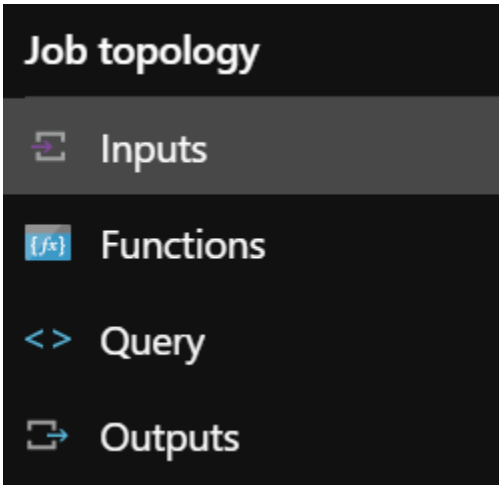
Ensure all your resources are in the same region or you will get charged extra.

Ensure you use the cheapest option for all the resources you created (Use Basic instead of Standard whenever applicable)

1. Create Blob
    a. Create a storage account
    b. Create a blob container
2. Create Event Hub
    a. Create **Namespace**
    b. Create event hub **instance** within namespace
3. Create Stream Analytics Job
    a. Set the event hub instance you have created as the **input**
    b. Set the Blob container you have created as the **output**(use Parquet as format)

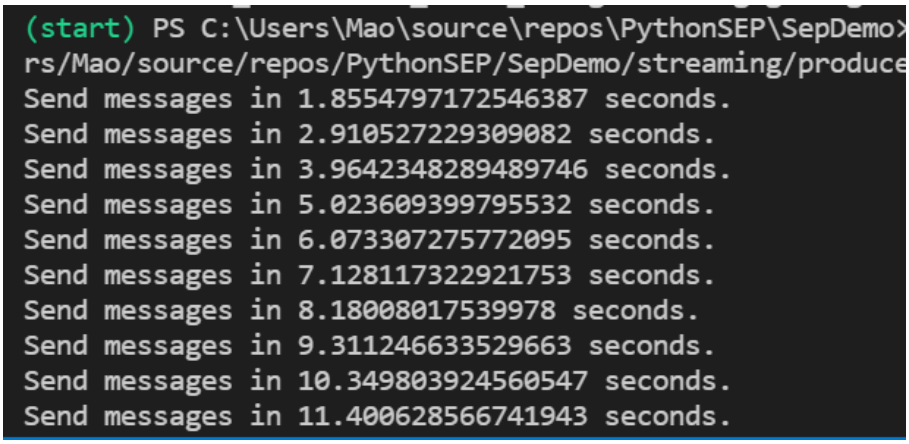


4. Producer setup
    a. Open producer.py
    b. Install event hubs library for Python using pip (might also be available through other tools like Anaconda but no guarantees)
    c. Fill **conn_str** and **eventhub_name** with the connection string and event hub name of your event hub **instance**(not the namespace)

```
#create a producer client
producer = EventHubProducerClient.from_connection_string(
        conn_str="YOUR HUB CONNECTION STRING",
        eventhub_name="YOUR HUB NAME"
        )
```

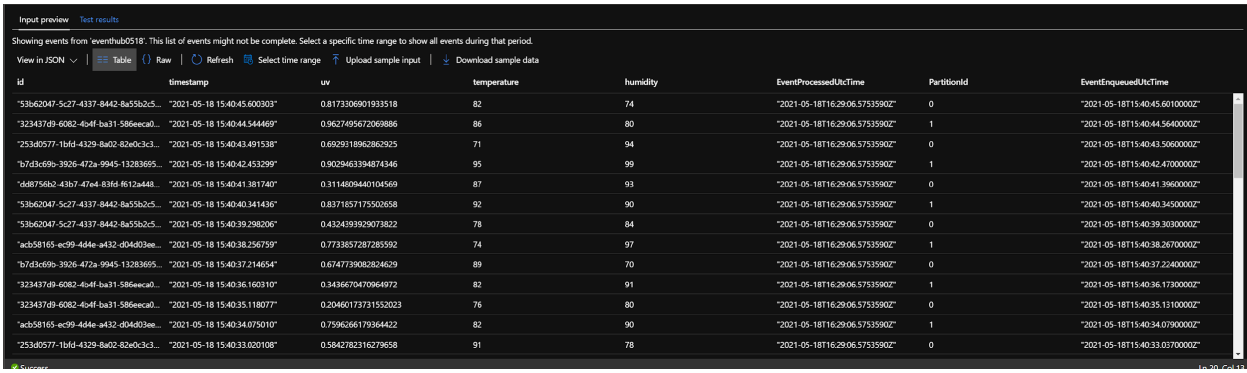
5. Test connection
    a. Run producer.py

b. If you set up everything correctly, you will see “Send messages in xx seconds.” In your terminal output. There will be roughly one message sent per second for a total of 250 messages

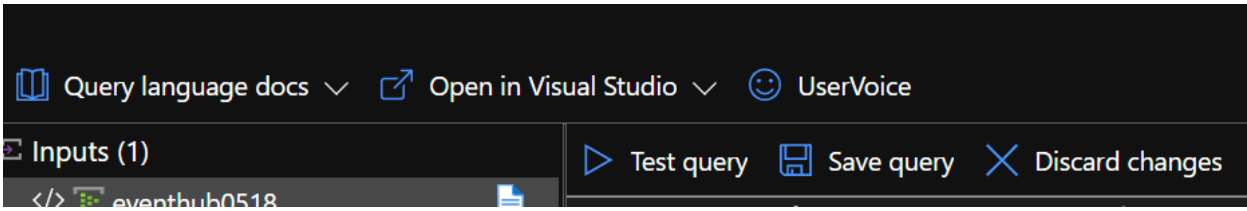


c. You should be able to see data coming in the overview dashboard for both your event hub instance and you stream analytics job.

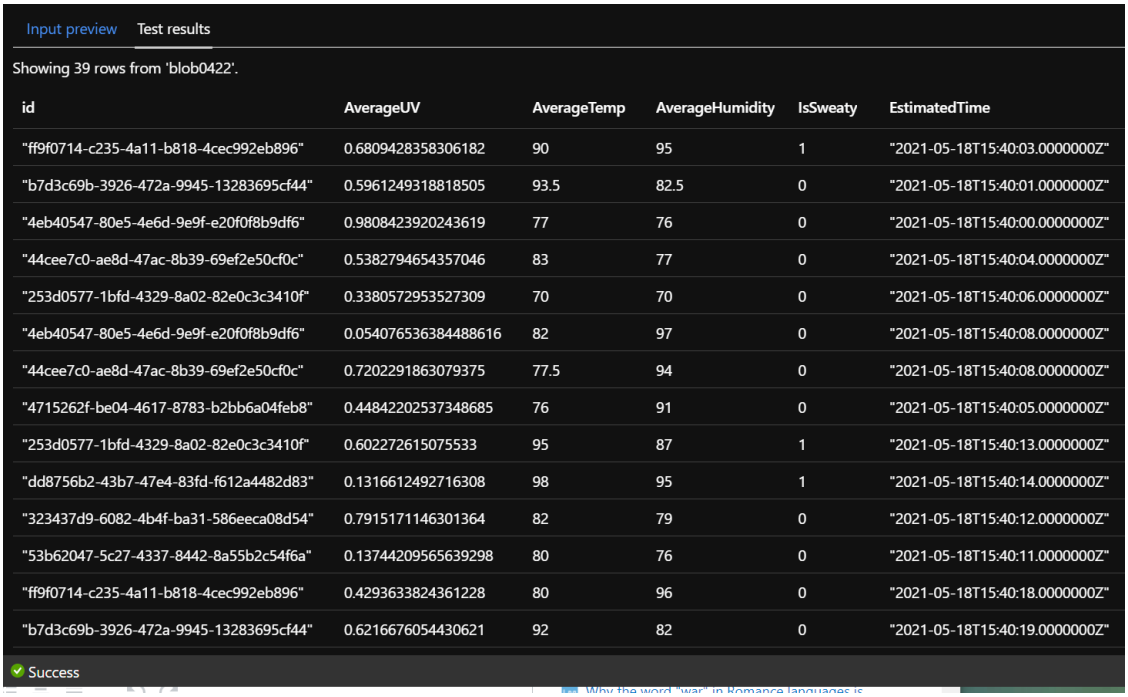
6. Create query
   a. In the “Query” tab in your stream analytics job, check the input preview, it should look like this



   b. Within your stream analytics job, write a SAQL window to do a Tumbling Window of 5 seconds
   c. Get **average uv, temperature, humidity and time** for each device within each window
      i. Hint: estimated time is calculated by first converting the datetime to seconds since the Epoch(unix time stamp), get the average for each device within the current window, and then convert the unix time stamp back to datetime type.
   d. If average humidity and temperature for a device in a window are **both** higher than 85, mark that window as “Sweaty”
   e. Test your query with the data received from event hub



f. Your test result should look like this



Input preview | Test results

Showing 39 rows from 'blob0422'.

id	AverageUV	AverageTemp	AverageHumidity	IsSweaty	EstimatedTime
"ff9f0714-c235-4a11-b818-4cec992eb896"	0.6809428358306182	90	95	1	"2021-05-18T15:40:03.0000000Z"
"b7d3c69b-3926-472a-9945-13283695cf44"	0.5961249318818505	93.5	82.5	0	"2021-05-18T15:40:01.0000000Z"
"4eb40547-80e5-4e6d-9e9f-e20f0f8b9df6"	0.9808423920243619	77	76	0	"2021-05-18T15:40:00.0000000Z"
"44cee7c0-ae8d-47ac-8b39-69ef2e50cf0c"	0.5382794654357046	83	77	0	"2021-05-18T15:40:04.0000000Z"
"253d0577-1bfd-4329-8a02-82e0c3c3410f"	0.3380572953527309	70	70	0	"2021-05-18T15:40:06.0000000Z"
"4eb40547-80e5-4e6d-9e9f-e20f0f8b9df6"	0.054076536384488616	82	97	0	"2021-05-18T15:40:08.0000000Z"
"44cee7c0-ae8d-47ac-8b39-69ef2e50cf0c"	0.7202291863079375	77.5	94	0	"2021-05-18T15:40:08.0000000Z"
"4715262f-be04-4617-8783-b2bb6a04feb8"	0.44842202537348685	76	91	0	"2021-05-18T15:40:05.0000000Z"
"253d0577-1bfd-4329-8a02-82e0c3c3410f"	0.602272615075533	95	87	1	"2021-05-18T15:40:13.0000000Z"
"dd8756b2-43b7-47e4-83fd-f612a4482d83"	0.1316612492716308	98	95	1	"2021-05-18T15:40:14.0000000Z"
"323437d9-6082-4b4f-ba31-586eeca08d54"	0.7915171146301364	82	79	0	"2021-05-18T15:40:12.0000000Z"
"53b62047-5c27-4337-8442-8a55b2c54f6a"	0.13744209565639298	80	76	0	"2021-05-18T15:40:11.0000000Z"
"ff9f0714-c235-4a11-b818-4cec992eb896"	0.4293633824361228	80	96	0	"2021-05-18T15:40:18.0000000Z"
"b7d3c69b-3926-472a-9945-13283695cf44"	0.6216676054430621	92	82	0	"2021-05-18T15:40:19.0000000Z"

Success

g. After you are happy with the test result, go back to your stream analytics dashboard, click on “Start” and wait for the job to start running

h. Now, rerun producer.py and check your blob container for the resulted Parquet files.

   i. You now have a pipeline that looks like this:

Producer.py -> Event Hubs -> Stream Analytics -> Blob Storage