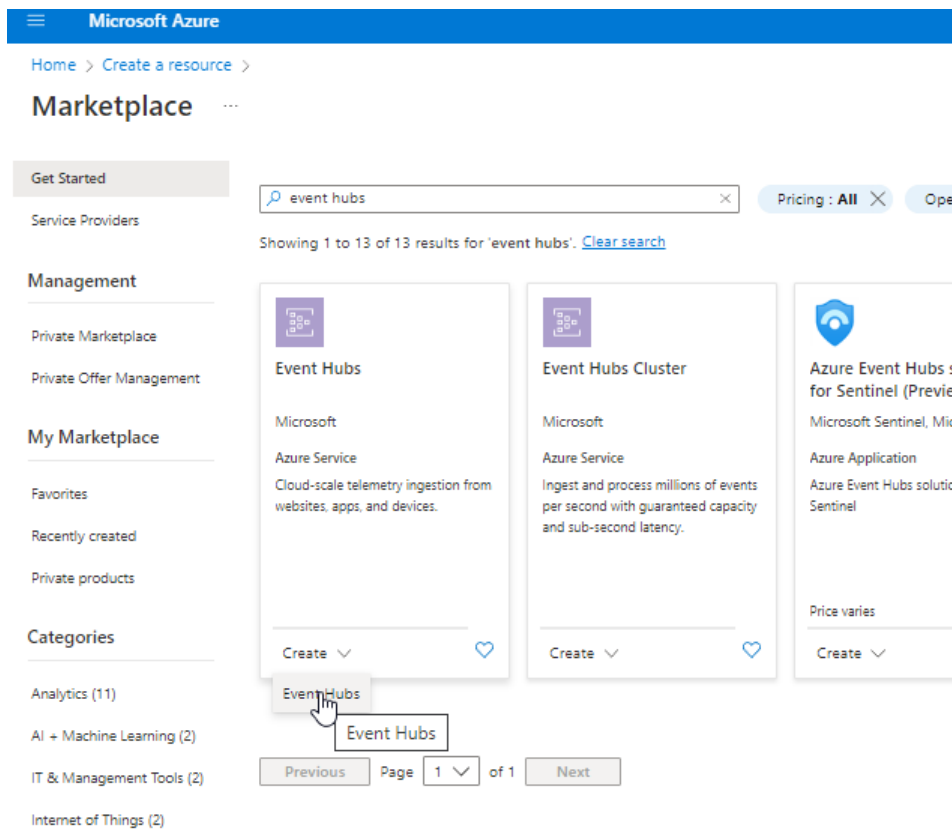




Assignment 3

MODERN DATA MANAGEMENT & BUSINESS INTELLIGENCE

After creating the account, first we add the Event Hubs module from marketplace.



We set up the resource group, namespace of the hub etc. and proceed by creating it.

Basics Advanced Networking Tags Review + create

Project Details

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

Subscription * Azure for Students

Resource group * MyRG

[Create new](#)

Instance Details

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

Namespace name * KOSTAS

.servicebus.windows.net

Location * West Europe

i The region selected supports Availability zones. Your namespace will have Availability Zones enabled. [Learn more.](#)

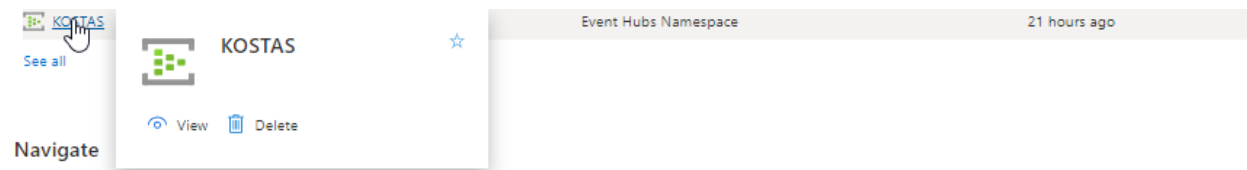
Pricing tier * Standard (~\$22 USD per TU per Month)

[Browse the available plans and their features](#)

Throughput Units * 1

Enable Auto-Inflate **i** ☐

We then select the hub namespace and create a new event hub.



Create Event Hub ...

Event Hubs

Basics Capture Review + create

Event Hub Details

Enter required settings for this event hub, including partition count and message retention.

Name * ⓘ eventhubdemo ✓

Partition count ⓘ 1

Retention

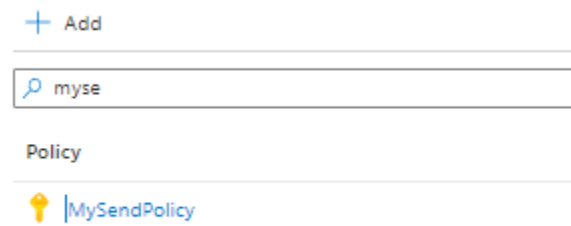
Configure retention settings for this Event Hub. [Learn more](#)

Cleanup policy ⓘ Delete

Retention time (hrs) ⓘ 1

min. 1 hour, max. 168 hours (7days)

On the newly created event hub, we create a sent policy.



...

☐ Manage
☒ Send
☐ Listen

Primary key

Zg8DU+3xyUsHk7vllkOyKNFveQPelzFLaTgE7vxZZrY=

Secondary key

HhxG+H0Uhybzhq7VNNAs8Lqo6cho5DpJfl+s/Fp23UI=

Connection string-primary key

Endpoint=sb://kostas.servicebus.windows.net;/SharedAccessKeyName=MySendPolicy;S...

Connection string-secondary key

Endpoint=sb://kostas.servicebus.windows.net;/SharedAccessKeyName=MySendPolicy;S...

SAS Policy ARM ID

/subscriptions/92398771-94bb-412f-928a-264bbfb74322/resourcegroups/MyRG/provi...

Using the primary key produced by the sent policy, we generate a signature using the Event Hub-Signature Regenerate.

Event Hubs - Signature Generator

Hub	Credentials
Namespace: KOSTAS	Sender Key Name: MySendPolicy
Hub Name: eventhubdemo	Sender Key: uxpcTGsLoo85VF/3GD0o5Sb5TYbX4Dfh7PU=
Publisher: Laptop	Token TTL (minutes): 60
Mode: Http	

Signature

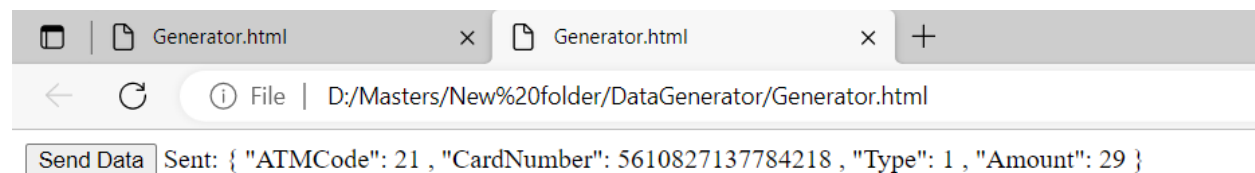
SharedAccessSignature sr=https%3a%2f%2fkostas.servicebus.windows.net%2feventhubdemo%2fpublishers%2flaptop%2fmessages&sig=W7OUjy2wVEqt5BnxPoTkaPN7Or6oDRf036YvHa5uAc%3d&se=1673706819&skn=MySendPolicy

Generate

We then fill in the produced signature on the Generator file as well as the service namespace, the hub name and the device name.

```
1 <html>
2 <head>
3   <script src="js/lodash.js"></script>
4 </head>
5 <body>
6   <input type="button" value="Send Data" onclick="sendDummyData()" />
7   <div id="status" style="display: inline-block;"></div>
8   <script type="text/javascript">
9     function sendDummyData() {
10
11       /*****
12       *** CONFIG ***
13       *****/
14
15       //Use the signature generator: https://github.com/sandrinodimattia/RedDog/releases
16       var sas = "SharedAccessSignature sr=https%3a%2f%2fkostas.servicebus.windows.net%2feventhubdemo%2fpublishers%2f1";
17       var serviceNamespace = "mscba-aueb";
18       var hubName = "eventhubdemo";
19       var deviceName = "Laptop";
20
21       /*****
22       *** GENERATOR ***
23       *****/
24       var atms = [{"atm_code":1,"area_code":20},{ "atm_code":2,"area_code":17},{ "atm_code":3,"area_code":18},{ "atm_code":
25
26       var customers = [{"card_number":5446210381593272,"first_name":"Eugene","last_name":"Mason","age":67,"gender":"Mal
27
28       var jsonData;
29
30       var RND_Customer = 0;
31       var RND_ATMCode = 0;
32       var RND_CardNumber = 0;
33       var RND_Type = 0;
34       var RND_Amount = 0;
35
36       setInterval(function(){
37
38         RND_Customer = _.random(0,19);
39
40         if( _.random(0,1) == 1 ) {
```

We then open the Generator file and by clicking on Send Date, the process of streaming data on our hub begins.



We then proceed by creating a SAS Policy with the Listen configuration.

+

Add

myrec

Policy

MyRecPolicy

Save

Discard

Delete

Regenerate Primary Key

...

☐

Manage

☐

Send

☒

Listen

Primary key

D61ktKaauUuXOVPRKnXhiZTMnx0yvlsiDcsa+Dkh0Co=

Secondary key

YwzR5p0w0WQJjZhNNP7NUMtN2rZswruZp2tEufel/I=

Connection string-primary key

Endpoint=sb://kostas.servicebus.windows.net/;SharedAccessKeyName=MyRecPolicy;Sh...

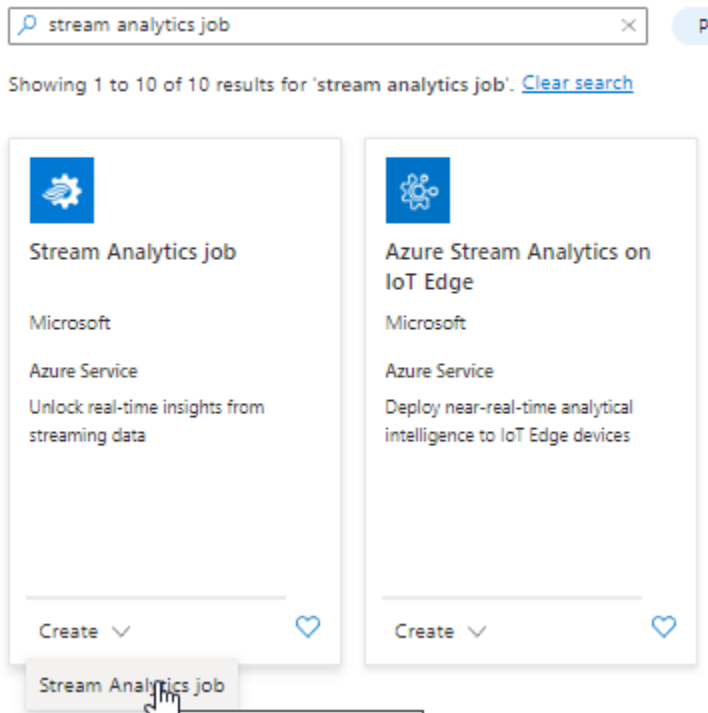
Connection string-secondary key

Endpoint=sb://kostas.servicebus.windows.net/;SharedAccessKeyName=MyRecPolicy;Sh...

SAS Policy ARM ID

/subscriptions/92398771-94bb-412f-928a-264bbfb74322/resourcegroups/MyRG/provi...

We then create a stream analytics job which we find from the marketplace.



After making all the required configuration, we click on create.

New Stream Analytics job

Basics Storage Tags Review + create

Azure Stream Analytics is a fully managed, SQL-based stream processing engine designed to help you tackle scenarios like streaming ETL to Azure Data Lake Storage, real-time dashboarding with Power BI, event driven applications with Azure SQL DB & Cosmos DB, remote monitoring, predictive maintenance, and more. [Learn more](#)

Project details

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

Subscription * ⓘ Azure for Students ▼

Resource group * ⓘ MyRG ▼
[Create new](#)

Instance details

Name * streamdemo

Region * ⓘ West Europe ▼

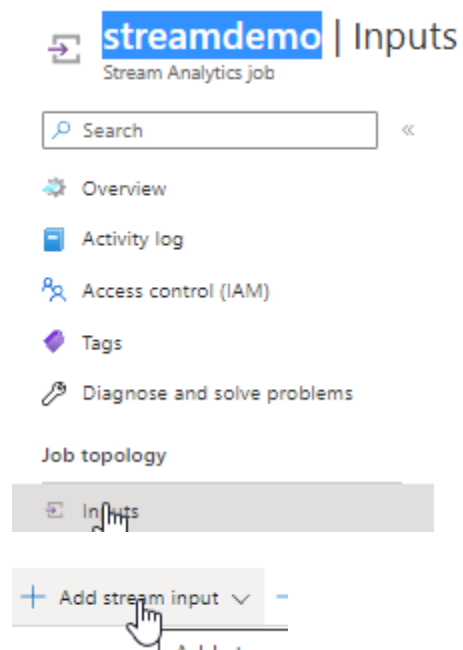
Hosting environment ⓘ
☒ Cloud
☐ Edge

Streaming unit details

Streaming units (SUs) represents the computing resources that are allocated to execute a Stream Analytics job. The higher the number of SUs, the more CPU and memory resources are allocated for your job. The number of SUs can be modified once you create the job. You will be charged for the job's Streaming Units only when the job runs. [Learn more](#)

Streaming units * 3 ▼




On the Stream Analytics Job that we created, on the Inputs section we select to add a stream input.



We configure the new input and we select to receive sample data every 3 minutes.

Input details ✕

input

 Test  Delete  Open Event Hub

Input alias

input

☐ Provide Event Hub settings manually

☒ Select Event Hub from your subscriptions

Subscription

Azure for Students

Event Hub namespace * ⓘ

KOSTAS

Event Hub name * ⓘ

☐ Create new ☒ Use existing

eventhubdemo

Event Hub consumer group * ⓘ

☐ Create new ☒ Use existing

streamdemo_input_consumer_group

Authentication mode

Managed Identity: System assigned

Managed Identity has permissions to the selected Event Hub.

Partition key ⓘ

Event serialization format * ⓘ

JSON

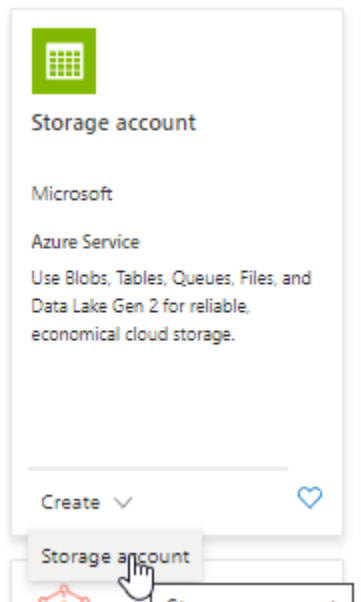
Encoding ⓘ

UTF-8

Event compression type ⓘ

None

We then create a new Storage Account from the marketplace.



After the appropriate configuration, we click on create.

[Home](#) > [Create a resource](#) > [Marketplace](#) >

Create a storage account ...

[Basics](#) [Advanced](#) [Networking](#) [Data protection](#) [Encryption](#) [Tags](#) [Review](#)

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

Project details

Select the subscription in which to create the new storage account. Choose a new or existing resource group to organize and manage your storage account together with other resources.

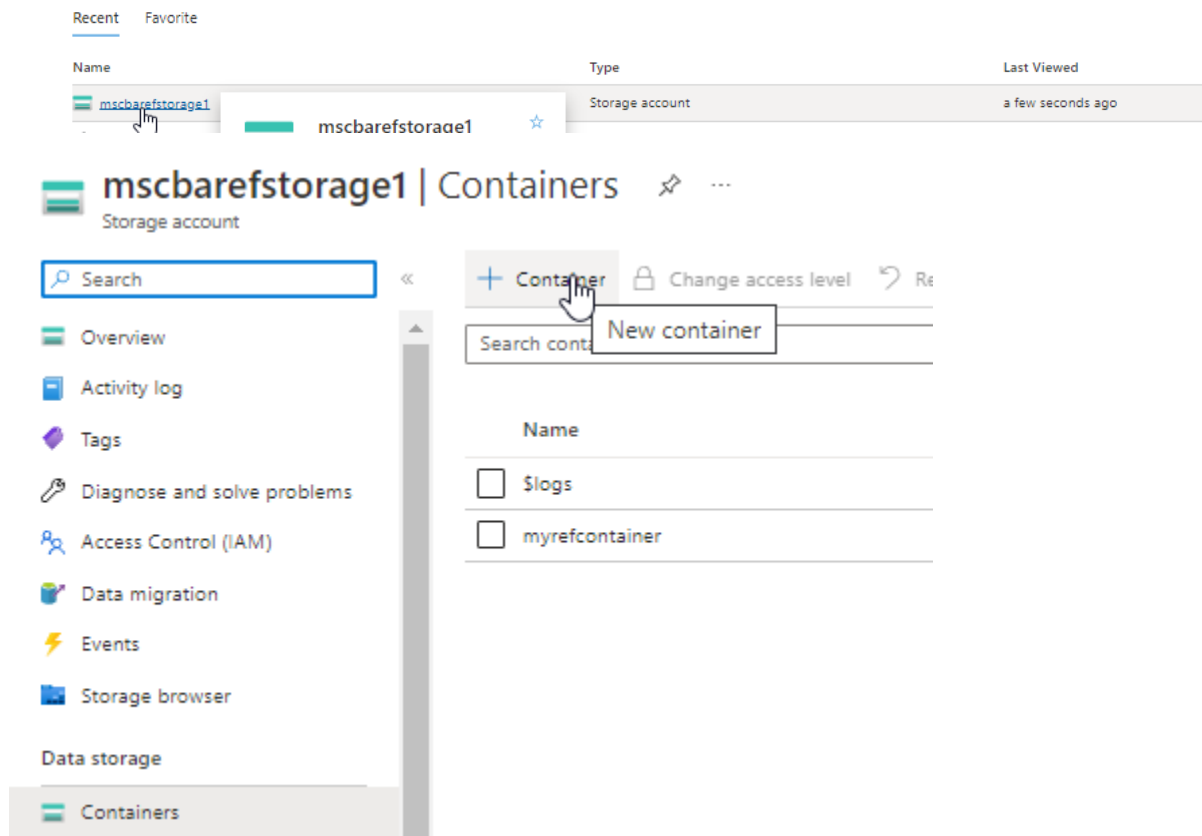
Subscription *	<div>Azure for Students</div>
Resource group *	<div>MyRG</div> <div>Create new</div>

Instance details

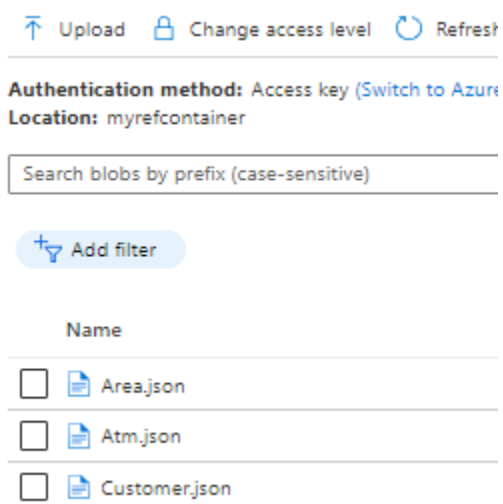
If you need to create a legacy storage account type, please click [here](#).

Storage account name ⓘ *	<div>mscvbarefstorage</div>
Region ⓘ *	<div>(Europe) West Europe</div> <div>Deploy to an edge zone</div>
Performance ⓘ *	<div><input checked="" type="radio"/> Standard: Recommended for most scenarios (general-purpose v2 account)</div> <div><input type="radio"/> Premium: Recommended for scenarios that require low latency.</div>
Redundancy ⓘ *	<div>Geo-redundant storage (GRS)</div> <div><input checked="" type="checkbox"/> Make read access to data available in the event of regional unavailability.</div>

We then select the newly created storage account and on the Container section we click on Create New Container.



After the container is created, we upload the three JSON files.

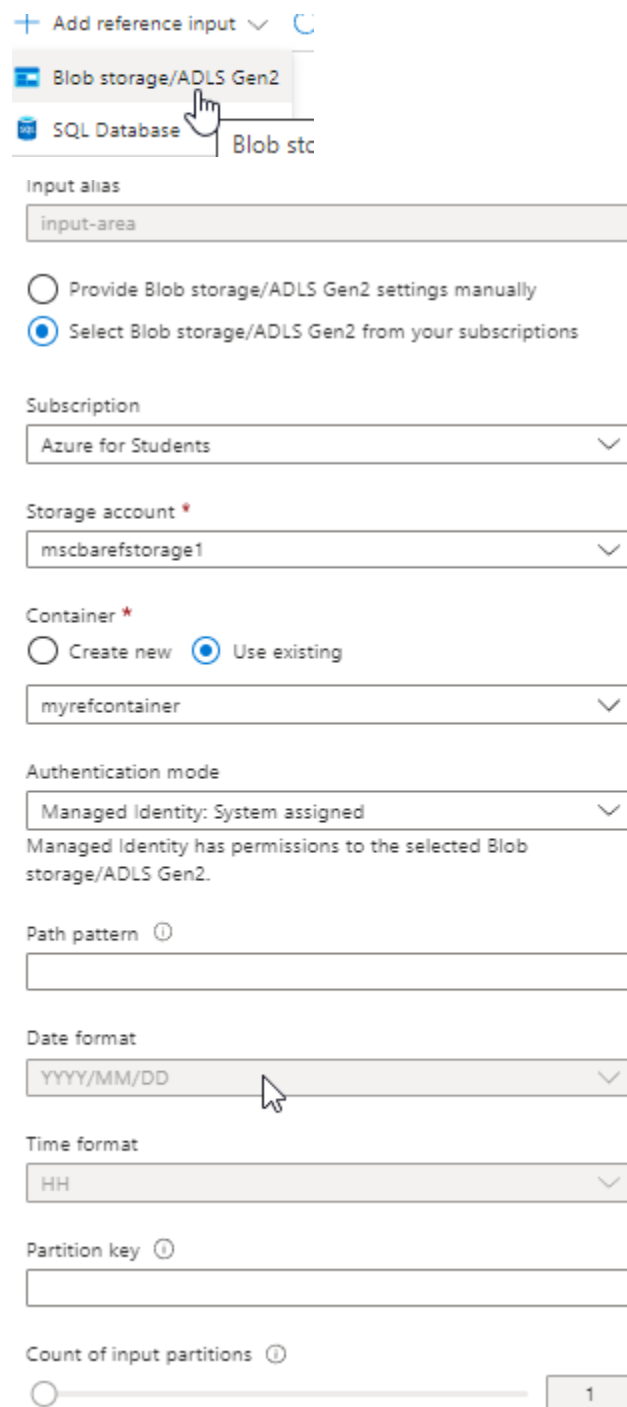


Back on the Stream Analytics Job, on the output section, we add a Blob storage and make the appropriate configuration.

The screenshot shows the 'Output' section of an Azure Stream Analytics job configuration. At the top, there are buttons for '+ Add', 'Refresh', 'Test', 'Delete', and 'Open Blob storage/ADLS Gen2'. Below these, the 'Output alias' is set to 'output'. Two radio buttons are present: 'Provide Blob storage/ADLS Gen2 settings manually' (unselected) and 'Select Blob storage/ADLS Gen2 from your subscriptions' (selected). A dropdown menu is open under the 'Add' button, showing options: 'Azure Data Explorer', 'Azure Function', 'Azure Synapse Analytics', 'Blob storage/ADLS Gen2' (highlighted), and 'Cosmos DB'. Below the menu, the configuration fields are as follows:

- Subscription:** Azure for Students
- Storage account:** msccarestorage1
- Container:** myrefcontainer (Selected: Use existing)
- Authentication mode:** Managed identity: System assigned
- Event serialization format:** JSON
- Format:** Line separated
- Encoding:** UTF-8
- Write mode:** Append as results arrive (Selected)
- Path pattern:** (Empty field)
- Date format:** YYYY/MM/DD
- Time format:** HH
- Minimum rows:** (Empty field)
- Maximum time:** Hours: 0, Minutes: 2, Seconds: 0

On the input section of the Stream Analytic Job, we click on Add reference input/Blob storage and we make the appropriate configuration.



The screenshot shows the 'Add reference input' configuration window for 'Blob storage/ADLS Gen2'. The window has a title bar with a plus icon, the text 'Add reference input', a dropdown arrow, and a refresh icon. Below the title bar, there are three options: 'Blob storage/ADLS Gen2' (selected), 'SQL Database', and 'Blob storage'. The 'Blob storage/ADLS Gen2' option is highlighted with a mouse cursor. Below the options, there is a section for 'Input alias' with a text box containing 'input-area'. There are two radio buttons: 'Provide Blob storage/ADLS Gen2 settings manually' (unselected) and 'Select Blob storage/ADLS Gen2 from your subscriptions' (selected). Below the radio buttons, there is a 'Subscription' dropdown menu with 'Azure for Students' selected. There is a 'Storage account' dropdown menu with 'mscbarefstorage1' selected. There is a 'Container' section with two radio buttons: 'Create new' (unselected) and 'Use existing' (selected). Below the radio buttons, there is a text box containing 'myrefcontainer'. There is an 'Authentication mode' dropdown menu with 'Managed Identity: System assigned' selected. Below the dropdown menu, there is a text box containing 'Managed Identity has permissions to the selected Blob storage/ADLS Gen2.'. There is a 'Path pattern' text box with a help icon. There is a 'Date format' dropdown menu with 'YYYY/MM/DD' selected. There is a 'Time format' dropdown menu with 'HH' selected. There is a 'Partition key' text box with a help icon. There is a 'Count of input partitions' section with a slider and a text box containing '1'.

+ Add reference input ▾ ↻

Blob storage/ADLS Gen2

SQL Database

Input alias

input-area

☐ Provide Blob storage/ADLS Gen2 settings manually

☒ Select Blob storage/ADLS Gen2 from your subscriptions

Subscription

Azure for Students ▾

Storage account *

mscbarefstorage1 ▾

Container *

☐ Create new ☒ Use existing

myrefcontainer ▾

Authentication mode

Managed Identity: System assigned ▾

Managed Identity has permissions to the selected Blob storage/ADLS Gen2.

Path pattern ⓘ

Date format

YYYY/MM/DD ▾

Time format

HH ▾

Partition key ⓘ

Count of input partitions ⓘ

1

We create three such inputs for Area, Customer and ATM JSON files.

+ Add stream input ▾ + Add reference input ▾ Refresh				
Alias ↑	Source type ↑	Type ↑	Authentication mode ↑	Resource
input	Stream	Event Hub	Managed Identity	eventhubdemo (KOSTAS/eventhubdemo)
input-area	Stream	Blob storage/ADLS Gen2	Managed Identity	mscbarefstorage1
input-atm	Stream	Blob storage/ADLS Gen2	Managed Identity	mscbarefstorage1
input-customer	Stream	Blob storage/ADLS Gen2	Managed Identity	mscbarefstorage1

On the Query section of the Stream Analytic Job, we upload the respectively JSON files for area, atm and customer.

Query language docs ▾
Open in VS Code
Share feedback
Refresh

Inputs (5)

input
 input-area
 input-atm
 input-customer
 inputref

Outputs (1)

output

Functions (0)

Test query
Save query
Discard changes

```

1 SELECT
2 *
3 INTO
4 [output]
5 FROM
6 [input]

```

Input preview
Test results

No data was found for preview from 'input-area'. Make sure the input has recently received data and the cor

Table
Raw
Refresh
Select time range
Upload sample input
Download

Upload sample input from a file

We then proceed by writing the appropriate queries:

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

Inputs (5)

input
 input-area
 input-atm
 input-customer
 inputref

Outputs (1)

output

Functions (0)

Test query
Save query
Discard changes

```

1 SELECT SUM(Amount) AS Sum_of_Amount
2 FROM [input]
3 TIMESTAMP BY EventEnqueuedUtcTime
4 GROUP BY Type, ATMCode, SlidingWindow(minute, 10)
5 HAVING Type = 0 AND ATMCode = 21

```

Input preview
Test results

Showing 3 rows from 'output'.

Sum_of_Amount
bigint

43

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

Inputs (5)

Outputs (1)

Functions (0)

Test selected query

Save query

Discard changes

```

1 SELECT SUM(Amount) AS Sum_of_Amount
2 FROM [input]
3 TIMESTAMP BY EventEnqueuedUtcTime
4 GROUP BY Type, ATMCode, TumblingWindow(minute, 60)
5 HAVING Type = 1 AND ATMCode = 21

```

Input preview

Test results

Showing 1 rows from 'output'.

Sum_of_Amount
37

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

Inputs (5)

Outputs (1)

Functions (0)

Test query

Save query

Discard changes

```

1 SELECT SUM(Amount) AS Sum_of_Amount
2 FROM [input]
3 TIMESTAMP BY EventEnqueuedUtcTime
4 GROUP BY Type, ATMCode, HoppingWindow(minute, 60, 30)
5 HAVING Type = 1 AND ATMCode = 21

```

Input preview

Test results

Showing 2 rows from 'output'.

Sum_of_Amount
37

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

Inputs (5)

☐ input

☒ input-area
 ☐ input-atm
 ☐ input-customer
 ☐ inputref

Outputs (1)

☒ output

Functions (0)

Test query
 Save query
 Discard changes

```

1 SELECT ATMCode, SUM(Amount) AS Sum_of_Amount
2 FROM [input]
3 TIMESTAMP BY EventEnqueuedUtcTime
4 GROUP BY Type, ATMCode, SlidingWindow(minute, 60)
5 HAVING Type = 1
    
```

Input preview

Test results

Showing 39 rows from 'output'.

ATMCode <i>bigint</i>	Sum_of_Amount <i>bigint</i>
19	11
21	20
18	34
21	37
12	44
15	11
10	26

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

Inputs (5)

☐ input

☐ input-area
 ☐ input-atm
 ☒ input-customer
 ☐ inputref

Outputs (1)

☒ output

Functions (0)

Test query
 Save query
 Discard changes

```

1 SELECT [input-customer].[area_code] AS Area_Code, SUM([input].[Amount]) AS Sum_of_Amount
2 FROM [input]
3 TIMESTAMP BY EventEnqueuedUtcTime
4 LEFT JOIN [input-customer]
5 ON [input].[CardNumber]=[input-customer].[card_number]
6 WHERE [input].[Type] = 1
7 GROUP BY [input-customer].[area_code], TumblingWindow(minute, 60)
    
```

Input preview

Test results

Showing 7 rows from 'output'.

Area_Code <i>bigint</i>	Sum_of_Amount <i>bigint</i>
6	74
1	171
7	11
10	115
2	106
8	164
4	24

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

Inputs (5)

Outputs (1)

Functions (0)

Test query

Save query

Discard changes

```

1 SELECT [input-area].[area_city], [input-customer].[gender], SUM([input].[Amount]) AS Sum_of_Amount
2 FROM [input]
3 TIMESTAMP BY [input].[EventEnqueuedUtcTime]
4 LEFT JOIN [input-customer]
5 ON [input-customer].[card_number] = [input].[CardNumber]
6 LEFT JOIN [input-atm]
7 ON [input-atm].[atm_code] = [input].[ATMCode]
8 LEFT JOIN [input-area]
9 ON [input-area].[area_code] = [input-atm].[area_code]
10 GROUP BY [input-area].[area_city], [input-customer].[gender], TumblingWindow(minute, 60)

```

Input preview

Test results

Showing 14 rows from 'output'.

area_city <i>string</i>	gender <i>string</i>	Sum_of_Amount <i>bigint</i>
"Vancouver"	"Female"	46
"Schaumburg"	"Male"	51
"Baltimore"	"Male"	37
"Memphis"	"Male"	123
"Springfield"	"Female"	124
"Memphis"	"Female"	100
"Schaumburg"	"Female"	173

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

Inputs (5)

Outputs (1)

Functions (0)

Test query

Save query

Discard changes

```

1 SELECT 1 AS Alert, [input-customer].[last_name]
2 FROM [input]
3 TIMESTAMP BY [input].[EventEnqueuedUtcTime]
4 LEFT JOIN [input-customer]
5 ON [input-customer].[card_number] = [input].[CardNumber]
6 GROUP BY [input].[Type], [input-customer].[last_name], SlidingWindow(minute, 60)
7 HAVING [input].[Type]=1 AND COUNT(*)=2

```

Input preview

Test results

Showing 5 rows from 'output'.

Alert <i>bigint</i>	last_name <i>string</i>
1	"Mason"
1	"Stone"
1	"Fuller"
1	"Carroll"
1	"Perez"

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

Inputs (5)

input

input-area

input-atm

input-customer

inputref

Outputs (1)

output

Functions (0)

Test query Save query Discard changes

```
1 SELECT 1 AS Alert, [input-atm].[area_code] AS 'ATM_Area_Code', [input-customer].[area_code] AS 'Customer_Area_Code'
2 FROM [input]
3 TIMESTAMP BY [input].[EventEnqueuedUtcTime]
4 LEFT JOIN [input-customer]
5 ON [input-customer].[card_number] = [input].[CardNumber]
6 LEFT JOIN [input-atm]
7 ON [input-atm].[atm_code] = [input].[ATMCode]
8 WHERE [input-atm].[area_code] <> [input-customer].[area_code]
```

Input preview Test results

Showing 45 rows from 'output'.

Alert <i>bigint</i>	ATM_Area_Code <i>bigint</i>	Customer_Area_Code <i>bigint</i>
1	14	8
1	11	8
1	11	8
1	11	8
1	11	8
1	11	8
1	11	8
1	11	8