|  |  |
| --- | --- |
|  |  |
|  | Cortana Intelligence  Suite  Lab CIS010  Real-Time Analytics  with Azure  Stream Analytics |

Contents

[Overview 3](#_Toc455054442)

[Create and Configure an Event Hub 5](#_Toc455054443)

[Create and Configure a Stream Analytics Job 9](#_Toc455054444)

[Simulate an Event Hub Data Stream 19](#_Toc455054445)

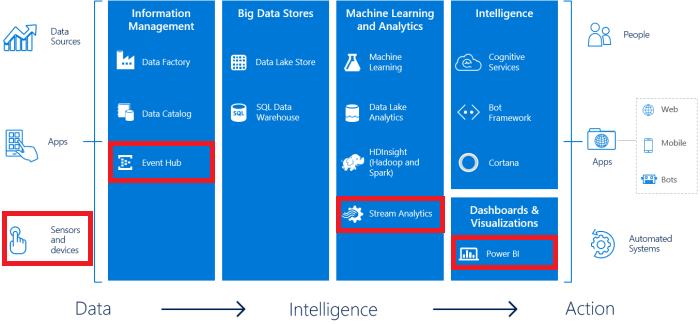
[Visualize your Streaming Data in Power BI 24](#_Toc455054446)

[Terms of Use 27](#_Toc455054447)

|  |
| --- |
| Overview |

### Summary

### This lab provides an introduction to Azure Event Hubs and processing streaming data in real-time with Azure Stream Analytics. An Event Hub is a publish-subscribe service that handles high volume and high velocity data streams. Stream Analytics allows real-time analytics for streaming data.



In this lab, you will configure an Event Hub and use PowerShell to send streaming sample data to it. You will create a Stream Analytics job, use the Event Hub as input, and configure Power BI as an output. Finally, you will visualize an aggregated view of your sample data stream in Power BI.

### Business Case

“How do freeway speeds differ across locations and times of day?”

In this lab, you will use a simulated sensor dataset consisting of time and speed readings for three locations. Ten minutes of randomly generated readings ranging from 65 miles per hour to 85 miles per hour will be sent to an Event Hub and aggregated in a five second window using Stream Analytics. You will also visualize the data in real-time in Power BI to determine how the average speed may differ between locations for that timespan.

### Learning Objectives

Upon completing this lab, you will have hands-on experience with the following functions and concepts related to Azure Event Hubs, Stream Analytics and Cortana Intelligence:

* Creating an Event Hub using the Azure Management Portal
* Configuring an Access Policy for an Event Hub
* Creating a Stream Analytics job using the Azure Management Portal
* Configuring input and output for a Stream Analytics job
* Creating a Stream Analytics query
* Sending data to an Event Hub using PowerShell
* Visualizing real-time data in Power BI

### Lab Requirements/Prerequisites

* An Azure subscription is required to create and use Azure Stream Analytics and Event Hubs. If you do not have an Azure Subscription, you can create one with a free credit at the following link:

<https://azure.microsoft.com/en-us/free/>

* A Power BI account is required. If you do not yet have access to Power BI, you can sign up at the following link:

<http://www.powerbi.com/>

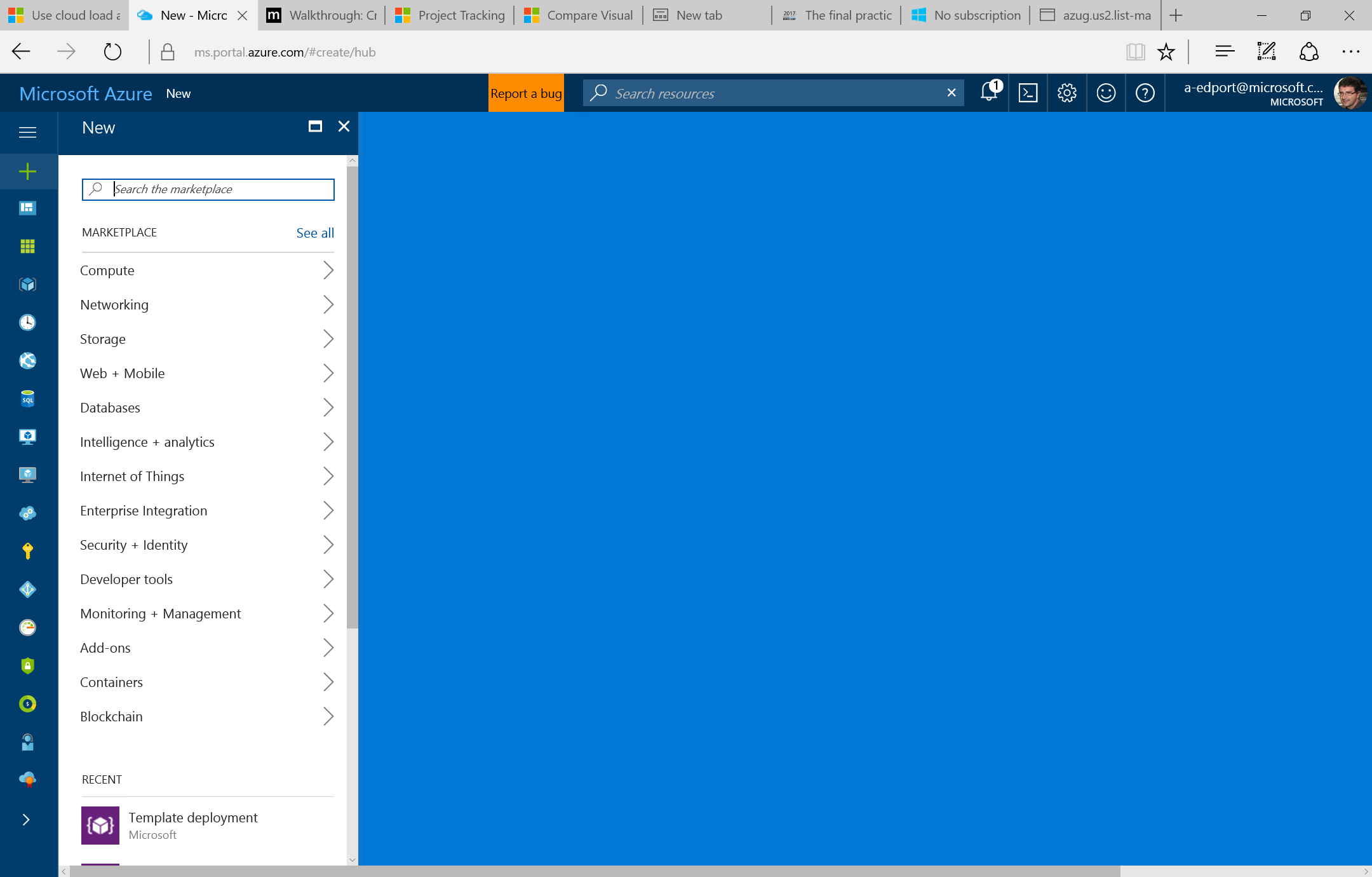
* Windows PowerShell is required to run a local script file. PowerShell should be enabled by default on your computer, but you may need to turn it on in the Control Panel by going to *Programs and Features* and enabling the PowerShell feature.

|  |
| --- |
| Create and Configure an Event Hub |

### Create an Event Hub within a Service Bus Namespace

To get started, you will need to log in to the Azure Management Portal. This differs from other labs, which use the new Portal. Functionality for Event Hubs is not yet present in the new Portal, and Stream Analytics configuration is still limited. As a result, this lab will use the legacy Portal to create and configure both an Event Hub and Stream Analytics job.

1. Open Microsoft Edge or another preferred browser and go to the Azure Management Portal: <http://manage.windowsazure.com>.
2. If you are not automatically signed in to the Management Portal, sign in using the Microsoft Account that is associated with your Azure Subscription.
3. Click on (+) NEW in the toolbar at the bottom.



1. Select Internet of Things, then select EVENT HUB.



1. Use the following information for your new event hub:

A warning will appear if the Event Hub Name or Namespace Name are not available, in which case you will need to enter an alternative name.

* 1. A unique *Event Hub Name*. ASALabEventHub1 is the sample event hub name that will appear in references throughout this lab.
  2. Select an appropriate *Region*, or use the region that is geographically closest to you
  3. Verify the *Subscription* value
  4. Use the default *Namespace Name*, which defaults to your Event Hub Name with the addition of “-ns”

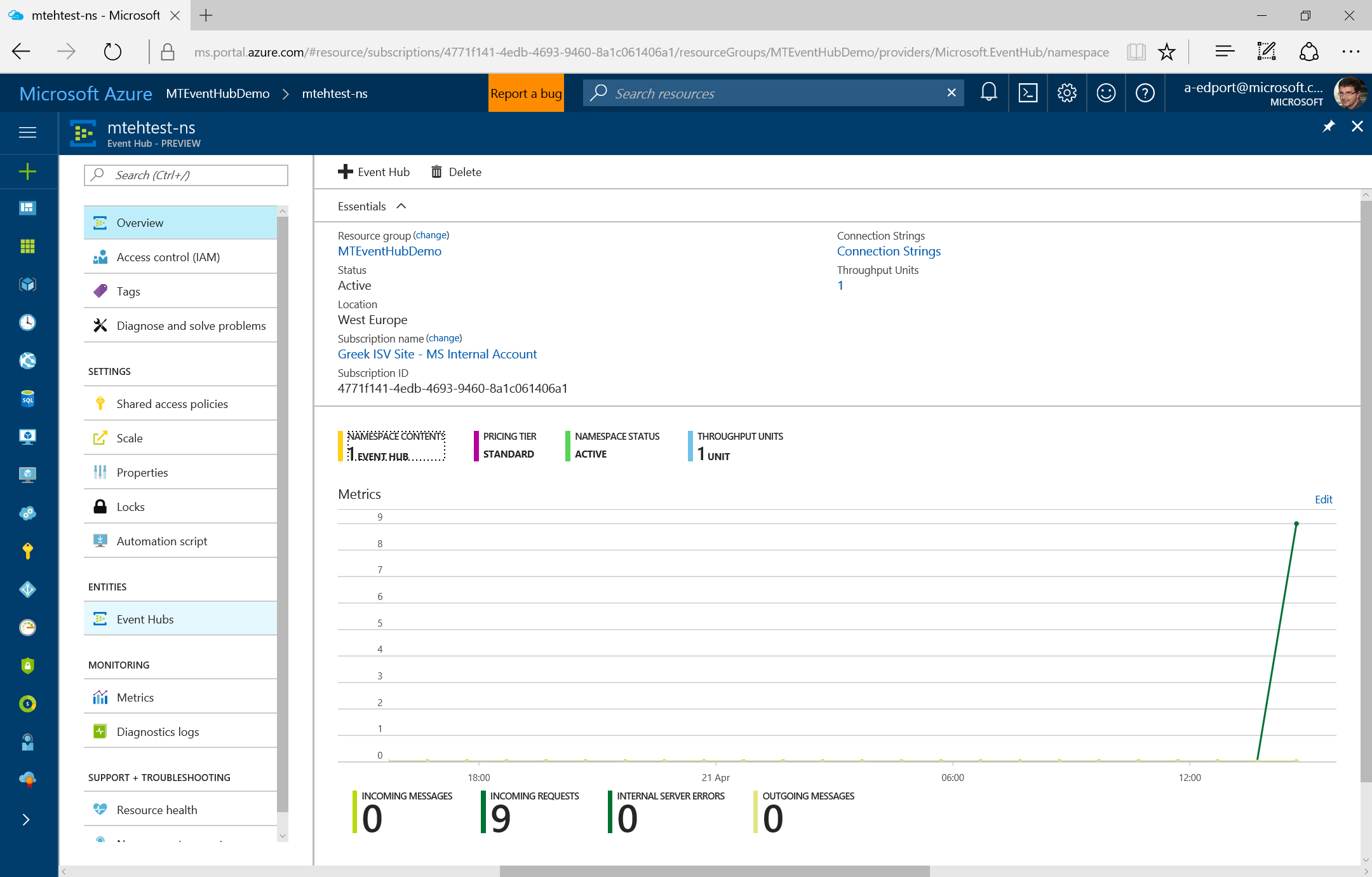


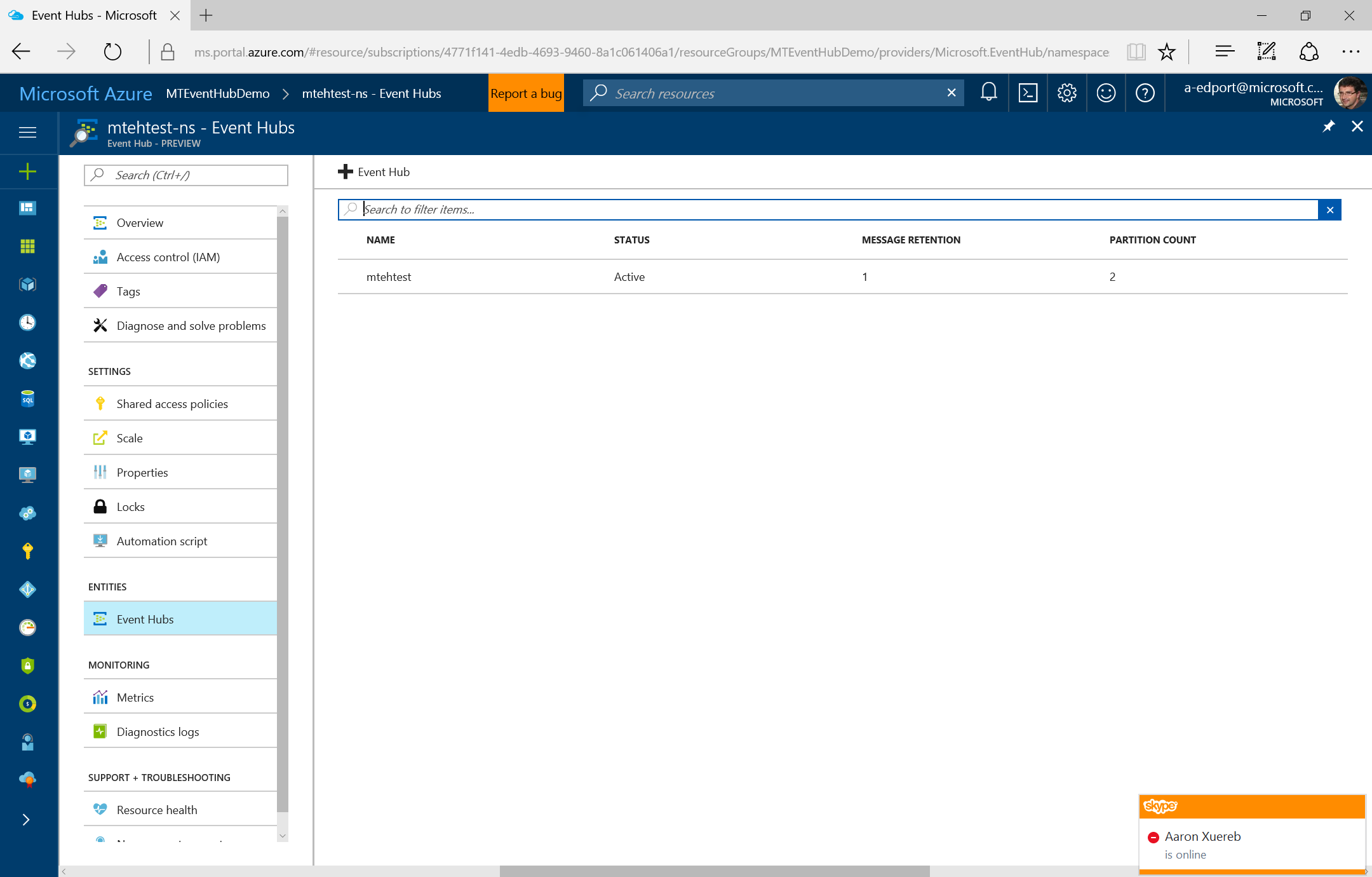
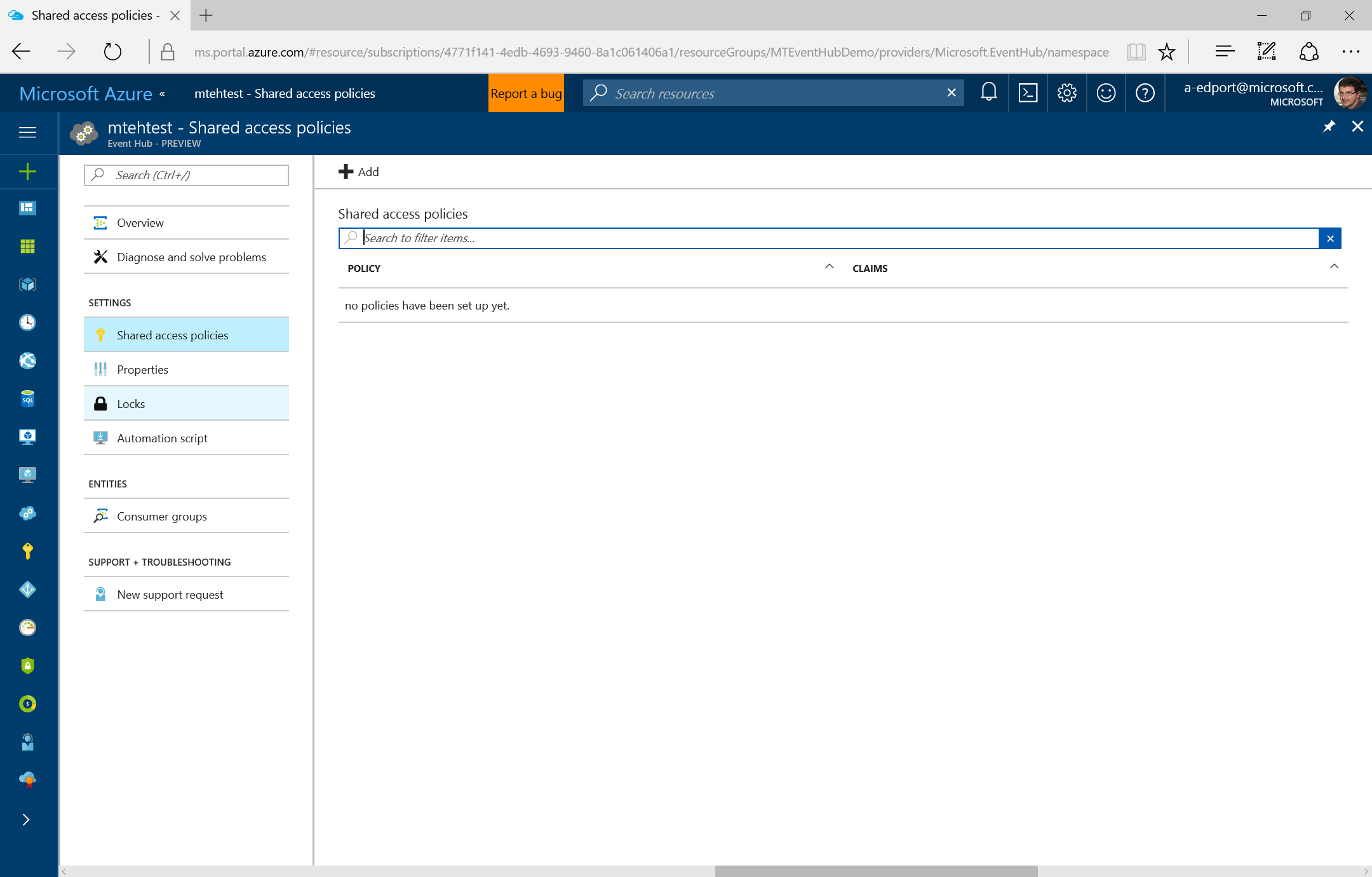
1. Click CREATE when ready.
2. It should take less than one minute to create the event hub, during which you should see the following notification.

### Configure an Event Hub

Once your event hub is ready, you should see your namespace listed in the *Service Bus* menu. You will now configure an access policy so that you can send data to your event hub in a later lab activity.

1. Click on the Events Hub namespace.
2. Select EVENT HUBS from the menu at the side of your service bus namespace. Note that EVENT HUBS may not appear initially. You may need to click ALL or refresh your browser.



1. Click on the name listed in the event hub window. 
2. Select Shared access policies from the left side of the event hub dashboard.

Shared Access Signature (SAS) is the authentication mechanism for Event Hubs. In a later activity, you will generate a SAS token and send data to your Event Hub.

1. Click *Add* under *Shared access policies*, and enter Lab as the policy *NAME.* Click the menu arrow under *PERMISSIONS* and then check the box next to Manage. After selecting the manage option, the Send and Listen options will be checked automatically.
2. When ready, click the Save changes icon in the bottom menu. You will see a brief message stating that your event hub is being updated.

IMPORTANT: You will need to know your Service Bus Namespace, Event Hub Name, Policy Name, and Primary Key in order to send data to your Event Hub.

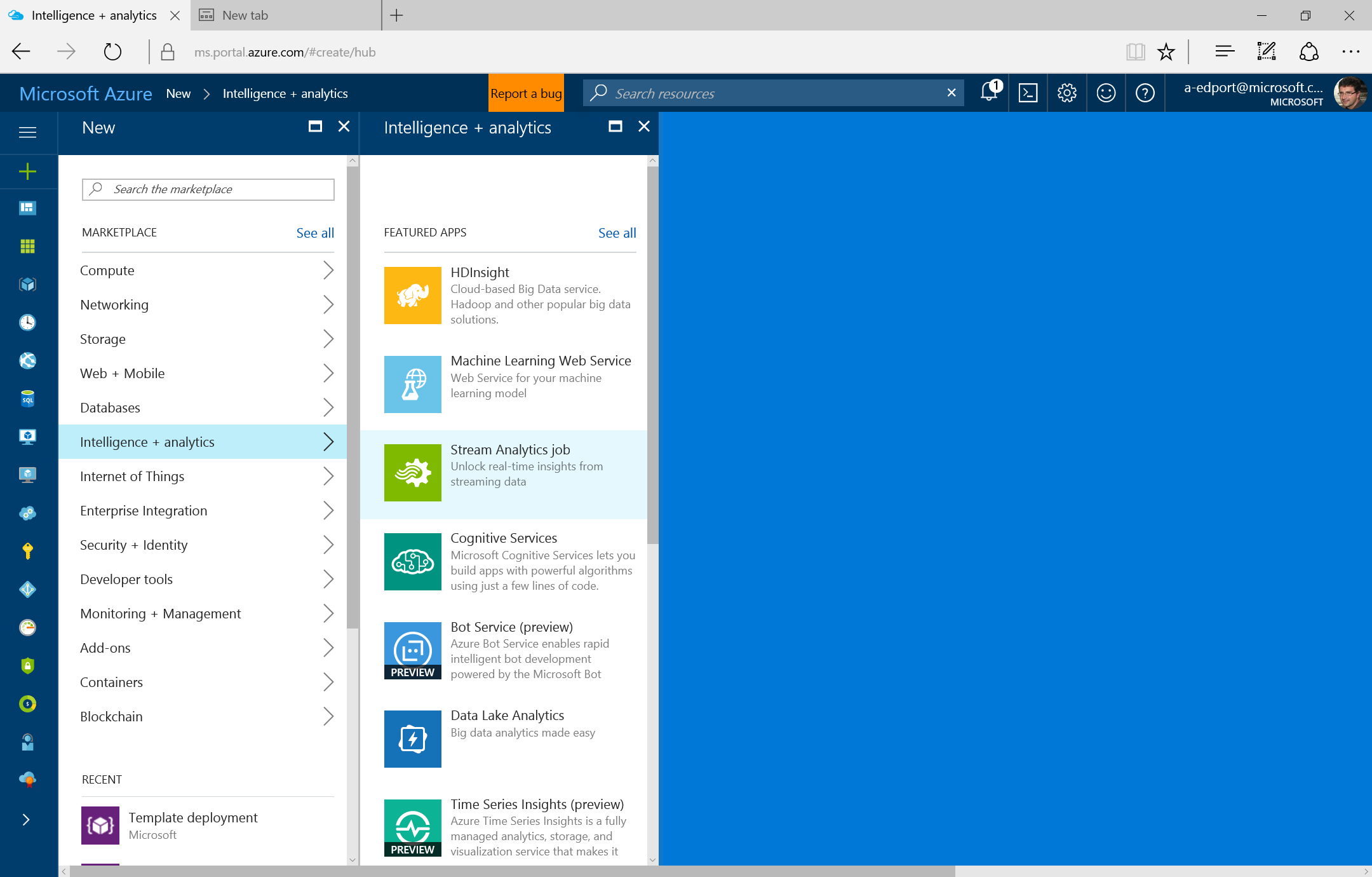
1. Scroll down after the event hub updates and locate the *shared access key generator* section. Retain the *POLICY NAME* of Lab and the *PRIMARY KEY* value by copying and pasting them into a text editor for later use.

|  |
| --- |
| Create and Configure a Stream Analytics Job |

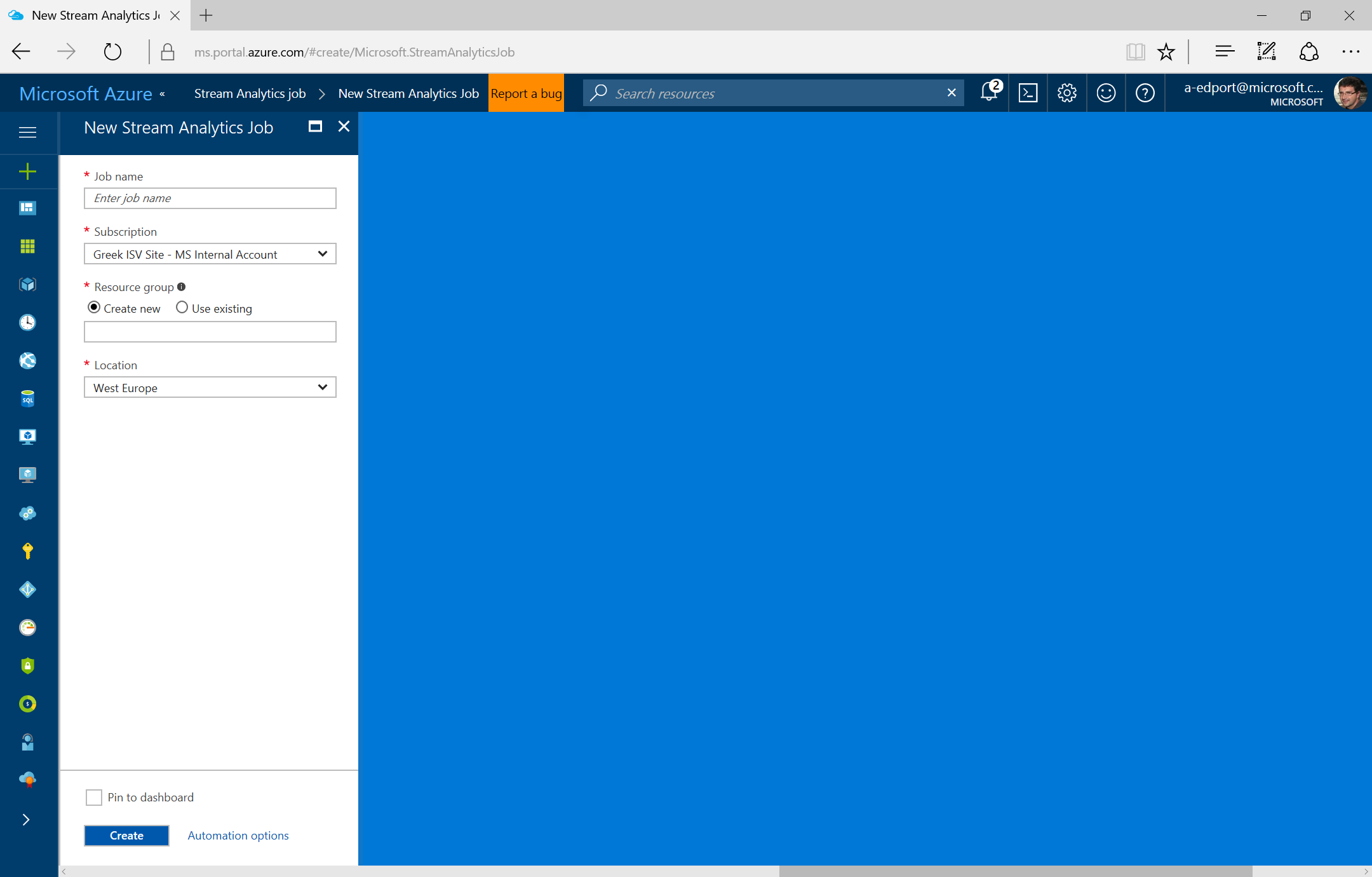
### Create a Stream Analytics Job

With an Event Hub in place and ready to accept a data stream, you will now create a Stream Analytics job that will define your Event Hub as input, query the data using a modified SQL language, and output the query results to an output sink. It is possible to create a Stream Analytics job in the new Portal, but you will do so in the legacy Management Portal for consistency with the Event Hub creation as well as the use of Power BI as an output sink.

1. Click on (+) in the left hand bar.



1. Select Intelligence + analytics, then select STREAM ANALYTICS job.
2. In the *STREAM ANALYTICS* menu, select CREATE.
3. Use the following details for your new Stream Analytics job:
   1. *Job Name* (MTLabJob is the sample job name that will appear in references throughout this lab)
   2. Select an appropriate *Region*, or use the region that is geographically closest to you
   3. Verify the *Subscription* value, which appears by default

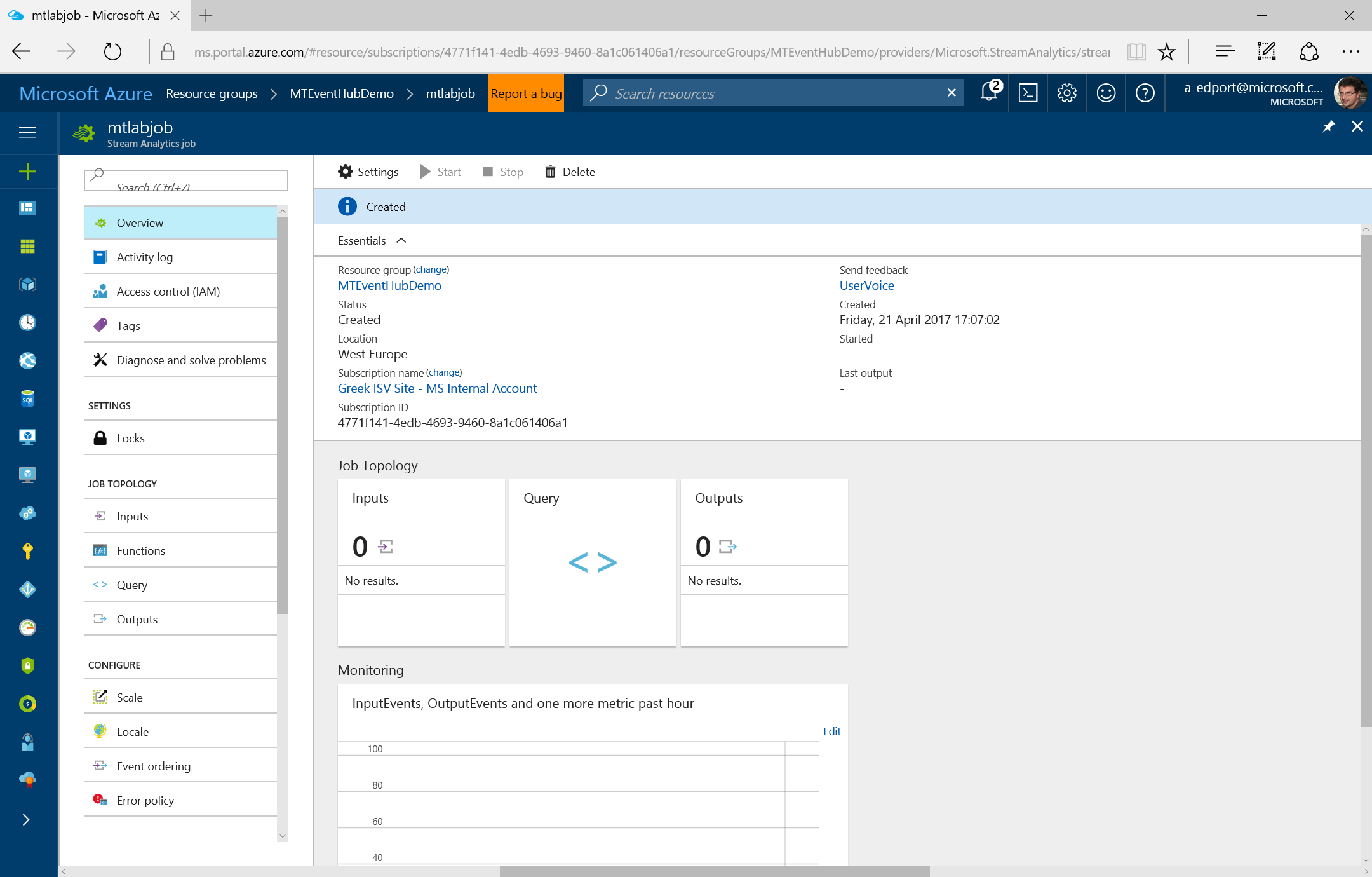


1. Click *Create*

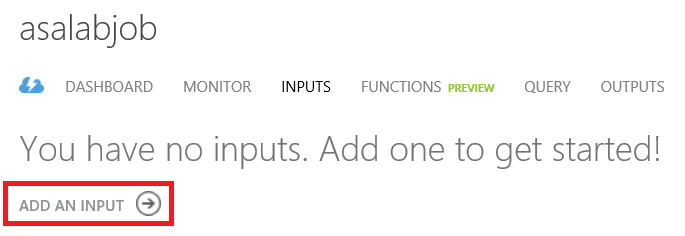
### Define Job Input

Next, you will define your Event Hub as the input for your Stream Analytics job.

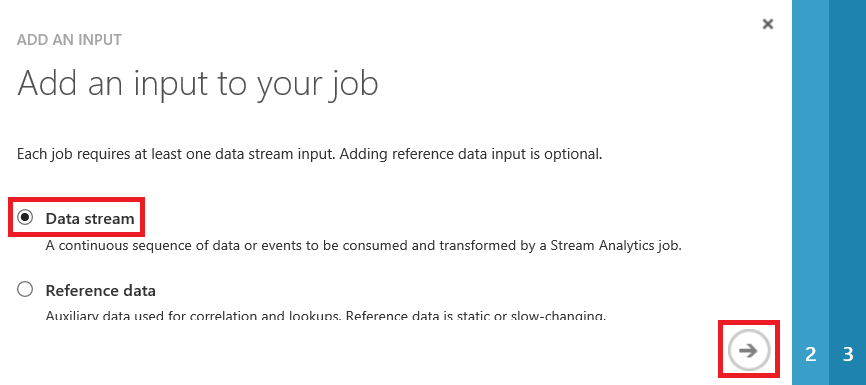
1. Once your Stream Analytics job is ready, you should see it in the job list. Click on the job name.



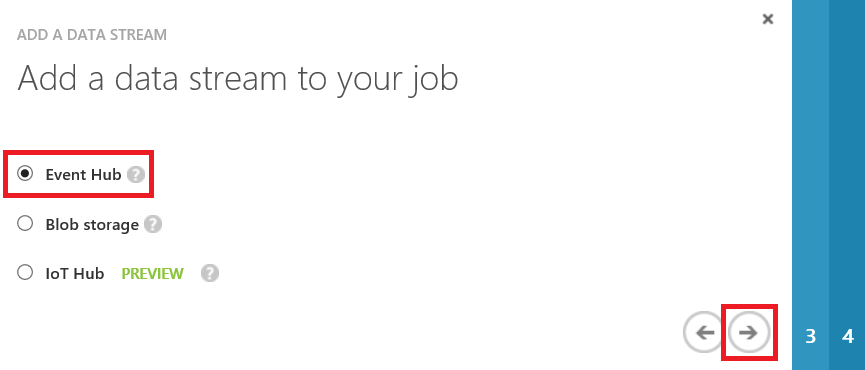
1. Select INPUTS from the left menu of your Stream Analytics job screen.
2. Select ADD



1. On the *Add an input* *to your job* setup screen, leave Data stream as the default since you will query streaming data from your event hub. While it is outside the scope of this lab, it is also possible to use Reference Data from a storage account to blend with streaming input for lookups. Click the arrow button when ready to progress to screen 2.



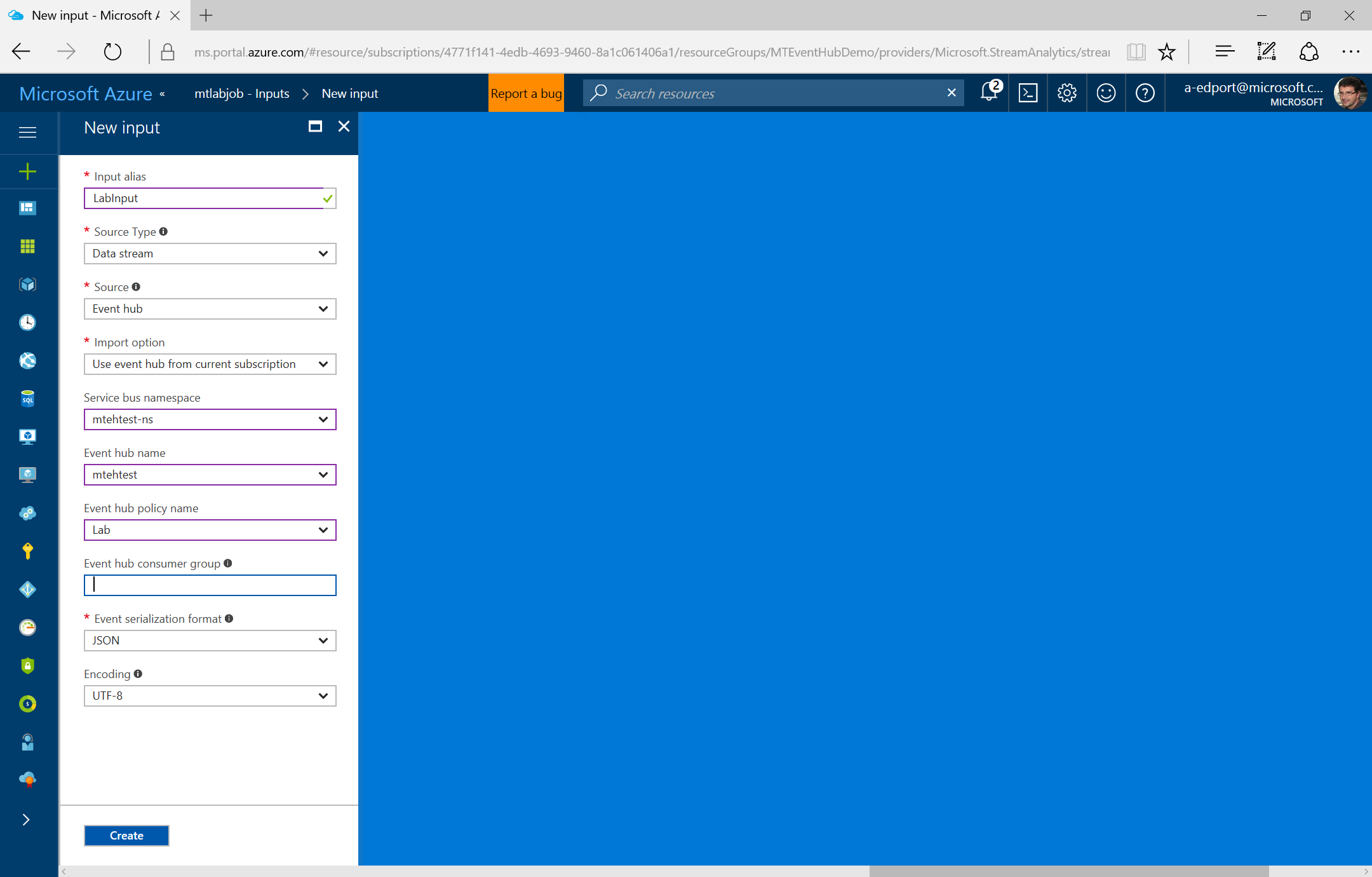
1. On the *Add a data stream to your job* screen, leave Event Hub as the default. Note the additional inputs available. Click the arrow button when ready to progress to screen 3.



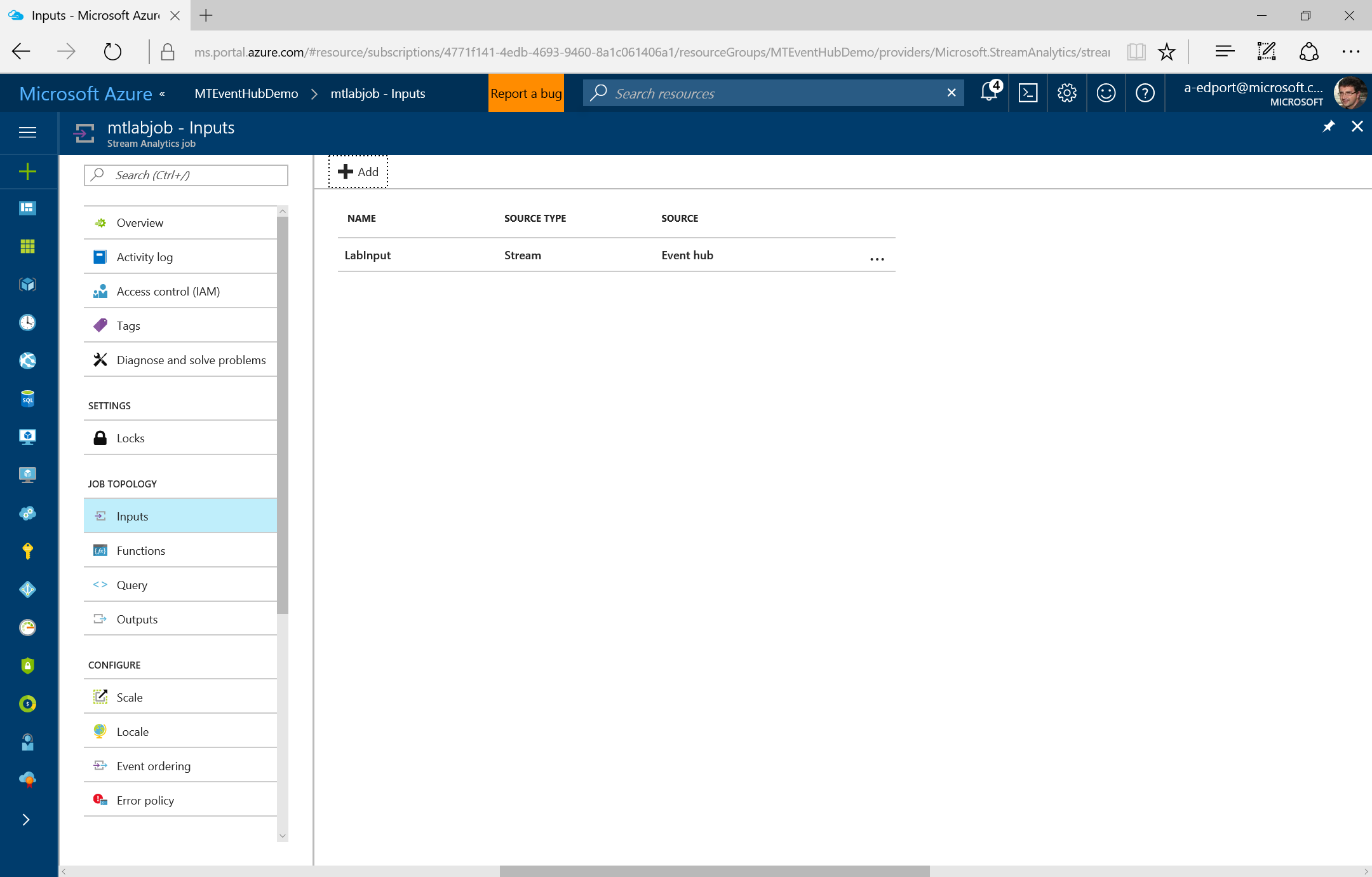
For more detail on how Event Hubs differ from Azure IoT Hub, see the Azure documentation:

<https://azure.microsoft.com/documentation/articles/iot-hub-compare-event-hubs/>

1. On the *Event Hub settings* screen, perform the following steps:
   1. Enter LabInput as the *INPUT ALIAS.*
   2. Leave the default *SUBSCRIPTION* value as Use Event Hub from Current Subscription. Unless there are other event hubs that are associated with your subscription, the remaining defaults on this screen should correspond to the event hub that you previously created for this lab. If you do not see the correct event hub information, however, select it as needed.
   3. Modify the *EVENT HUB POLICY NAME* from the default of *RootManagedSharedAccessKey* to the Lab policy that you created.
   4. Click the arrow button when ready.



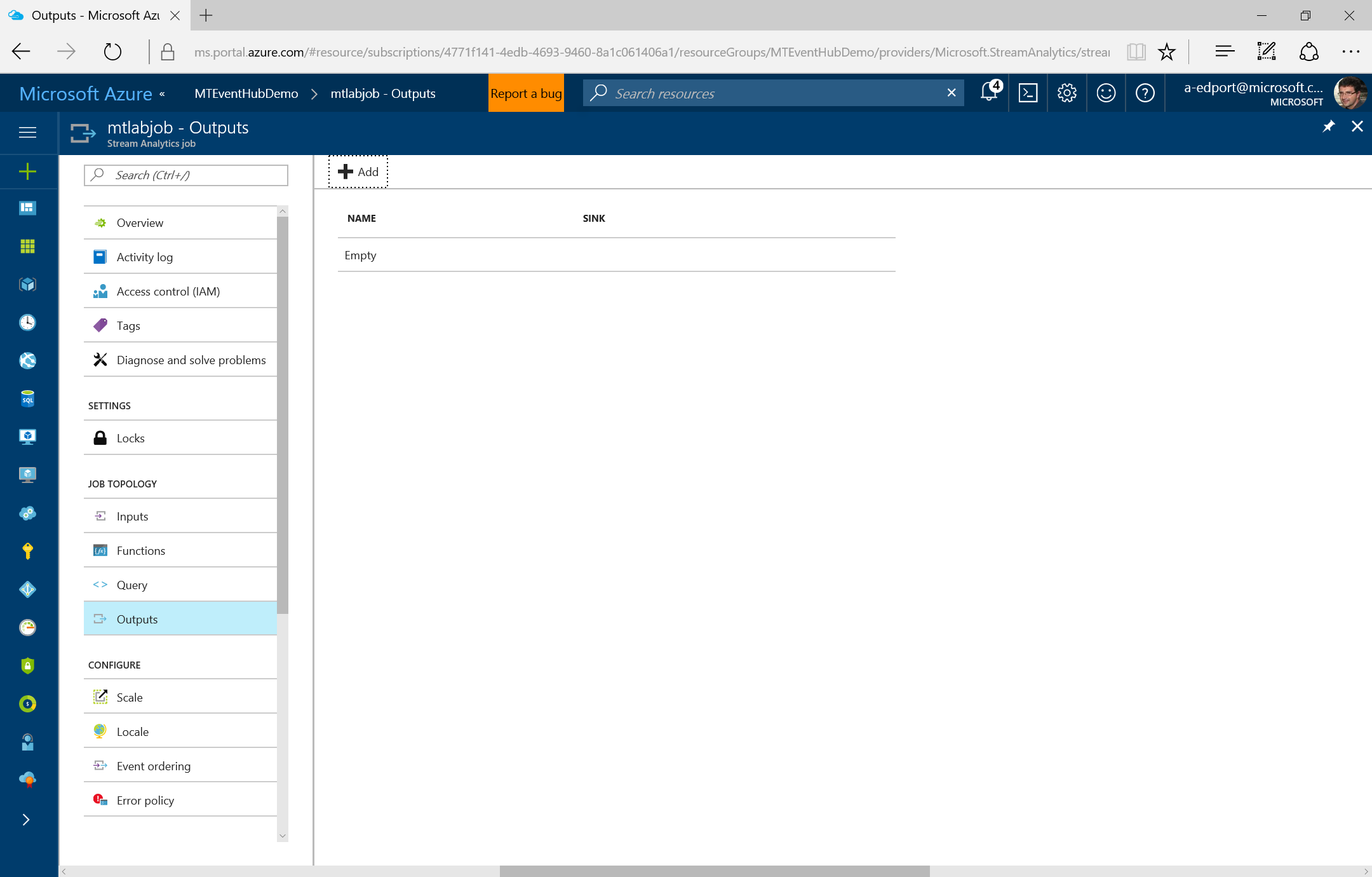
1. For the Event *Serialization settings*, leave the default values and click the Create button to finish defining your job input
2. Your input data stream labeled LabInput should now appear in the list.



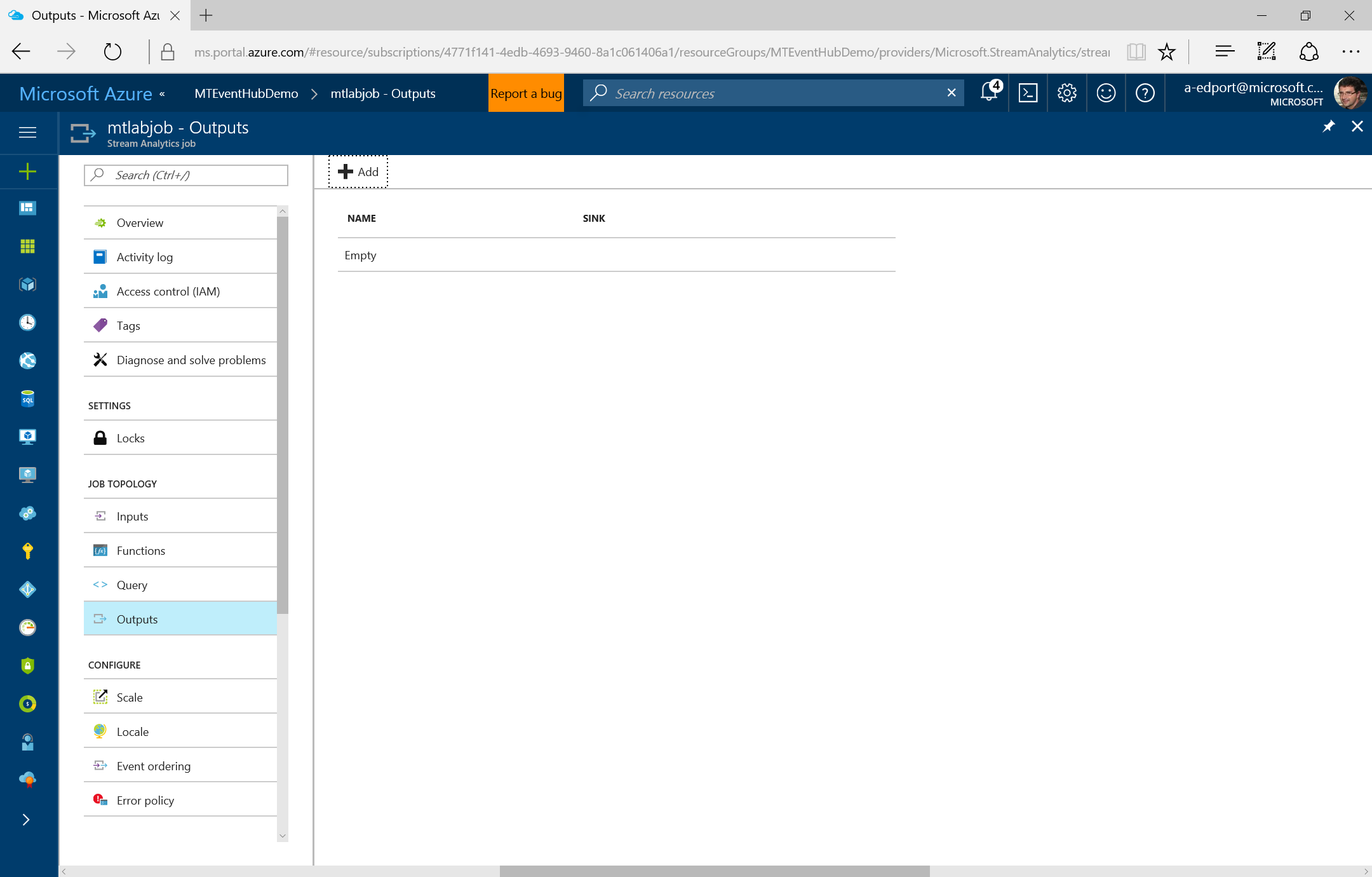
### Define Job Output

Next, you will define an output sink for the results of the Stream Analytics job. There are numerous available sinks, but you will use Power BI for this lab.

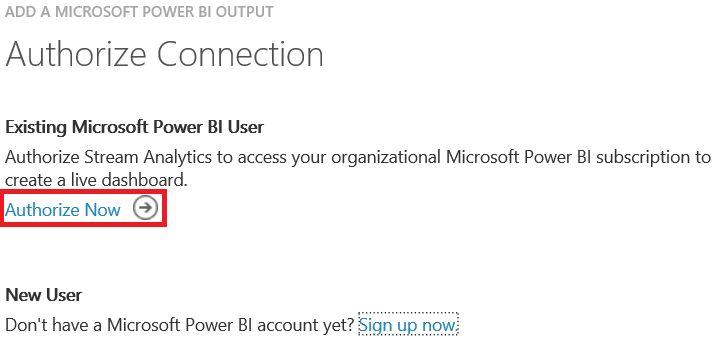
1. Select OUTPUTS from the menu at the left sid of your Stream Analytics job screen.



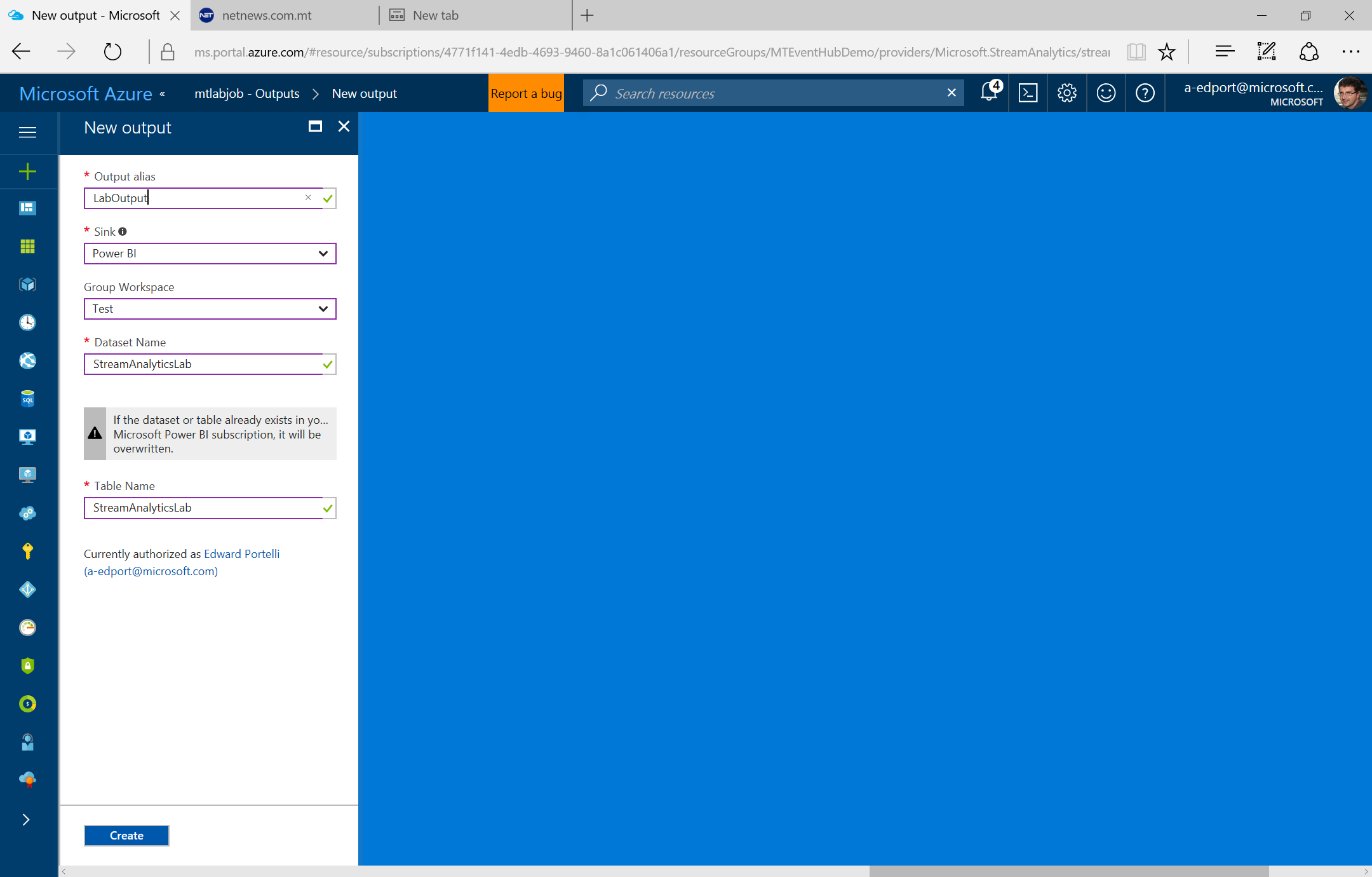
1. Select ADD at the top of the list.



1. On the *Add an output to your job* screen, note the variety of output sinks available. Select Power BI and then click the arrow button when ready to progress to screen 2.
2. On the *Authorize Connection* screen, select Authorize Nowand login with your Power BI account. If you do not already have an account, you should see an option to sign up on the window as well.



1. After logging in, you should see a *Microsoft Power BI Settings* screen.
   1. Enter LabOutput under *OUTPUT ALIAS*.
   2. Enter StreamAnalyticsLab under *DATASET NAME.*
   3. Enter StreamAnalyticsLab under *TABLE NAME*.
   4. Click the check / OK button when ready.



1. Your Power BI output sink labeled LabOutput should now appear in the list.

### Define a Query and Start your Job

Stream Analytics uses a specialized query language that is closely related to SQL. While maintaining a basic SELECT / FROM / WHERE format, it also includes concepts such as time windows to manage and aggregate high velocity input. In your job, you will use what is called a *tumbling window* to aggregate data over time. Tumbling windows are sequential blocks of time that do not overlap. Any events occurring within the same window are grouped together.

There are three window options:

1) Tumbling

2) Hopping

3) Sliding

For more detail on windows, see the Azure documentation: <https://msdn.microsoft.com/en-us/library/azure/dn835019.aspx>



1. Select QUERY from the left menu your Stream Analytics job screen.
2. Delete any existing code in the *query* window, then copy/paste the following code into the *query* window.

SELECT SensorId,

Location,

System.TimeStamp AS Time,

AVG(Speed) AS Speed

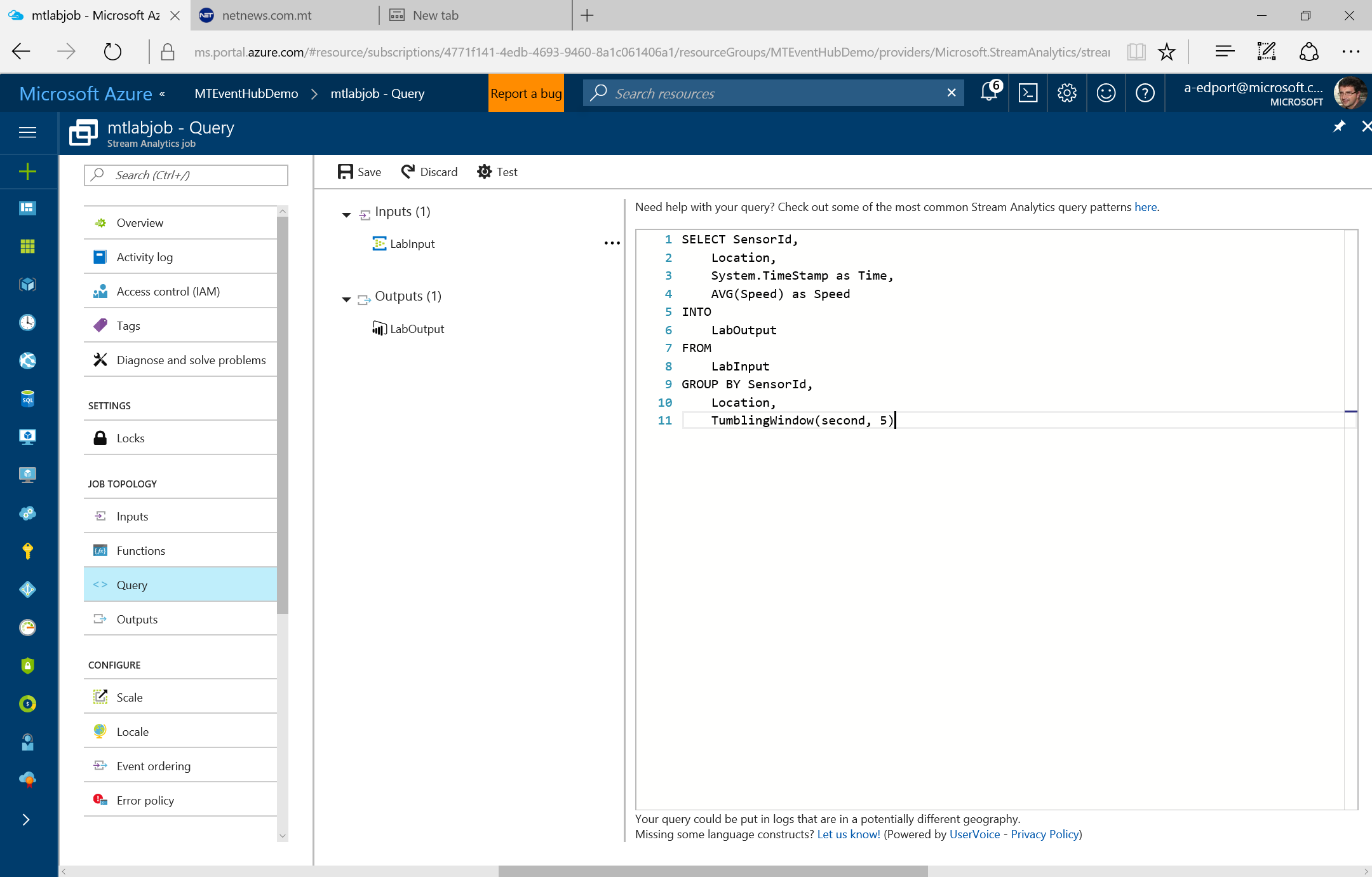
INTO LabOutput

FROM LabInput

GROUP BY SensorId,

Location,

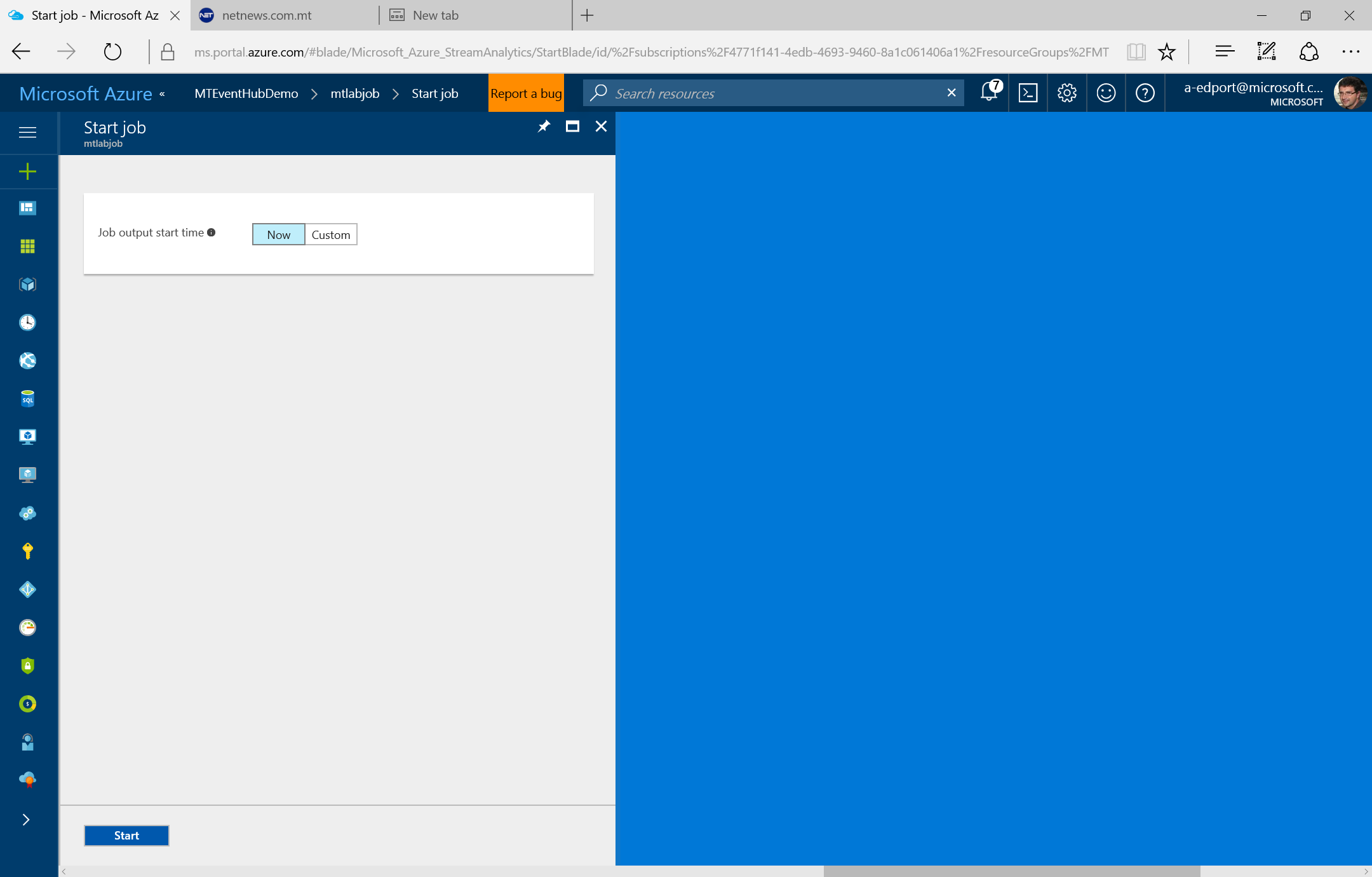
TumblingWindow(second, 5)



1. Click the SAVE button on the top toolbar and click Yes if prompted to confirm.

By default, your job will start with one “streaming unit”, which roughly corresponds to 1 MB per second of throughput. The data stream for this lab only requires one streaming unit, but if you wish to increase the processing power for jobs in production scenarios, you can do so by clicking on the **SCALE** option on the top menu and increasing your units.

1. Click the START button on the bottom toolbar to start your Stream Analytics job.
2. On the *START Job* window that appears, leave the default start time as Now and click the check / OK button in the bottom right corner.



1. It will take a few minutes to start your Stream Analytics job, during which you should see the following notification. Once the job has started, the notification will change, and you can move on to the next activity.

|  |
| --- |
| Simulate an Event Hub Data Stream |

### Simulate Event Hub Input with PowerShell

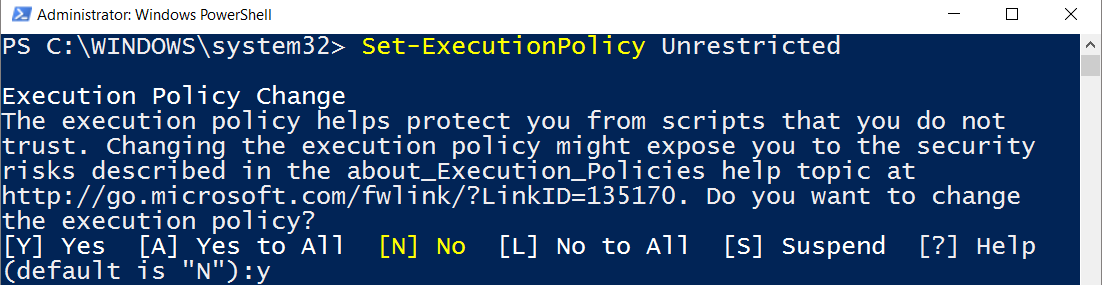
You will now use PowerShell to send events to your Event Hub. In the real world, you may have numerous devices sending data to an Event Hub. For this lab, however, the PowerShell script simulates three sensors sending randomly generated data at five second intervals.

1. Open Windows PowerShell as an administrator (right-click and select **Run as administrator**).



1. Set the execution policy to unrestricted. If prompted at any point with a message, select **Y / Yes** to proceed. When a step completes, you will see the command prompt again (i.e. “C:\[…]>”).

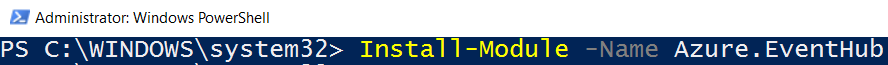
Set-ExecutionPolicy Unrestricted



In PowerShell, you may receive additional security prompts for various steps. Throughout this lab, continue to select **Y / Yes** to proceed.

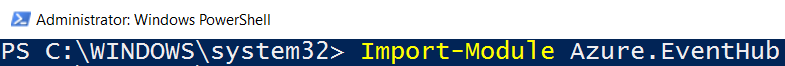
1. Install the Event Hub module from the PowerShell Gallery.

Install-Module –Name Azure.EventHub



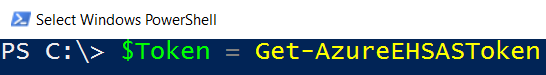
1. Import the Event Hub module.

Import-Module –Name Azure.EventHub



1. Type or copy/paste the following code in sequence to obtain a shared access signature (SAS) token for your Event Hub.
   1. Enter the PowerShell cmdlet to obtain your event hub token, then hit Enter.

$Token = Get-AzureEHSASToken



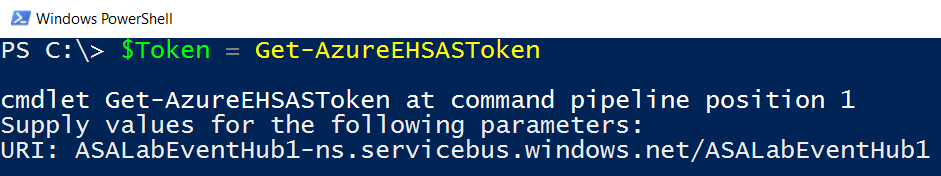
* 1. Enter your event hub URI and then hit Enter (replace this sample value with your own service bus namespace and event hub name).

sbus = service bus namespace

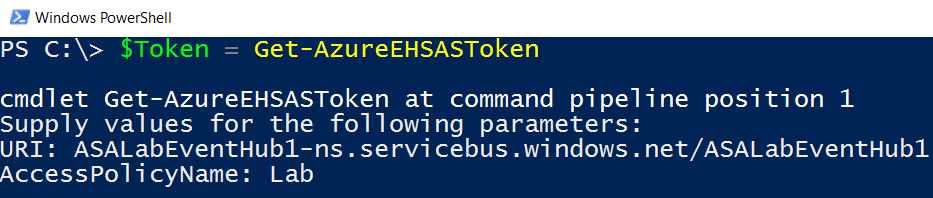
ehub = event hub name

<sbus>.servicebus.windows.net/<ehub>

(i.e. ASALabEventHub1-ns.servicebus.windows.net/ASALabEventHub1)



* 1. Enter Lab for the *AccessPolicyName* and then hit Enter. This corresponds to the shared access policy that you setup after creating your event hub.

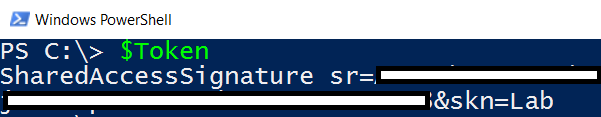


* 1. Enter the associated Primary Key value for your Lab policy as the *AccessPolicyKey*, and then hit Enter (this is the key that you copied earlier from the Azure Portal). The following example is only a truncated example, and your own key will differ. After hitting the Enter key, PowerShell will return to the command prompt.



1. Your SAS token is now stored in a variable named $Token. To verify, type $Token into the command prompt, and then hit Enter.

$Token



1. Store your event hub URI in a variable for use in the main script. Enter the following code and replace with your own service bus namespace and event hub name, and then hit Enter. [Although the code appears on two lines in the lab due to space constraints, there should be no line break in the code. Prior to hitting Enter, also verify that no spaces exists in the URI and that the quotation marks appear correctly.]

Remember to include the single quotes surrounding the URI string.

$EventHubUri = '<sbus>.servicebus.windows.net/<ehub>'



1. Copy/paste the following code. This code sends randomly generated speed values from three fake sensors to your event hub every five seconds for ten minutes. Note the number of *–Minutes* in the first line of code. Feel free to change the total duration by altering the number of minutes. You can also change the frequency of how often events are sent to your event hub by altering the number of seconds in the *Start-Sleep* line toward the end. When ready, hit Enter. As long as your $EventHubUri variable was set properly in the previous steps, you should not otherwise need to modify the code block.

$EventHubTimer = new-timespan -Minutes 10

$StopWatch = [diagnostics.stopwatch]::StartNew()

$APIUri = "https://"+ $EventHubUri +"/messages"

While ($StopWatch.elapsed -lt $EventHubTimer){

$RandomDetroit = Get-Random -minimum 65 -maximum 85

$RandomChicago = Get-Random -minimum 65 -maximum 85

$RandomKalamazoo = Get-Random -minimum 65 -maximum 85

$LabData = '[{ "SensorId":"101", "Location":"Detroit, MI", "Speed": ' + $RandomDetroit + ' },

{ "SensorId":"102", "Location":"Chicago, IL", "Speed": ' + $RandomChicago + ' },

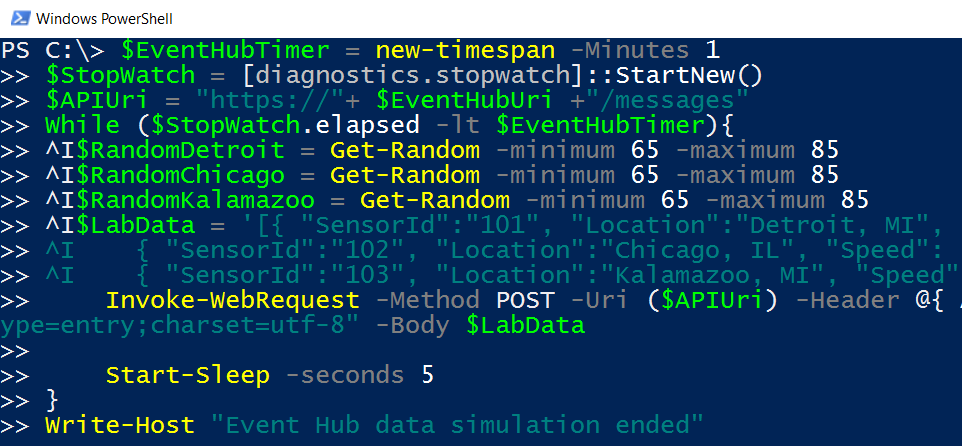
{ "SensorId":"103", "Location":"Kalamazoo, MI", "Speed": ' + $RandomKalamazoo + ' }]'

Invoke-WebRequest -Method POST -Uri ($APIUri) -Header @{ Authorization = $Token} -ContentType "application/json;type=entry;charset=utf-8" -Body $LabData

Start-Sleep -seconds 5

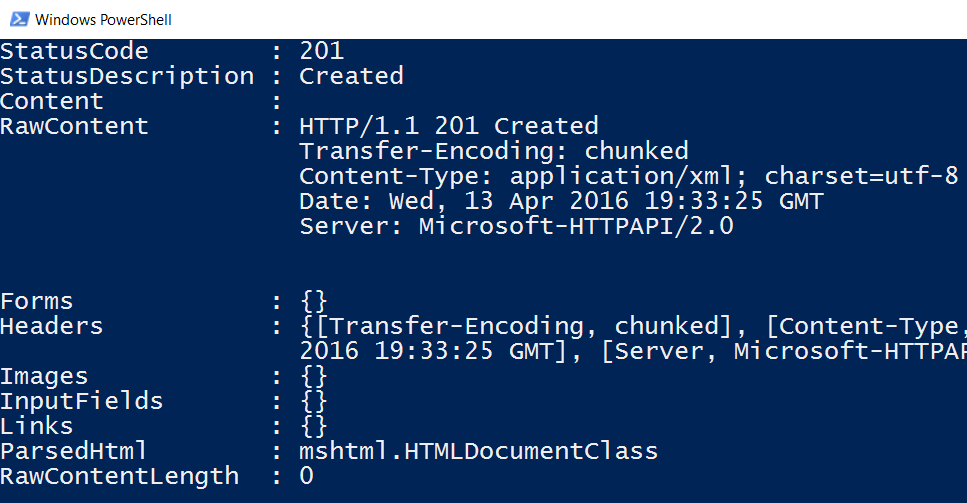
}

Write-Host "Event Hub data simulation ended"

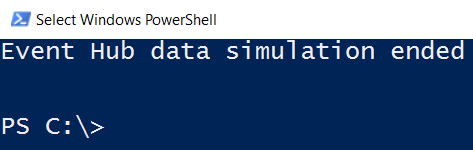


1. While the script executes, you should see a StatusCode of 201 and other data associated with the web request appear every five seconds. If you would like to resend more data later on, you may need to revisit all of the previous steps and obtain a new token. Prior to resending your request from the final code block, your SAS token may expire.

If there were any issues with your token or event hub URI, you may see recurring errors while the script runs. In this case, use **Ctrl + C** to force the script to stop executing. Make any required corrections and try running the script again.



1. You should not wait for the data simulation to finish before moving to the next lab activity. Note that after the PowerShell script finishes executing, however, you will see a message stating that the Event Hub data simulation has ended, and the PowerShell command prompt will return.



1. If you no longer need to run unrestricted scripts in PowerShell after this lab, it would be beneficial to change the execution policy again. You can choose from *Restricted*, *AllSigned*, or *RemoteSigned* as appropriate. If prompted at any point with a message, select Y / Yes to proceed.

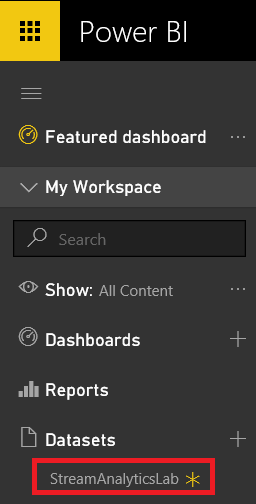
Set-ExecutionPolicy Restricted

|  |
| --- |
| Visualize your Streaming Data in Power BI |

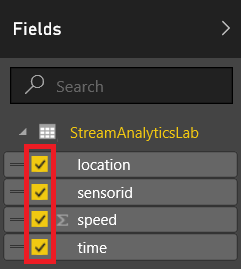
### Visualize Stream Analytics Output in Power BI

The Stream Analytics job should now be ingesting the data stream from the Event Hub, aggregating it, and then sending it to a Power BI dataset in real-time. You will now open Power BI, view the data, and have the opportunity to create sample visualizations.

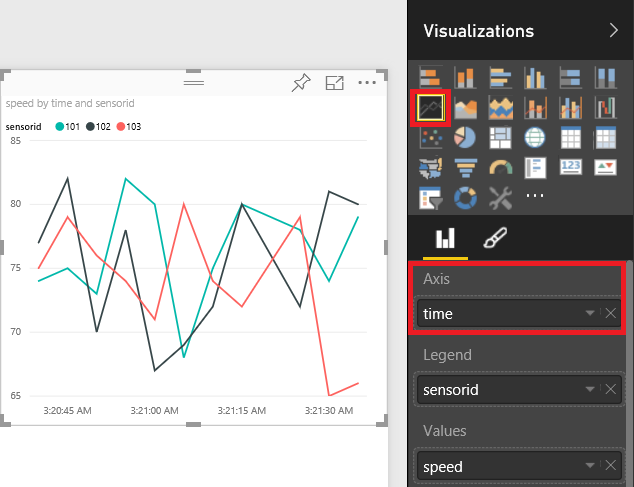
1. In a new browser tab, open Power BI by going to <http://www.powerbi.com>. If you are prompted to login to Power BI, please do so (if your Power BI account is the same as the Microsoft account that you use for Azure, you may not be prompted to sign in).
2. Go to My Workspace in the menu on the left and find *StreamAnalyticsLab* under Datasets. Select the new dataset, and a new report will appear.



1. In the *Fields* pane to the right, check all four boxes for location, sensorid, speed, and time. As you check each box, a table visual will update in the main report window.



1. In the *Visualizations* pane, click the icon for Line chart, and the table will change to a line chart. Replace the value of *Axis* with time instead of location, and the line chart should change to show a time series.



1. Add a Slicer visual to the report and add location to the slicer. Check the box for each location to view its effect on the line chart.
2. Save your report. Click Refresh at the top on occasion to see new data updates.
3. Feel free to continue adding and altering visuals in Power BI as desired.

### Stop Your Stream Analytics Job

When finished with Power BI, it is important to stop your Stream Analytics job to avoid unnecessary charges. If desired, you can delete your Stream Analytics job and Event Hub in the Azure Management Portal. Otherwise, you can keep the resources for later use. Simply make sure that your Stream Analytics job is stopped so that your Azure subscription will not incur charges.

1. Return to the browser tab that contains your Stream Analytics job in the Azure Management Portal. If necessary, re-open the portal and search for your Stream Analytics job.
2. Click the STOP button on the bottom toolbar to stop your Stream Analytics job.

For more detail on Azure Stream Analytics and Azure Event Hubs, see the Azure documentation:

**Stream Analytics**

<https://azure.microsoft.com/documentation/services/stream-analytics/>

**Event Hubs**

<https://azure.microsoft.com/documentation/services/event-hubs/>

### Conclusion

You have now completed the *Real-Time Analytics with Azure Stream Analytics (and Event Hub)* lab. In this lab, you learned how to create an Event Hub, send simulated road sensor data stream to it, process the data in Stream Analytics, and visualize near real-time freeway speeds in Power BI.

To expand on what you learned, some suggested paths include:

* Alter the output to gain experience with additional sinks such as Azure SQL Database and Azure Storage
* Change the sample PowerShell script to create variations in your simulated data stream, or learn to send input from a real device to an Event Hub
* Add reference data as an additional input and join it to your real-time input in a Stream Analytics query
* Experiment with changes to your Stream Analytics query and create more advanced output

|  |
| --- |
| Terms of Use |

© 2016 Microsoft Corporation. All rights reserved.

By using this Hands-on Lab, you agree to the following terms:

The technology/functionality described in this Hands-on Lab is provided by Microsoft Corporation in a “sandbox” testing environment for purposes of obtaining your feedback and to provide you with a learning experience. You may only use the Hands-on Lab to evaluate such technology features and functionality and provide feedback to Microsoft.  You may not use it for any other purpose. You may not modify, copy, distribute, transmit, display, perform, reproduce, publish, license, create derivative works from, transfer, or sell this Hands-on Lab or any portion thereof.

COPYING OR REPRODUCTION OF THE HANDS-ON LAB (OR ANY PORTION OF IT) TO ANY OTHER SERVER OR LOCATION FOR FURTHER REPRODUCTION OR REDISTRIBUTION IS EXPRESSLY PROHIBITED.

THIS HANDS-ON LAB PROVIDES CERTAIN SOFTWARE TECHNOLOGY/PRODUCT FEATURES AND FUNCTIONALITY, INCLUDING POTENTIAL NEW FEATURES AND CONCEPTS, IN A SIMULATED ENVIRONMENT WITHOUT COMPLEX SET-UP OR INSTALLATION FOR THE PURPOSE DESCRIBED ABOVE.  THE TECHNOLOGY/CONCEPTS REPRESENTED IN THIS HANDS-ON LAB MAY NOT REPRESENT FULL FEATURE FUNCTIONALITY AND MAY NOT WORK THE WAY A FINAL VERSION MAY WORK.  WE ALSO MAY NOT RELEASE A FINAL VERSION OF SUCH FEATURES OR CONCEPTS.  YOUR EXPERIENCE WITH USING SUCH FEATURES AND FUNCITONALITY IN A PHYSICAL ENVIRONMENT MAY ALSO BE DIFFERENT.

**FEEDBACK**.  If you give feedback about the technology features, functionality and/or concepts described in this Hands-on Lab to Microsoft, you give to Microsoft, without charge, the right to use, share and commercialize your feedback in any way and for any purpose.  You also give to third parties, without charge, any patent rights needed for their products, technologies and services to use or interface with any specific parts of a Microsoft software or service that includes the feedback.  You will not give feedback that is subject to a license that requires Microsoft to license its software or documentation to third parties because we include your feedback in them.  These rights survive this agreement.

MICROSOFT CORPORATION HEREBY DISCLAIMS ALL WARRANTIES AND CONDITIONS WITH REGARD TO THE HANDS-ON LAB, INCLUDING ALL WARRANTIES AND CONDITIONS OF MERCHANTABILITY, WHETHER EXPRESS, IMPLIED OR STATUTORY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT.  MICROSOFT DOES NOT MAKE ANY ASSURANCES OR REPRESENTATIONS WITH REGARD TO THE ACCURACY OF THE RESULTS, OUTPUT THAT DERIVES FROM USE OF THE VIRTUAL LAB, OR SUITABILITY OF THE INFORMATION CONTAINED IN THE VIRTUAL LAB FOR ANY PURPOSE.

DISCLAIMER

This lab contains only a portion of the features and enhancements in Microsoft Azure Stream Analytics and Service Bus. Some of the features might change in future releases of the product.