Azure Discovery Days 2019

Data Analytics & Near Real Time Intelligence with Azure - Hands-On Lab Guide

Lab 4: Streaming Dashboard

Summary

In this hands-on lab, you will:

1. Create an Azure Stream Analytics Job which will process the enriched stream emitted by the Azure Function you built in lab 3
2. Create Stream Analytics functionality to send specific events to a streaming dashboard in PowerBI.com ("hot path analytics")
3. Create Stream Analytics functionality to send all events to the SQL database you created in lab 2 ("cold path analytics")

General Notes

Before starting this lab, please ensure you have completed the Power BI pre-requisite in Lab 0.

In this lab, you will work on the business use case that the taxi company wants to know about unhappy customers as soon as possible, so that customer service can reach out to the unhappy customers and attempt to improve the situation. The taxi company also wants all trip data to be available for later analysis in their traditional BI infrastructure.

References

- Azure Stream Analytics Documentation: https://docs.microsoft.com/azure/stream-analytics/
- Stream Analytics Query Language Reference: https://docs.microsoft.com/stream-analytics-query/stream-analytics-query-language-reference
- Stream Analytics Windowing Functions Reference: https://docs.microsoft.com/stream-analytics-query/windowing-azure-stream-analytics
- Stream Analytics output to Power BI Dashboard: https://docs.microsoft.com/azure/stream-analytics/stream-analytics-power-bi-dashboard
- Power BI Documentation: https://docs.microsoft.com/power-bi/

Architecture for this Lab

The tasks in this lab cover the following components of the overall architecture.
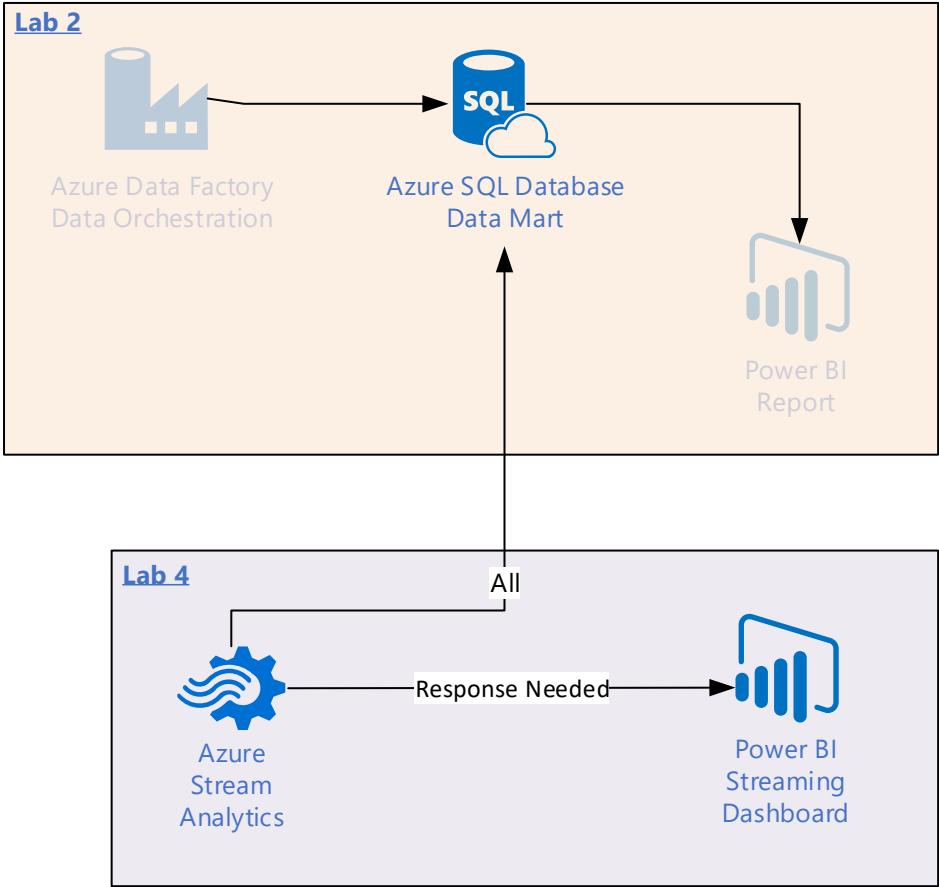
Task 1 – Create an Azure Stream Analytics Job and Send Data to Power BI

Start in your Resource Group in the Azure portal. As before, click "+ Add" to create a new Azure resource in your Resource Group. Type "stream analytics" in the search box, then click on the "Stream Analytics job" result. In the product blade, click "Create".

Provide appropriate information to create the new Stream Analytics job. Ensure you are using the same Resource Group and Azure region as in previous labs and tasks. Ensure that "Hosting environment" is set to "Cloud". Set a starting value for "Streaming units" to 6 (best practice for an initial setting per Azure Stream Analytics documentation; can be scaled as metrics/load indicate need). Then click "Create", return to your Resource Group, and wait for deployment to finish.
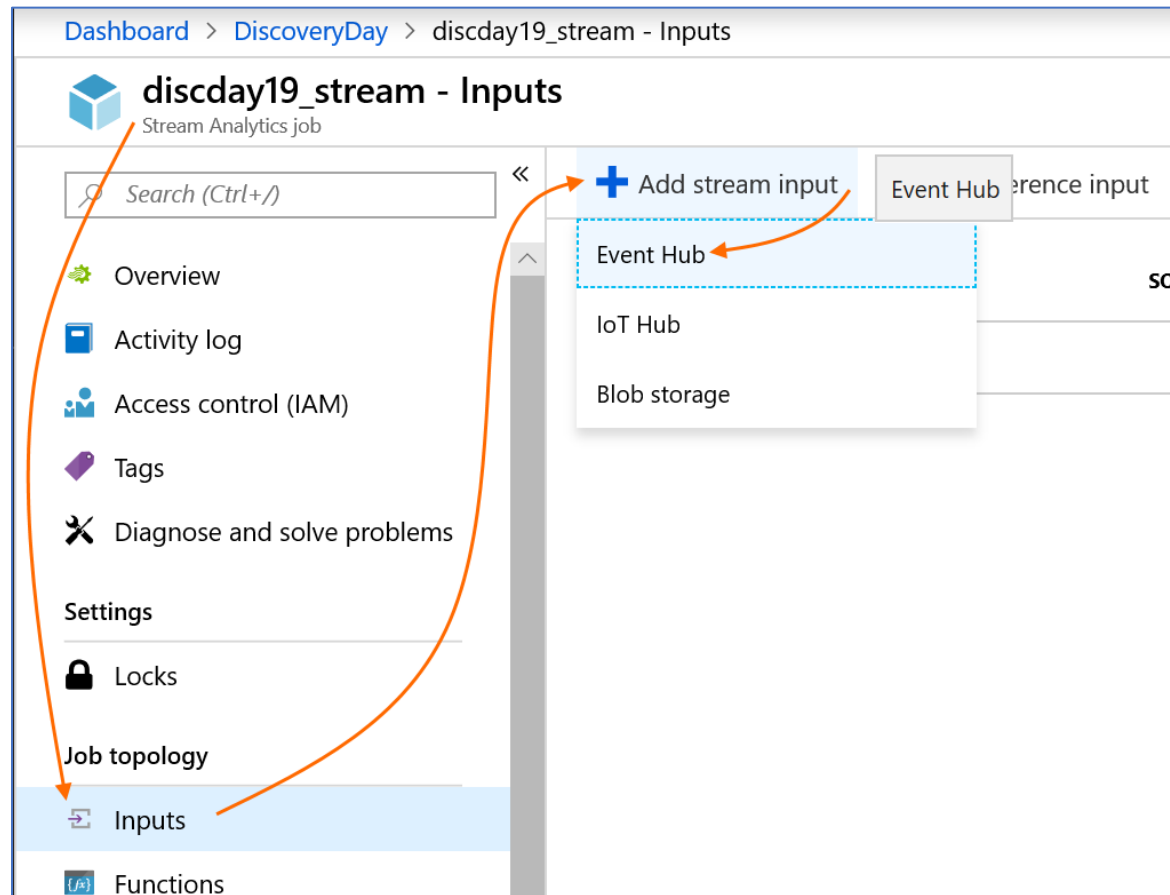
After deployment of your Stream Analytics Job completes, click on its entry in your Resource Group. In the left nav bar, under "Job topology", click "Inputs". Next, in the main view, click "+ Add stream input", then click "Event Hub".

On the new input screen, provide the required information: provide an input alias (this is a name by which you will refer to this input in analytic queries you will write later in this lab). Confirm that "Select Event Hub from your subscriptions" is selected, and then ensure that all the following settings are correct.

Ensure that you select the enriched Event Hub and the Send + Listen policy you created on the enriched Event Hub in lab 3.

You can leave "Event Hub consumer group" blank. The default ("$Default") will be used. In environments with multiple listeners for an Event Hub, you would have created additional consumer groups per listener purpose in lab 3, but we will keep it simple in this case.

When you have specified all input information, click "Save" to start the creation of this input.

Event Hub
New input

* Input alias

ehenriched ✓

○ Provide Event Hub settings manually
● Select Event Hub from your subscriptions

Subscription

Azure-FTE-201609 ⌄

* Event Hub namespace ⓘ

azdiscday2019 ⌄

* Event Hub name ⓘ
○ Create new   ● Use existing

ehenriched ⌄

* Event Hub policy name ⓘ

ehenriched_send_listen ⌄

Event Hub policy key

●●●●●●●●●●●●●●●●●●●●●●●●●●●●

Event Hub consumer group ⓘ

* Event serialization format ⓘ

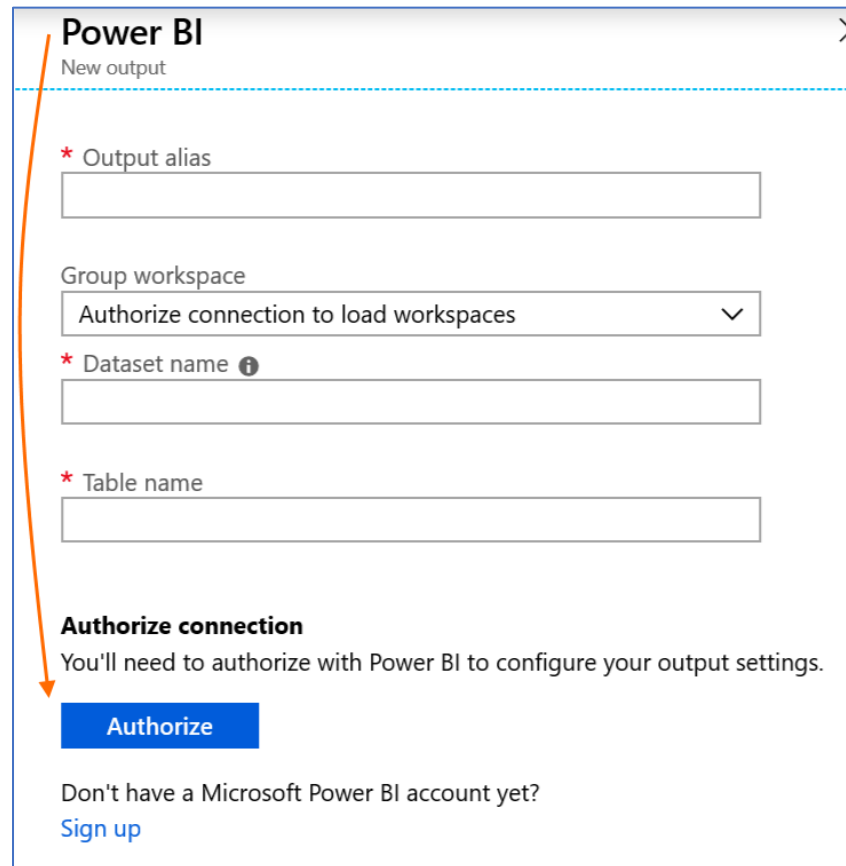JSON ⌄

Encoding ⓘ

UTF-8 ⌄

Event compression type ⓘ

None ⌄

Save

Next, in your Stream Analytics view left nav bar, again find "Job topology" and click "Outputs". Next, click "+ Add", then click on "Power BI".
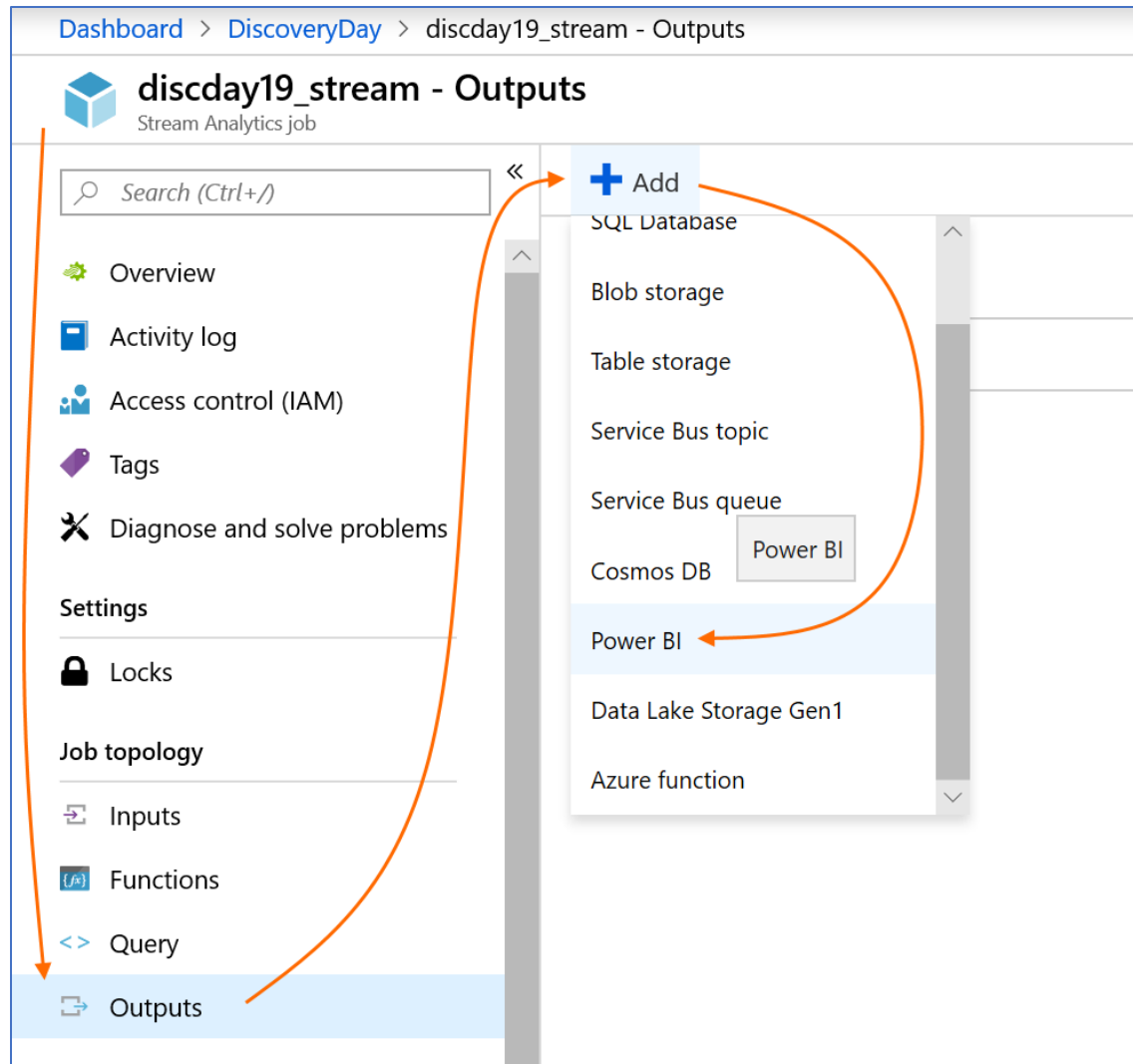
On the new output view, you will now need to authorize a connection to your PowerBI.com account, so that you can specify where Azure Stream Analytics should send enriched event messages. Click "Authorize".

After you click "Authorize", you will be asked to authenticate with credentials for PowerBI.com.

If you were given Azure credentials for this workshop and you logged into Azure with those credentials at the start of this event, please use those credentials now to authorize the connection to Power BI.

If you are using your own Azure credentials, note that this step requires you to have a valid PowerBI.com subscription. If you do not have one, please use the credentials you were given for the lab, or provision a free trial PowerBI.com subscription. Please see lab 0 for details.

After you successfully authorize the connection to PowerBI.com, provide an output alias (again, this is the name by which you will refer to this output in queries). Select a Power BI workspace (you can create one in PowerBI.com if needed, or select "My workspace"). Provide a name for the streaming dataset, then a name for the output of the first query you will write, which will project only customers with low sentiment scores. Then click "Save".

Please note: output alias names have very specific naming requirements. We suggest you use a name that is all lower case letters, without any other characters.

After you have created the input and the output, you can test them at any time (note: the Stream Analytics Job must be stopped for tests).

Click your input, then click "Test". A few seconds later, you will see a notification with the outcome. You can test your output the same way.

Next, we will add a simple Stream Analytics Query which will read incoming messages from the Event Hub input, and project selected messages to the Power BI output. Our intent is to create a streaming dashboard showing trips with unhappy customers, so our query will emit only trip messages where the sentiment score from text analytics is < 0.5. Naturally, you can adjust this threshold in your query.

In your Stream Analytics job view under "Job topology", click "<> Query". The query field will be pre-filled with a sample Stream Analytics SQL query; clear away that query text and replace it with this query:

```sql
SELECT
    *
INTO
    [pbiunhappycustomers]
FROM
    [ehenriched]
WHERE
    textanalytics_customer_sentiment_score < 0.5
```

Then click "Save" in the query editor, and "Yes" when asked to confirm the changes you made will be saved.

Note that you did not have to first create the streaming dataset or the table for this data in PowerBI.com (when you defined the Power BI output). Stream Analytics does this automatically as soon as data begins flowing.

The query simply selects all trip event messages whose sentiment score is below 0.5 (recall that the sentiment score range is 0 to 1, where 0 is negative sentiment and 1 is positive sentiment) and emits them into the Power BI output you configured previously. Feel free to adjust this value, and to elaborate the query (see References above for a link to the Stream Analytics Query Language).

Stream Analytics Query Language offers far richer capabilities than the above query uses, including custom timestamp management, aggregates, windowing functions, explicit inbound message schema management, and more.

discday19_stream - Query
Stream Analytics job

Search (Ctrl+/)

- Overview
- Activity log
- Access control (IAM)
- Tags
- Diagnose and solve problems

Settings

- Locks

Job topology

- Inputs
- Functions
- <> Query
- Outputs

💾 Save   ↻ Discard   ⚙ Test

Need help with your query? Check out some of the most common Stream Analytics query patt

▼ 🔁 Inputs (1)

    📇 ehenriched   ...

▼ 🔁 Outputs (1)

    📊 pbiunhappycustomers

```
1   SELECT
2       *
3   INTO
4       [pbiunhappycustomers]
5   FROM
6       [ehenriched]
7   WHERE
8       textanalytics_customer_sentiment_score < 0.5
9
```

Next, return to the Stream Analytics job Overview. Click "Start", then set the "Job output start time" to "Now", then click "Start". The job will take from several seconds to a minute to start; check the Notifications to track its status.

While you wait for the job to start, confirm that the device simulator you deployed in lab 3 is running, so that you have a data stream that this Stream Analytics job will use to send data to the Power BI output you configured above.

After waiting for a few seconds, in a new browser tab or window, open PowerBI.com. Go to the workspace you specified when you created the Power BI output above, select the "Datasets" tab, and verify that the streaming dataset is now shown.

On the dataset row, you can click the pencil icon ("Edit") to see the dataset's schema, including additional fields added by Stream Analytics: "EventProcessedUtcTime", "PartitionId" (this refers to the partition on the Event Hub from which Stream Analytics receives events), and "EventEnqueuedUtcTime".

Now that you have a streaming dataset in PowerBI.com, you can build a report with that data. On the dataset row, click the "Create report" icon.
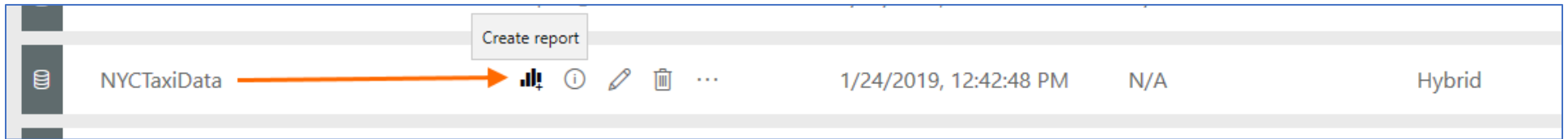
You will be taken to the report design canvas. Let's create two visualizations for our data.

First, let's create a table of individual trips so that users can see a list with data about individual unhappy customers. In the "VISUALIZATIONS" area, click on the Table visualization. Ensure that your focus is set to the table that is added to the canvas by clicking it.
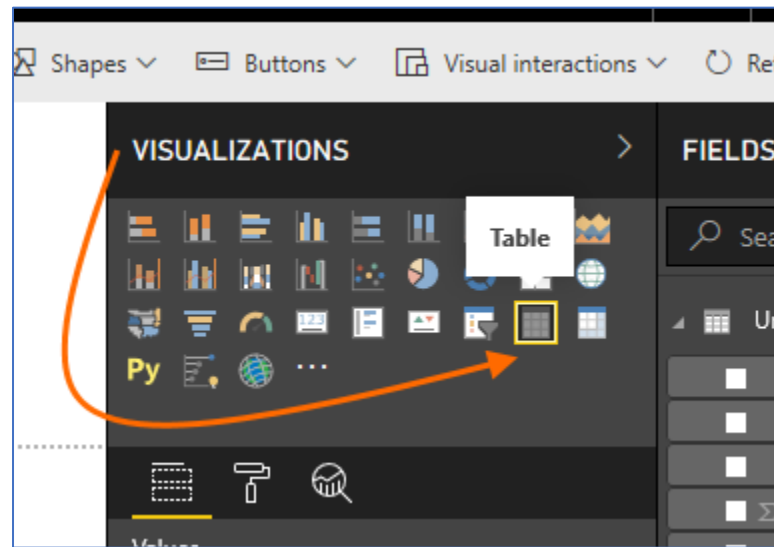
Next, in the "FIELDS" list, check the following fields: EventProcessedUtcTime, taxi_type, pickup_datetime, dropoff_datetime, trip_distance, total_amount, textanalytics_customer_sentiment_score, and customer_comments. (Feel free to try other fields too.)

FIELDS

🔍 Search

▲ 🏙 UnhappyCustomers
- ☑ customer_comments
- ☑ dropoff_datetime
- ☐ dropoff_latitude
- ☐ Σ dropoff_location_id
- ☐ dropoff_longitude
- ☐ Σ ehail_fee
- ☐ EventEnqueuedUtcTime
- ☑ EventProcessedUtcTime
- ☐ Σ extra
- ☐ Σ fare_amount
- ☐ Σ improvement_surcharge
- ☐ Σ mta_tax
- ☐ Σ PartitionId
- ☐ Σ passenger_count
- ☐ Σ payment_type
- ☑ pickup_datetime
- ☐ pickup_latitude
- ☐ Σ pickup_location_id
- ☐ pickup_longitude
- ☐ Σ rate_code_id
- ☐ store_and_fwd_flag
- ☑ taxi_type
- ☑ Σ textanalytics_customer_sentiment_score
- ☐ Σ tip_amount
- ☐ Σ tolls_amount
- ☑ Σ total_amount
- ☑ Σ trip_distance
- ☐ trip_month
- ☐ Σ trip_type
- ☐ trip_year
- ☐ Σ vendor_id

Your table visualization on the design canvas should update accordingly. Note, in the following screenshot the Totals row was turned off (under Format -> Total) but no other visual or data formatting has been applied. Feel free to experiment with your visualization's appearance.

| EventProcessedUtcTime	taxi_type	pickup_datetime	dropoff_datetime	trip_distance	total_amount	textanalytics_customer_sentiment_score	customer_comments
01/24/19 05:42:45 PM	Green	01/24/19 05:12:44 PM	01/24/19 05:41:47 PM	3.17	36.42	0.07	Car was dirty and smelled bad. Get it detailed
01/24/19 05:42:48 PM	Yellow	01/24/19 05:13:24 PM	01/24/19 05:42:07 PM	0.28	20.69	0.23	My driver tried to convert me - WTH? Just drive
01/24/19 05:42:51 PM	Green	01/24/19 04:59:46 PM	01/24/19 05:39:14 PM	2.69	1.29	0.02	Why do I have to fill out yet another survey? Leave me alone
01/24/19 05:42:52 PM	Yellow	01/24/19 05:06:46 PM	01/24/19 05:40:28 PM	3.31	27.63	0.23	My driver tried to convert me - WTH? Just drive
01/24/19 05:42:56 PM	Green	01/24/19 05:22:39 PM	01/24/19 05:41:30 PM	3.35	18.44	0.02	Why do I have to fill out yet another survey? Leave me alone
01/24/19 05:42:57 PM	Yellow	01/24/19 05:20:55 PM	01/24/19 05:41:35 PM	0.91	8.05	0.18	YOu need a better app!
01/24/19 05:43:00 PM	Yellow	01/24/19 05:31:41 PM	01/24/19 05:42:54 PM	0.15	29.44	0.05	Seats were wet from rain, no towel, cold, not happy!
01/24/19 05:43:06 PM	Green	01/24/19 04:55:54 PM	01/24/19 05:39:57 PM	2.86	33.11	0.06	Car was an ugly color, was embarrassing and dirty. Try a car wash!
01/24/19 05:43:06 PM	Green	01/24/19 04:57:04 PM	01/24/19 05:41:24 PM	1.33	22.59	0.13	Driver drove too fast, ran red lights, did not feel safe
01/24/19 05:43:09 PM	Green	01/24/19 05:20:17 PM	01/24/19 05:42:04 PM	-1.29	19.95	0.05	Seats were wet from rain, no towel, cold, not happy!
01/24/19 05:43:14 PM	Green	01/24/19 05:30:04 PM	01/24/19 05:40:39 PM	3.92	20.24	0.23	Why no coffee machines in cars? Make this happen!
01/24/19 05:43:20 PM	Green	01/24/19 04:57:22 PM	01/24/19 05:42:12 PM	4.44	10.56	0.03	Got stuck in heavy traffic, I was late. Driver should have checked GPS to find a
01/24/19 05:43:22 PM	Yellow	01/24/19 05:07:04 PM	01/24/19 05:39:28 PM	0.85	18.88	0.11	Is Penn Station really THAT hard to find? Incompetent - train your drivers
01/24/19 05:43:24 PM	Yellow	01/24/19 05:25:56 PM	01/24/19 05:41:13 PM	0.44	24.03	0.10	Driver played loud fast music. This woke me up too much!
01/24/19 05:43:33 PM	Green	01/24/19 05:14:27 PM	01/24/19 05:41:35 PM	4.40	24.06	0.03	Got stuck in heavy traffic, I was late. Driver should have checked GPS to find a
01/24/19 05:43:34 PM	Green	01/24/19 05:02:26 PM	01/24/19 05:42:02 PM	1.97	41.90	0.01	Taxi was late, driver was rude, weather was bad, and I have a zit! Ready for ton
01/24/19 05:43:40 PM	Yellow	01/24/19 05:38:31 PM	01/24/19 05:42:29 PM	3.34	15.92	0.02	Why do I have to fill out yet another survey? Leave me alone
01/24/19 05:43:43 PM	Green	01/24/19 05:25:56 PM	01/24/19 05:40:54 PM	3.86	21.30	0.18	YOu need a better app!
01/24/19 05:43:59 PM	Green	01/24/19 05:16:30 PM	01/24/19 05:41:04 PM	2.97	11.28	0.01	Taxi was late, driver was rude, weather was bad, and I have a zit! Ready for ton
01/24/19 05:44:04 PM	Green	01/24/19 04:59:32 PM	01/24/19 05:42:05 PM	-0.89	13.49	0.18	YOu need a better app!
01/24/19 05:44:05 PM	Green	01/24/19 05:30:52 PM	01/24/19 05:42:03 PM	8.11	28.09	0.00	Driver got upset with heavy traffic and yelled a lot. It was very unpleasant.
01/24/19 05:44:08 PM	Yellow	01/24/19 05:14:56 PM	01/24/19 05:41:57 PM	3.01	24.08	0.02	Why do I have to fill out yet another survey? Leave me alone

Next, in the top toolbar, click "Save".
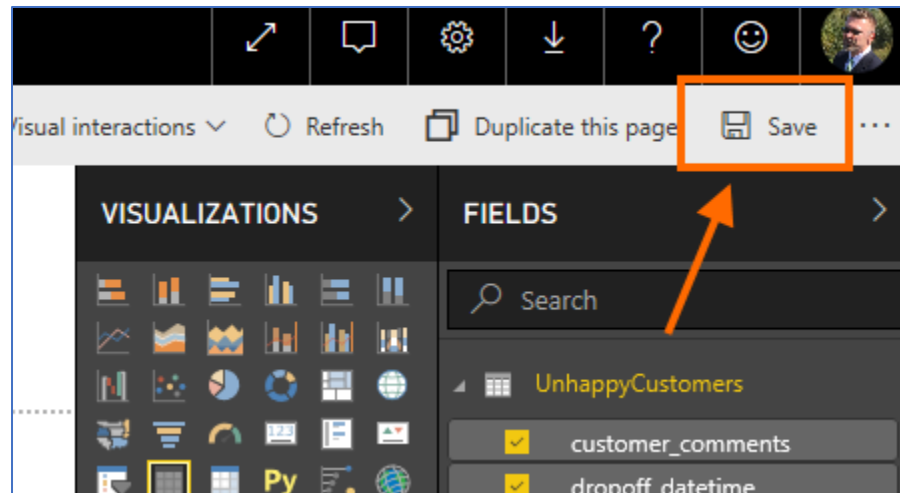
Provide a name for the Report, and click "Save".
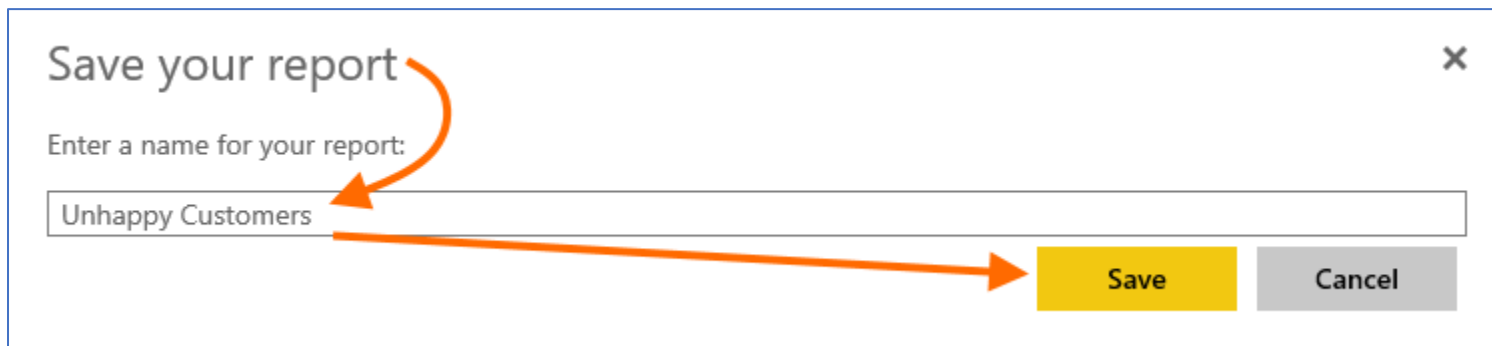
After the Report is saved, you can now click "Pin Live Page".
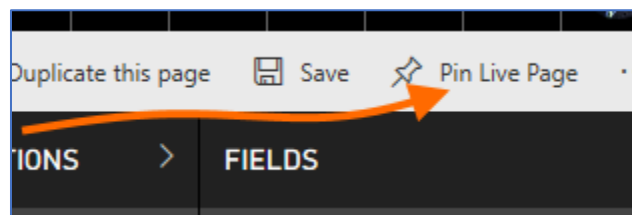
You can now create a new Dashboard. Provide a dashboard name, then click "Pin live".

After the dashboard is saved, navigate to it. You can do this from the notification shown after you click "Pin live", or by navigating to the workspace you selected, then to "Dashboards", then by clicking the Dashboard you just created.
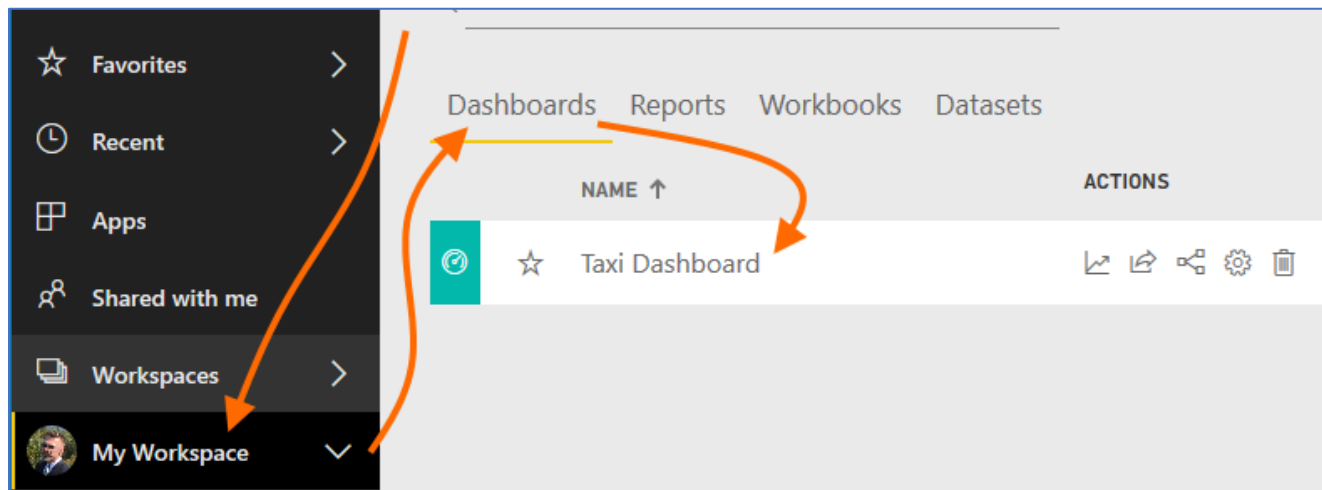
You now have a streaming dashboard in PowerBI.com which you can share (use the "Share" button). You can also add your streaming dataset to other Power BI dashboards. See the Power BI documentation for further details and feel free to experiment with adding more streaming datasets, editing the query, and so on.

Now, return to the Azure portal and to your Stream Analytics job's Overview. Click "Stop" to stop the job from running, and confirm that you want to stop the job. We cannot edit inputs, outputs, or queries while the job is running.

Task 2 –Send Data to Data Mart

In lab 2, you created an Azure SQL Database data mart. In this task, you will add an output to your Stream Analytics job and send all data received on the inbound data stream to that second output, concurrently with the Power BI output you created in task 1.

Return to your Stream Analytics job. Click on "Outputs" under "Job topology". Click "+ Add" and this time, select "SQL Database". Provide information including an output alias name (remember to use a name with all lower-case letters only), your database from lab 2, the username and password, and the destination database table. Confirm that input partitions will be merged and leave the max batch count at its default. Then click "Save".

SQL Database
New output   ✕

* Output alias

sqlallcustomers ✓

○ Provide SQL Database settings manually
● Select SQL Database from your subscriptions

Subscription

Azure-FTE-201609 ⌄

Database ⓘ

TaxiDb ⌄

Server name

discday19sql.database.windows.net

* Username

pelazem ✓

* Password

•••••••••••• ✓

* Table

dbo.trips_new ✓

● Merge all input partitions into a single writer
○ Inherit partition scheme of previous query step or input

Max batch count ⓘ

10000 ✓

Save

After the new output is saved, return to your query. Azure Stream Analytics supports multiple query statements, targeting multiple outputs. <u>Keep your previous query statement (see above) and append the following query after it – do not erase the initial query</u>.

```sql
SELECT
    trip_type,
    trip_year,
    trip_month,
    taxi_type,
    vendor_id,
    pickup_datetime,
    dropoff_datetime,
    passenger_count,
    trip_distance,
    rate_code_id,
    store_and_fwd_flag,
    pickup_location_id,
    dropoff_location_id,
    pickup_longitude,
    pickup_latitude,
    dropoff_longitude,
    dropoff_latitude,
    payment_type,
    fare_amount,
    extra,
    mta_tax,
    tip_amount,
    tolls_amount,
    improvement_surcharge,
    ehail_fee,
    total_amount,
    textanalytics_customer_sentiment_score,
    customer_comments
INTO
    [sqlallcustomers]
FROM
    [ehenriched]
```

Why is this second query explicit – i.e. why are we specifying the list of fields this time, rather than just using "SELECT *" as in the first query which outputs to Power BI? Because in Power BI, we used one of the extra fields added by Event Hub, whereas for our SQL data mart database, we are not persisting the Event Hub fields. To persist a subset of the fields in the event messages, we must specify the field list explicitly, as in this query.

Click "Save" to save the query, and confirm the save by clicking "Yes" when prompted.

Next, return to your Stream Analytics job's Overview and click "Start". This time, select to start the job "When last stopped" – this lets us avoid a gap in our data stream to Power BI due to our job stop in order to add the new SQL output and edit our query.
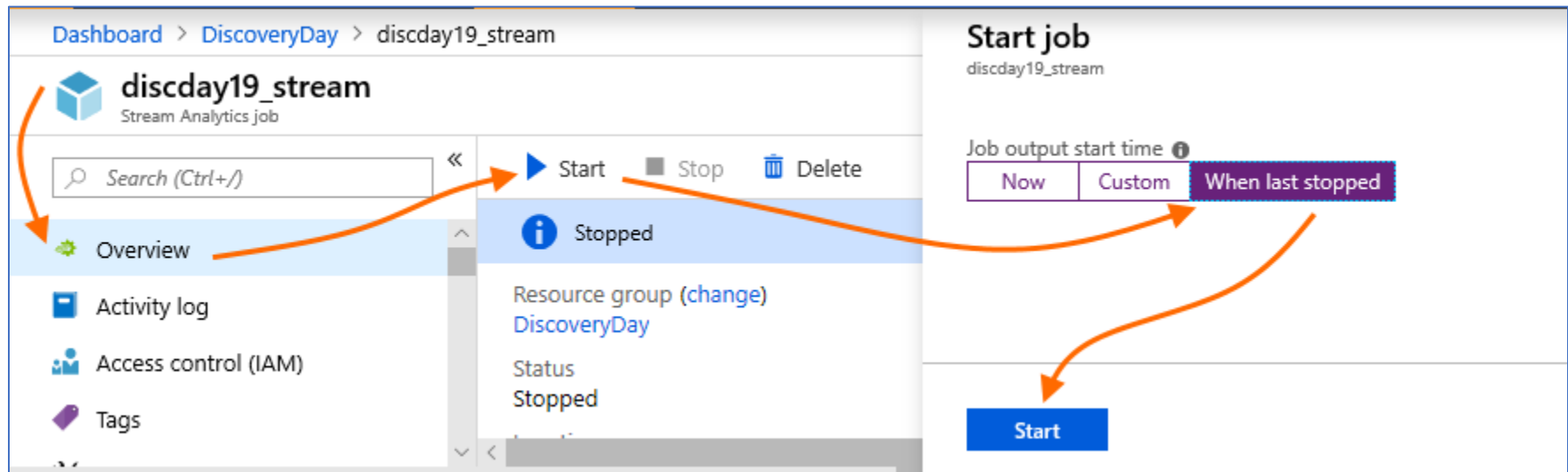
Wait until the job starts (see Notifications). Then, return to your Resource Group, select the SQL Database you created in lab 2, and click on "Query Editor" (in preview at the time of this writing). Authenticate if prompted. Then, confirm that data is now flowing into the table of streaming taxi events by entering queries like

```
SELECT COUNT(*) FROM dbo.trips_new;
```

```
SELECT TOP 100 * FROM dbo.trips_new ORDER BY dropoff_datetime DESC;
```

Feel free to modify these queries or write your own.

When you have verified that data is flowing into your SQL database table for new trips (flowing from taxi devices), this task is complete.

Conclusion

Congratulations! You have completed lab 4. In this lab, you built data analytics paths for both "hot" (streaming dashboard) and "cold" (data mart) analytics.

The paths you built in this lab are deliberately simple. There are extensive additional capabilities available for the Azure technologies you used, as well as performance tuning enhancements such as per-partition parallelized writes from Stream Analytics to SQL Database. This workshop and these labs were designed to have you build an end-to-end data analytics platform in just a few hours, and we encourage you to explore the additional capabilities.