Azure Fast Start for Mobile Application Development

Module 12: Real Time and Big Data Analysis

Student Lab Manual

Instructor Edition (Book Title Hidden Style)

Version 1.0

**Conditions and Terms of Use**

Microsoft Confidential

This training package is proprietary and confidential, and is intended only for uses described in the training materials. Content and software is provided to you under a Non-Disclosure Agreement and cannot be distributed. Copying or disclosing all or any portion of the content and/or software included in such packages is strictly prohibited.

The contents of this package are for informational and training purposes only and are provided "as is" without warranty of any kind, whether express or implied, including but not limited to the implied warranties of merchantability, fitness for a particular purpose, and non-infringement.

Training package content, including URLs and other Internet Web site references, is subject to change without notice. Because Microsoft must respond to changing market conditions, the content should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented after the date of publication. Unless otherwise noted, the companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted herein are fictitious, and no association with any real company, organization, product, domain name, e-mail address, logo, person, place, or event is intended or should be inferred.

© 2015 Microsoft Corporation. All rights reserved.

**Copyright and Trademarks**

© 2015 Microsoft Corporation. All rights reserved.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

For more information, see Use of Microsoft Copyrighted Content at  
<http://www.microsoft.com/en-us/legal/intellectualproperty/Permissions/default.aspx>

Azure, HDInsight, Internet Explorer, Microsoft, Skype, Windows, and Xbox are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. Other Microsoft products mentioned herein may be either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. All other trademarks are property of their respective owners.

Contents

[Lab 1: Real Time Analysis 1](#_Toc432693553)

[Exercise No. 1: Create an Event Hub input 2](#_Toc432693554)

[Exercise No. 2: Configure and start the Twitter client application 4](#_Toc432693555)

[Exercise No. 3: Create Stream Analytics Job 6](#_Toc432693556)

[Exercise No. 4: Create Power BI dashboard 11](#_Toc432693557)

[Lab 2: Big Data Analysis 15](#_Toc432693558)

[Exercise No. 1: Adding outputs to Stream Analytics 16](#_Toc432693559)

[Exercise No. 2: Create HDInsight Spark Cluster 21](#_Toc432693560)

[Exercise No. 3: Create Hive External Table 27](#_Toc432693561)

[Exercise No. 4: Run Interactive Spark SQL queries using a Zeppelin notebook 30](#_Toc432693562)

# Lab 1: Real Time Analysis

#### Introduction

In this tutorial, you will learn how to analyze in Real Time Tweets events solution by bringing real-time events into Event Hubs, writing Stream Analytics queries to analyze the data, and then storing the results or using a dashboard to provide insights in real time. Also, we will add sentiments Analysis to this result.

Social media analytics tools help organizations understand trending topics, meaning subjects and attitudes with a high volume of posts in social media. Sentiment analysis —also called **opinion mining**—uses social media analytics tools to determine attitudes toward a product, idea, etc.

#### Objectives

After completing this lab, you will be able to:

* Create an Azure Event Hub.
* Create a Stream Analytics Job.
* Analyze Tweets and their sentiment in real time into Power BI.

#### Prerequisites

* An Azure account is required for this tutorial.
* A Power BI account is required for this tutorial.
* A Twitter account is required for this tutorial.

#### Estimated Time to Complete This Lab

60 minutes

#### Scenario

Real Time Analysis on share products from the Windows 10 Application previously developed will help customers to make the correct choice regarding the products that they want to buy. Also, combining this Social data with a sentiment Analysis give them a better overview and a useful Business Intelligence tool to make the correct choice.

Exercise No. 1: Create an Event Hub input

#### Objectives

In this exercise, you will:

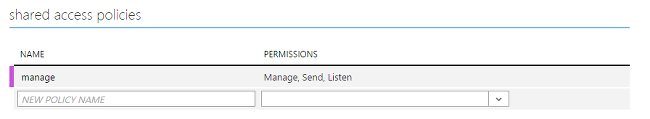
* Create an Event Hub Input.
* Configure a Consumer Group.

Task Description

The sample application will generate events and push them to an Event Hubs instance (an Event Hub, for short). Service Bus Event Hubs are the preferred method of event ingestion for Stream Analytics. See Event Hubs documentation in [Service Bus documentation](https://azure.microsoft.com/documentation/services/service-bus/).

Follow these steps to create an Event Hub:

1. In the Azure Portal, (<https://manage.windowsazure.com>) click **NEW** > **APP** **SERVICES** > **SERVICE** **BUS** > **EVENT** **HUB** > **QUICK** **CREATE** and provide a name, region, and new or existing namespace to create a new Event Hub.
2. As a best practice, each Stream Analytics job should read from a single Event Hubs Consumer Group. We will walk you through the process of creating a Consumer Group below and you can learn more about them here. To create a Consumer Group, navigate to the newly created Event Hub and click the **CONSUMER GROUPS** tab, then click **CREATE** on the bottom of the page and provide a name for your Consumer Group.
3. To grant access to the Event Hub, we will need to create a shared access policy. Click the **CONFIGURE** tab of your Event Hub.
4. Under **SHARED ACCESS POLICIES**, create a new policy with **MANAGE** permission.



1. Click **SAVE** at the bottom of the page.
2. Navigate to the **DASHBOARD**, click **View Connection String** at the bottom of the page, and copy and save the connection information. (Use the copy icon that appears under the search icon).
3. Because the creation of an HDInsight Spark Cluster can take time, at this step, create the HDInsight Spark Cluster (Step: [Exercise No. 2: Create HDInsight Spark Cluster](#_Exercise_No._2:)).

Exercise No. 2: Configure and Start the Twitter Client Application

#### Objectives

In this exercise, you will:

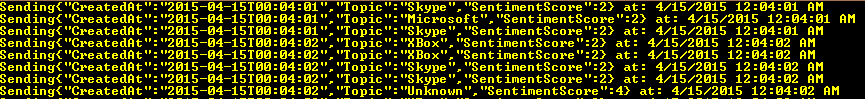
* Configure the application Twitter.
* Run the application and verify Tweets and their sentiment.

Task Description

We have provided a client application that will tap into Twitter data through [Twitter's Streaming APIs](https://dev.twitter.com/streaming/overview) to collect Tweet events about a parameterized set of topics. The third-party open source tool [Sentiment140](http://help.sentiment140.com/) is used to assign a sentiment value to each tweet (0: negative, 2: neutral, 4: positive) and then Tweet events are pushed to Event Hub.

Follow these steps to set up the application:

1. Open the TwitterClient solution.
2. Open App.config and replace oauth\_consumer\_key, oauth\_consumer\_secret, oauth\_token, oauth\_token\_secret with Twitter tokens with your values.
3. [Steps to generate an OAuth access token](https://dev.twitter.com/oauth/overview/application-owner-access-tokens).
4. Note that you will need to make an empty application to generate a token.
5. Replace the EventHubConnectionString and EventHubName values in App.config with your Event Hub connection string and name.
6. Optional: Adjust the keywords to search for. As a default, this application looks for **Azure, Skype, XBox, Microsoft, Seattle**. You can adjust the values for twitter\_keywords in App.config, if desired.
7. Build the solution.
8. Start the application. You will see Tweet events with the CreatedAt, Topic, and SentimentScore values being sent to your Event Hub:



Keep the solution running as we will analyze those Dates during the following exercises.

Exercise No. 3: Create Stream Analytics Job

#### Objectives

In this exercise, you will:

* Create a Stream Analytics Job.
* Configure an Input.
* Create a Query.
* Configure an Output.

Now that we have Tweet events streaming in real-time from Twitter, we can set up a Stream Analytics Job to analyze these events in real time.

Task 1: Provision a Stream Analytics Job

1. In the [Azure Portal](https://manage.windowsazure.com/), click **NEW** > **DATA SERVICES** > **STREAM ANALYTICS** > **QUICK CREATE**.
2. Specify the following values, and then click **CREATE STREAM ANALYTICS JOB**.

* **JOB NAME** Enter a job name.
* **REGION** Select the region where you want to run the job. Consider placing the job and the event hub in the same region to ensure better performance and to ensure that you will not be paying to transfer data between regions.
* **STORAGE ACCOUNT** Choose the Storage account that you would like to use to store monitoring data for all Stream Analytics jobs running within this region. You have the option to choose an existing Storage account or to create a new one.

1. Click **STREAM ANALYTICS** in the left pane to list the Stream Analytics jobs.

Stream Analytics service icon

1. The new job will be shown with a status of **CREATED**. Notice that the **START** button on the bottom of the page is disabled. You must configure the job input, output, and query before you can start the job.

Task 2: Specify Job Input

1. In your Stream Analytics Job click **INPUTS** from the top of the page, and then click **ADD INPUT**. The dialog box that opens will walk you through a number of steps to set up your input.
2. Select **DATA STREAM**, and then click the button on the right side.
3. Select **EVENT** **HUB**, and then click the button on the right side.
4. Type or select the following values on the third page:

* **INPUT ALIAS** Enter the following name for this job input **TwitterStream**. Note that you will be using this name in the query later on.
* **EVENT HUB** If the Event Hub you created is in the same subscription as the Stream Analytics job, select the namespace that the event hub is in.
* If your event hub is in a different subscription, select **Use Event Hub** from Another Subscription, and then manually enter information for SERVICE BUS NAMESPACE, EVENT HUB NAME, EVENT HUB POLICY NAME, EVENT HUB POLICY KEY, and EVENT HUB PARTITION COUNT.
* **EVENT HUB NAME** Select the name of the Event Hub
* **EVENT HUB POLICY NAME** Select the event-hub policy created earlier in this tutorial.
* **EVENT HUB CONSUMER GROUP** Type in the Consumer Group created earlier in this tutorial.

1. Click the button on the right side.
2. Specify the following values:

* **EVENT SERIALIZER FORMAT** JSON
* **ENCODING** UTF8

1. Click the **check button** to add this source and to verify that Stream Analytics can successfully connect to the event hub.

Task 3: Specify Job Query

Stream Analytics supports a simple, declarative query model for describing transformations. To learn more about the language, see the [Azure Stream Analytics Query Language Reference](https://msdn.microsoft.com/library/azure/dn834998.aspx). This tutorial will help you author and test several queries over Twitter data.

1. To validate your query against actual job data, you can use the SAMPLE DATA feature to extract events from your stream and create a JSON file of the events for testing.
2. Select your **Stream Analytics Job**, **INPUTS**, and click **SAMPLE DATA** at the bottom of the page.
3. In the dialog box that appears, specify a START TIME to start collecting data from and a DURATION for how much additional data to consume.
4. Click the **DETAILS** button, and then the **Click** **here** link to download and save the .JSON file that is generated.

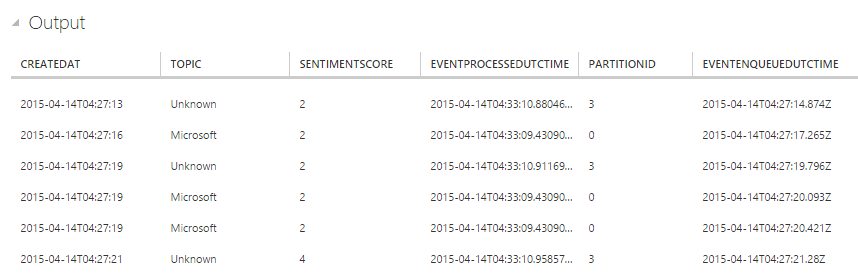
To start with, we will do a simple pass-through query that projects all the fields in an event.

1. Click **QUERY** from the top of the Stream Analytics Job page.
2. In the code editor, replace the initial query template with the following:

|  |
| --- |
| SELECT \* FROM TwitterStream |

Ensure that the name of the input source matches the name of the input you specified earlier.

1. Click **TEST** under the query editor.
2. Browse to your sample .JSON file.
3. Click the **check button** and see the results displayed below the query definition.



To compare the number of mentions between topics, we will use a [TumblingWindow](https://msdn.microsoft.com/library/azure/dn835055.aspx) to get the count of mentions by topic every 5 seconds.

1. Change the query in the code editor to:

|  |
| --- |
| SELECT System.Timestamp as Time, Topic, COUNT(\*)  FROM TwitterStream TIMESTAMP BY CreatedAt  GROUP BY TUMBLINGWINDOW(s, 5), Topic |

Note that this query uses the TIMESTAMP BY keyword to specify a timestamp field in the payload to be used in the temporal computation. If this field was not specified, the windowing operation would be performed using the time each event arrived at Event Hub. Learn more under **Arrival Time Vs Application Time** in the [Stream Analytics Query Reference](https://msdn.microsoft.com/library/azure/dn834998.aspx).

This query also accesses a timestamp for the end of each window with System.Timestamp

1. Click **RERUN** under the query editor to see the results of the query.

To identify trending topics, we will look for topics that cross a threshold value for mentions in a given amount of time. For the purposes of this tutorial, we will check for topics that are mentioned more than 20 times in the last 5 seconds using a [SlidingWindow](https://msdn.microsoft.com/library/azure/dn835051.aspx).

1. Change the query in the code editor to:

|  |
| --- |
| SELECT System.Timestamp as Time, Topic, COUNT(\*) as Mentions  FROM TwitterStream TIMESTAMP BY CreatedAt  GROUP BY SLIDINGWINDOW(s, 5), topic  HAVING COUNT(\*) > 20 |

1. Click **RERUN** under the query editor to see the results of the query.



The final query we will test uses a TumblingWindow to obtain the number of mentions and average, minimum, maximum, and standard deviation of sentiment score for each topic every 5 seconds.

1. Change the query in the code editor to:

|  |
| --- |
| Copy to clipboardSELECT System.Timestamp as Time, Topic, COUNT(\*), AVG(SentimentScore), MIN(SentimentScore),  Max(SentimentScore), STDEV(SentimentScore)  FROM TwitterStream TIMESTAMP BY CreatedAt  GROUP BY TUMBLINGWINDOW(s, 5), Topic |

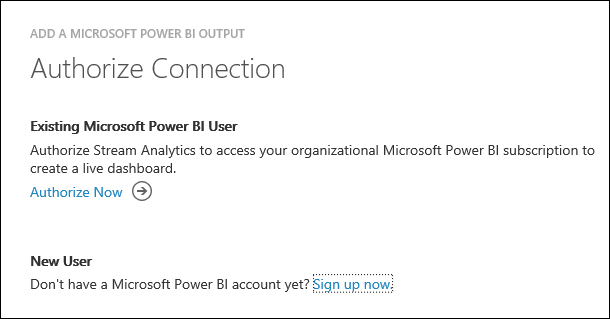
1. Click **RERUN** under the query editor to see the results of the query.
2. This is the query we will use for our dashboard. Click **SAVE** at the bottom of the page.

Exercise No. 4: Create Power BI dashboard

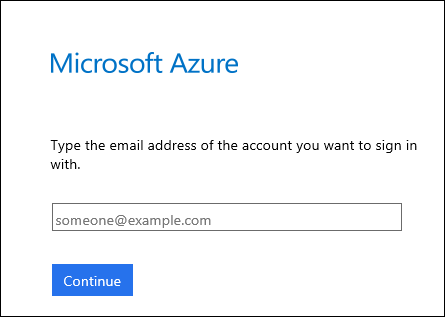
Now that we have defined an event stream, an Event Hub input to ingest events, and a query to perform a transformation over the stream, the last step is to define an output sink for the job. We will write the aggregated tweet events from our job query to an Azure Blob. You could also push your results to SQL Database, Table Store or Event Hub, depending on your specific application needs.

Power BI can be utilized as an output for a Stream Analytics job to provide for a rich visualization experience for Stream Analytics users. This capability can be used for operational dashboards, report generation, and metric driven reporting. For more information on Power BI visit the [Power BI](https://powerbi.microsoft.com/) site.

1. Click **Output** from the top of the page, and then click **Add Output**. Select **Power BI** as the output option.



1. A screen like the following is presented.



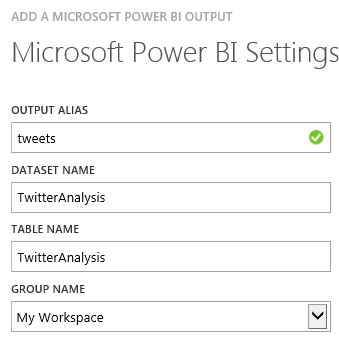
1. In this step, provide the work or school account for authorizing the Power BI output. If you are not already signed up for Power BI, choose **Sign up now**.
2. Next, a screen like the following will be presented:

There are a few parameters that are needed to configure a Power BI output.

**Output Alias** Any friendly-named output alias that is easy to refer to. This output alias is particularly helpful if it is decided to have multiple outputs for a job. In that case, this alias will be referred to in your query. For example, use the output alias value = **OutPbi**.

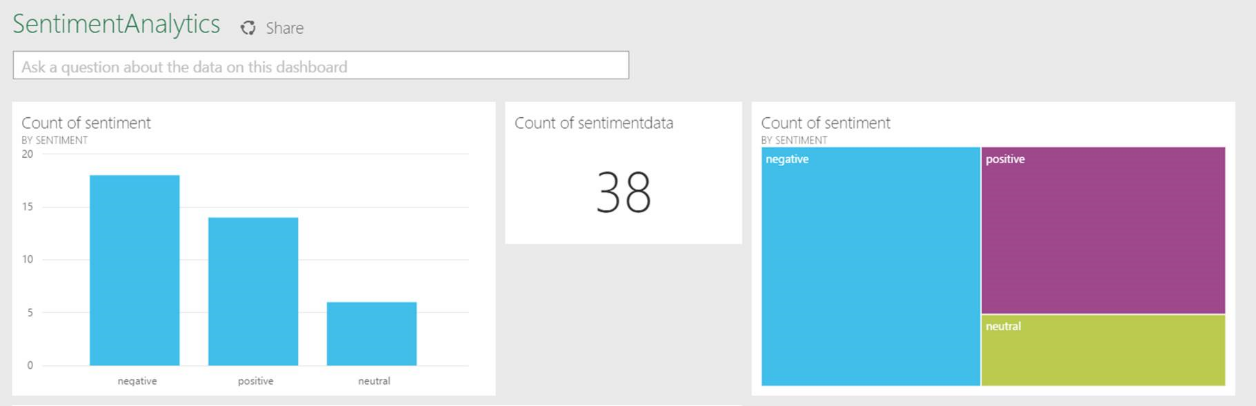
**Dataset Name** Provide a dataset name that it is desired for the Power BI output to use. For example, use **pbidemo**.

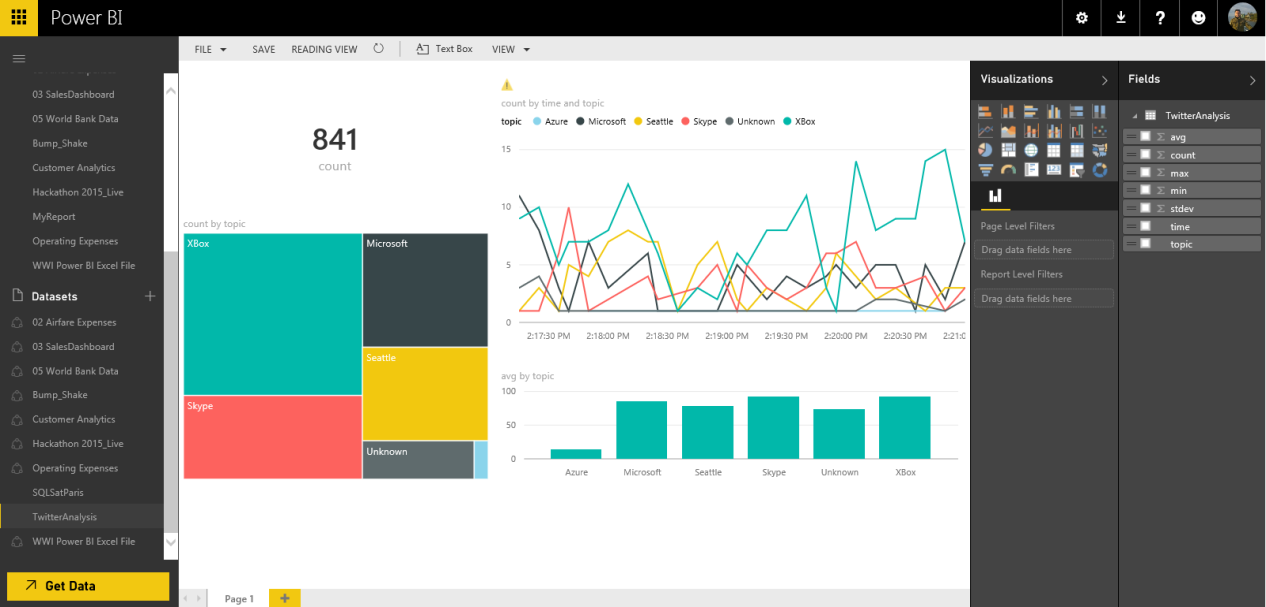
**Table Name** Provide a table name under the dataset of the Power BI output. For example, use **pbidemo**. Currently, Power BI output from Stream Analytics jobs may only have one table in a dataset.

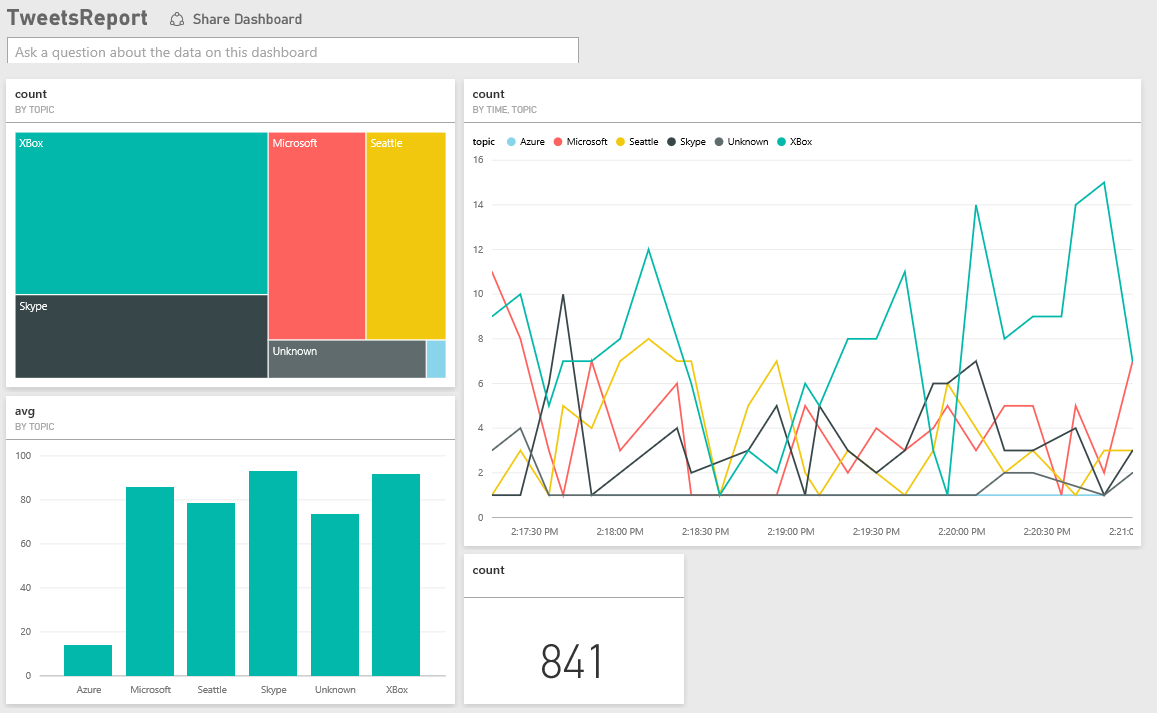


**Note** One should not explicitly create the dataset and table in the Power BI dashboard. The dataset and table will be automatically populated when the job is started and the job starts pumping output into Power BI. Note that if the job query does not return any results, the dataset and table will not be created. Also, be aware that if Power BI already had a dataset and table with the same name as the one provided in this Stream Analytics job, the existing data will be overwritten.

1. Click **OK**, Test Connection and now the output configuration is complete.
2. Connect to PowerBI.com and create a Dashboard dragging the datas in the center of the page and choosing some charts representations:







# Lab 2: Big Data Analysis

#### Introduction

In this tutorial, you will learn how to analyze Tweets events with HDInsight Spark.

#### Objectives

After completing this lab, you will be able to:

* Create an HDInsight cluster.
* Create Hive Table.
* Run interactive Spark SQL queries using a Zeppelin notebook.

#### Prerequisites

* An Azure account is required for this tutorial.

#### Estimated Time to Complete This Lab

50 minutes

#### Scenario

Real Time Analysis on share products from the Windows 10 Application previously developed will help customers to make the right choice regarding the products that they want to buy. Saving this data for further analysis can help knowing tendencies, consumer habits, etc. It also allows to combine more data, do complex queries, do predictive analysis, and gain insight.

Exercise No. 1: Adding outputs to Stream Analytics

#### Objectives

In this exercise, you will:

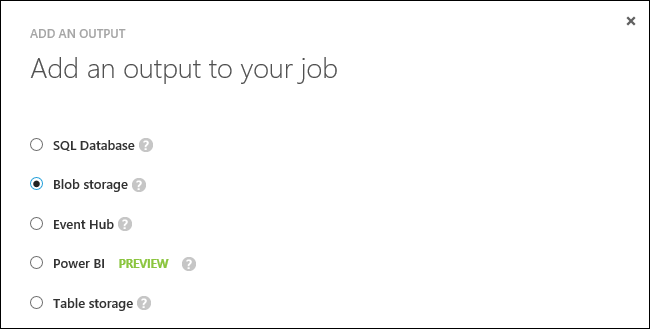
* Configure the application Twitter.
* Run the application and verify Tweets and their sentiment.

Task Description

Now that we have defined an event stream, an Event Hub input to ingest events, and a query to perform a transformation over the stream, we will define another output sink for the job. We will write the non-aggregated tweet events from our job query to an Azure Blob.

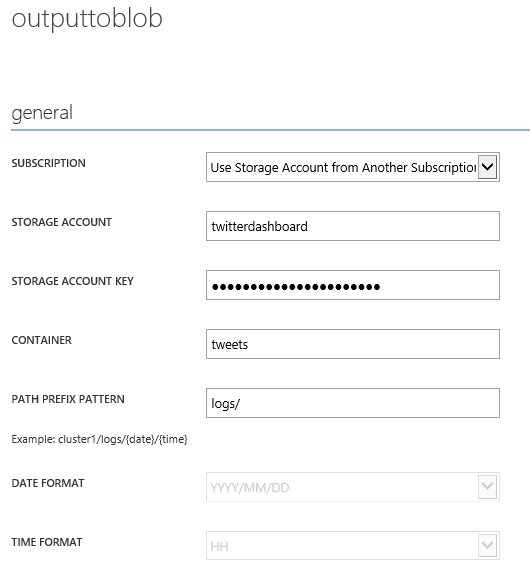
Follow the steps below to create a container for Blob storage, if you do not already have one:

1. Use an existing Storage account or create a new Storage account by clicking **NEW** > **DATA** **SERVICES** > **STORAGE** > **QUICK** **CREATE** > and following the instructions on the screen.
2. Select the **Storage account** and then click **CONTAINERS** at the top of the page, and then click **ADD**.
3. Specify a **NAME** for your container and set its ACCESS to Public Blob.
4. In your Stream Analytics job, click **OUTPUT** at the top of the page, and then click **ADD** **OUTPUT**. The dialog box that opens will walk you through a number of steps to set up your output.
5. Select **BLOB** **STORAGE**, and then click the button on the right side.



1. Type or select the following values on the third page:

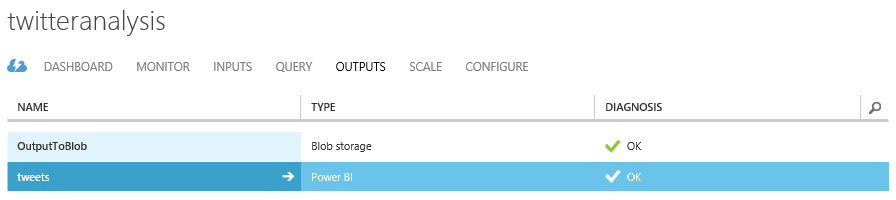
* **OUTPUT** **ALIAS**: Enter a friendly name for this job output (OutputToBlob).
* **SUBSCRIPTION:** If the Blob Storage you created is in the same subscription as the Stream Analytics job, select Use Storage Account from Current Subscription. If your storage is in a different subscription, select Use Storage Account from Another Subscription and manually enter information for STORAGE ACCOUNT, STORAGE ACCOUNT KEY and CONTAINER.
* **STORAGE** **ACCOUNT**: Select the name of the Storage Account.
* **CONTAINER**: Select the name of the Container.
* **FILENAME** **PREFIX:** Type in a file prefix to use when writing blob output.



1. Click the button on the right side.
2. Specify the following values:

* **EVENT** **SERIALIZER** **FORMAT:** CSV
* **DELIMITER**: Semicolon (;)
* **ENCODING**: UTF8

1. Click the **check** **button** to add this source and to verify that Stream Analytics can successfully connect to the storage account.



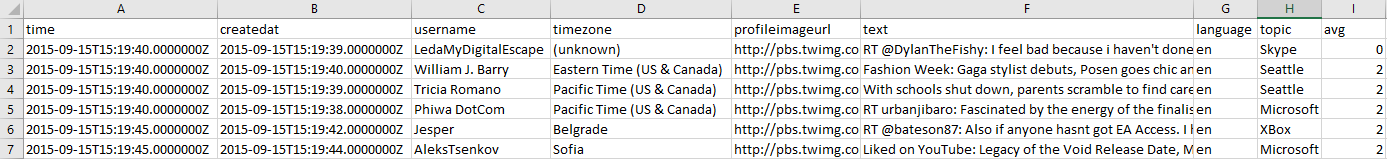
1. Change the query in the code editor to:

|  |
| --- |
| SELECT  System.Timestamp as Time,  Topic,  COUNT(\*),  AVG(SentimentScore),  MIN(SentimentScore),  MAX(SentimentScore),  STDEV(SentimentScore)  INTO tweets  FROM TwitterStream TIMESTAMP BY CreatedAt  GROUP BY TUMBLINGWINDOW(s, 5), Topic  SELECT  System.Timestamp as Time,  CreatedAt,  UserName,  TimeZone,  ProfileImageUrl,  Text,  Language,  Topic,  AVG(SentimentScore) AS SentimentScore  INTO  OutputToBlob  FROM TwitterStream TIMESTAMP BY CreatedAt  WHERE LANGUAGE = 'en'  GROUP BY TUMBLINGWINDOW(s, 5), CreatedAt, UserName, TimeZone, ProfileImageUrl, Text, Language, Topic |

Note that this query uses the TIMESTAMP BY keyword to specify a timestamp field in the payload to be used in the temporal computation. If this field were not specified, the windowing operation would be performed using the time each event arrived at Event Hub.

Since a job input, query and output have all been specified, we are ready to start the Stream Analytics job.

1. From the job DASHBOARD, click **START** at the bottom of the page.
2. In the dialog that appears, select **JOB START TIME**, and then click the **checkmark** button on the bottom of the dialog. The job status will change to Starting and will shortly move to Running.
3. Verify that the job generates files in the Blob Storage and open one of them:



Exercise No. 2: Create HDInsight Spark Cluster

#### Objectives

In this exercise, you will:

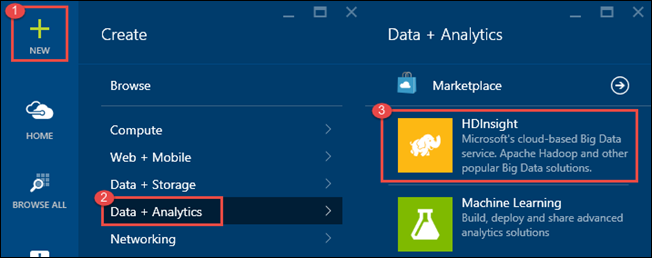
* Create an HDInsight Spark Cluster.

Task Description

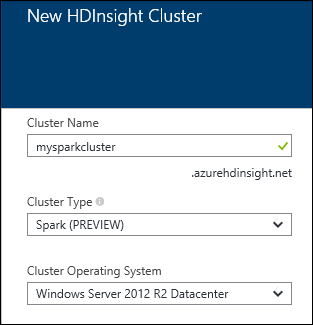
Spark clusters on Microsoft HDInsight use an Azure Blob storage container as the default file system. An Azure Storage account located on the same data center is required before you can create an HDInsight cluster. For more information, see [Use Azure Blob Storage with HDInsight](https://azure.microsoft.com/en-us/documentation/articles/hdinsight-hadoop-use-blob-storage/). For details on creating an Azure Storage account, see [How to Create a Storage Account](https://azure.microsoft.com/en-us/documentation/articles/storage-create-storage-account/).

**To create an HDInsight cluster by using the Custom Create option**

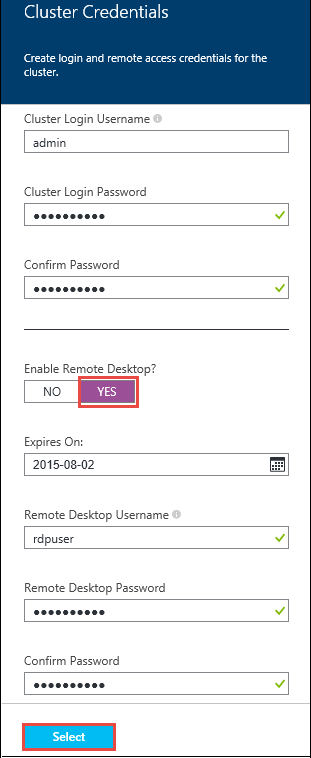
1. Sign in to the [Azure Preview Portal](https://portal.azure.com/).
2. Click **NEW**, click **Data + Analytics**, and then click **HDInsight**.



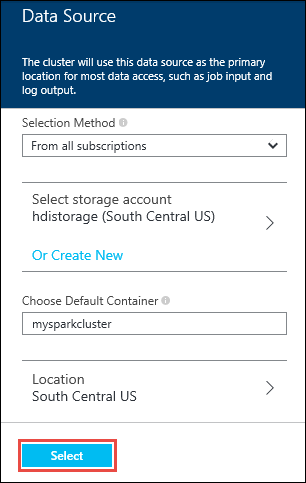
1. Enter a **Cluster Name**, select **Spark** for the **Cluster Type**, and from the **Cluster Operating System** drop-down menu, select **Windows Server 2012 R2 Datacenter**. A green check mark will appear beside the cluster name if it is available.



1. If you have more than one subscription, click the **Subscription** entry to select the Azure subscription that will be used for the cluster.
2. Click **Resource Group** to see a list of existing resource groups and then select the one to create the cluster in. Alternatively, you can click **Create New** and then enter the name of the new resource group. A green check mark will appear to indicate if the new group name is available.
3. Click **Credentials** and then enter a Cluster Login Username and Cluster Login Password. If you want to enable remote desktop on the cluster node, for Enable Remote Desktop, click **Yes**. Select a date when remote desktop access to the cluster expires, and provide the username/password for the remote desktop user. To save the credentials configuration click **Select** at the bottom.



1. Click **Data** **Source** to choose an existing data source for the cluster, or create a new one.



1. Currently, you can select an Azure Storage Account as the data source for an HDInsight cluster. Use the following to understand the entries on the Data Source blade:

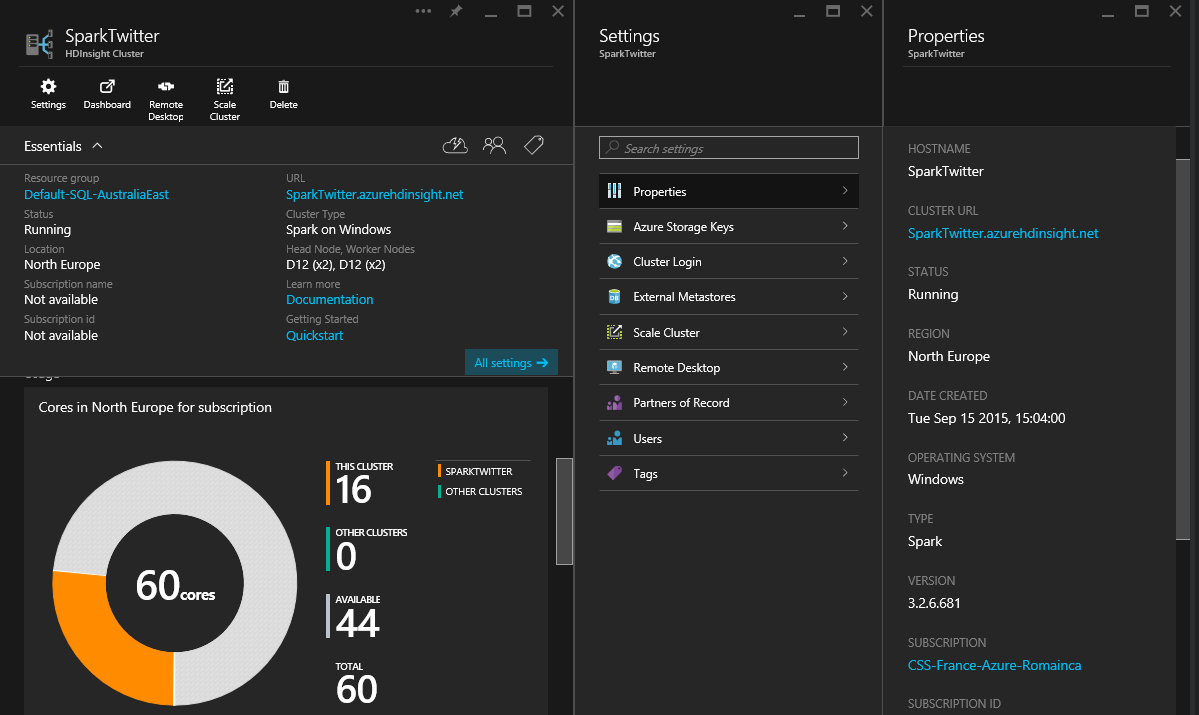
* **Selection** **Method:** Set this to enable browsing of storage accounts from all your subscriptions. Set this to Access Key if you want to enter the Storage Name and Access Key of an existing storage account.
* **Select storage account / Create New:** Click **Select storage** account to browse and select the **existing storage account** (created before).
* **Choose** **Default** **Container**: Use this to enter the name of the default container to use for the cluster. While you can enter any name here, we recommend using the same name as the cluster so that you can easily recognize that the container is used for this specific cluster.
* **Location:** The geographic region that the storage account is in, or will be created in. Selecting the location for the default data source will also set the location of the HDInsight cluster. The cluster and default data source must be located in the same region. Click **Select** to save the data source configuration.

1. Click **Node** **Pricing** **Tiers** to display information about the nodes that will be created for this cluster. Set the number of worker nodes that you need for the cluster. The estimated cost of the cluster will be shown within the blade. Click **Select** to save the node pricing configuration.
2. On the **New HDInsight Cluster** blade, ensure that **Pin to Startboard** is selected, and then click **Create**. This will create the cluster and add a tile for it to the Startboard of your Azure Portal. The icon will indicate that the cluster is provisioning, and will change to display the HDInsight icon once provisioning has completed.

|  |  |
| --- | --- |
| While provisioning | Provisioning complete |
| Provisioning indicator on startboard | Provisioned cluster tile |

It will take some time for the cluster to be created, usually around 15 minutes. Use the tile on the Startboard or the **Notifications** entry on the left side of the page to check on the provisioning process.

1. After the provisioning completes, click the **tile** for the Spark cluster from the Startboard to launch the cluster blade.



Exercise No. 3: Create Hive External Table

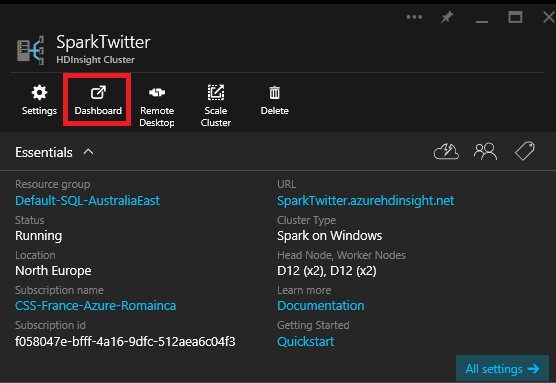
#### Objectives

In this exercise, you will:

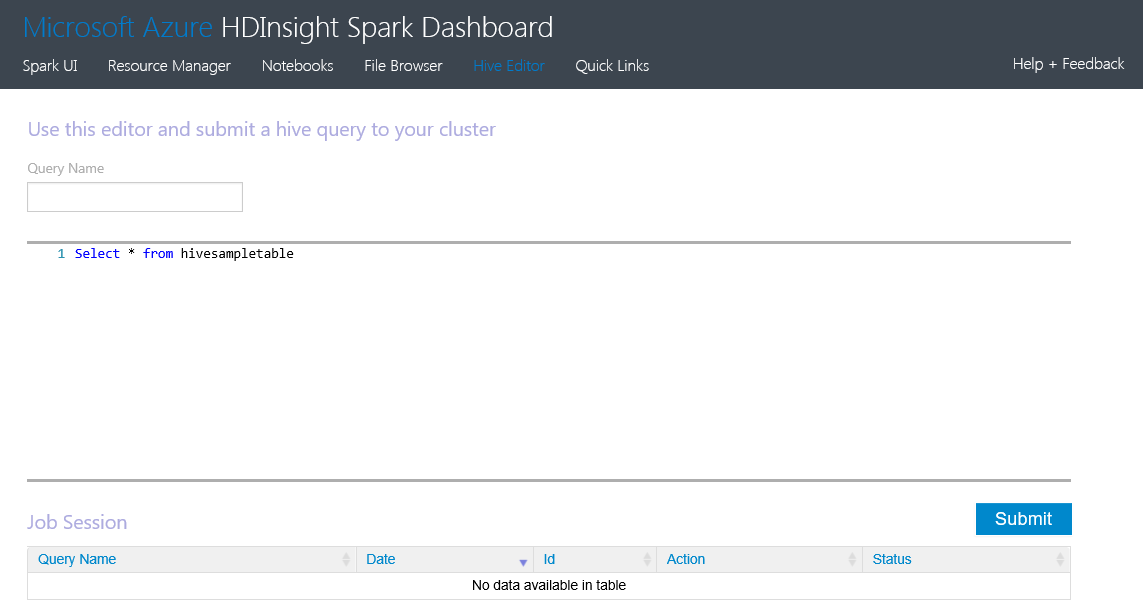
* Create an External Hive Table.

Task Description

1. Click on **Dashboard** and logon with your credentials.



1. On the Hive Editor, execute the following HQL query:

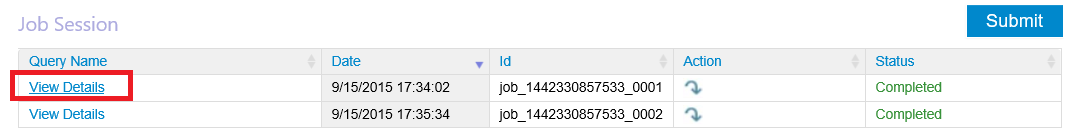


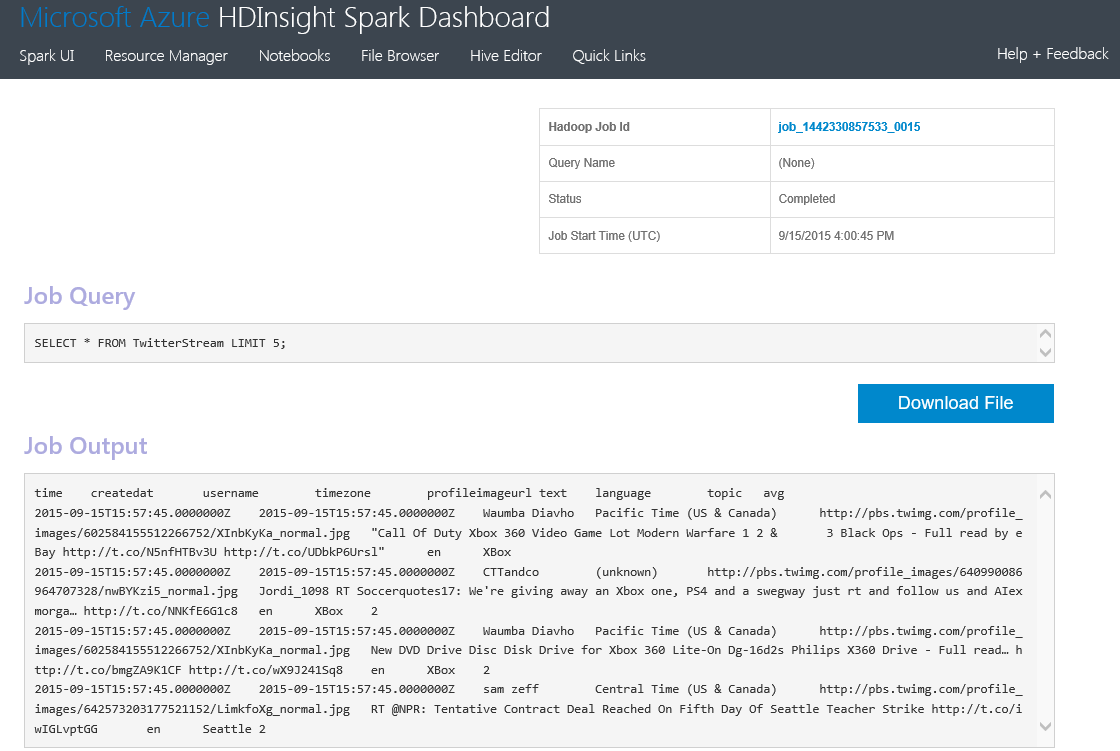
|  |
| --- |
| CREATE EXTERNAL TABLE TwitterStream (  Time STRING,  CreatedAt STRING,  UserName STRING,  TimeZone STRING,  ProfileImageUrl STRING,  Text STRING,  Language STRING,  Topic STRING,  SentimentScore STRING  )  ROW FORMAT DELIMITED FIELDS TERMINATED BY '\;'  STORED AS TEXTFILE LOCATION 'wasb://tweets@twitterdashboard.blob.core.windows.net/logs' |

**Note**: The yellow part need to be changed regarding your Blob Storage.

1. Verify the output of the following query, viewing the Job details:

|  |
| --- |
| SELECT \* FROM TwitterStream LIMIT 5; |





Exercise No. 4: Run Interactive Spark SQL queries using a Zeppelin notebook

#### Objectives

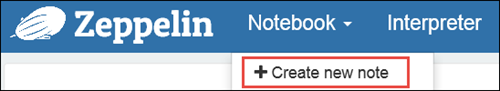
In this exercise, you will: Run interactives Spark SQL queries using Zeppelin.

Task Description

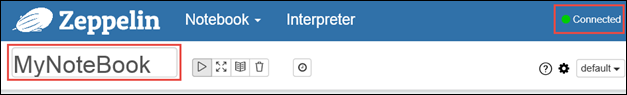
After you have provisioned a cluster, you can use a web-based Zeppelin notebook to run Spark SQL interactive queries against the Spark HDInsight cluster. In this section, we will use a sample data file (hvac.csv) available by default on the cluster to run some interactive Spark SQL queries.

Launch the Zeppelin notebook. From the Spark cluster blade, click Quick Links, and then from the Cluster Dashboard blade, click **Zeppelin Notebook**. When prompted, enter the admin credentials for the cluster. Follow the instructions on the page that opens to launch the notebook:

1. Create a new notebook. From the header pane, click **Notebook**, and then click **Create New Note**.



1. On the same page, under the Notebook heading, you should see a new notebook with the name starting with Note XXXXXXXXX. Click the **new notebook**.
2. On the web page for the new notebook, click the **heading**, and change the name of the notebook if you want to. Press the Enter key to save the name change. Also, ensure the notebook header shows a Connected status in the top-right corner.

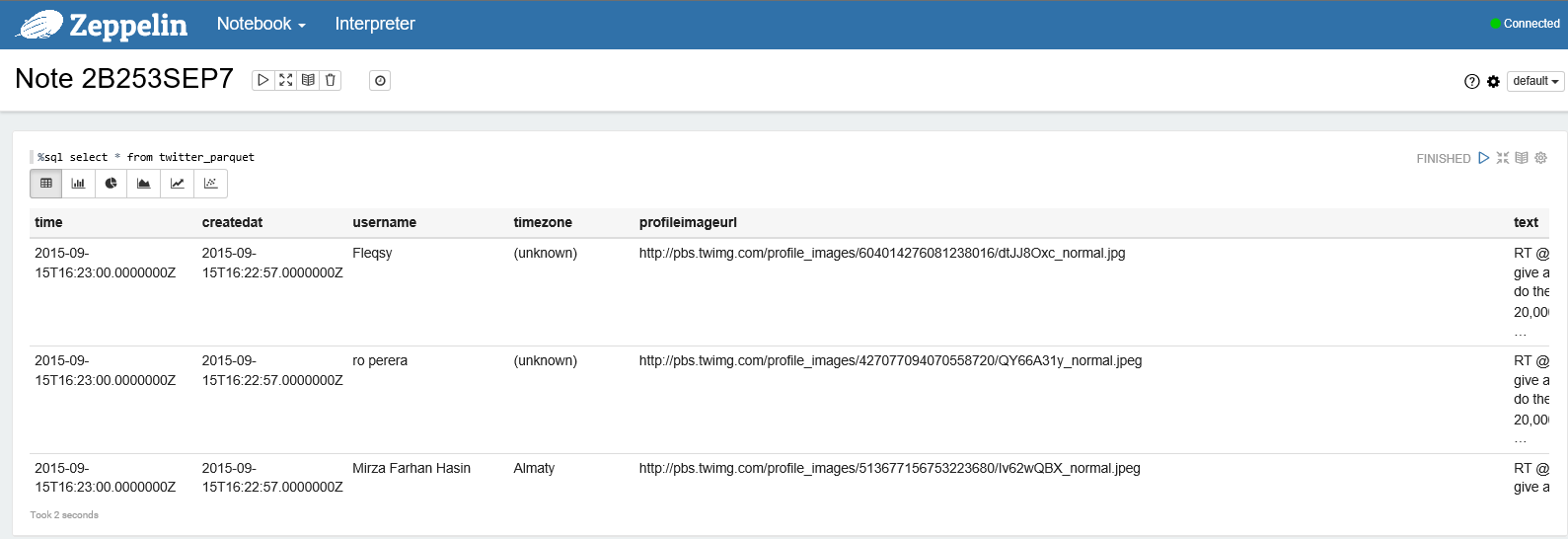


1. Return on the Hive Editor, execute the following query:

|  |
| --- |
| CREATE EXTERNAL TABLE twitter\_parquet (  Time STRING,  CreatedAt STRING,  UserName STRING,  TimeZone STRING,  ProfileImageUrl STRING,  Text STRING,  Language STRING,  Topic STRING,  SentimentScore STRING)  ROW FORMAT DELIMITED FIELDS TERMINATED BY '\;' STORED AS PARQUET;  INSERT OVERWRITE TABLE twitter\_parquet SELECT \* FROM TwitterStream WHERE Time <> 'time'; |

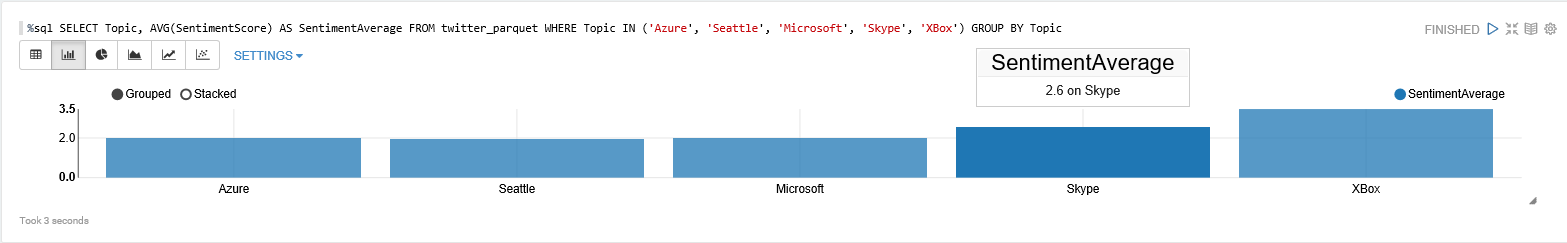
1. On the Zeppelin notebook, run the following query to view the data stored in parquet format:

|  |
| --- |
| %sql SELECT \* FROM twitter\_parquet limit 5 |

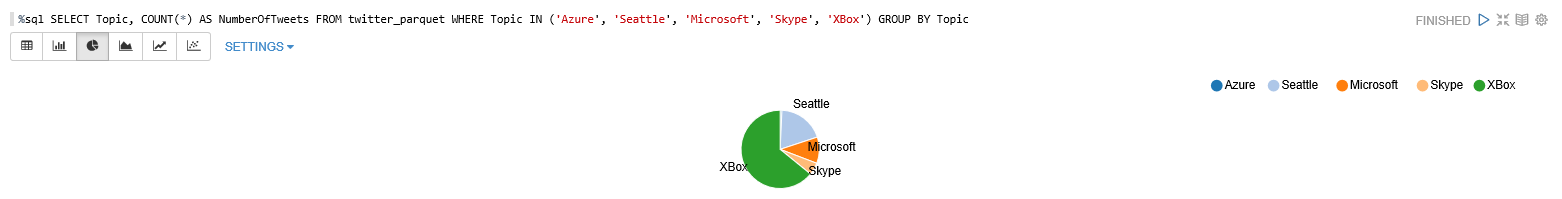


1. Analyze the data executing some of the following queries. Do not hesitate to create another one.

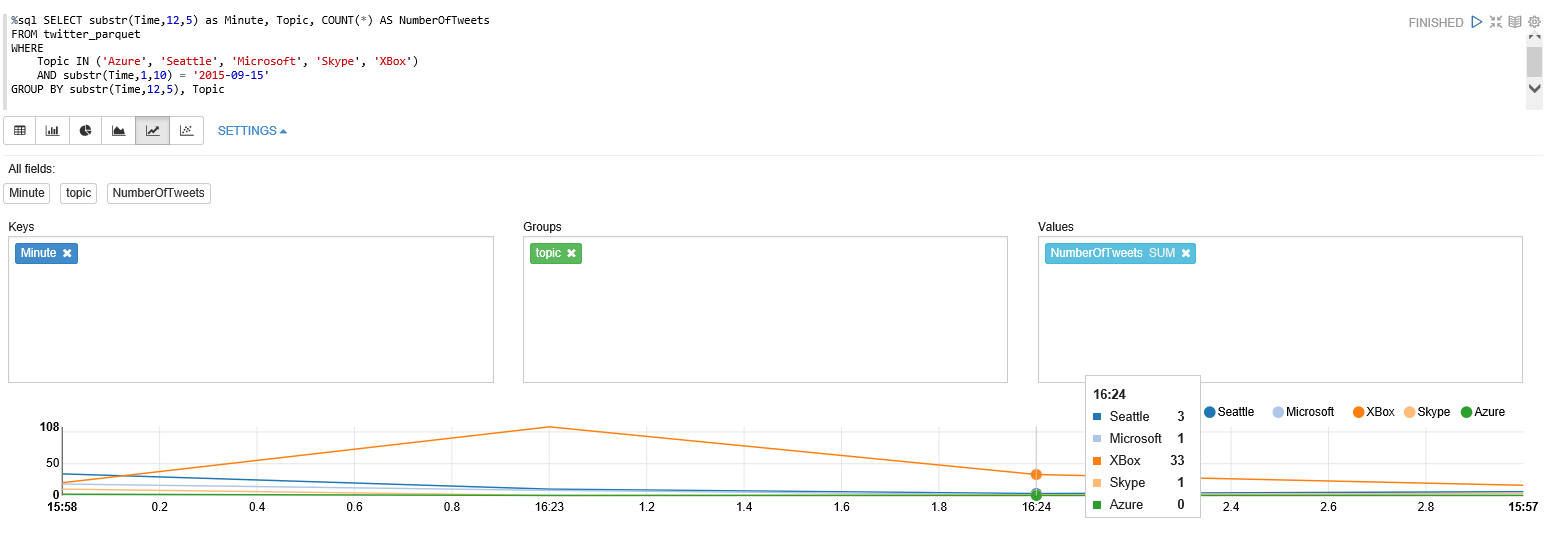
|  |
| --- |
| %sql SELECT Topic, AVG(SentimentScore) AS SentimentAverage FROM twitter\_parquet WHERE Topic IN ('Azure', 'Seattle', 'Microsoft', 'Skype', 'XBox') GROUP BY Topic |



|  |
| --- |
| %sql SELECT Topic, COUNT(\*) AS NumberOfTweets FROM twitter\_parquet WHERE Topic IN ('Azure', 'Seattle', 'Microsoft', 'Skype', 'XBox') GROUP BY Topic |



|  |
| --- |
| %sql SELECT substr(Time,12,5) as Minute, Topic, COUNT(\*) AS NumberOfTweets  FROM twitter\_parquet  WHERE  Topic IN ('Azure', 'Seattle', 'Microsoft', 'Skype', 'XBox')  AND substr(Time,1,10) = '2015-09-15'  GROUP BY substr(Time,12,5), Topic |



**Note:** Adjust the date regarding Tweets date.

|  |
| --- |
| %sql SELECT TimeZone, AVG(SentimentScore) AS SentimentAverage FROM twitter\_parquet GROUP BY TimeZone |

