

Sentiment Analysis

The purpose of this lab is to show how you can apply in real-time a machine learning model on streaming data. This use case will apply sentiment analysis on an incoming stream of Twitter tweets.

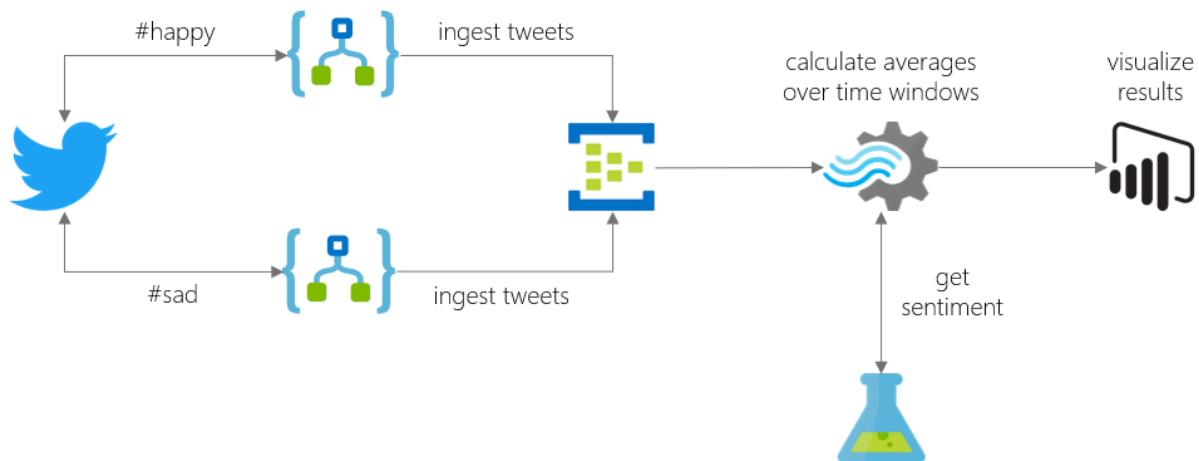
Prerequisites

To execute this lab successfully, you need the following:

- An Azure subscription. You can create a free one over [here](#).
- An Azure Machine Learning Studio workspace. you can create a free one over [here](#)
- A Power BI pro subscription. You can create a 60-day trial over [here](#).

Solution design

The high level solution design of this lab looks like this.



- Two Logic Apps are capturing tweets that contain *#happy* or *#sad*
- These tweets are ingested into Event Hubs
- Azure Stream Analytics performs the sentiment analysis against an Azure Machine Learning web service

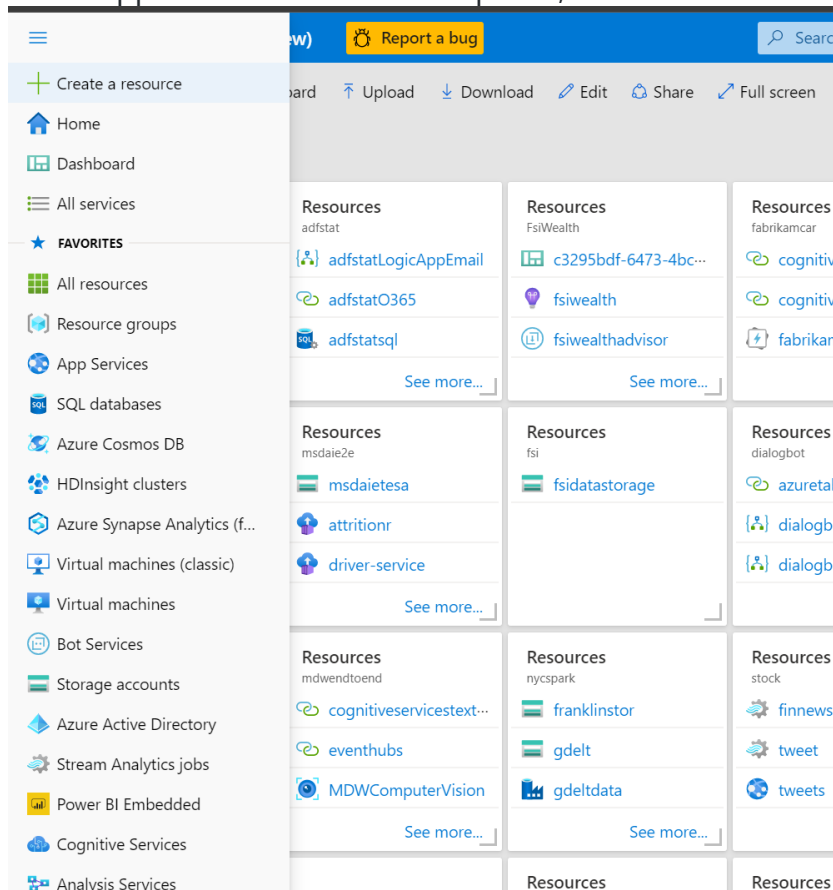
- Azure Stream Analytics also calculates the average value over a specific time window
- The results are outputted to Power BI, where they can be easily visualized

Ingest tweets

Create an Event Hub

First of all, we need a messaging service that can handle huge amounts of streaming data. Azure Event Hubs is a great service that offers all features to build a realtime data ingestion pipeline.

- Sign in to the [Azure portal](#) by using the credentials for your Azure subscription.
- In the upper-left corner of Azure portal, select + Create a resource.



- Use the search bar to find Event Hubs
- Select Event Hubs
- Click Create

Dashboard > New >

Event Hubs

Microsoft



Event Hubs

Microsoft

Save for later

Create

Overview Plans

Azure Event Hubs is a highly scalable publish-subscribe service that can ingest millions of events per second and stream them into multiple applications. This lets you process and analyze the massive amounts of data produced by your connected devices and applications.

Use Event Hubs to:

- Log millions of events per second in near real time.
- Connect devices using flexible authorization and throttling.
- Use time-based event buffering.
- Get a managed service with elastic scale.
- Reach a broad set of platforms using native client libraries.
- Pluggable adapters for other cloud services.

Useful Links

[Service overview](#)

[Documentation](#)

[Pricing details](#)

-
- Provide the following information to configure your new eventhub.

Field	Description
Resource group	Use an existing resource group in your subscription or enter a name to create a new resource group. A resource group holds related resources for an Azure solution.
Namespace	Enter a unique name that identifies your event hub namespace. Names must be unique across the resource group. <i>{prefix}-sentiment-analysis-ingestion</i>
Subscription	Select the Azure subscription that you want to use.
Location	Select the location closest to your users and the data resources to create your workspace.
Pricing Tier	Basic
Throughput Units	1



Create Namespace

Event Hubs

Basics Features 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 *

Resource group *

[Create new](#)

INSTANCE DETAILS

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

Namespace name * ✓
.servicebus.windows.net

Location *

Pricing tier ([View full pricing details](#)) *

Throughput Units *

[Review + create](#)

[< Previous](#)

[Next: Features >](#)

-
- Click Next:Features
- Leave Defaults
- Click Next:Tags
- Click Next:Review+Create
- If the validation is successful, click create

[Dashboard](#) > [New](#) > [Event Hubs](#) >



Create Namespace

Event Hubs

✓ Validation succeeded.

Basics Features Tags Review + create

Event Hubs Namespace
by Microsoft

Basics

Namespace name	astalati-sentiment-analysis-ingestion
Subscription	astalati_microsoft
Resource group	ehublab
Location	South Central US
Pricing tier	Basic
Throughput Units	1

Features

Availability Zones	Disabled
--------------------	----------

- [Create](#) [< Previous](#) [Next >](#)

- To view the new workspace, select **Go to resource**.

Microsoft Azure (Preview) [Report a bug](#)

[Dashboard](#) >
NoMarketplace | Overview [✕](#)
Deployment

« [Delete](#) [Cancel](#) [Redeploy](#) [Refresh](#)

[Overview](#)
[Inputs](#)
[Outputs](#)
[Template](#)

✓ Your deployment is complete

Deployment name: NoMarketplace
Subscription: [astalati_microsoft](#)
Resource group: [ehublab](#)

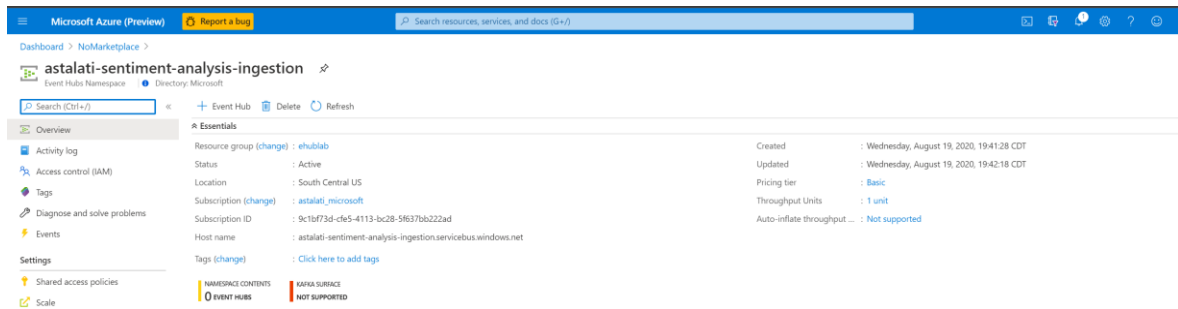
Start time: 8/19/2020, 7:41:25 PM
Correlation ID: e9caa8a8-3a76-4873-9627-232fa3deebec

[Deployment details](#) [\(Download\)](#)

[Next steps](#)

[Go to resource](#)

- Click on +Event Hub



- Create a new EventHub *ingestion-eventHubs*. A partition count of 2 and 1 day of message retention is sufficient. No need to enable the *capture* feature.

Microsoft Azure (Preview) Report a bug Search

Dashboard > NoMarketplace > astalati-sentiment-analysis-ingestion >

Create Event Hub

Event Hubs

Name *

Partition Count

Message Retention

Capture ☐ On ☐ Off

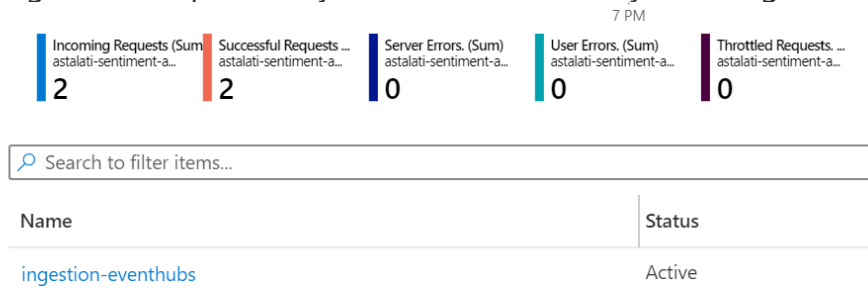
Create

Click Create

Create an access policy

Each client that reads from the Event Hub needs to be assigned to a particular consumer group. It's a good practice to give each consumer group a separate access policy, so you can revoke each one of them separately.

- Navigate to the previously created Event Hub by selecting the event hub.



-
- Click on Shared access policies (under settings)

Microsoft Azure (Preview) Report a bug Search resources

Dashboard > NoMarketplace > astalati-sentiment-analysis-ingestion >

ingestion-eventhubs (astalati-sentiment-analysis-ingestion)

Event Hubs Instance | Directory: Microsoft

Search (Ctrl+/) << + Consumer group Delete Refresh

Overview

Access control (IAM)

Diagnose and solve problems

Settings

Shared access policies

Properties

Locks

Export template

Entities

Consumer groups

Features

Capture

Process data

Support + troubleshooting

New support request

Essentials

Resource group (change) : ehublal

Location : South Central US

Subscription (change) : astalati_microsoft

Subscription ID : 9c1bf73d-cfe5-4113-bc28-5f637bb222ad

Partition Count : 2

Capture events

Use Capture to save your events to persistent storage.

Event Hub Contents: 1 CONSUMER GROUP

Event Hub Status: ACTIVE

Message Retention: 1 DAY

Partitions: 2

Requests

100

90

80

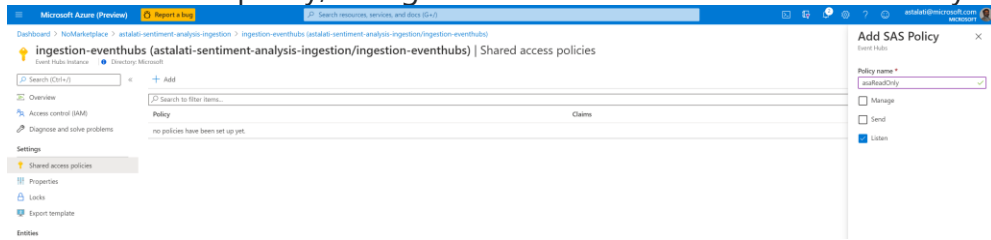
70

60

50

40

- Add here a new policy, that gives read access to Azure Stream Analytics.



- Click Create
- Click on the created access policy and copy the connection string with primary key. You'll need this later in this lab.

SAS Policy: asaReadOnly ×

Save Discard Delete ...

☐ Manage

☐ Send

☒ Listen

Primary key

+hg2KhkFrU4hgs1GhgY+XjUm+nm0AmC0cu9...

Secondary key

DVQzS4ZiswGn+DilCIJ+N7lLwCC0raWoMv40y...

Connection string–primary key

Endpoint=sb://astalati-sentiment-analysis-ing...

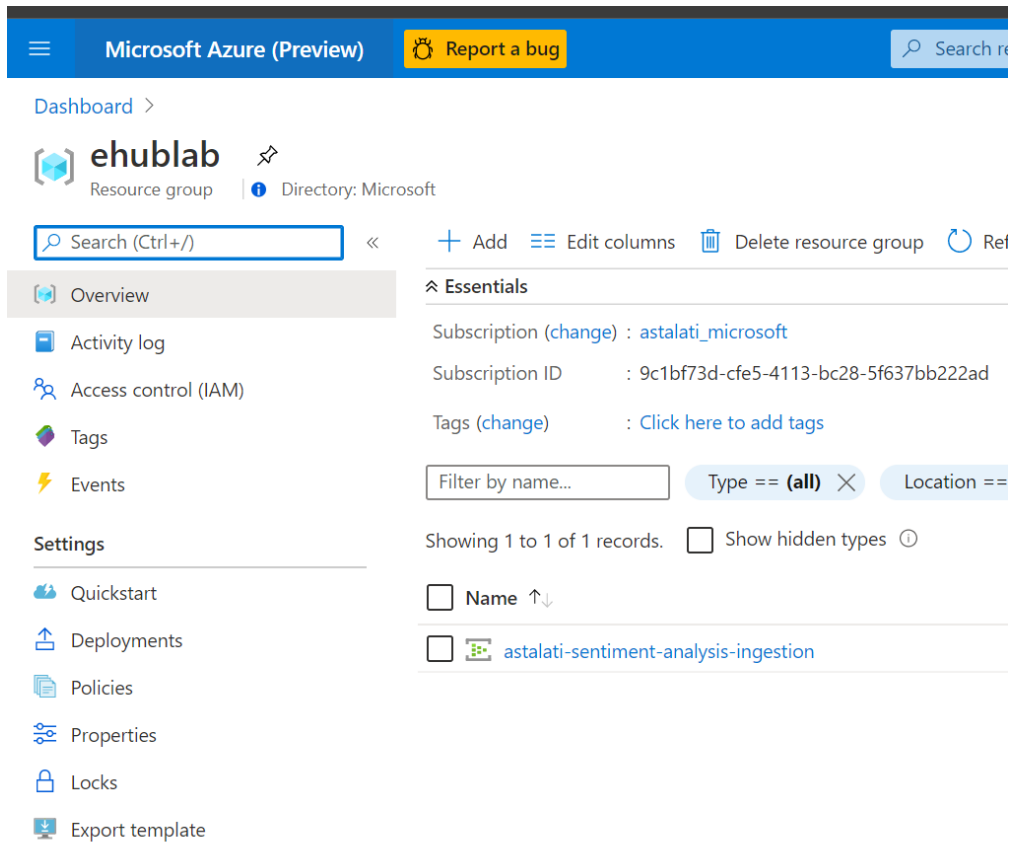
Connection string–secondary key

Endpoint=sb://astalati-sentiment-analysis-ing...

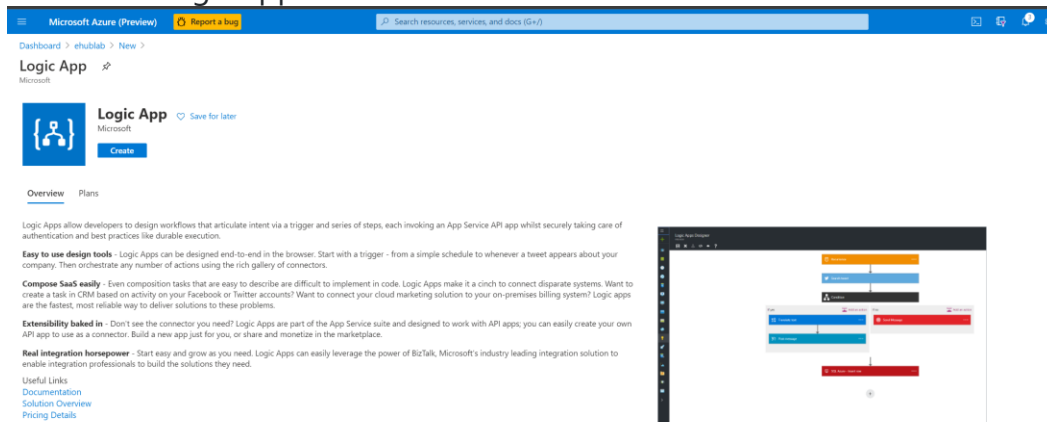
Create a Logic App

In order to provide a simplified way to ingest tweets, we will use Azure Logic Apps.

- Go to the resource group you created earlier (you can search the resource group in search bar on the portal page)
- Click on +Add



-
- Search for Logic app.



-
- Click Create

- Create a Logic App, named *{prefix}-sentiment-analysis-ingestion-happy*, choose the same region as previously.

[Dashboard](#) > [ehublab](#) > [New](#) > [Logic App](#) >

Logic App

* Basics 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 *

Resource group * [Create new](#)

Instance details

Logic App name * ✓

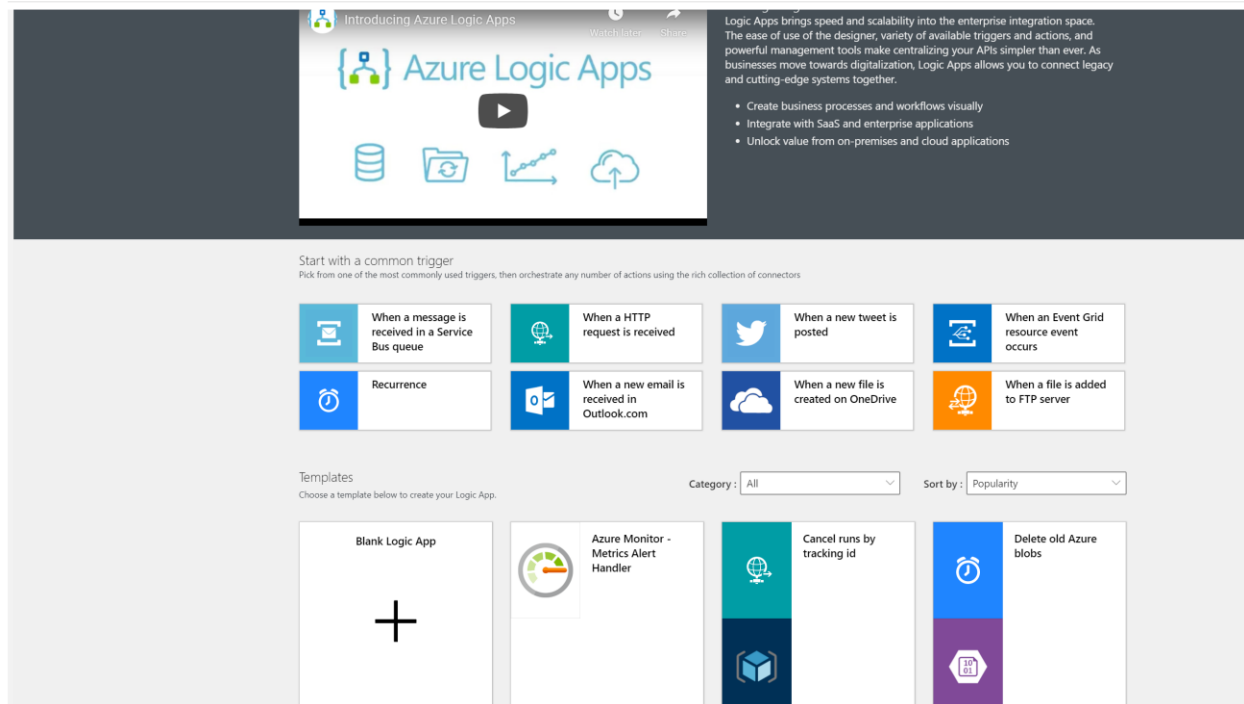
Select the location ☒ Region ☐ Integration Service Environment

Location *

Log Analytics ☐ On ☒ Off

[Review + create](#) < Previous : Basics Next : Tags > [Download a template for automation](#)

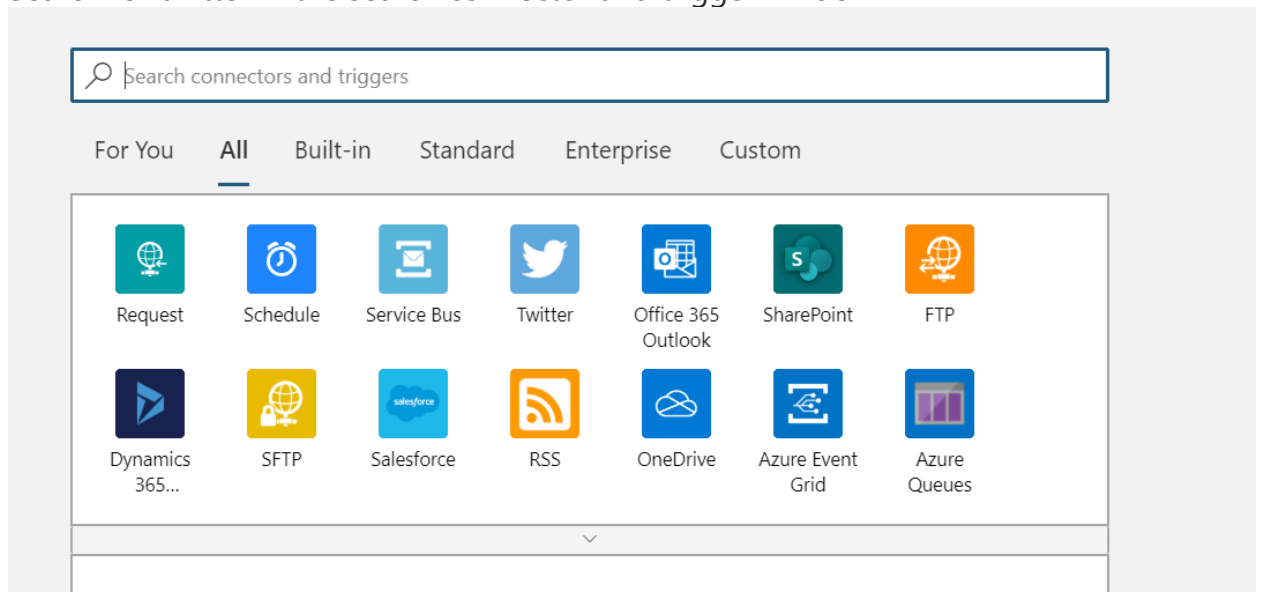
-
- Click Review+Create
- Click Create.
- Once deployed, click on "Go to resource"
- Choose to start from *Blank Logic App*.



Add a trigger that receives specific tweets

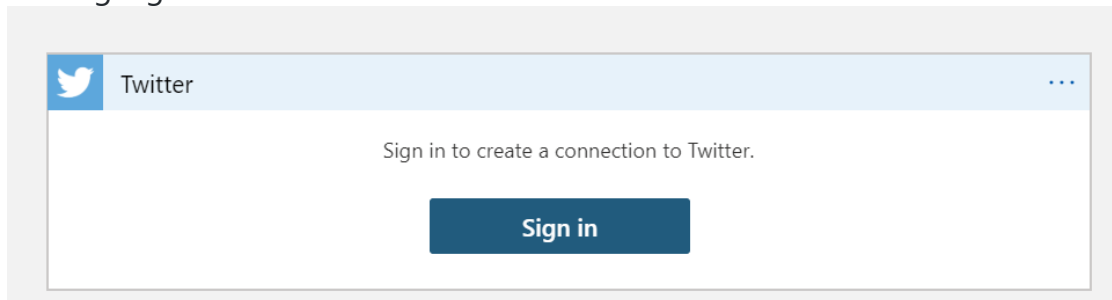
This Logic App must fire each time a tweet contains a certain key word.

- Search for twitter in the search connector and trigger window

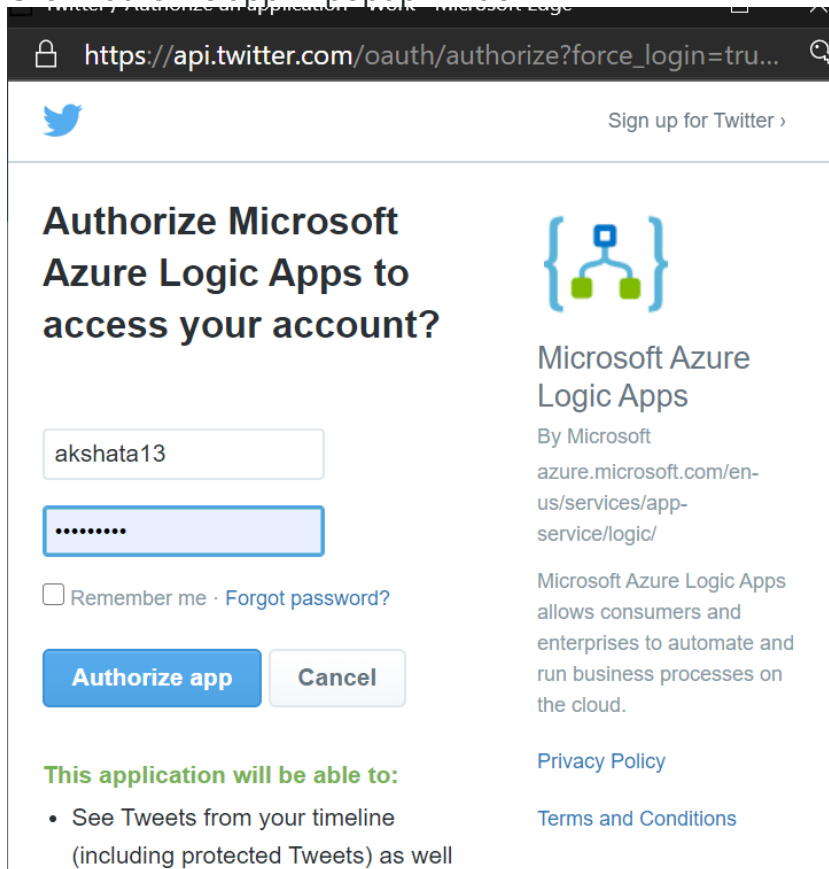


- Select Twitter

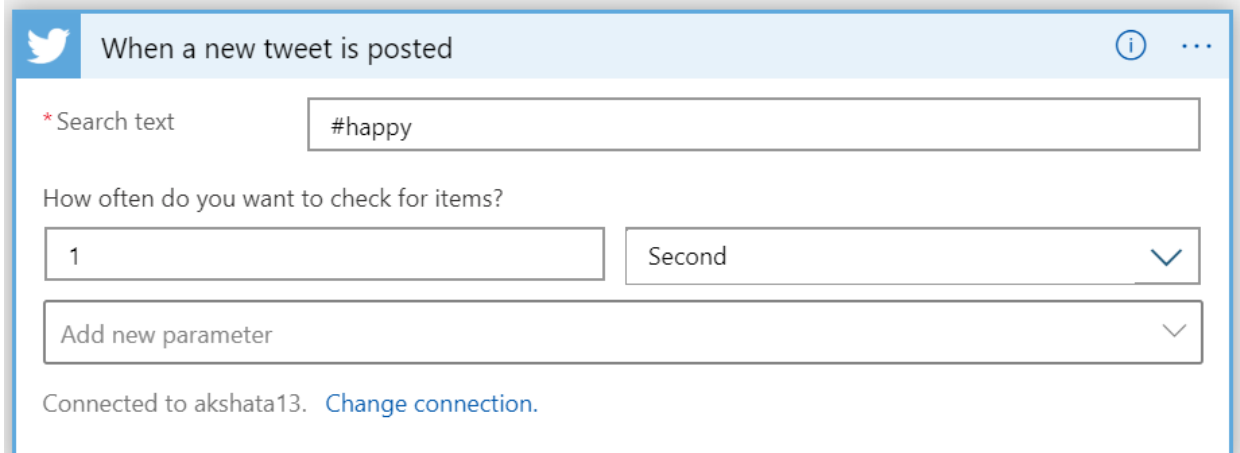
- Select *When a new tweet is posted* and authenticate with your Twitter account, by clicking Signin.



- Click Authorize app in popup window



- Provide *#happy* as the hashtag to search for and poll every second.



When a new tweet is posted

* Search text

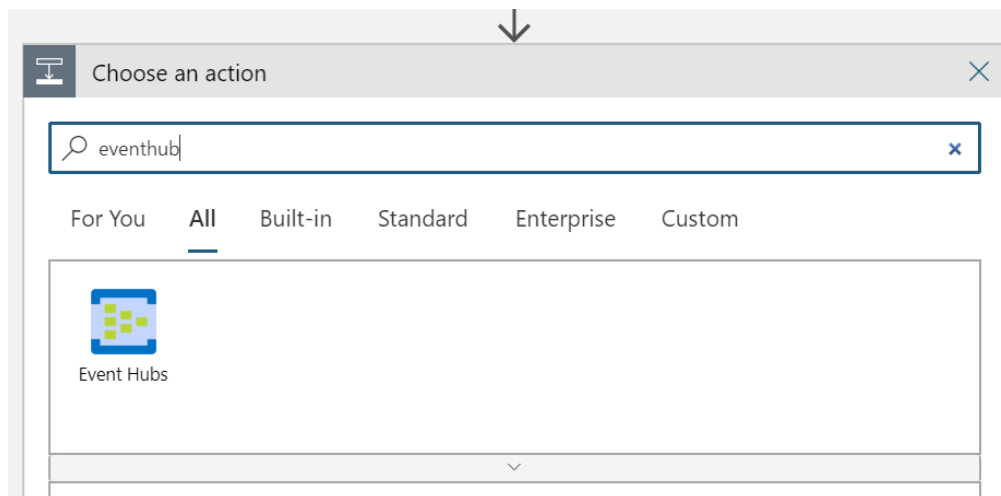
How often do you want to check for items?

Connected to akshata13. [Change connection.](#)

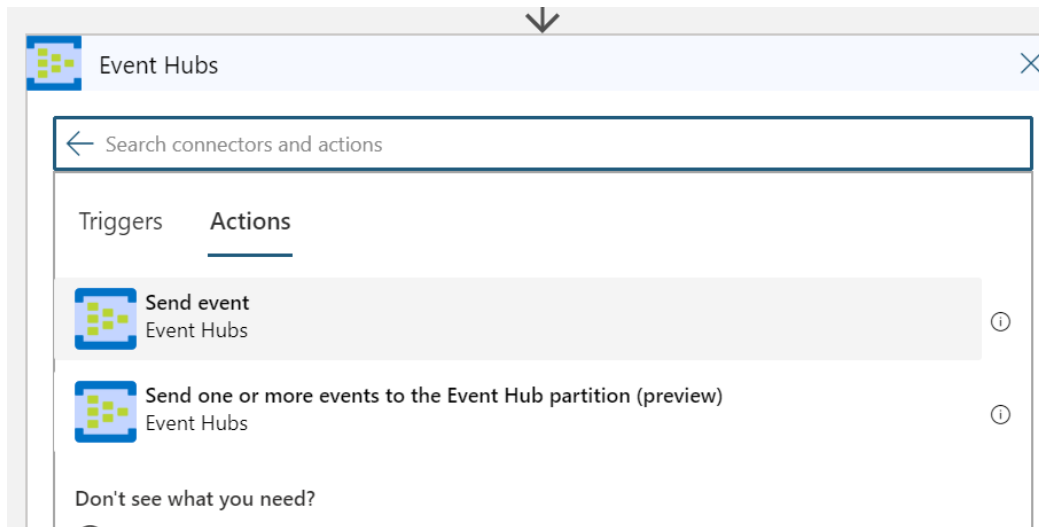
Send the tweets to Event Hubs

This Logic App has to send the captured tweets to Event Hubs.

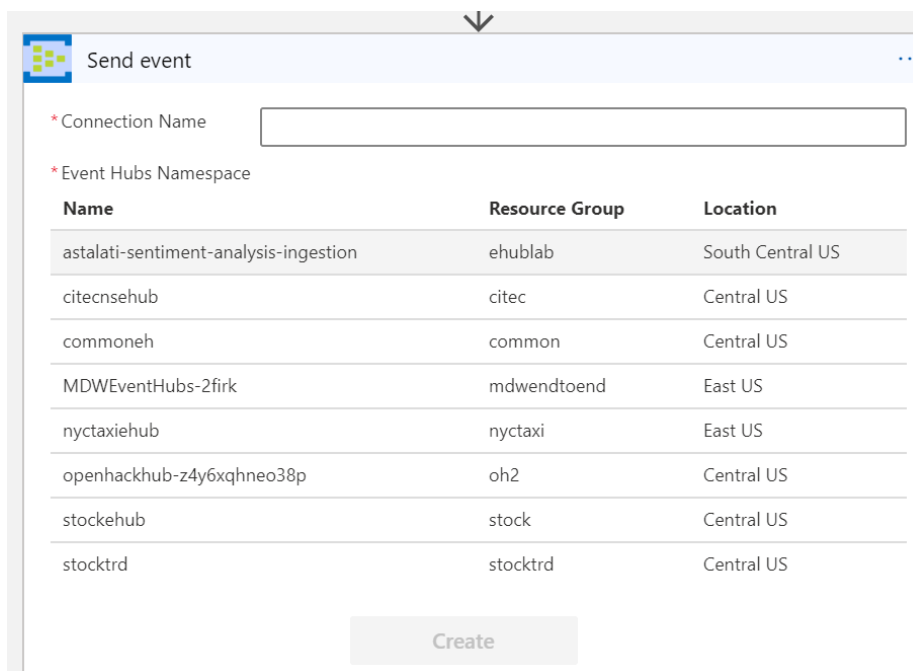
- Below the trigger, click on *New step* to add an action to send to Event Hubs via the *Send event* action.
- Search for event hub in search connector and action window



- Select event hub -> Send event.



- Connect the action to the previously created Event Hub namespace and provide connection name



- Select the event hub policy and click create

Send event

* Connection Name: sentimentanalysis

* Event Hubs Policy

Name	Rights
RootManageSharedAccessKey	Listen, Manage, Send

[Manually enter connection information](#)

Create

- In the parameter drop-down select content

Send event

* Event Hub name: Name of the Event Hub.

Add new parameter

Search or filter parameters...

- ☒ Content: Content of the event
- ☐ Properties: Key-value pairs for each application property
- ☐ Partition key: Partition key.

- Select the eventhub and add the following JSON structure:

```
{
  "text" : "@{triggerBody()['TweetText']}",
  "hashtag" : "#happy",
  "time" : "@{utcNow()}"
}
```

- This should result in the following Logic App:

When a new tweet is posted

+

Send event Show options

* Event Hub name ingestion-eventhubs

Content

```
{
  "text": "🐦 Tweet text x",
  "hashtag": "#happy",
  "time": "📅 utcNow() x"
}
```

Add new parameter

Connected to sentimentanalysis. [Change connection.](#)

- Click *Save*

Microsoft Azure (Preview) Report a

[Dashboard](#) > [Microsoft.EmptyWorkflow](#) > [astalati-se](#)

Logic Apps Designer

Save Discard Run Designer </>

•

- Go to the *Overview* blade and click *Refresh*. After a while, you should see successful Logic App runs. All tweets that contain *#happy* are from now on being ingested into your Event Hub.

Microsoft Azure (Preview) Report a bug

Search resources, services, and do

Dashboard > Microsoft.EmptyWorkflow > astalati-sentiment-analysis-ingestion-happy

astalati-sentiment-analysis-ingestion-happy

Logic app | Directory: Microsoft

<<

Run Trigger Refresh Edit Delete Disable Update Schema

Overview

Activity log

Access control (IAM)

Tags

Diagnose and solve problems

⌵ Essentials

Resource group [\(change\)](#) : ehublab

Location : South Central US

Subscription [\(change\)](#) : astalati_microsoft

Subscription ID : 9c1bf73d-cfe5-4113-bc28-5f637bb222ad

Summary

Runs history

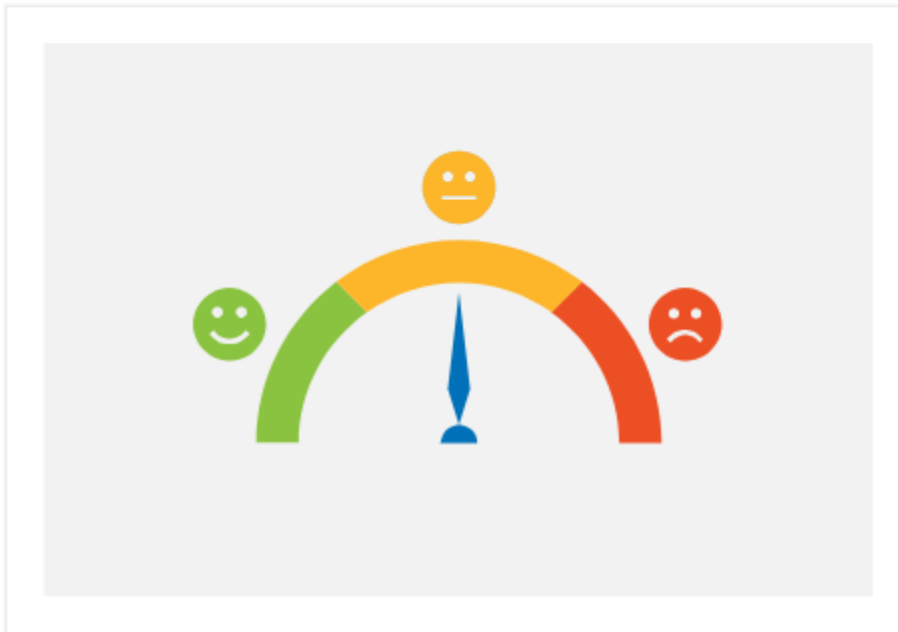
All	Start time earlier than
Specify the run identifier to open monitor view directly	
Status	Start time
Succeeded	8/19/2020, 8:08 PM

Repeat the above steps to create another Logic App that ingests tweets that contain *#sad*.

Create a web service that performs sentiment analysis

In this step, we will create an Azure Machine Learning (AML) web service that performs the sentiment analysis.

- Navigate to the Azure AI Gallery [experiment for sentiment analysis](#).
- Click on *Open in studio*.



Open in Studio

- Select and/or create a AML Studio workspace. Click OK if you get a warning about upgrading the experiment to a later version.

×

Copy experiment from Gallery

REGION:

South Central US▼

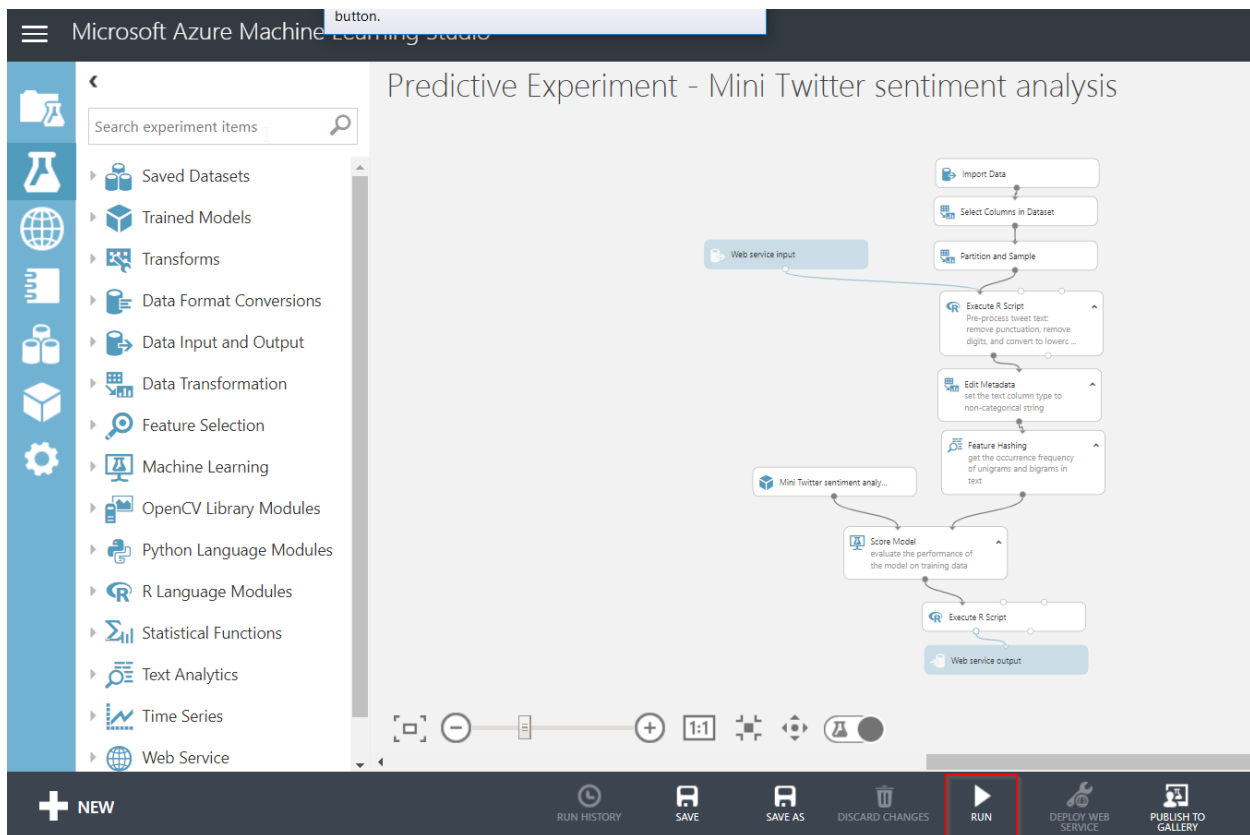
WORKSPACE:

astalati (Current)▼

✓

4/29/2019 2:44:51 PMNone

- Run the experiment, via the command at the bottom of the page, in order to train the model. This can take several minutes.



- Next, you can click *Deploy web service*. After a while, you get redirected to the overview page of the created web service. Copy already the *API key*, as you will need this later in the lab.

Microsoft Azure Machine Learning Studio

predictive experiment - mini twitter sentiment analysis

DASHBOARD CONFIGURATION

General [New Web Services Experience](#) preview


Published experiment

[View snapshot](#) [View latest](#)

Description

No description provided for this web service.

API key



Default Endpoint

[API HELP PAGE](#) TEST

[REQUEST/RESPONSE](#) [Test](#) [Test](#) preview

[BATCH EXECUTION](#) [Test](#) preview

- Via the *Test* button, you can easily provide a value to be analyzed:

Test Predictive Experiment - Mini Twitter sentiment analysis Service

Enter data to predict

TWEET_TEXT

happy



- At the bottom of the page, the result appears.

✓ 'Predictive Experiment - Mini Twitter sentiment analysis' test returned ["positive","0.709294199943542"]...

- Click now on the *REQUEST/RESPONSE* link, to go to the *API Help Page*, where you need to copy the web service URL for later usage.

Request

Method	Request URI	HTTP Version
POST	https://ussouthcentral.services.azureml.net/workspaces/787d1108244f4cd6861103366a09e868/services/3ec5d9dac6184cf9ad338adc25f22b48/execute?api-version=2.0&details=true	HTTP/1.1

Note: You may omit the **details** parameter from the query string. This would cause **ColumnTypes** to be omitted from the output




Process tweets in realtime

Create an Azure Stream Analytics Job

We need an Azure Stream Analytics Job to process the incoming stream of tweets in realtime.

- Go to the resource group you created earlier (you can search the resource group in search bar on the portal page)
- Click on +Add

Dashboard >

 **ehublab** 
Resource group |  Directory: Microsoft

Search (Ctrl+/) << [+ Add](#) [Edit columns](#) [Delete resource group](#) [Refresh](#) [→](#)

[Overview](#)

- [Activity log](#)
- [Access control \(IAM\)](#)
- [Tags](#)
- [Events](#)

Settings

- [Quickstart](#)
- [Deployments](#)
- [Policies](#)
- [Properties](#)
- [Locks](#)
- [Export template](#)

Cost Management

- [Cost analysis](#)

Essentials






Subscription ([change](#)) : [astalati_microsoft](#)

Subscription ID : 9c1bf73d-cfe5-4113-bc28-5f637bb222ad

Tags ([change](#)) : [Click here to add tags](#)

Filter by name... Type == **(all)** × Location == **(all)** ×

Showing 1 to 5 of 5 records. ☐ Show hidden types ⓘ

<input type="checkbox"/> Name ↑↓
<input type="checkbox"/>  astalati-sentiment-analysis-ingestion
<input type="checkbox"/>  astalati-sentiment-analysis-ingestion-happy
<input type="checkbox"/>  astalati-sentiment-analysis-ingestion-sad
<input type="checkbox"/>  eventhubs
<input type="checkbox"/>  twitter

- [Cost analysis](#)
- Search for stream analytics job.
- Click create

[Dashboard](#) > [ehublab](#) > [New](#) >

Stream Analytics job

Microsoft



Stream Analytics job [Save for later](#)

Microsoft

Create

Overview [Plans](#)

[Azure Stream Analytics](#) is a fully managed, cost effective real-time event processing engine that helps to unlock deep insights up real-time analytic computations on data streaming from devices, sensors, web sites, social media, applications, infrastructure, and more.

With a few clicks in the Azure portal, you can author a Stream Analytics job specifying the input source of the streaming data transformation expressed in a SQL-like language. You can monitor and adjust the scale/speed of your job in the Azure portal or more of events processed per second.

Stream Analytics leverages years of Microsoft Research work in developing highly tuned streaming engines for time-sensitive intuitive specifications of such.

Useful Links

[What is Azure Stream Analytics?](#)


[Learning Path for Stream Analytics](#)

[Get Started](#)

-
- Create a Stream Analytics Job, named *{prefix}-sentiment-analysis-asa*. Select the resource group you created and identical location as the previously created services. Keep *Cloud* as the hosting environment and set the *Streaming units* to 1. The latter will save you some costs.

[Dashboard](#) > [ehublab](#) > [New](#) > [Stream Analytics job](#) >

New Stream Analytics job

 This will create a new Stream Analytics job. You will be charged according to Azure

Job name *

astalati-sentiment-analysis-asa ✓

Subscription *

astalati_microsoft ▼

Resource group *

ehublab ▼

[Create new](#)

Location *

Central US ▼

Hosting environment ⓘ

Cloud

Edge

Streaming units (1 to 192) ⓘ



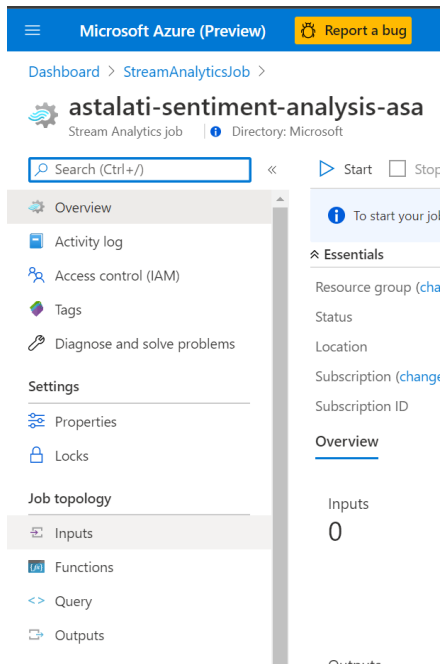
1

-
- Click Create
- Click on Go to resource

Configure the Event Hubs Input

Let's now create a new *Input*, which should refer to the Event Hub that we created.

- Go to the *Inputs* blade and click *Add stream input*. Choose *Event Hub*.



+ Add stream input ▾

Event Hub

IoT Hub

Blob storage

- In case you created the Event Hub yourself, you can use the *Select Event Hub from your subscription* option. If not, provide the settings manually. You can retrieve all these settings from the Event Hubs connection string.

Event Hub

New input

Input alias *

twitterfeed

Provide Event Hub settings manually

Select Event Hub from your subscriptions

Subscription

astalati_microsoft

Event Hub namespace * ⓘ

astalati-sentiment-analysis-ingestion

Event Hub name * ⓘ

Create new

Use existing

ingestion-eventhubs

Event Hub policy name * ⓘ

Create new

Use existing

astalati-sentiment-analysis-asa_twitterfeed_policy

Event Hub policy key

Event Hub consumer group * ⓘ

Create new

Use existing

\$Default

Partition key ⓘ

Event serialization format * ⓘ

JSON

Encoding ⓘ

UTF-8

Event compression type ⓘ

None

Save

The selected resource and the stream

- Click Save

Create AML Web Service function

To be able to connect to the AML web service, we must create a new *Function*.

- Go to the *Funtions* blade and click *Add*. Choose *Azure ML Studio*.

: Microsoft

+ Add

Azure ML Service

Javascript UDF

Javascript UDA

Azure ML Studio

-
- Provide the function alias *getSentiment*. Provide the settings manually by specifying the *Url* and *API Key* that you copied previously.

Azure Machine Learning function

New function

* Function alias

getSentiment



- ☒ Provide Azure Machine Learning function settings manually
- ☐ Select Azure Machine Learning function from your subscriptions

Subscription

Subscription information not needed

* URL

https://ussouthcentral.services.azureml.net/workspaces/787c



Key

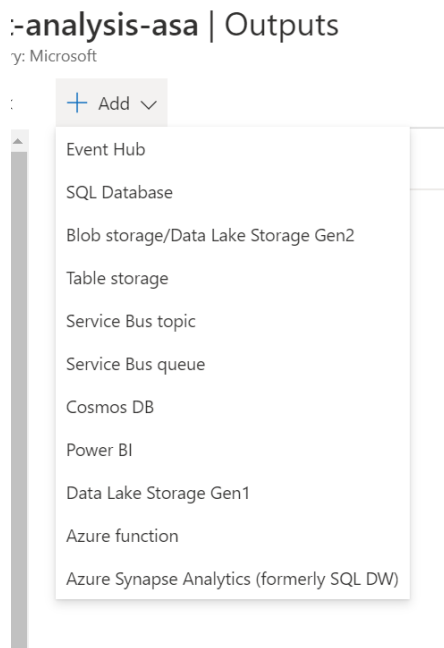
.....



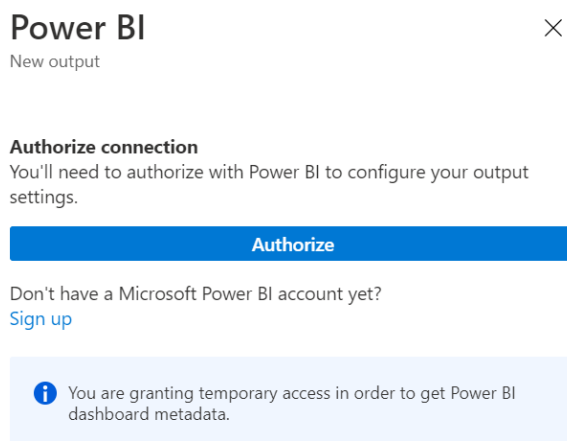
Configure the Power BI output

We need to send the result to Power BI, which means creating an *Output*.

- Go to the *Outputs* blade and click *Add*. Choose Power BI.



- Click on *Authorize* if it prompts you to authenticate



-
- Provide the output alias *powerbi*. Specify a meaningful *Dataset name* and *Table name*. These names will be used to create automatically a data set in Power BI.

Power BI

New output

×

Currently authorized as [Ashish Talati \(astalati@microsoft.com\)](#)

Output alias *

powerbi ✓

Group workspace

astalati ▼

Dataset name * ⓘ

tweets ✓

Table name *

tweettbl ✓

Authentication mode

User token ▼

ⓘ

Note: You are granting this output permanent access to your Power BI dashboard. Should you need to revoke this access in the future you can do one of the following:
1. Change the user account password.
2. Delete this output.
3. Delete this job.

Save

Configure the query

Now, we must write a query that calls the AML function to get the sentiment score for each tweet and aggregates the results per 10 seconds.

- Go to the *Query* blade and paste the following SQL statement in the query window.

```
--Create a temp table that contains the sentiment score (via the getSentiment
function)
WITH
scoredData AS (
    SELECT time, hashtag, getSentiment(text) as result
    FROM twitterfeed
)
--Select average score over a window of 10 seconds and send it to Power BI
```

```

SELECT
    System.Timestamp as time,
    hashtag,
    AVG(result.[Score]) as score
INTO
    powerbi
FROM
    scoredData
GROUP BY
    hashtag, TumblingWindow(second,10)

```

The screenshot shows the Azure Stream Analytics Query Editor interface. The query is as follows:

```

1 --Create a temp table that contains the sentiment score (via the getsentiment function)
2 SET
3   scoredData AS (
4     SELECT time, hashtag, getsentiment(text) as result
5     FROM twitterfeed
6   )
7
8 --Select average score over a window of 10 seconds and send it to Power BI
9 SELECT
10    System.Timestamp as time,
11    hashtag,
12    AVG(result.[Score]) as score
13 INTO
14    powerbi

```

Below the query, the 'Test results' section shows a table of events. The table has columns: text, hashtag, time, EventProcessedUtcTime, PartitionId, and EventEnqueuedUtcTime. The data shows various tweets and their sentiment scores.

-
- Click **Save**.

Start and monitor the job

- Go to the **Overview** blade and start the job from now.

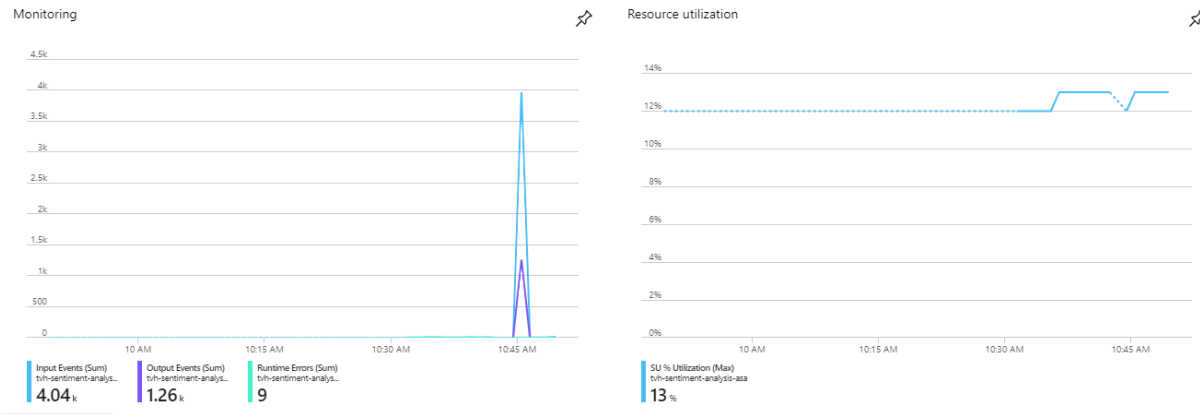
The screenshot shows the Azure Stream Analytics job overview page for the job named 'astalati-sentiment-analysis-asa'. The page includes a search bar, a list of tabs (Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Settings), and a section for job details. The job is currently in a 'Created' state.

Job Details:

- Resource group (change): ehublab
- Status: Created
- Location: Central US
- Subscription (change): astalati_microsoft
- Subscription ID: 9c1bf73d-cfe5-4113-bc28-5f637bb222ad





Overview

- Click Start
- It takes a while before the job is completed up and running, but after 5 minutes, you should see that the first events are getting processed.






Visualize results in Power BI

In your Power BI namespace, you should see under the *Datasets* tab, that a data set has been automatically created by Azure Stream Analytics.

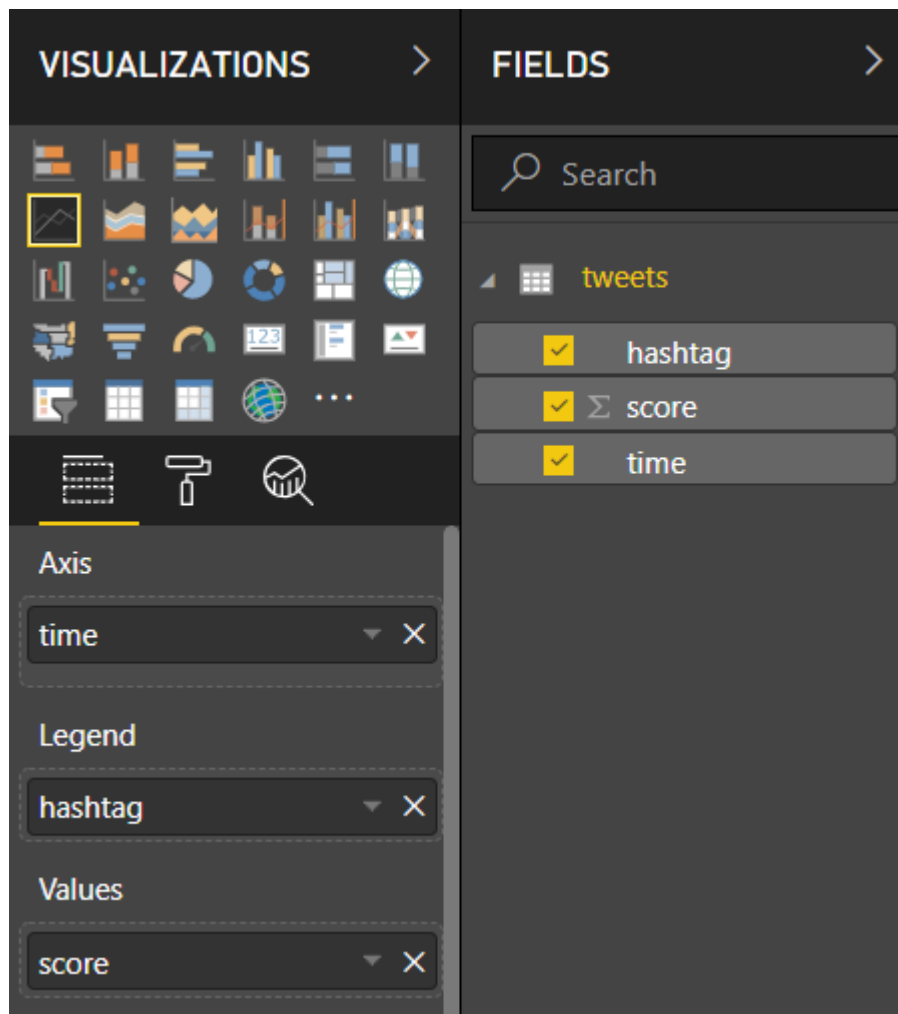
Showing 1 item(s)					
NAME ↑	ACTIONS	REFRESHED	NEXT REFRESH	API ACCESS	
twitterSentimentResults	    ...	2/7/2019, 10:42:54 AM	N/A	Hybrid	

- In the *Actions* of your data set, choose *Create report*.

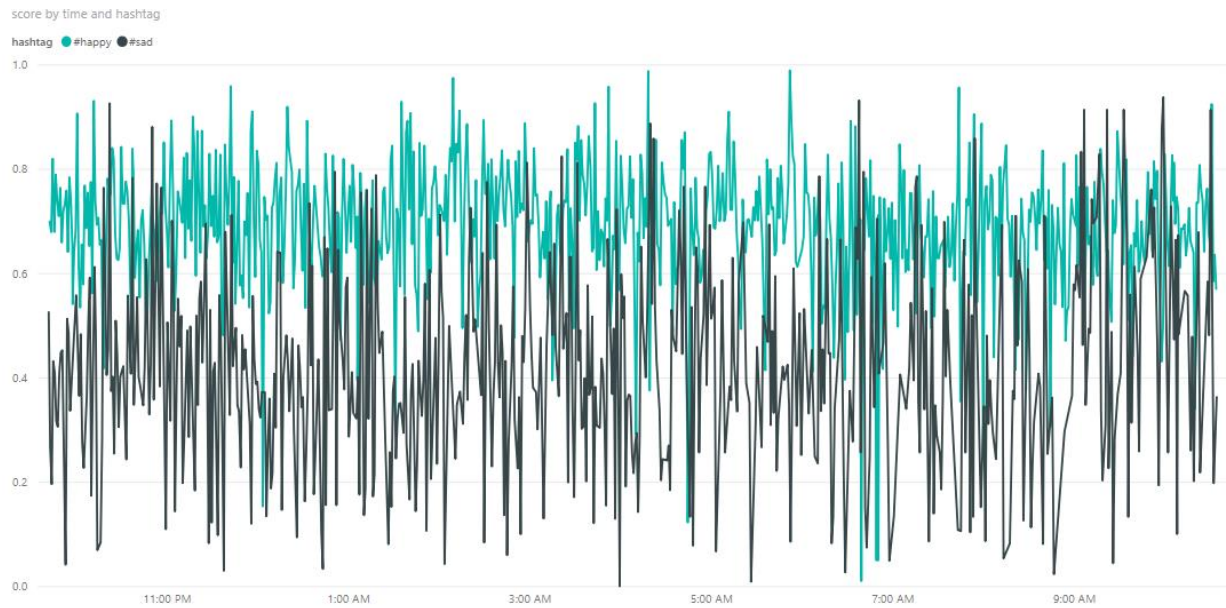
	Name		Type	Owner	Refreshed
	PBI Embed		Dataset	astalati	4/1/19, 1:27
	tweetds		Dataset	astalati	8/19/20, 9:00

Create report
Delete
Manage permissions
Quick insights
Edit
API Info

-
- Select the *Line chart* as the chart type. Take *time* as the *Axis*, *hashtag* as the *Legend* and *score* as the *Values*.



- Make the chart itself bigger, so it nicely fits your screen. You should see the results by now. Normally, *#happy* should have a significantly better sentiment score, compared to *#sad*.



- Save the report and give it a meaningful name.