to-batch joins, etc.	Not applicable.	Can extend its SQL language with JavaScript and C# user-defined functions (UDFs).
Recovery	Ensure end-to-end exactly-once semantics under any failure with the help of checkpointing and write-ahead logs mechanisms.	Keep processing data during emergencies using the geo-disaster recovery and geo-replication features.	• Has built-in checkpoints to maintain the state of your job and provides repeatable results in case the delivery of an event fails. • Guarantees exactly once event processing and at-least-once delivery of events, so events are never lost.
Scalability	Is scalable and received data in a trigger is appended to a continuously flowing unbounded data stream.	Easy scaling from streaming megabytes of data to terabytes while keeping control over when and how much to scale.	Adding more Streaming Units to a job easily. But it can't do auto scaling.
Security	By default, it's not enabled and need additional configurations.	It supports the option of encrypting data at rest with either Microsoft-managed keys or customer-managed keys.	It encrypts all incoming and outgoing communications and built-in checkpoints.

2. Describe briefly 3 benefits of Azure U-SQL over Apache Spark. Illustrate them briefly with some use cases. Ref [3]

Azure U-SQL is a Microsoft query language that combines a declarative SQL-like syntax with C# programming, enabling it to be used to process both structured and unstructured data in big data environments with batch mode only.

Benefits of Azure U-SQL over Apache Spark:

1. With Azure U-SQL, you can query data where it lives instead of having to move/copy all the data to one location. For external systems, such as Azure SQL DB/DW and SQL Server in a VM, this is achieved using federated queries against those data sources where the query is “pushed down” to the remote data source and executed on that data source, with only the results being returned.

2. In Microsoft’s Azure cloud, Data Lake Analytics offers U-SQL as a serverless job service which can be cheaper because you aren't charged while it's provisioning. Unlike Apache Saprk that costs money as long as the cluster is running (cluster service). Also, with Apache Spark, you have to take into account the implications of creating, sizing, scaling, and decommissioning the clusters.

3. U-SQL allows writing custom code easily in C# to express complex business requirements:
    - Inline C# functions,
    - User-Defined Functions (UDF),
    - User- Defined Operators (UDO) such as extractors, outputters, reducers, processors, appliers, and combiners,
    - and User-Defined Aggregators (UDAGG) such as aggregate to perform custom math calculations or manipulations with strings.

    Unlike Spark that doesn't offer the same extensibility model for operators, but has equivalent capabilities for some. Also, if the developer is familiar with C# more than Scala or Python, this would be an additional benefit.

4. It unifies querying structured and unstructured data, data viewing (regardless of physical location), and data copying.

3. What are the 5 characteristics of Azure Data Lake that distinguish it from other Distributed Dataset Storage infrastructure such as Hadoop?

- Security and Encryption:
    - Data is encrypted and secured both in transit and rest. Data Lake safeguards the data assets while allowing you to simply extend your on-premises security and governance policies to the cloud. Azure Active Directory has features like single sign-on (SSO) and multi-factor authentication. Unlike Hadoop is designed without considering security of data. Data stored at HDFS is in plaintext. This data is prone to be accessed by unauthorized user.
- Combining data from multiple sources into a single location:
    - Azure Data Lake combines all of big data from many sources across cloud and on-premises infrastructures into a single location. Unlike Hadoop, the analysis of data across different sources, can be extremely difficult to achieve
- Enabling Hadoop for the Cloud:
    - Because of the perceived constraints of cloud infrastructure, many organisations select for an on-premises Hadoop solution. Azure Data Lake brings Hadoop's cloud of constraints closer to typical deployments.
- Unlimited Data Size:
    - File and account sizes are limited to a few terabytes in other cloud storage choices. This may appear to be a large number, but in big data situations, it is actually modest. This limitation is removed with Azure Data Lake: There is no fixed file or account size limit. Hadoop can store and process single or many files at terabyte and petabyte scales. This is a perfect scenario because Hadoop shines when dealing with massive files.
- Very High-Speed Throughput:
    - Internet of Things (IOT) devices such as phones, sensors and equipment are streaming data in very small transactions at very high volume. At massive scale, Azure Data Lake supports incredibly high-speed data ingestion.
- Developed with Parallel Processing:
    - Azure Data Lake is built to facilitate parallel processing throughput in order to deliver performance comparable to an on-premises Hadoop solution.

4. What 2 factors influence the use of Azure AUs for U-SQL query processing. Describe with a simple example.

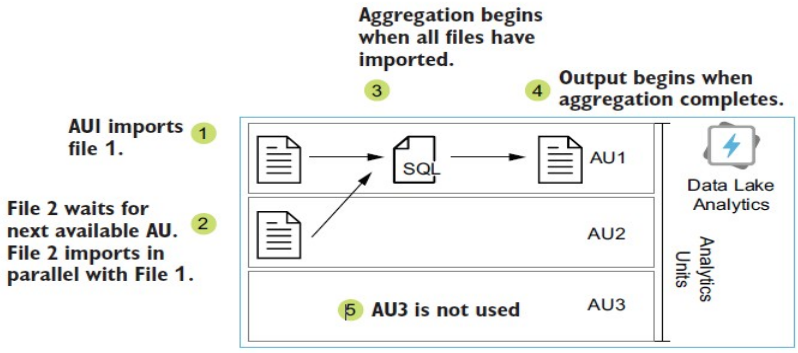


Figure 2.10 Parallel processing in Azure Data Lake Analytics jobs with multiple analytics units

Figure 1: *Source*: Nuckolls, R. L. (2020). "Azure Storage, Streaming, and Batch Analytics. A guide for data engineers". Manning *Publications Co*. ISBN: 9781617296307

The processing can be done by one, two, or more analytics units:

- Single Unit:
  - Both files will be imported consecutively using one unit, the consolidation step will be run, and the summary file will be created.
- Two or more Units:
  - Both files will be imported in parallel using two units, then the aggregation step will be run, and finally the summary file will be created.

2 Factors influence the use of Azure AUs for U-SQL:

- Time:
  - The time which the Analytical Unit takes. So if we want to accelerate the process we can use two or many Analytical Units which the files will be imported in parallel.
- Cost:
  - The cost is the same for running one or more Analytical Units.

Example:

- Time:
  - If we run a job that takes 16 hours with one AUs
  - It would take 8 hours with two AUs.

- It would take one hour with 16 AUs.

- Cost:
  - All three above runs will cost the same.
  - 16hr*1AUs = 8hr*2AUs = 1hr*1AUs = the same cost.

References:

[1] Lab code and lecture notes.

[2] https://stackoverflow.com/questions/52135478/deserialize-array-of-arrays-in-c-sharp/52135640

[3] https://superuser.com/questions/34451/notepad-find-and-replace-string-with-a-new-line

[4] Nuckolls, R. L. (2020). "Azure Storage, Streaming, and Batch Analytics. A guide for data engineers". *Manning Publications Co.* ISBN: 9781617296307

[5] https://docs.microsoft.com/en-us/azure/stream-analytics/stream-analytics-introduction

[6] https://azure.microsoft.com/en-us/services/event-hubs/#overview

[7] https://devblogs.microsoft.com/visualstudio/introducing-u-sql-a-language-that-makes-big-data-processing-easy/

[8] https://xo.xello.com.au/blog/azure-data-lake-benefits

[9] https://www.bluegranite.com/blog/6-key-features-from-microsoft-s-azure-data-lake

[10] https://www.techopedia.com/2/31730/trends/big-data/hadoop-analytics-not-so-easy-across-multiple-data-sources

[11] https://stackoverflow.com/questions/6467216/is-it-possible-to-use-aggregate-function-in-a-select-statment-without-using-grou