Real-Time Big Data Processing

Lab 3 – Processing Real-Time Data with Stream Analytics

Overview

In this lab, you will create an Azure Stream Analytics job to process simulated device data from the applications you generated in labs 1 and 2.

What You'll Need

To complete the labs, you will need the following:

- A web browser
- A Microsoft account
- A Microsoft Azure subscription
- A Windows, Linux, or Mac OS X computer
- The lab files for this course
- The Azure resources created in the previous labs

Important: This lab depends on resources created in Lab 1 and Lab 2. If you have not completed the previous labs (or if you have deleted the resources you created), complete these labs now.

Exercise 1: Processing a Stream of Data

To process real-time data as it arrives in an event hub or IoT hub, you can use an Azure Stream Analytics job. In this procedure, you will create a simple Stream Analytics job that reads device readings from your event hub, and stores them in a blob store container.

Create a Storage Account

Your streaming solution will store its output in Azure blob storage, so you will need to create an Azure Storage account.

1. In a web browser, navigate to http://portal.azure.com, and if prompted, sign in using the Microsoft account that is associated with your Azure subscription.
2. In the Microsoft Azure portal, in the Hub Menu, click **New**. Then in the **Storage** menu, click **Storage account**.
3. In the **Create storage account** blade, enter the following settings and click **Create**:
    - **Name**: *Enter a unique name (and make a note of it!)*
    - **Deployment model**: Resource manager
    - **Account kind**: Storage (General purpose v1)
    - **Location**: *Select the region where you created your event hub*

- **Replication**: Locally-redundant storage (LRS)
- **Performance**: Standard
- **Secure transfer required**: Disabled
- **Subscription**: *Select your Azure subscription*
- **Resource group**: *Use the existing resource group you created in the previous procedure*
- **Virtual Networks:** Disabled

4. In the Azure portal, view **Notifications** to verify that deployment has started. Then wait for the storage account to be deployed (this can take a few minutes.)

Create a Stream Analytic Job

The first step in using Stream Analytics to process real-time data is to create a Stream Analytics job.

1. In the Microsoft Azure portal, in the Hub Menu, click **New**. Then in the **Internet of Things** menu, click **Stream Analytics job**.
2. In the **New Stream Analytics Job** blade, enter the following settings, and then click **Create**:
   - **Job Name**: *Enter a unique name (and make a note of it!)*
   - **Subscription**: *Select your Azure subscription*
   - **Resource Group**: *Select the resource group containing your existing resources*
   - **Location:** *Select any available region*
   - **Hosting environment:** Cloud
   - **Streaming units:** 1
   - **Pin to dashboard**: *Not selected*
3. In the Azure portal, view **Notifications** to verify that deployment has started. Then wait for the job to be deployed (this can take a few minutes.)

Add an Input

Stream Analytics jobs get their data from one or more *inputs*. In this procedure, you will create and sample an input for the IoT hub you created in the previous lab.

1. In the Azure portal, browse to the Stream Analytics job you created previously.
2. In the blade for your Stream Analytics job, in the **Job Topology** section, click **Inputs**.
3. In the **Inputs** blade, click **Add Stream Input** and select **IoT Hub**.
4. In the **New input** blade, enter the following settings, and then click **Save**:
   - **Input alias**: DeviceData
   - **Select IoT Hub from your subscriptions**: Selected
   - **Subscription:** *Select your subscription*
   - **IoT hub:** *Select your IoT hub*
   - **Endpoint:** Messaging
   - **Shared access policy name:** service
   - **Shared access policy key:** *Automatically selected*
   - **Consumer group**: $Default
   - **Event serialization format:** JSON
   - **Encoding:** UTF-8
   - **Event compression type:** None
4. Wait for the input to be created and tested.

Add an Output

Stream Analytics jobs return their results to an *output*. In this procedure, you will add an output to your Stream Analytics job so that the processed results are stored in Azure Blob storage.

1. In the Azure portal, browse to the Stream Analytics job you created previously.

2. In the blade for your Stream Analytics job, in the **Job Topology** section, click **Outputs**.
3. In the **Outputs** blade, click **Add** and select **Blob Storage**.
4. In the **New output** blade, enter the following settings, and then click **Save**:
    - **Output alias**: DeviceReadings
    - **Select Blob storage from your subscriptions**: Selected
    - **Subscription:** *Select your subscription*
    - **Storage account**: *Select your storage account*
    - **Container:** Create a new container named **device-readings**
    - **Path pattern:** readings/{date}
    - **Date format:** YYYY/MM/DD
    - **Time format:** *Should be unavailable*
    - **Event serialization format:** CSV
    - **Delimiter:** comma (,)
    - **Encoding:** UTF-8
5. Wait for the output to be created and tested.

Add a Query

Now that you have defined an input and an output for your Stream Analytics job, you can connect them by defining a query that will process the data stream.

1. In the Azure portal, browse to the Stream Analytics job you created previously.
2. In the blade for your Stream Analytics job, in the **Job Topology** section, click **Query**.
3. In the query blade, modify the default query that is provided for you, replacing ***YourOutputAlias*** with the output alias you specified for your output, and ***YourInputAlias*** with the input alias you specified for your input (the available inputs and outputs are shown on the left of the query editor pane):

```
SELECT
    *
INTO
    [DeviceReadings]
FROM
    [DeviceData]
```

4. Save the query.

View the Job Diagram

Your Stream Analytics job now consists of an input, connected to an output by a query. You can verify this by viewing the job diagram.

1. In the blade for your Stream Analytics job, click **Job diagram**.
2. In the **Job diagram** blade, verify that your job consists of an Event Hub input, followed by a query step, followed by a Blob Storage output.
3. Close the **Job diagram** blade.

Start the Job

A Stream Analytics job runs perpetually, processing data as it arrives.

1. In the blade for your Stream Analytics job, select the **Overview** page and click **Start**. Then in the **Start job** blade, ensure that **Now** is selected and click **Start**.
2. Wait for the streaming job to start – this can take a minute or so.

3. When the job has started, in the Node.JS console, in the **eventclient** folder, enter the following command to run device simulation script and start submitting messages to the event hub:

   ```
   node eventclient.js
   ```

4. While the script is running, start Azure Storage Explorer, and if necessary, sign into your azure subscription using your Microsoft account.
5. Expand your storage account, and then expand **Blob Containers**.
6. Double-click the **device-readings** container, and then browse through the **readings** folder, and the year, month, and date folder to view the most recent blob that has been generated by your job.
7. Download the blob to open it in a text editor or spreadsheet application, and verify that in addition to **device**, **reading** fields, it contains values for **EventProcesssedUtcTime**, **PartitionId**, **EventEnqueuedUtcTime**, and **IotHub**.
8. Close the downloaded file, and in the Node.JS console, press CTRL+C to stop the script.

Stop the Job

When you want to stop processing events, you can stop the job.

1. In the blade for your Stream Analytics job, click **Stop**. When prompted to confirm, click **Yes**.
2. Wait for the job to stop running.

Exercise 2: Extending a Streaming Solution

A Stream Analytics job can include multiple inputs and outputs, enabling you to combine data streams from multiple sources and send processed results to multiple destinations. Additionally, you can create a streaming topology that uses multiple Stream Analytics jobs to filter and route messages through event hubs. In this exercise, you will extend your streaming solution to filter readings greater than 0.5 and route them to a second Stream Analytics job, which will store then in blob storage in an alerts folder.

Add a Second Output

In this procedure, you will add a second output to your Stream Analytics job.

1. In the Azure portal, browse to the Stream Analytics job you created previously.
2. In the blade for your Stream Analytics job, in the **Job Topology** section, click **Outputs**.
3. In the **Outputs** blade, click **Add** and select **Event Hub**.
4. In the **New output** blade, enter the following settings, and then click **Create**:
   - **Output alias**: HighReadings
   - **Select Event Hub from your subscriptions**: Selected
   - **Subscription:** *Select your subscription*
   - **Event hub namespace**: *Select your event hub namespace*
   - **Event hub name:** *Select your event hub*
   - **Event hub policy name:** DeviceAccess *(this should be the name of the shared access policy you created in lab 1)*
   - **Event hub policy key:** *Automatically selected*
   - **Partition key column:** *Leave blank*
   - **Event serialization format:** JSON
   - **Encoding:** UTF-8
   - **Format:** Line separated
5. Wait for the output to be created and tested.

Modify the Query

Now that you have defined a second output for your Stream Analytics job, you can send processed data to it from your query.

1. In the Azure portal, browse to the Stream Analytics job you created previously.
2. In the blade for your Stream Analytics job, in the **Job Topology** section, click **Query**.
3. Modify the query as shown below, creating a common table expression for all readings, a SELECT statement that routes the **device**, **reading**, and **EventEnqueuedEtcTime** fields to the blob store output, and a second SELECT statement that filters rows with a reading greater than 0.5 and writes them to the new event hub output:

```
WITH [AllReadings] AS
(SELECT * FROM [DeviceData])

SELECT device, reading, EventEnqueuedUtcTime
INTO [DeviceReadings]
FROM [AllReadings]

SELECT device, reading, EventEnqueuedUtcTime
INTO [HighReadings]
FROM [AllReadings]
WHERE CAST(reading AS float) > 0.5
```

4. Save the query, and then close the query pane.

View the Job Diagram

Your Stream Analytics job now consists of an input, connected to two outputs by two steps. You can verify this by viewing the job diagram.

1. In the blade for your Stream Analytics job, click **Job diagram**.
2. In the **Job diagram** blade, verify that your job consists of an IoT Hub input, followed by two query steps, followed by a Blob Storage output and an Event Hub output.
3. Close the **Job diagram** blade.

Add a Second Stream Analytics Job

The Stream Analytics job you have created routes readings with a high value to an event hub. You will now add a second stream analytics job to process these high readings.

1. In the Microsoft Azure portal, in the Hub Menu, click **New**. Then in the **Internet of Things** menu, click **Stream Analytics job**.
5. In the **New Stream Analytics Job** blade, enter the following settings, and then click **Create**:
    - **Job Name**: *Enter a unique name (and make a note of it!)*
    - **Subscription**: *Select your Azure subscription*
    - **Resource Group**: *Select the resource group containing your existing resources*
    - **Location:** *Select any available region*
    - **Hosting environment:** Cloud
    - **Streaming units:** 1
    - **Pin to dashboard**: *Not selected*
2. In the Azure portal, view **Notifications** to verify that deployment has started. Then wait for the job to be deployed (this can take a few minutes.) Then browse to the blade for the new Stream Analytics job.
3. Add an **Event Hub** stream input to the new job, with the following settings:

- **Input alias**: HighReadings
- **Select Event Hub from your subscriptions**: Selected
- **Subscription:** *Select your subscription*
- **Event hub namespace**: *Select your event hub namespace*
- **Event hub name:** *Select your event hub*
- **Event hub policy name:** DeviceAccess *(this should be the name of the shared access policy you created in lab 1)*
- **Event hub policy key:** *Automatically selected*
- **Partition key column:** *Leave blank*
- **Event serialization format:** JSON
- **Encoding:** UTF-8
- **Format:** Line separated

6. Add a **Blob Storage** output to the stream analytics job, with the following settings:
    - **Output alias**: DeviceAlerts
    - **Select Blob storage from your subscriptions**: Selected
    - **Subscription:** *Select your subscription*
    - **Storage account**: *Select your storage account*
    - **Container:** device-readings
    - **Path pattern:** alerts/{date}
    - **Date format:** YYYY/MM/DD
    - **Time format:** *Should be unavailable*
    - **Event serialization format:** CSV
    - **Delimiter:** comma (,)
    - **Encoding:** UTF-8
5. Add the following query to the stream analytics job:

```
SELECT
    *
INTO
    [DeviceAlerts]
FROM
    [HighReadings]
```

Start the Jobs

Now you're ready to start both jobs and test the streaming topology.

1. Start the new stream analytics job and wait for it to start – this can take a minute or so.
2. Start the original stream analytics job and wait for it to start – this can take a minute or so.
3. When the jobs have started, in the Node.JS console, in the **iotdevice** folder, enter the following command to run device simulation script and start submitting messages to the IoT hub:

```
node iotdevice.js
```

4. While the script is running, start Azure Storage Explorer, and if necessary, sign into your azure subscription using your Microsoft account.
5. Expand your storage account, and then expand **Blob Containers**.
6. Double-click the **device-readings** container, and then browse through the **alerts** folder, and the year, month, and date folders to view the most recent blob that has been generated by your job.
7. Download the blob to open it in a text editor or spreadsheet application and verify that it contains readings with a value greater than 0.5.
8. Close the downloaded file, and in the Node.JS console, press CTRL+C to stop the script.

Stop the Jobs

When you want to stop processing events, you can stop the jobs.

1. In the Azure portal, stop both stream analytics jobs.

Exercise 3: Using Static Reference Data

Many real-time data processing solutions use static reference data to augment the streaming data. In this exercise, you will add a static dataset containing details of the devices submitting readings to your streaming solution.

Upload Reference Data to Azure

The device details data is provided as a text file, which you will upload to your Azure blob storage container.

1. In the folder where you extracted the lab files, open the **devices.csv** file in a text editor or spreadsheet application.
2. Review the device data, noting that the first value in each row is a device ID (in the format dev*n*) and the second value is the full name of the device (in the format Device *n*). Then close the file without saving any changes.
3. Use Azure Storage Explorer to view the **device-readings** container you created previously.
4. Create a new folder named **static-data**.
5. In the **Upload** drop-down list, click **Files**, and then upload the **devices.csv** file as a block blob to the **static-data**. folder.

Modify a Stream Analytics Job

Now that you have uploaded the static reference data you can modify the query in the first of your Stream Analytics jobs to look up the device name.

1. In the Azure portal, browse to the first Stream Analytics job you created, which reads device data from an IoT hub input, and routes the processed results to a blob storage output and an event hub output.
2. In the blade for your Stream Analytics job, in the **Job Topology** section, click **Inputs**.
3. In the **Inputs** blade, click **Add reference input** and select **Blob storage**.
4. In the **New input** blade, enter the following settings, and then click **Create**:
    - **Input alias**: DeviceDetails
    - **Select Blob storage from your subscriptions**: Selected
    - **Subscription:** *Select your subscription*
    - **Storage account**: *Select your storage account*
    - **Container:** device-readings
    - **Path pattern:** static-data/devices.csv
    - **Partition key column:** *Leave blank*
    - **Event serialization format:** CSV
    - **Delimiter:** comma (,)
    - **Encoding:** UTF-8
5. Wait for the input to be created and tested.
6. In the blade for your Stream Analytics job, in the **Job Topology** section, click **Query**.
7. Verify that the existing query looks like this:

```
WITH [AllReadings] AS
(SELECT * FROM [DeviceData])
```

```
SELECT device, reading, EventEnqueuedUtcTime
INTO [DeviceReadings]
FROM [AllReadings]

SELECT device, reading, EventEnqueuedUtcTime
INTO [HighReadings]
FROM [AllReadings]
WHERE CAST(reading AS float) > 0.5
```

8. Modify the query as shown below, joining the streaming data source to the static reference data:

```
WITH [AllReadings] AS (
 SELECT strm.*, stat.DeviceName
 FROM [DeviceData] AS strm
 JOIN [DeviceDetails] AS stat
 ON strm.device = stat.DeviceID
)

SELECT device, DeviceName, reading, EventEnqueuedUtcTime
INTO [DeviceReadings]
FROM [AllReadings]

SELECT device, DeviceName, reading, EventEnqueuedUtcTime
INTO [HighReadings]
FROM [AllReadings]
WHERE CAST(reading AS float) > 0.5
```

9. Save the query, and then close the query pane.

View the Job Diagram

Your Stream Analytics job now consists of two inputs, connected to two outputs by two steps. You can verify this by viewing the job diagram.

1. In the blade for your Stream Analytics job, click **Job diagram**.
2. In the **Job diagram** blade, verify that your job consists of an IoT Hub input and a blob storage input, followed by two query steps, followed by a Blob Storage output and an Event Hub output.
3. Close the **Job diagram** blade.

Start the Jobs

Now you're ready to start both jobs and test the streaming topology.

1. Start both analytics jobs and wait for them to start – this can take a minute or so.
2. When the jobs have started, in the Node.JS console, in the **iotdevice** folder, enter the following command to run device simulation script and start submitting messages to the IoT hub:

```
node iotdevice.js
```

3. While the script is running, start Azure Storage Explorer, and if necessary, sign into your azure subscription using your Microsoft account.
4. Expand your storage account, and then expand **Blob Containers**.

5. Double-click the **device-readings** container, and then browse through the **readings** folder, and the year, month, and date folder to view the most recent blob that has been generated by your job.
6. Download the blob to open it in a text editor or spreadsheet application and verify that it contains the device ID and name for each reading.
7. Close the downloaded file, and in the Node.JS console, press CTRL+C to stop the script.

Stop the Jobs

When you want to stop processing events, you can stop the jobs.

1. In the Azure portal, stop both stream analytics jobs.

Note: You will use the resources you created in this lab when performing the next lab, so do not delete them. Ensure that all stream analytics jobs and Node.js scripts are stopped to minimize ongoing resource usage costs.