

## Sending data to Event Hub

In this lab, the goal is to establish a data pipeline for monitoring and analyzing Blob storage activity. Initially, Blob diagnostic data is sent to Azure Event Hub for real-time processing. Then, using Stream Analytics, the data is parsed and transformed into a structured format. Finally, the processed data is stored in an SQL Database for further analysis, reporting, or integration with other systems. The end goal is to gain insights into Blob storage usage patterns, optimize performance, and ensure compliance with security and regulatory requirements.

1. In this lab we want to send our Blob diagnostic data onto Azure Event Hub and then try and consume that data from our stream analytics job.
2. The first thing that we are going to do is to create an Event Hub.
3. Now you need to give your event a name then you can increase the partition count.
4. After that you need to increase the retention time to 24 hours.

### 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 \* blobhub ✓

Partition count 4 4

Retention

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

Cleanup policy Delete ▼

Retention time (hrs) \* 24 ✓

min. 1 hour, max. 24 hours (1day)

5. Then just create your Event Hub.
6. Once you have this in place, then you move to your storage account where you created your diagnostic settings. Open your blob diagnostic settings.
7. Then click on edit settings.

Refresh Feedback

Subscription \* Azure Pass - Sponsorship Resource group demo-grp Resource type Storage accounts Resource appstorage120/blob

Azure Pass - Sponsorship > demo-grp > appstorage120/blob

Diagnostic settings are used to configure streaming export of platform logs and metrics for a resource to the destination of your choice. You may create up to five different diagnostic settings to send different logs and metrics to independent destinations. [Learn more about diagnostic settings](#)

Diagnostic settings

Name	Storage account	Event hub	Log Analytics workspace	Partner solution	Edit setting
Logsettings	destinationstorage120	-	-	-	<a href="#">Edit setting</a>

+ Add diagnostic setting

Click 'Add Diagnostic setting' above to configure the collection of the following data:

- Storage Read
- Storage Write
- Storage Delete
- Transaction

8. Now in there you have to enable Stream to an event hub. Then you need to choose your event hub name and click on save.

Stream to an event hub

For potential partner integrations, [click to learn more about event hub integration](#).

Subscription

Azure Pass - Sponsorship

Event hub namespace \*

demonamespace120

Event hub name (optional) ⓘ

blobhub

Event hub policy name

RootManageSharedAccessKey

9. Once your diagnosis is complete then you have to wait for 10 mins.

10. Or what you can do is again you can create containers and upload files on them.

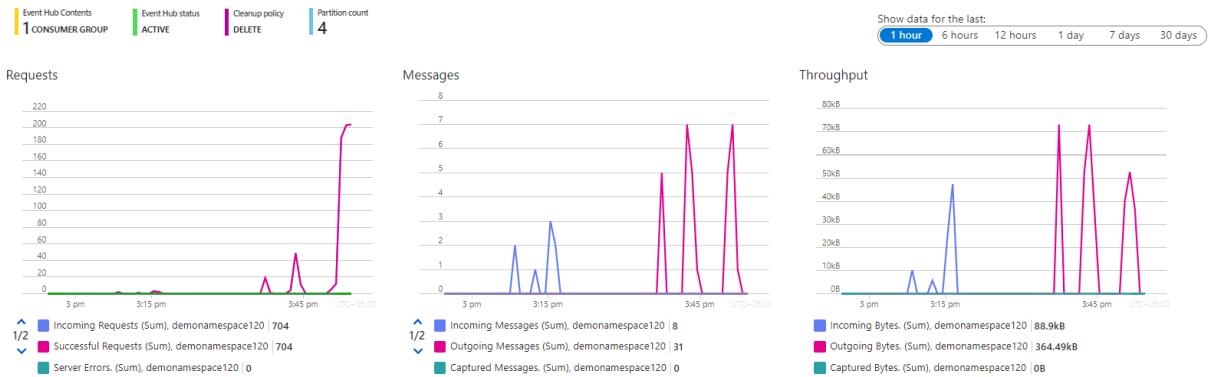
Updating diagnostics

X

Successfully updated diagnostics for 'appstorage120/blob'.

a few seconds ago

11. After some time, you need to go to your Event hub and open the blob hub event and you will see that the data was requested here.



12. Now you need to go to Stream analytics then go towards inputs and add an input for the event hub which will be the blob hub.

13. Just keep things to default and click on save.

14. After that you need to go to query and when you select blob hub you will be able to see the information. But this time it is different than the last time.

Start job Open in VS Code Diagnostic settings Refresh Query language docs Share feedback Tutorial Job ready to start

Inputs (2)

- blobhub
- demohub

Outputs (1)

- demodata1201-1

Functions (0)

Test query Save query Discard changes

```

5  */
6  SELECT
7   OrderID,Quantity,UnitPrice,DiscountCategory
8  INTO
9   [demodata1201-1]
10 FROM
11 [demohub]

```

Input preview Test results SQL table schema (preview) Job simulation (preview)

Showing sample events from 'blobhub'.

records	EventProcessedUtcTime	PartitionId	EventEnqueuedUtcTime
array	datetime	bigint	datetime
[{"time": "2024-05-09T19:00:27.239446Z", "re...}	"2024-05-09T19:20:33.4641897Z"	2	"2024-05-09T19:12:25.0080000Z"
[{"time": "2024-05-09T19:00:16.834888Z", "re...}	"2024-05-09T19:20:33.4641897Z"	2	"2024-05-09T19:12:26.2580000Z"
[{"time": "2024-05-09T18:11:26.092466Z", "re...}	"2024-05-09T19:20:33.4641897Z"	3	"2024-05-09T19:12:57.5820000Z"
[{"time": "2024-05-09T18:11:25.5278816Z", "re...}	"2024-05-09T19:20:33.4641897Z"	2	"2024-05-09T19:13:02.0080000Z"
[{"time": "2024-05-09T18:11:23.3565619Z", "re...}	"2024-05-09T19:20:33.4641897Z"	2	"2024-05-09T19:13:03.0080000Z"
[{"time": "2024-05-09T18:12:06.6839102Z", "re...}	"2024-05-09T19:20:33.4641897Z"	1	"2024-05-09T19:13:54.0340000Z"
[{"time": "2024-05-09T18:12:06.1426018Z", "re...}	"2024-05-09T19:20:33.4641897Z"	1	"2024-05-09T19:14:03.8930000Z"
[{"time": "2024-05-09T18:12:04.9069528Z", "re...}	"2024-05-09T19:20:33.4641897Z"	0	"2024-05-09T19:14:04.2510000Z"

Success

15. Here you can see that if you click on Raw then you will see the data in JSON format. Now you will also notice that the data is in the form of records. Also, this is in the form of Array.

Table Raw Refresh Select time range Upload sample input Send events Download sample data

```
1 [ {"records": [ 2 { 3 "time": "2024-05-09T19:00:27.2394467Z", 4 "resourceId": "/subscriptions/3541d15a-44aa-4f6e-a120-1b7a6d5925bf/resourceGroups/demo-grp/providers/Microsoft.Storage/storageAccount", 5 "category": "StorageWrite", 6 "operationName": "PutBlob", 7 "operationVersion": "2022-11-02", 8 "schemaVersion": "1.0", 9 "statusCode": 201, 10 "statusText": "Success", 11 "durationMs": 19, 12 "callerIpAddress": "192.140.153.27:53926", 13 "correlationId": "a6e885f4-801e-0078-4b43-a2b409000000", 14 "identity": { 15 "type": "SAS", 16 "tokenHash": "key1(1FF19981232FBF5612B952769AB23F77691A8E520623C0F7B7EDC180B2946D13),SasSignature(85E3A91EE02DBA5367F38883994F016DE", 17 }, 18 "location": "northeurope", 19 "properties": { 20 "accountName": "anstorageacc123", 21 } 22 ] } ] }
```

16. Then if we click on the test query you will see that we got all the Null values.

The screenshot shows the Azure Data Factory interface with the following details:

- Start job**, **Open in VS Code**, **Diagnostic settings**, **Refresh**, **Query language docs**, **Share feedback**, **Tutorial** buttons.
- Job status**: Job ready to start.
- Inputs (2)**:
  - `blobhub`
  - `demohub`
- Outputs (1)**:
  - `demodata1201-1`
- Functions (0)**
- Test query** pane:

```
5  */
6  SELECT
7  |    OrderID,Quantity,UnitPrice,DiscountCategory
8  INTO
9  |    [demodata1201-1]
10 FROM
11  |    [demohub]
```
- Test results** tab selected, showing an empty table preview.
- SQL table schema (preview)** and **Job simulation (preview)** tabs.
- Download results** button.
- Table Preview** (empty):

OrderID	Quantity	UnitPrice	DiscountCategory
unknown	unknown	unknown	unknown
null	null	null	null
null	null	null	null
null	null	null	null
null	null	null	null
null	null	null	null
null	null	null	null
null	null	null	null
null	null	null	null
- Footer**: Showing 110 rows from 'demodata1201-1'. Ln 1. Col 1.

## Formulating Our Query:

1. Now inside of your query editor you will write this query.

```
*/  
SELECT  
    Records.ArrayValue.time AS TimeGenerated,  
    Records.ArrayValue.resourceId AS ResourceId  
INTO  
    [demodata1201-1]  
FROM  
    [blobhub] b  
    CROSS APPLY GetArrayElements(b.records) As Records
```

▷ Test query ⌂ Save query ✎ Discard changes

```

1  /*
2  Here are links to help you get started with Stream Analytics Query Language:
3  Common query patterns - https://go.microsoft.com/fwlink/?LinkID=619153
4  Query language - https://docs.microsoft.com/stream-analytics-query/query-language-elements-azure-stream-analytics
5  */
6  SELECT
7      Records.ArrayValue.time AS TimeGenerated,
8      Records.ArrayValue.resourceId AS ResourceId
9  INTO
10     [demodata1201-1]
11    FROM
12        [blobhub] b
13        CROSS APPLY GetArrayElements(b.records) As Records

```

## 2. Then you need to click on the test query and below are the rest for that query.

Input preview    Test results    SQL table schema (preview)    Job simulation (preview)

Download results

TimeGenerated	ResourceId
"2024-05-09T19:00:27.2394467Z"	"/subscriptions/3541d15a-44aa-4f6e-a120-1b7a6d5925bf/resourceGroups/demo-grp/prov...
"2024-05-09T19:00:16.8348887Z"	"/subscriptions/3541d15a-44aa-4f6e-a120-1b7a6d5925bf/resourceGroups/demo-grp/prov...
"2024-05-09T19:00:17.0202768Z"	"/subscriptions/3541d15a-44aa-4f6e-a120-1b7a6d5925bf/resourceGroups/demo-grp/prov...
"2024-05-09T19:00:26.2784455Z"	"/subscriptions/3541d15a-44aa-4f6e-a120-1b7a6d5925bf/resourceGroups/demo-grp/prov...
"2024-05-09T19:00:26.4486803Z"	"/subscriptions/3541d15a-44aa-4f6e-a120-1b7a6d5925bf/resourceGroups/demo-grp/prov...
"2024-05-09T19:00:27.0564926Z"	"/subscriptions/3541d15a-44aa-4f6e-a120-1b7a6d5925bf/resourceGroups/demo-grp/prov...
"2024-05-09T19:00:27.4450164Z"	"/subscriptions/3541d15a-44aa-4f6e-a120-1b7a6d5925bf/resourceGroups/demo-grp/prov...
"2024-05-09T18:11:26.0924663Z"	"/subscriptions/3541d15a-44aa-4f6e-a120-1b7a6d5925bf/resourceGroups/demo-grp/prov...
"2024-05-09T18:11:25.5278816Z"	"/subscriptions/3541d15a-44aa-4f6e-a120-1b7a6d5925bf/resourceGroups/demo-grp/prov...

Showing 42 rows from 'demodata1201-1'.    Ln 12, Col 16

3. Now from above we can understand that we selected just two records to display, and it did that. But now we will select all the records in place and display them.
4. For that we will use the query shown below.

```

*/
SELECT
    Records.ArrayValue.time AS TimeGenerated,
    Records.ArrayValue.category AS Category,
    Records.ArrayValue.operationName AS OperationName,
    Records.ArrayValue.statusCode AS StatusCode,
    Records.ArrayValue.callerIpAddress AS CallerIpAddress,
    Records.ArrayValue.[identity].type AS IdentityType
INTO
    [demodata1201-1]
FROM
    [blobhub] b
    CROSS APPLY GetArrayElements(b.records) As Records

```

Test query Save query Discard changes

```

1  /*
2  Here are links to help you get started with Stream Analytics Query Language:
3  Common query patterns - https://go.microsoft.com/fwlink/?LinkID=619153
4  Query language - https://docs.microsoft.com/stream-analytics-query/query-language-elements-azure-stream-analytics
5  */
6  SELECT
7      Records.ArrayValue.time AS TimeGenerated,
8      Records.ArrayValue.category AS Category,
9      Records.ArrayValue.operationName AS OperationName,
10     Records.ArrayValue.statusCode AS StatusCode,
11     Records.ArrayValue.callerIpAddress AS CallerIpAddress,
12     Records.ArrayValue.[identity].type AS IdentityType
13  INTO
14      [demodata1201-1]
15  FROM
16      [blobhub] b
17      CROSS APPLY GetArrayElements(b.records) As Records

```

5. Now for the above query run the test and you will see the data.

Input preview Test results SQL table schema (preview) Job simulation (preview)

[Download results](#)

TimeGenerated datetime	Category string	OperationName string	StatusCode bigint	CallerIpAddress string	IdentityType string
"2024-05-09T19:00:27.2394...	"StorageWrite"	"PutBlob"	201	"192.140.153.27:53926"	"SAS"
"2024-05-09T19:00:16.8348...	"StorageRead"	"BlobPreflightRequest"	200	"192.140.153.27:53926"	"AnonymousPreflight"
"2024-05-09T19:00:17.0202...	"StorageRead"	"ListBlobs"	200	"192.140.153.27:53926"	"SAS"
"2024-05-09T19:00:26.2784...	"StorageRead"	"BlobPreflightRequest"	200	"192.140.153.27:53926"	"AnonymousPreflight"
"2024-05-09T19:00:26.4486...	"StorageRead"	"GetBlobProperties"	404	"192.140.153.27:53926"	"SAS"
"2024-05-09T19:00:27.0564...	"StorageRead"	"BlobPreflightRequest"	200	"192.140.153.27:53926"	"AnonymousPreflight"
"2024-05-09T19:00:27.4450...	"StorageRead"	"ListBlobs"	200	"192.140.153.27:53926"	"SAS"
"2024-05-09T18:11:26.0924...	"StorageRead"	"GetContainerProperties"	404	"10.0.11.28:41352"	"TrustedAccess"
"2024-05-09T18:11:25.5278...	"StorageWrite"	"CreateContainer"	201	"10.0.11.28:40472"	"TrustedAccess"

Showing 42 rows from 'demodata1201-1'. Ln 17, Col 55

## 😊 Reading Blob Diagnostic Data

1. Now to read the blob diagnostic data first we need to create a table.
2. We will create a table based on the information that we provided in the query.
3. Now you can either create a table using SSMS or just open the query editor for in your SQL Database.

## Query 1 ×

Run Cancel query Save query Export data as Show only Editor

```
1 CREATE TABLE [dbo].[BlobDiagnostics]
2 (
3     [TimeGenerated] datetime,
4     [Category] varchar(300),
5     [OperationName] varchar(300),
6     [StatusCode] int,
7     [CallerIpAddress] varchar(200),
8     [IdentityType] varchar(5000)
9 )
```

Results Messages

Query succeeded: Affected rows: 0

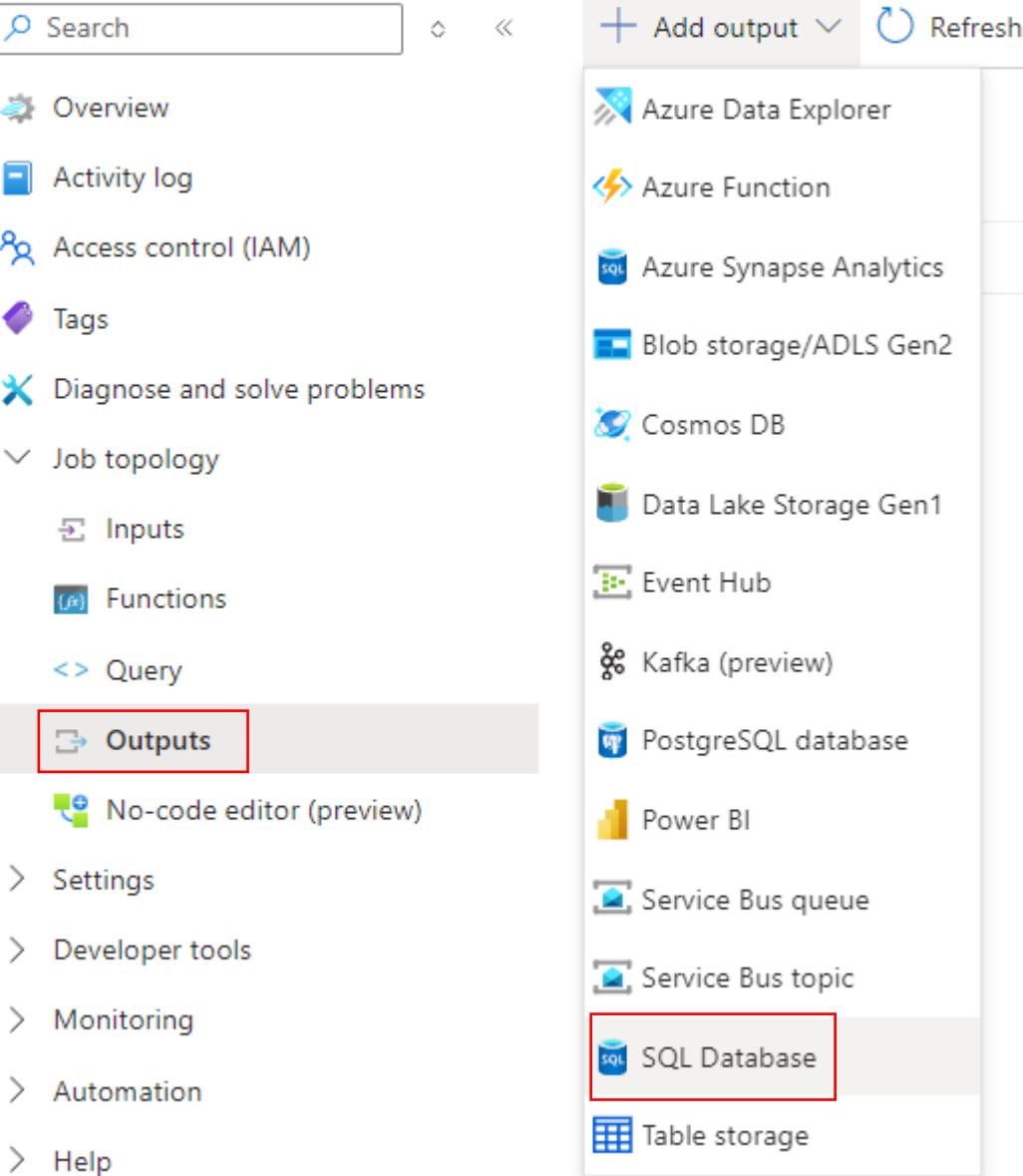
4. Once your table is created, then you need to navigate to Stream Analytics and create an output.
5. Click on Add Output and choose SQL Database.

 **demostream120 | Outputs** ☆ ...  
Stream Analytics job

⌂ ⌂ ⌂ Add output Refresh

Overview  
Activity log  
Access control (IAM)  
Tags  
Diagnose and solve problems  
Job topology  
Inputs  
Functions  
Query  
**Outputs** No-code editor (preview)  
Settings  
Developer tools  
Monitoring  
Automation  
Help

Azure Data Explorer  
Azure Function  
Azure Synapse Analytics  
Blob storage/ADLS Gen2  
Cosmos DB  
Data Lake Storage Gen1  
Event Hub  
Kafka (preview)  
PostgreSQL database  
Power BI  
Service Bus queue  
Service Bus topic  
**SQL Database** Table storage



6. Now just give the name of the table then write your username and password then click on save.

# SQL Database

X

New output

Output alias \*

demodata1201



Manual entry

Off

Subscription

Azure Pass - Sponsorship



Database \* ⓘ

demodata1201 (sqlserver120)



Table \*

BlobDiagnostics



Authentication mode

SQL server authentication



Username \*

sqladmin



Password \*

.....



7. Below you can see that output was added and the connection was successful.

 **Successful connection test** 

Connection to output 'demodata1201' succeeded.

4 minutes ago

 **Added output** 

Added output 'demodata1201' to Stream Analytics job  
'demostream120'.

4 minutes ago

8. Then go to query and use the query mentioned below.

```
SELECT
    Records.ArrayValue.time AS TimeGenerated,
    Records.ArrayValue.category AS Category,
    Records.ArrayValue.operationName AS OperationName,
    Records.ArrayValue.statusCode AS StatusCode,
    Records.ArrayValue.callerIpAddress AS CallerIpAddress,
    Records.ArrayValue.[identity].type AS IdentityType
INTO
    [BlobDiagnostics]
FROM
    [blobhub] b
CROSS APPLY GetArrayElements(b.records) AS Records
```



The screenshot shows the Azure Stream Analytics Query Editor interface. At the top, there are three buttons: 'Test query', 'Save query', and 'Discard changes'. Below these buttons, the query code is displayed in a code editor. The code is a Stream Analytics query (SAL) using T-SQL syntax. It starts with a 'SELECT' statement, followed by an 'INTO' clause pointing to '[BlobDiagnostics]', and a 'FROM' clause with a 'CROSS APPLY' clause. The code uses the 'GetArrayElements' function to process records from a blob storage container named 'blobhub'.

```
1 /*
2  Here are links to help you get started with Stream Analytics Query Language:
3  Common query patterns - https://go.microsoft.com/fwlink/?LinkID=619153
4  Query language - https://docs.microsoft.com/stream-analytics-query/query-language-elements-azure-stream-analytics
5 */
6 SELECT
7     Records.ArrayValue.time AS TimeGenerated,
8     Records.ArrayValue.category AS Category,
9     Records.ArrayValue.operationName AS OperationName,
10    Records.ArrayValue.statusCode AS StatusCode,
11    Records.ArrayValue.callerIpAddress AS CallerIpAddress,
12    Records.ArrayValue.[identity].type AS IdentityType
13 INTO
14    [BlobDiagnostics]
15 FROM
16    [blobhub] b
17    CROSS APPLY GetArrayElements(b.records) AS Records
```

9. Then first you need to test the query, save this query, and go back to the overview of stream analytics.

Test query Save query Discard changes

```

6   SELECT
7       Records.ArrayValue.time AS TimeGenerated,
8       Records.ArrayValue.category AS Category,
9       Records.ArrayValue.operationName AS OperationName,
10      Records.ArrayValue.statusCode AS StatusCode,
11      Records.ArrayValue.callerIpAddress AS CallerIpAddress,
12      Records.ArrayValue.[identity].type AS IdentityType
13  INTO

```

Input preview Test results SQL table schema (preview) Job simulation (preview)

Download results

TimeGenerated datetime	Category string	OperationName string	StatusCode bigint	CallerIpAddress string	IdentityType string
"2024-05-11T08:42:11.70459..."	"StorageRead"	"BlobPreflightRequest"	200	"192.140.153.206:62287"	"AnonymousPreflight"
"2024-05-11T08:42:11.89741..."	"StorageRead"	"ListBlobs"	200	"192.140.153.206:62287"	"SAS"
"2024-05-11T08:43:25.19135..."	"StorageRead"	"BlobPreflightRequest"	200	"192.140.153.206:62356"	"AnonymousPreflight"
"2024-05-11T08:43:25.197664..."	"StorageRead"	"BlobPreflightRequest"	200	"192.140.153.206:62356"	"AnonymousPreflight"
"2024-05-11T08:43:26.50234..."	"StorageRead"	"BlobPreflightRequest"	200	"192.140.153.206:62356"	"AnonymousPreflight"
"2024-05-11T08:43:26.93817..."	"StorageRead"	"BlobPreflightRequest"	200	"192.140.153.206:62356"	"AnonymousPreflight"
"2024-05-11T08:43:27.55103..."	"StorageRead"	"BlobPreflightRequest"	200	"192.140.153.206:62356"	"AnonymousPreflight"
"2024-05-11T08:43:27.73403..."	"StorageRead"	"BlobPreflightRequest"	200	"192.140.153.206:62356"	"AnonymousPreflight"

Showing 46 rows from 'BlobDiagnostics'. Ln 17, Col 55

10. From there you are going to start this, Job.

11. Then wait for 10-15 minutes.

12. Go back to the Query editor in the SQL database OR use SSMS and run the select statement for your table.

13. Below you can see that data in place.

165 %

Results Messages

```
SELECT * FROM [dbo].[BlobDiagnostics]
```

TimeGenerated	Category	OperationName	StatusCode	CallerIpAddress	IdentityType
2024-05-11 08:35:28.520	StorageWrite	PutBlob	201	192.140.153.206:61836	SAS
2024-05-11 08:35:28.280	StorageRead	BlobPreflightRequest	200	192.140.153.206:61836	AnonymousPreflight
2024-05-11 08:35:26.487	StorageRead	GetBlobProperties	404	192.140.153.206:61836	SAS
2024-05-11 08:35:27.230	StorageRead	BlobPreflightRequest	200	192.140.153.206:61836	AnonymousPreflight
2024-05-11 08:35:28.770	StorageRead	ListBlobs	200	192.140.153.206:61836	SAS
2024-05-11 08:35:28.050	StorageRead	BlobPreflightRequest	200	192.140.153.206:61836	AnonymousPreflight
2024-05-11 08:35:32.250	StorageRead	GetBlob	200	192.140.153.206:61836	SAS
2024-05-11 08:35:40.803	StorageRead	BlobPreflightRequest	200	192.140.153.206:62197	AnonymousPreflight
2024-05-11 08:40:55.007	StorageRead	GetBlobServiceProperties	403	192.140.153.206:62197	OAuth
2024-05-11 08:40:55.600	StorageRead	BlobPreflightRequest	200	192.140.153.206:62197	AnonymousPreflight
2024-05-11 08:40:55.797	StorageRead	ListBlobs	200	192.140.153.206:62197	SAS
2024-05-11 08:41:1.703	StorageRead	BlobPreflightRequest	200	192.140.153.206:62287	AnonymousPreflight
2024-05-11 08:42:1.897	StorageRead	ListBlobs	200	192.140.153.206:62287	SAS
2024-05-11 08:43:25.190	StorageRead	BlobPreflightRequest	200	192.140.153.206:62356	AnonymousPreflight
2024-05-11 08:43:25.877	StorageRead	BlobPreflightRequest	200	192.140.153.206:62356	AnonymousPreflight
2024-05-11 08:43:26.503	StorageRead	BlobPreflightRequest	200	192.140.153.206:62356	AnonymousPreflight
2024-05-11 08:43:26.997	StorageRead	BlobPreflightRequest	200	192.140.153.206:62356	AnonymousPreflight
2024-05-11 08:43:27.559	StorageRead	BlobPreflightRequest	200	192.140.153.206:62356	AnonymousPreflight
2024-05-11 08:43:27.773	StorageRead	BlobPreflightRequest	200	192.140.153.206:62356	AnonymousPreflight
2024-05-11 08:43:28.153	StorageRead	BlobPreflightRequest	200	192.140.153.206:62356	AnonymousPreflight
2024-05-11 08:43:28.000	StorageWrite	PutBlob	201	192.140.153.206:62356	SAS
2024-05-11 08:43:26.659	StorageWrite	PutBlob	201	192.140.153.206:62356	SAS
2024-05-11 08:43:26.273	StorageWrite	PutBlob	201	192.140.153.206:62356	SAS
2024-05-11 08:43:26.710	StorageWrite	PutBlob	201	192.140.153.206:62356	SAS

Query executed successfully. sqlserver120.database.window... sqladmin (33) | database120 | 00:00:00 | 57 rows