Microsoft

Perform sentiment analysis using Stream Analytics and Machine Learning

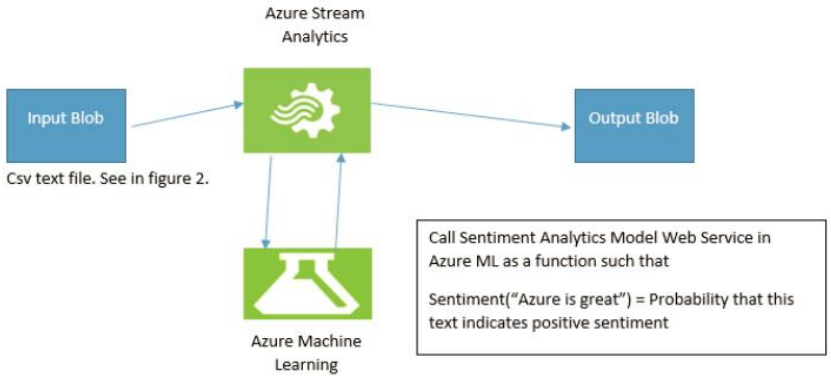
Contents

Overview

Summary

In this lab we will demonstrate how we can quickly setup a simple Stream Analytics job that integrates with Machine Learning. We will leverage a Sentiment Analytics Machine Learning Model from Cortana Analytics Gallery to analyze streaming text data and get determine the sentiment score in real time.

This is a good tutorial to understand scenarios such as real time sentiment analytics on streaming twitter data, customer chat record analysis with support staff, comments on forums/blogs/videos and many other real-time predictive scoring scenarios.



For a more realistic scenario, you can replace Blob storage with streaming Twitter data from an Azure Event Hubs input. Additionally, you could build a Microsoft Power BI real-time visualization of the aggregate sentiment.

Scenario

Contoso is interested in discovering what Social media is saying about its products and services. From ingesting a twitter feed into stream analytics, we are able automatically determine the sentiment using Azure Machine learning.

Solution Summary

Building on the experience from the Stream Analytics and Azure Machine labs we will combine both services using the Machine Learning function within Stream Analytics.

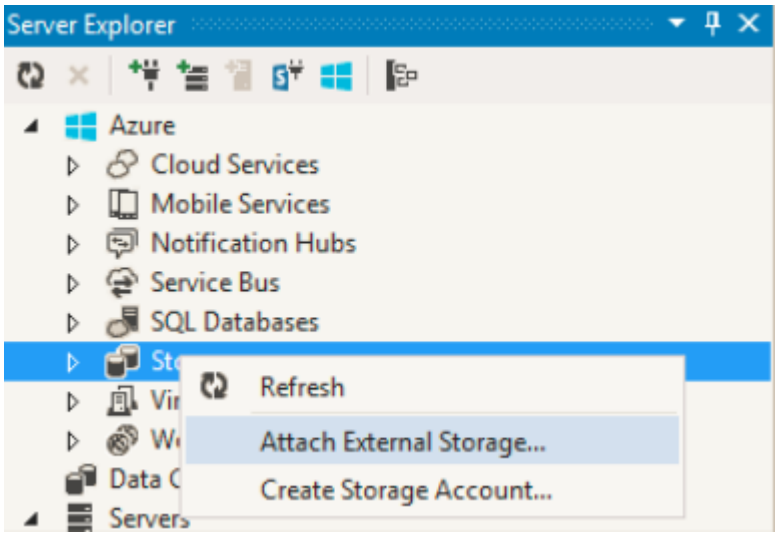
StreamAnalytics will pick up a file which has been inserted into a Storage Container. It will score each of the rows and write the results to an output container.

Upload the CSV input file to Blob Storage

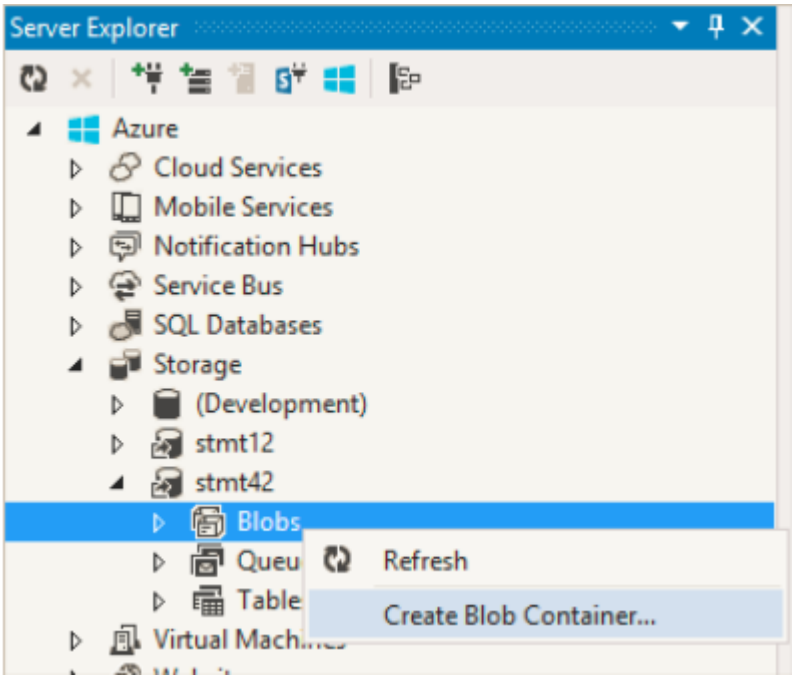
The sample application will take a CSV file from an azure blob container and feed it into Stream Analytics. Stream Analytics will take the data and use Azure ML to score the Sentiment for the twitter messages and place the results in an outgoing blob container.

For this step you can use any CSV file like this one. To upload the file, Azure Storage Explorer or Visual Studio may be used as well as custom code. For this tutorial examples are provided for Visual Studio using the extension Cloud Explorer.

1. Expand Azure and right click on the **Storage**. Choose **Attach External Storage** and provide **Account Name** and **Account Key**.



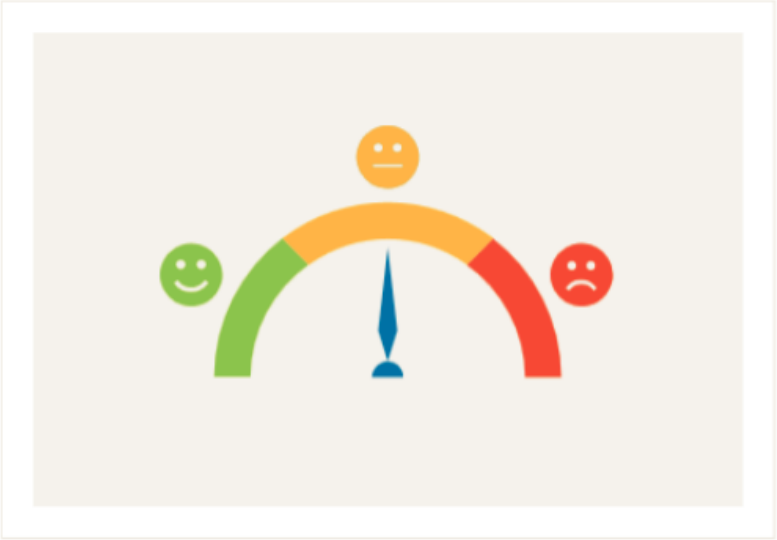
2. Expand the storage you just attached and choose **Create Blob Container** and provide a logical name of your choice. Once created, double click on the container to view its contents (which will be empty at this point).



3. Upload the CSV file by clicking the **Upload Blob** icon and then choose **file from the local disk**.

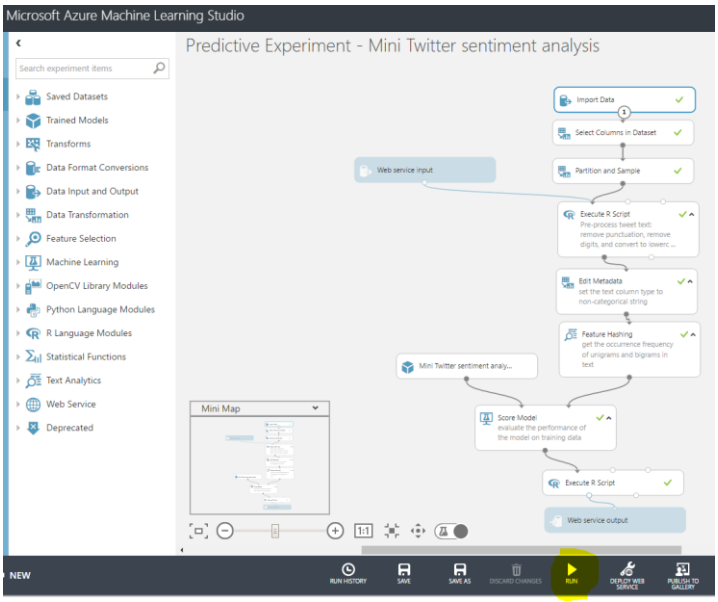
Add the Sentiment Analytics Model from Cortana

1. Download the predictive sentiment analytics model in Cortana Analytics Gallery.
2. Click Open in the Studio

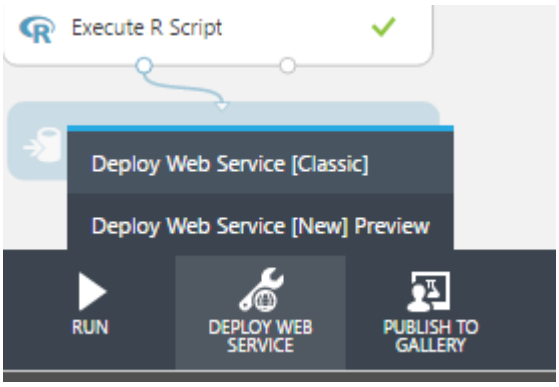


3. Sign in to be taken to the workspace. Choose the location that best suits your location.

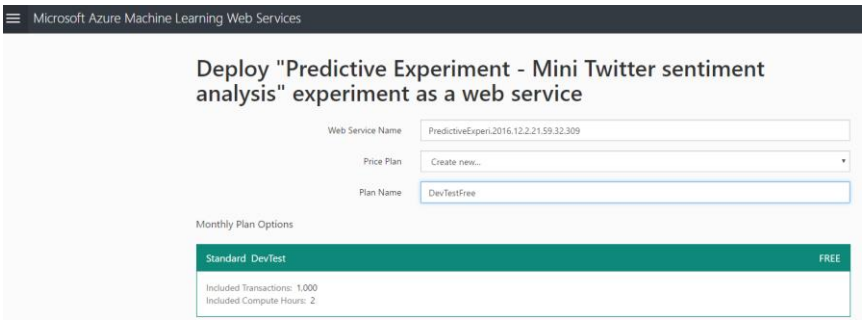
4. Now click on Run at the bottom of the Studio



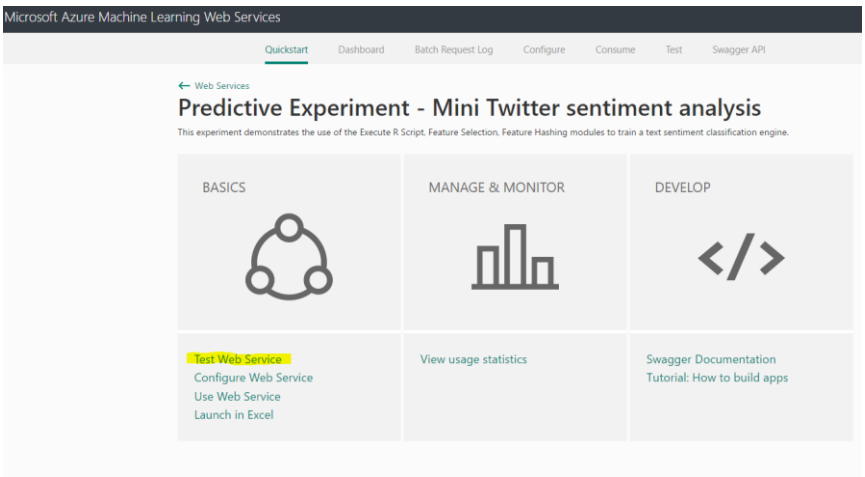
5. Once it runs successfully, click on Deploy Web Service. You can use either [Classical] or [New] in this step, but [New] is recommendable.



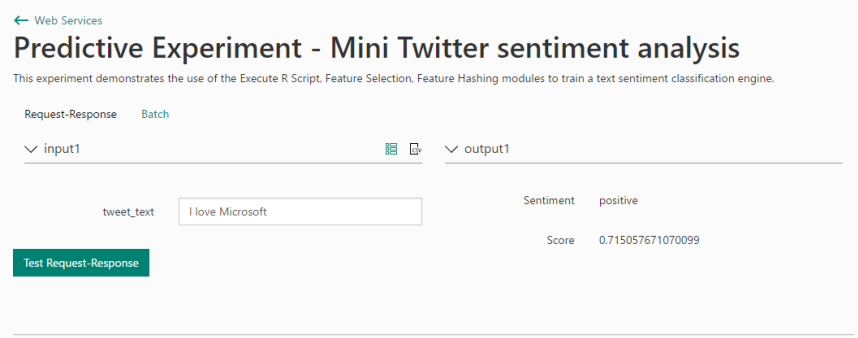
6. If you have chosen the “Deploy Web Service [New] previous step, you will need to choose a Plan. For this Lab the Free Plan is enough.



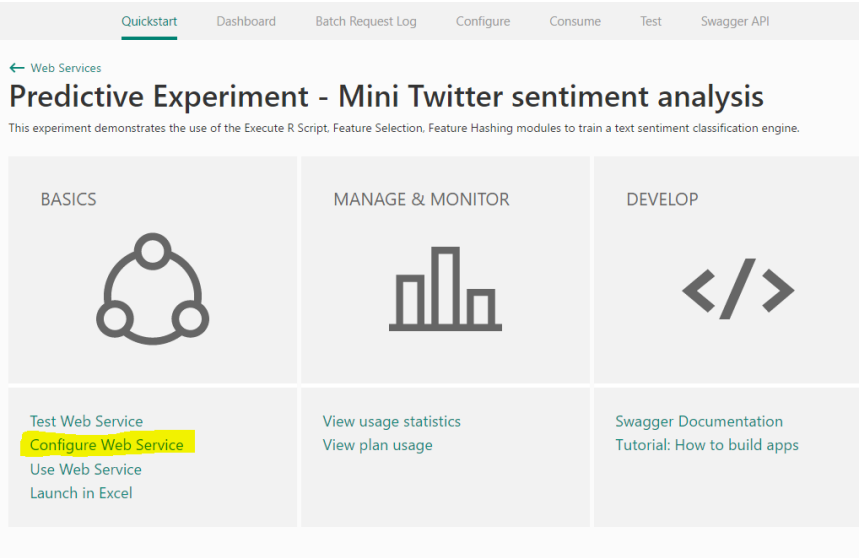
7. Now the sentiment analytics model is ready for use. To validate, click the “Test Web Service”.



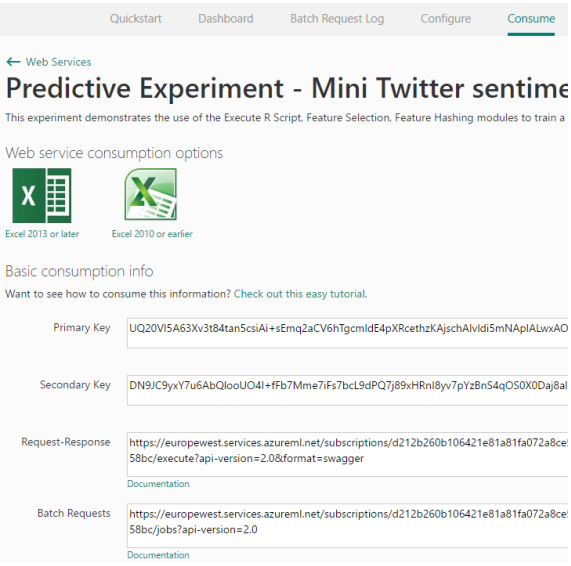
8. Providing text input such as “I love Microsoft” and the test should return a similar result as shown below



9. Go back to “Quickstart” and choose the option “Configure Web Service”



10. Take note for the API key and URL, you’ll need later for seeting up the Stream Analytics job (This step is only required to leverage a machine learning model from another Azure account's workspace. This tutorial assumes this is the case to address this scenario).

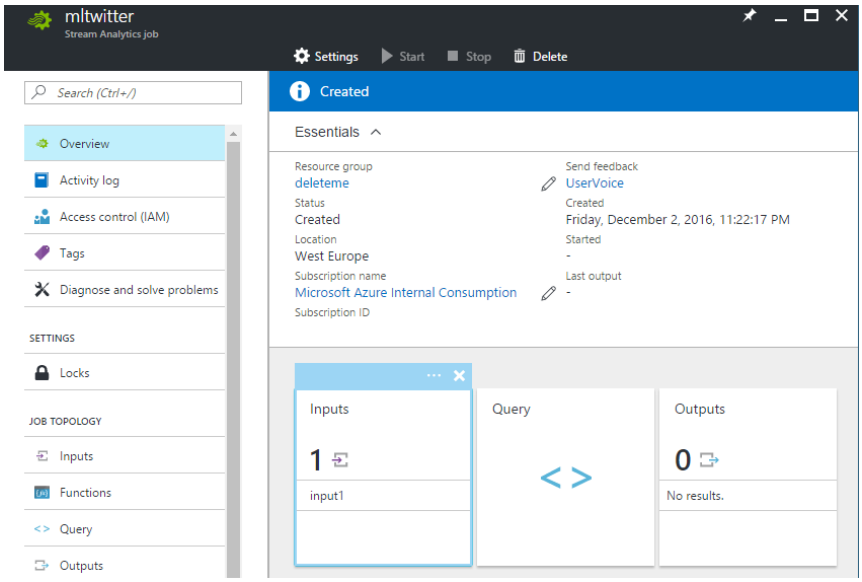


Create an Stream Analytics job which uses the machine

1. **Navigate to the Azure Management Portal.**

2. **Click** New**, Search** for “Stream Analytics jobs” **and** choose Create**.** Provide the **Job Name,** choose the appropriate **Resource Group** (or create a new one), choose the appropriate **Region Location** for the job.

3. Once the job is created, navigate **Stream Analytics jobs** section and choose the job name you just created in the previous step. Click on the **Inputs** to add an Input.



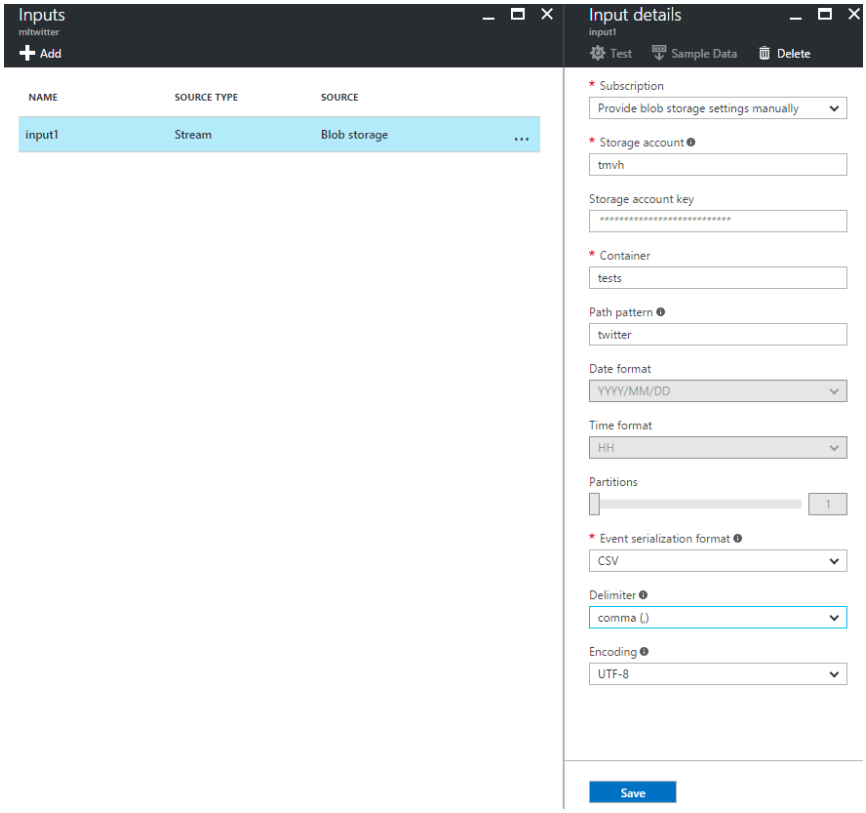
4. Specific a name for the **Input alias**.

   Select **Data stream**

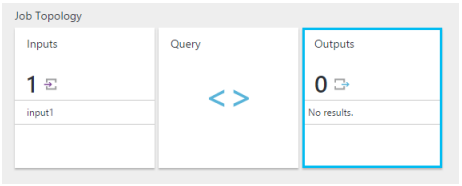
   Select **Blob Storage** as the input. Provide the storage account blob container name defined earlier when the data was uploaded.

   Choose **CSV** as **Event Serialization Format**. Accept the defaults for the rest of the **Serialization settings**.

   Click **Create**.



5. Go back to the previous panel and select “Outputs” to add an output.

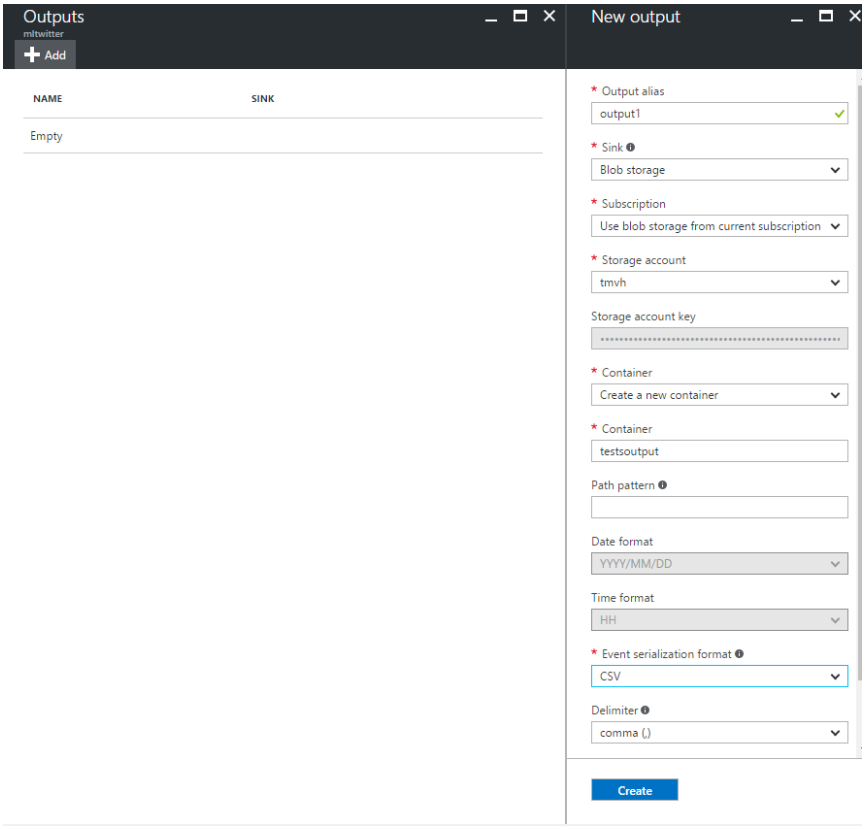


6. Choose **Blob Storage** and provide the same parameters with the exception of the container.

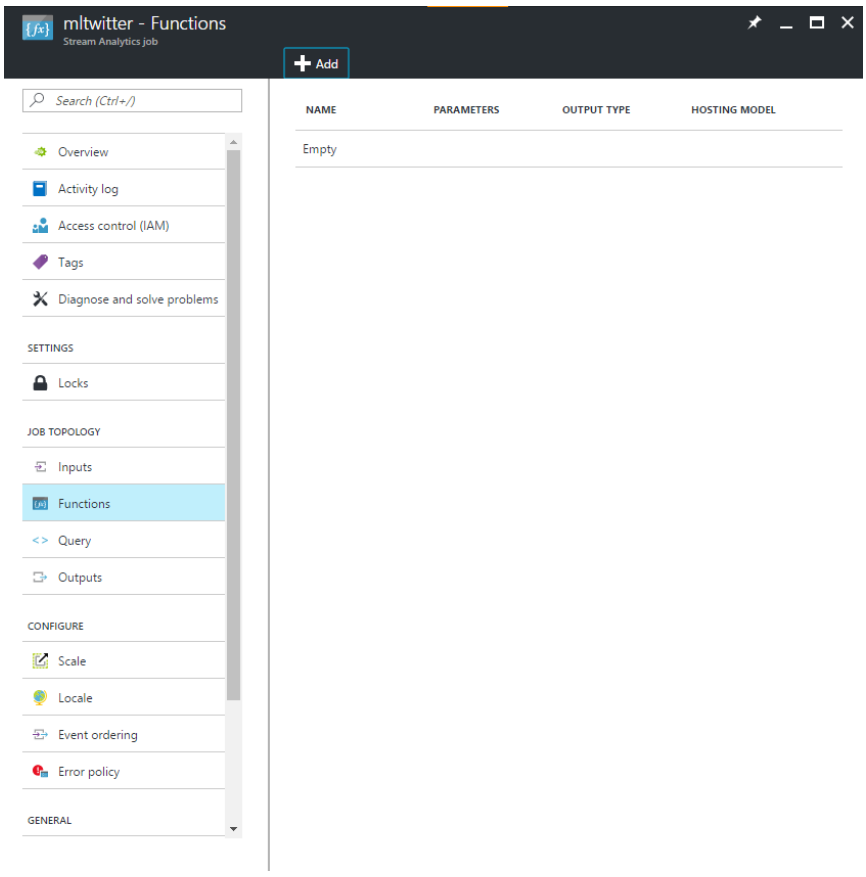
7. The Input was configured to read from the container named “tests” where the CSV file was uploaded. For the Output, put “testoutput”. The container names need to be different, and verify this container exists.

   Configure output’s **Serialization settings**. As with Input, choose **CSV**.

   Finally click the **Create** button.



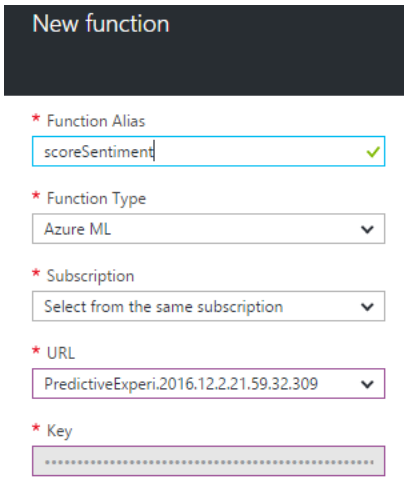
8. Navigate to **Functions** tab and click **Add**



9. Choose a name for the **Function alias**.

On **Function type** type choose AzureML

On **URL** and **Key** it should automatically populate with the name of the previously Deployed Web Service.

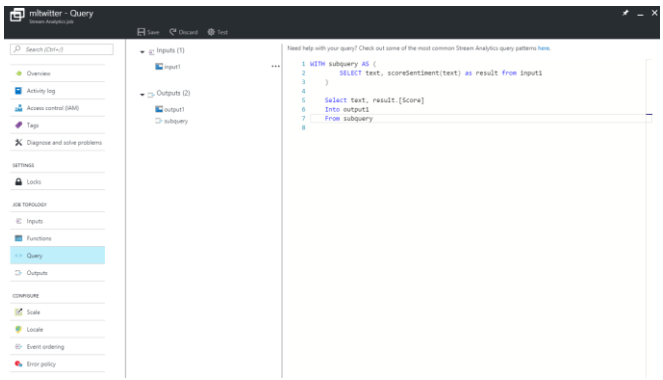


Alternatively, if you know the URL and you have the KEY you can manually introduce them.

10. Navigate to Query tab and modify the query as below:

```
WITH subquery AS (
      SELECT text, sentiment(text) as result from input
)

Select text, result.[Score]
Into output
From subquery
```



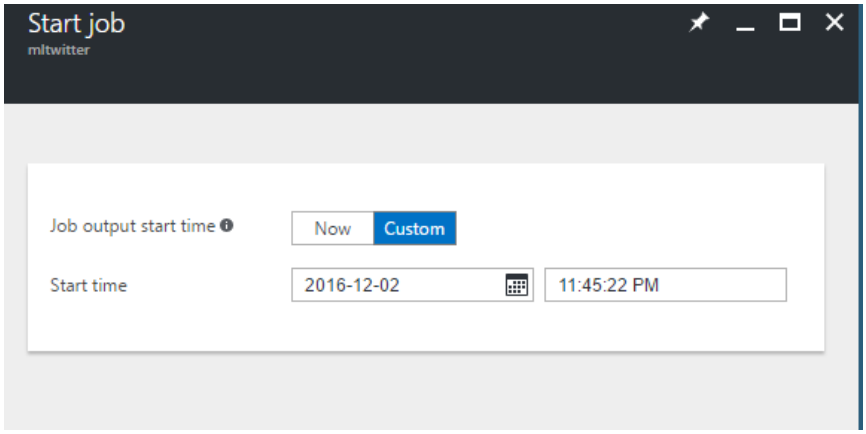
Note that the highlighted parts in the previous SQL sentence must match the input, output and function names that you attributed in the previous steps.

Click **Save** to save the query.

Start the Stream Analytics Job and observe the output

1. Click **Start** to start the job.

2. On the Start Query Dialog, choose Custom Time and select a time **prior** to when the CSV was uploaded to Blob Storage. Click OK.



3. If you want to archive every event, you can use a pass through query to read all the fields in the payload of the event or message. To start with, do a simple pass through query that projects all the fields in an event.

4. Navigate to the Blob Storage using the tool used when the CSV file was uploaded. This tutorial used Visual Studio.

5. In few minutes after the job is started, the output container is created and a CSV file uploaded into it.

6. Double clicking on the file will open the default CSV editor and should show something as below:

text	scored probabilities
Azure is so cool	0.733864188
Cortana Analytics rocks	0.590195894
I hate going to Gym	0.403451264

Stretch Activities

If you have time, consider the following stretch activities.

- Add another CSV file with a different name into the Input Blob container.
- Use Event hubs as Input an Input to Stream Analytics.
- Use PowerBI as an Output to Steam Analytics.

Terms of Use

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. Some of the features might change in future releases of the product.