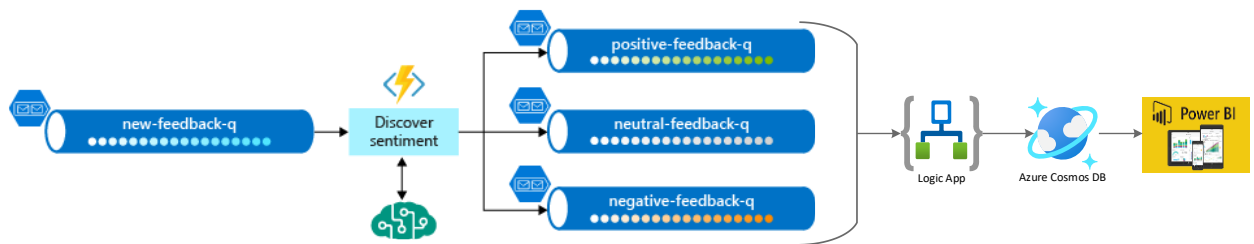


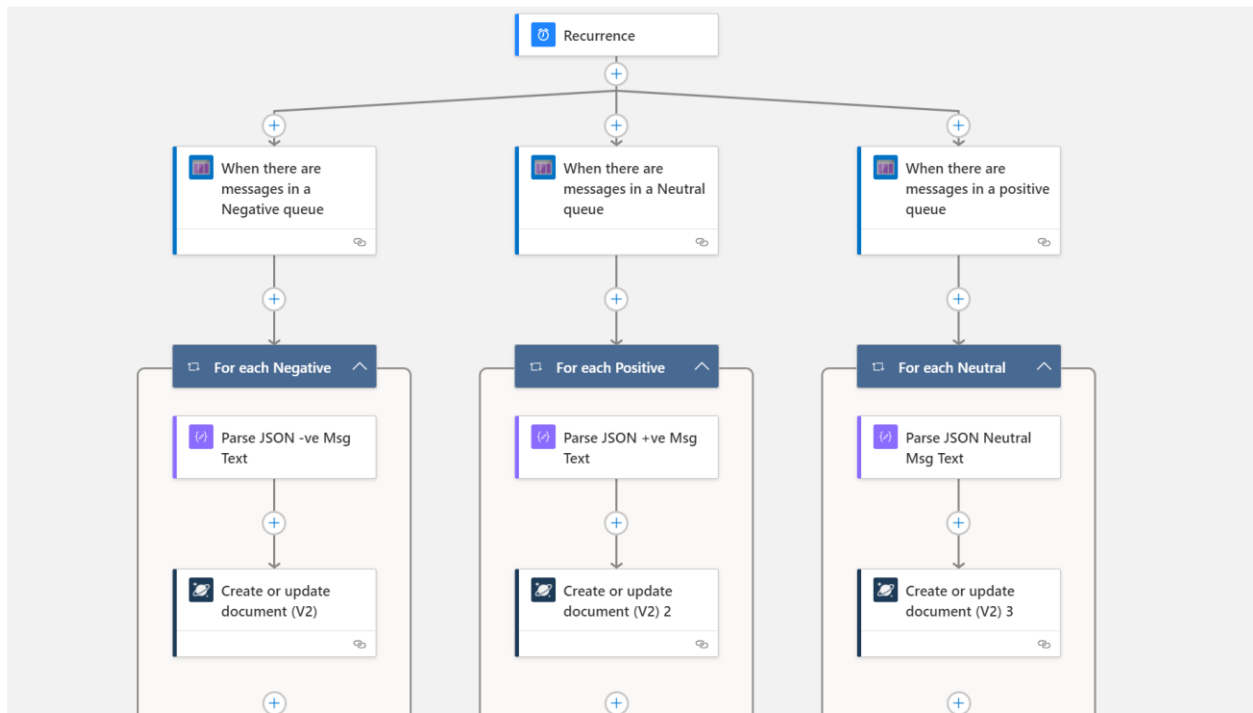
Sentiment Analysis with Storage Queues, Azure Function, Logic Apps, Cosmos DB, and Power BI

High-Level Overview of the solution

Sentiment Analysis Using Storage Queues, Azure Functions, Logic App, Cosmos DB and Power BI



1. In the [Azure portal](#), create an empty logic app by following the instructions in [Create your logic app](#). When you see the **Logic Apps Designer**, return to this tutorial.
2. Once you are done with the logic app, it will look like this



3. The cosmos DB account and database have already been created. You have to create the collection with the name of your choice. For example, I created a collection with the name queue1.

Microsoft Azure | Cosmos DB | kdqueuetestinsert

SQL API | Items

DATA

sentimentanalysisqueue

Scale

queue1

NOTEBOOKS

Gallery

My Notebooks

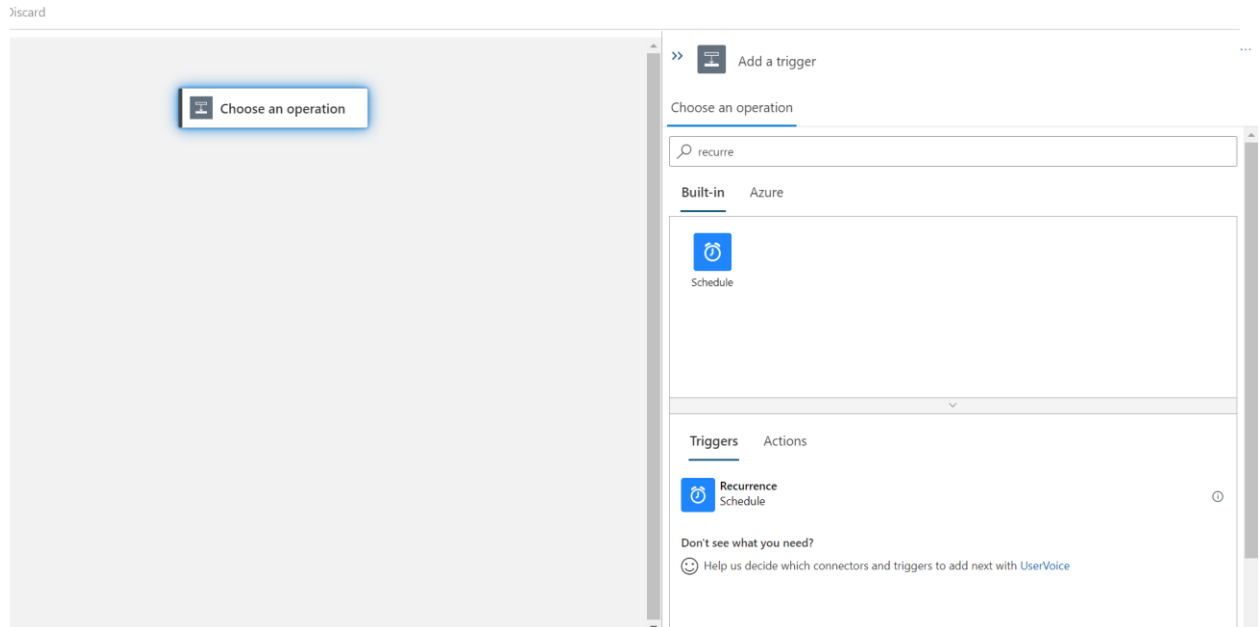
SELECT * FROM c

id	/id
ef35d7c5-76d4-4...	ef35d7c5-76d4-4557-a...
c38dc640-528d-4...	c38dc640-528d-4988-...
b7fbc099-8849-4...	b7fbc099-8849-463c-8...
e5410c5b-0459-4...	e5410c5b-0459-434f-a...
9e996d7d-d225-4...	9e996d7d-d225-428e-...
8f9f380a-3810-45...	8f9f380a-3810-452a-b...
f3ccb687-9b39-4...	f3ccb687-9b39-486f-8...
62c33e03-64dd-4...	62c33e03-64dd-4426-...

```

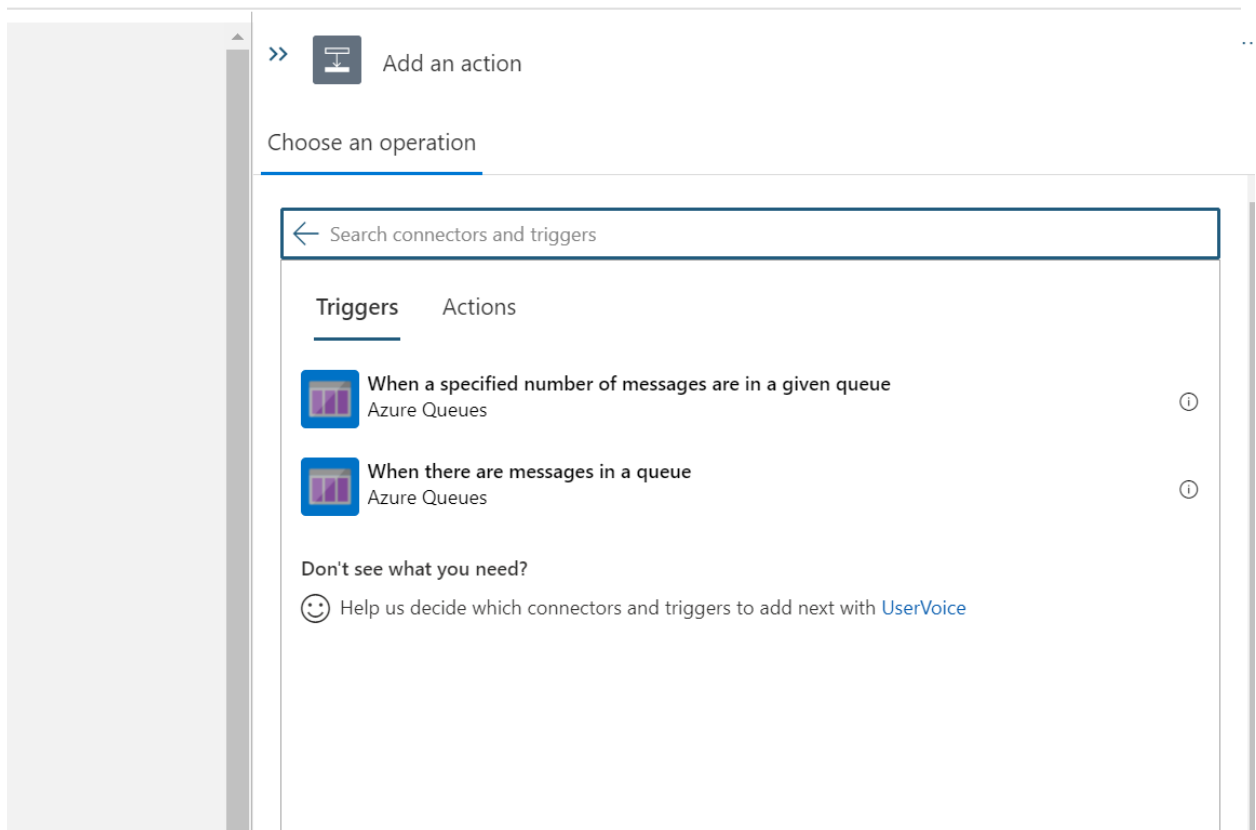
1 {
2   "originalMessage": "Time to go to bed",
3   "score": 0.7976418137550354,
4   "id": "f3ccb687-9b39-486f-8609-227cddbef46c",
5   "MessageArrivalTime": "2021-03-20T03:21:15.5065883Z",
6   "_rid": "uml+AKLVSwGHAAAAAAAAA==",
7   "_self": "dbs/uml+AA==/colls/uml+AKLVSwG=/docs/uml+AKLVSwGHAAAAAAAAA==/",
8   "_etag": "\"0800c031-0000-0100-0000-60556a2b0000\"",
9   "_attachments": "attachments/",
10  "_ts": 1616210475
11 }
  
```

4. Make sure you select stateful workflow in the logic app, not the stateless. In the Logic Apps Designer, add recurrence action. It will run will logic app based on the time interval we want it to run.

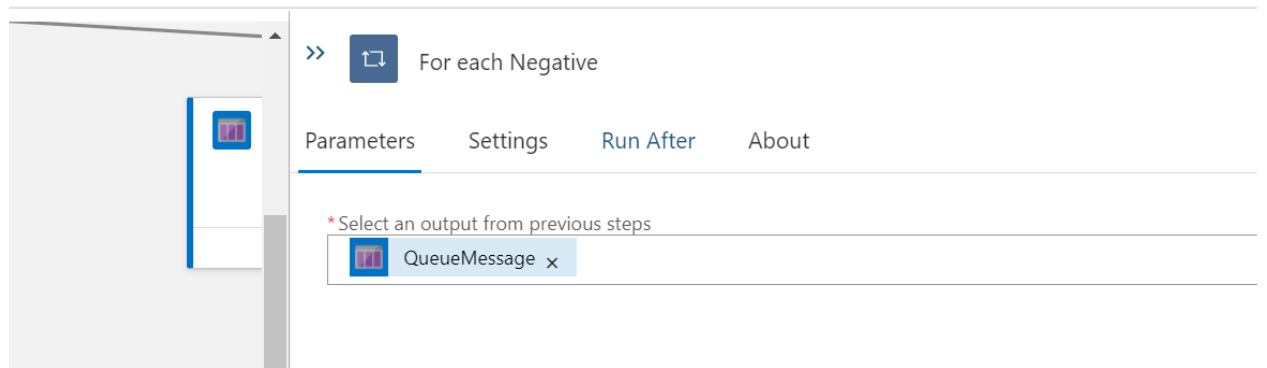


5. Select trigger "when there are messages in the queue" and provide the queue name; for example, For positive messages, provide queue name positive-feedback-q. Add a parallel step by clicking + sign and create the trigger for

negative and neutral queue also.



6. Add "For Each" loop as we have to process all arrived messages in the queue. Inside 'For Each' add ParseJson action. As queue messages are stored as JSON, we have to parse and select the fields we want for our analysis. We are ignoring some default fields added by queue metadata.
 1. Add "queuemessage" from the previous step as a parameter in For Each loop



2. For ParseJSON, select the message text and Provide the below schema

```
{
  "properties": {
    "originalMessage": {
      "type": "string"
    },
    "score": {
      "type": "number"
    }
  },
  "type": "object"
}
```















>>  Add an action



Choose an operation

 Search connectors and actions

Triggers Actions

- | | | |
|---|---|---|
|  | Compose
Data Operations |  |
|  | Create CSV table
Data Operations |  |
|  | Create HTML table
Data Operations |  |
|  | Filter array
Data Operations |  |
|  | Join
Data Operations |  |
|  | Parse JSON
Data Operations |  |
|  | Select
Data Operations |  |

Don't see what you need?

>> Parse JSON -ve Msg Text ...

Parameters Settings Code View Static Result ...

* Content Message Text x

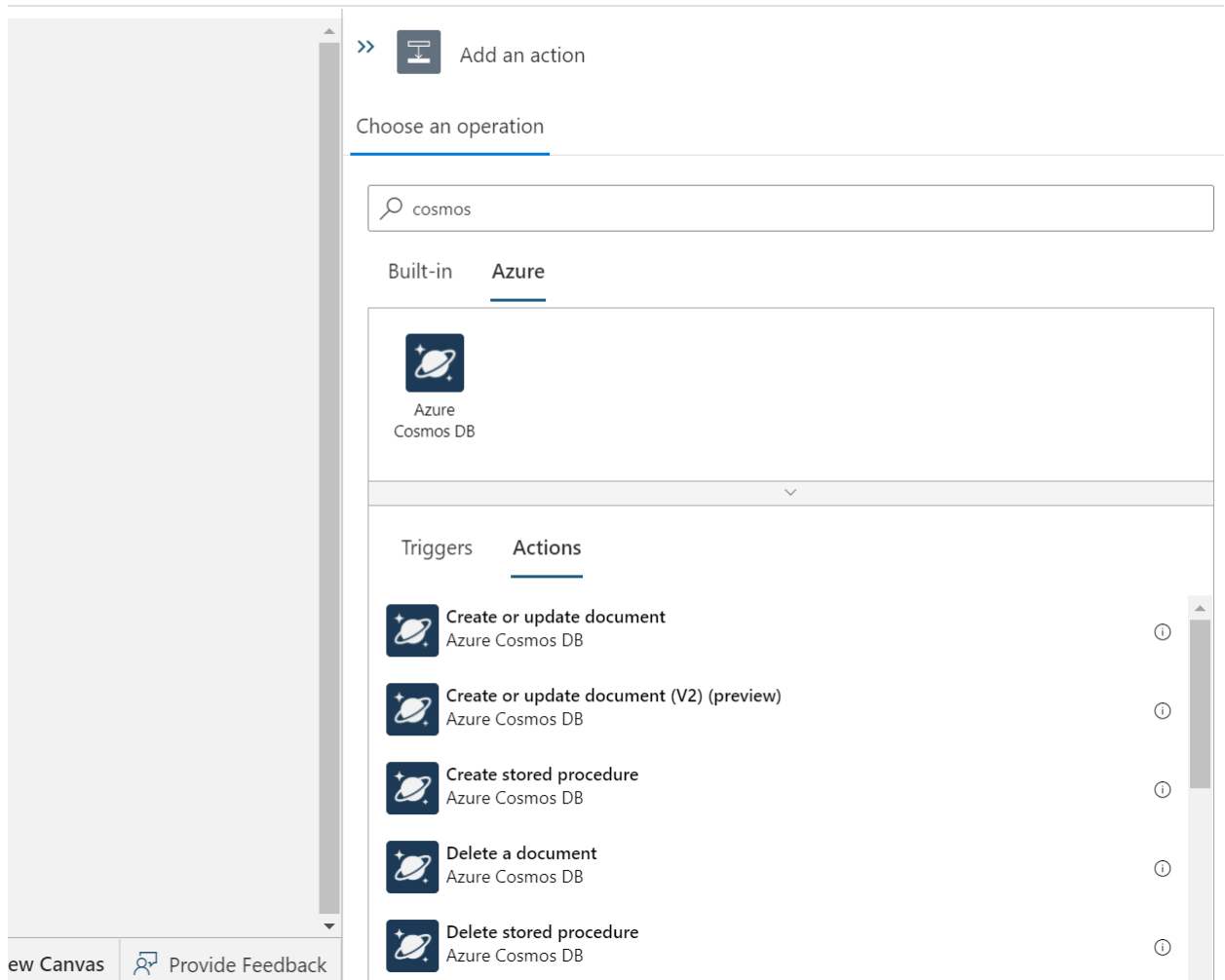
* Schema

```
{
  "properties": {
    "originalMessage": {
      "type": "string"
    },
    "score": {
      "type": "number"
    }
  }
}
```

[Use sample payload to generate schema](#)

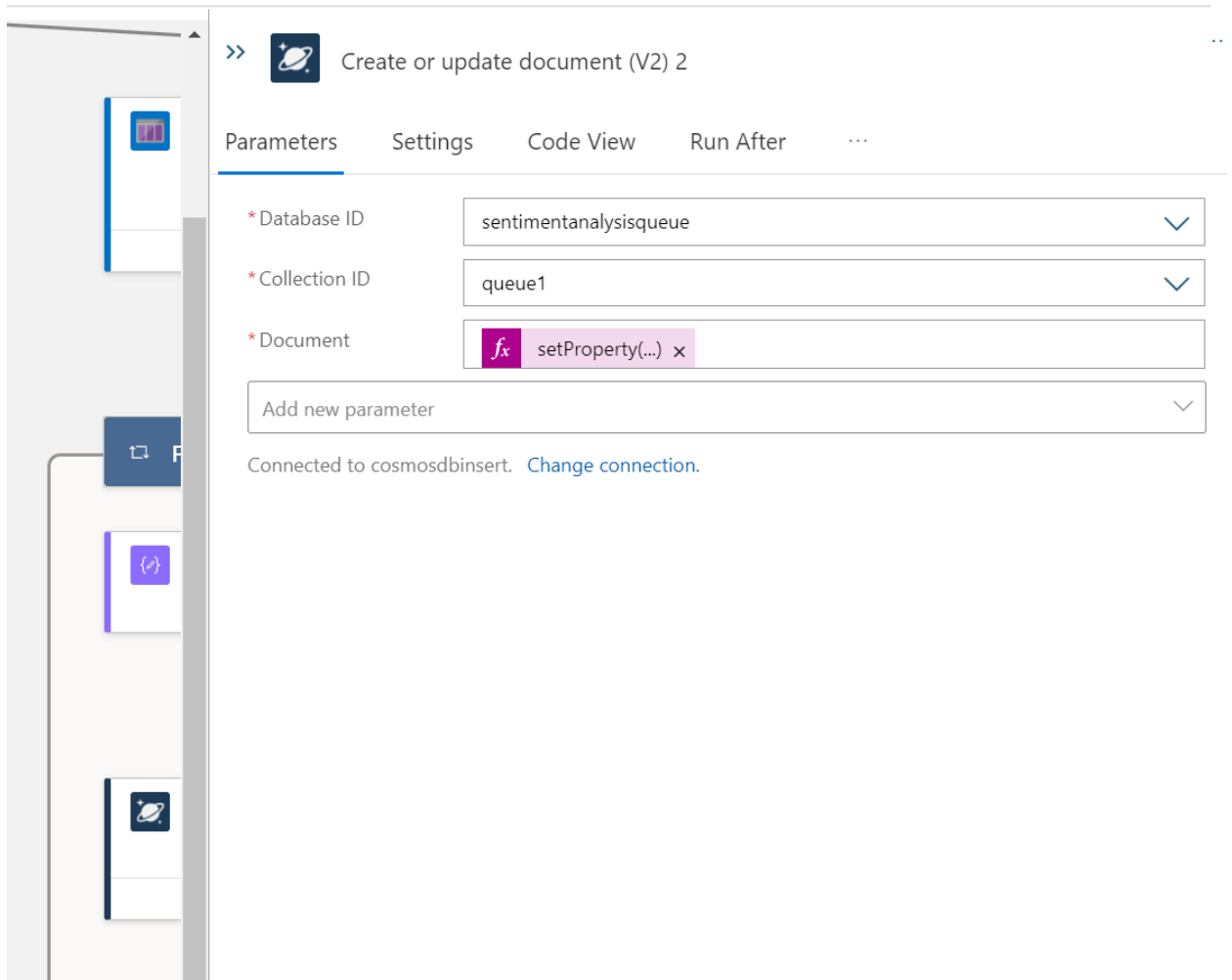
Complete these steps for all three message types.

7. Inside 'For Each' after ParseJSON action, add 'Create or update document (V2) (preview)' to insert messages into cosmos DB collection.



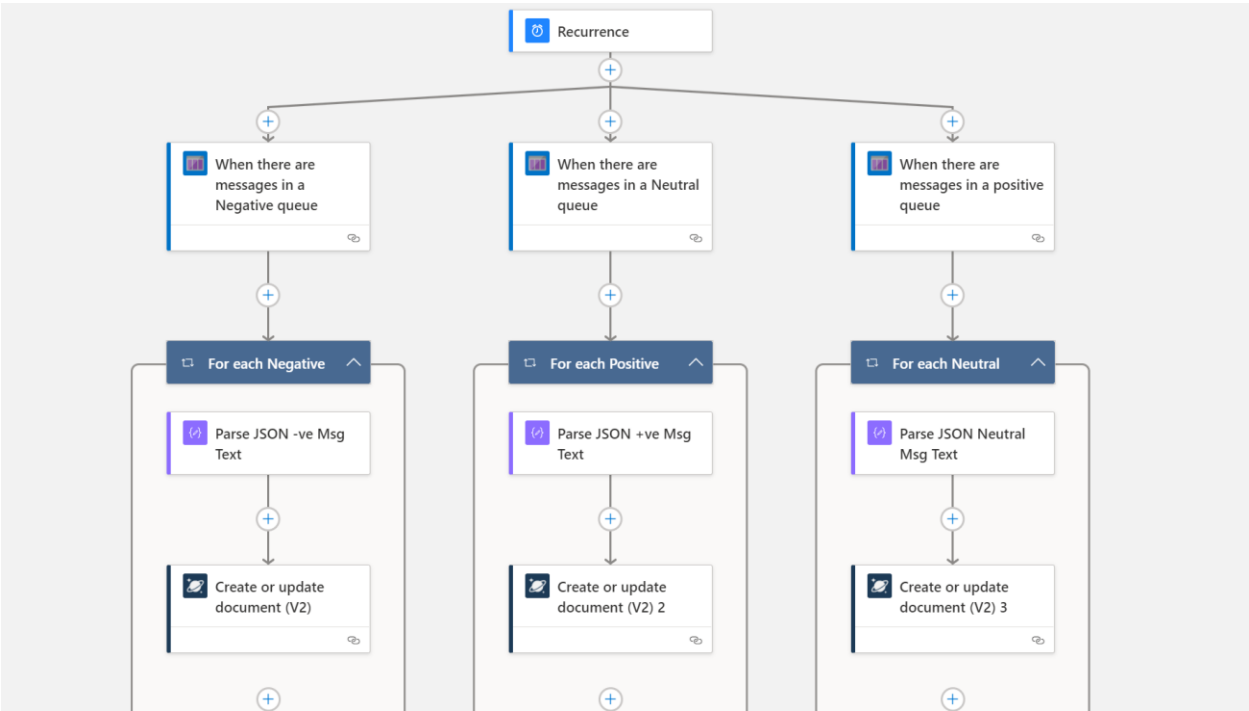
8. In parameter for 'Create or update document (V2) (preview)' Select database ID as "sentimentanalysisqueue" and Collection as queue1 and under document user

dynamic content and function to insert message and add ID field for uniqueness.
The formula looks like this
`setProperty(setProperty(body('Parse_JSON_+ve_Msg_Text'),'id',guid()),'MessageArrivalTime',utcNow())`

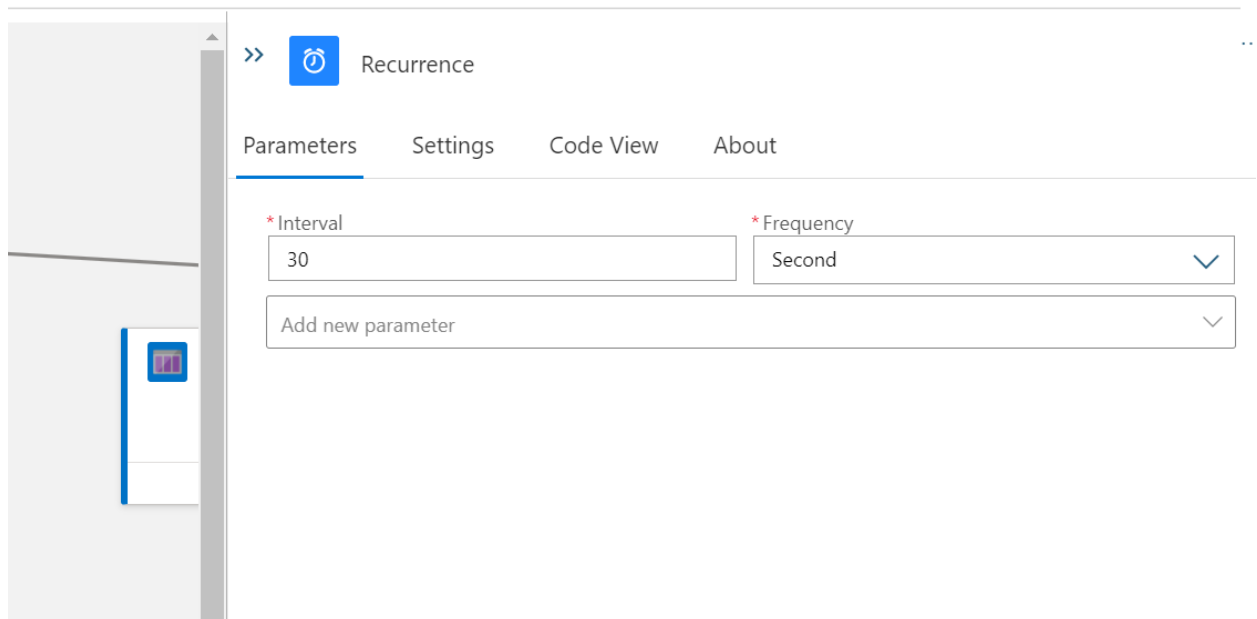



Complete above cosmos DB steps for all three message types.

9. Once you are done with step 8, your logic app should look like this



10. If you want to change the recurrence time, then go to recurrence trigger.



>>  Recurrence

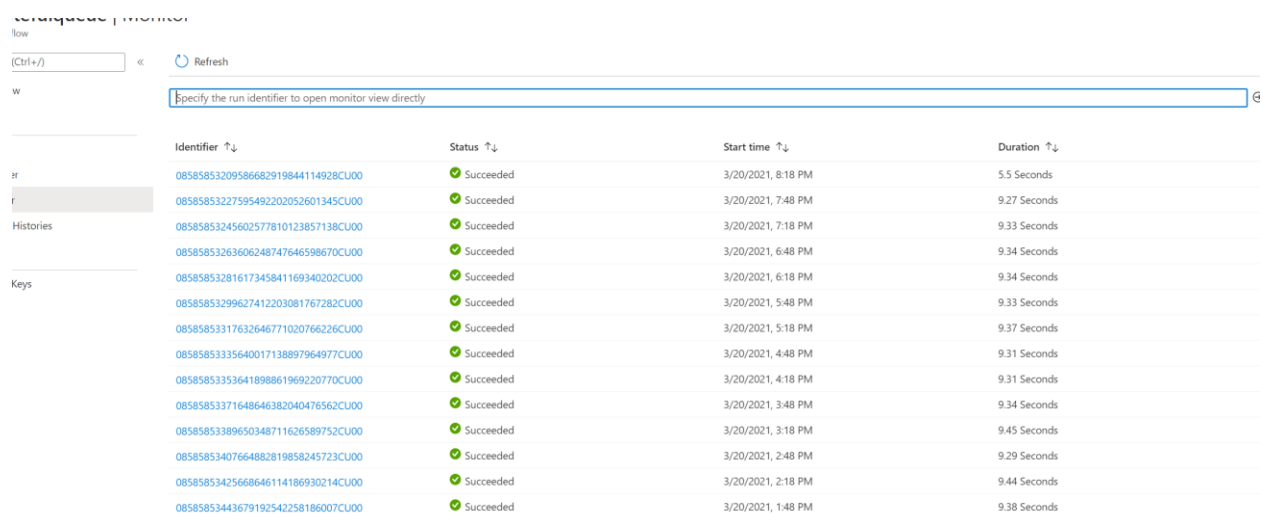
Parameters Settings Code View About

*Interval 30

*Frequency Second

Add new parameter

11. Save the Logic App and Monitor its run by going to monitor on the left side.



Specify the run identifier to open monitor view directly

Identifier ↑↓	Status ↑↓	Start time ↑↓	Duration ↑↓
08585853209586682919844114928CU00	✓ Succeeded	3/20/2021, 8:18 PM	5.5 Seconds
08585853227595492202052601345CU00	✓ Succeeded	3/20/2021, 7:48 PM	9.27 Seconds
08585853245602577810123857138CU00	✓ Succeeded	3/20/2021, 7:18 PM	9.33 Seconds
08585853263606248747646598670CU00	✓ Succeeded	3/20/2021, 6:48 PM	9.34 Seconds
08585853281617345841169340202CU00	✓ Succeeded	3/20/2021, 6:18 PM	9.34 Seconds
08585853299627412203081767282CU00	✓ Succeeded	3/20/2021, 5:48 PM	9.33 Seconds
08585853317632646771020766226CU00	✓ Succeeded	3/20/2021, 5:18 PM	9.37 Seconds
08585853335640017138897964977CU00	✓ Succeeded	3/20/2021, 4:48 PM	9.31 Seconds
08585853353641898861969220770CU00	✓ Succeeded	3/20/2021, 4:18 PM	9.31 Seconds
08585853371648646382040476562CU00	✓ Succeeded	3/20/2021, 3:48 PM	9.34 Seconds
08585853389650348711626589752CU00	✓ Succeeded	3/20/2021, 3:18 PM	9.45 Seconds
08585853407664882819858245723CU00	✓ Succeeded	3/20/2021, 2:48 PM	9.29 Seconds
08585853425668646114186930214CU00	✓ Succeeded	3/20/2021, 2:18 PM	9.44 Seconds
08585853443679192542258186007CU00	✓ Succeeded	3/20/2021, 1:48 PM	9.38 Seconds

12. You can to CosmosDB and look for messages inserted in the collection.

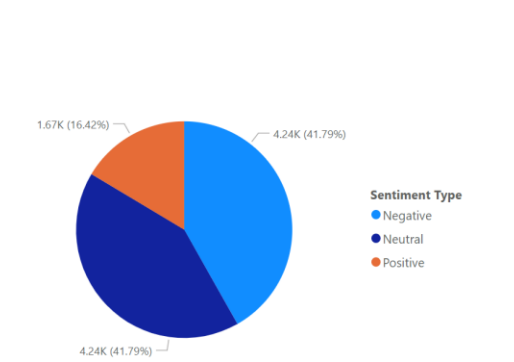
The screenshot shows the Microsoft Azure Cosmos DB portal interface. The top navigation bar includes 'Microsoft Azure', 'Cosmos DB', and a dropdown menu with 'kdqueuetestinsert'. Below the navigation bar, there's a toolbar with icons for 'New Item', 'Update', 'Discard', 'Delete', and 'Upload Item'. The main content area is divided into two sections: 'DATA' and 'NOTEBOOKS'. The 'DATA' section is active, showing a table of items. The table has columns 'id' and '/id'. The 'NOTEBOOKS' section is collapsed. On the right side, a detailed view of an item is shown, displaying its JSON structure. The JSON object contains fields like 'originalMessage', 'score', 'id', 'MessageArrivalTime', 'rid', 'self', 'etag', 'attachments', and 'ts'.

id	/id
ef35d7c5-76d4-4...	ef35d7c5-76d4-4557-a...
c38dc640-528d-4...	c38dc640-528d-4988-...
b7fbc099-8849-4...	b7fbc099-8849-463c-8...
e5410c5b-0459-4...	e5410c5b-0459-434f-a...
9e996d7d-d225-4...	9e996d7d-d225-428e-...
8f9f380a-3810-45...	8f9f380a-3810-452a-b...
f3ccb687-9b39-4...	f3ccb687-9b39-486f-8...
62c33e03-64dd-4...	62c33e03-64dd-4426-...
92b6b4ea-5ab5-4...	92b6b4ea-5ab5-498a-...
2b08b300-773f-4...	2b08b300-773f-45e1-...
57470f95-e83e-4...	57470f95-e83e-49cd-a...
c69c6c57-7592-4...	c69c6c57-7592-4e24-9...
4dee004b-ce1b-4...	4dee004b-ce1b-4864-...
31269a68-37f6-4...	31269a68-37f6-4c87-9...
dcb1bd08-374c-4...	dcb1bd08-374c-4db7-...
f7771a42-318e-4...	f7771a42-318e-4a30-9...
d15a056f-1420-4...	d15a056f-1420-4205-...
1494f77a-83ea-4...	1494f77a-83ea-4d05-...
2f8f5741-0479-40...	2f8f5741-0479-4057-9...

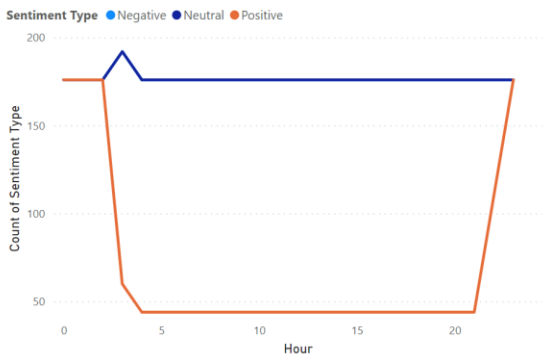
```
1
2
3
4
5
6
7
8
9
10
11
{
  "originalMessage": "Time to go to bed",
  "score": 0.7976418137550354,
  "id": "f3ccb687-9b39-486f-8609-227cddbef46c",
  "MessageArrivalTime": "2021-03-20T03:21:15.5065883Z",
  "_rid": "uml+AKLVSwgHAAAAAAAAA==",
  "self": "dbx/uml+AKLVSwgHAAAAAAAAA==/docs/uml+AKLVSwgHAAAAAAAAA==/",
  "etag": "\"06080c031-0000-0100-0000-60556a2b0000\"",
  "attachments": "attachments/",
  "ts": 1616210475
}
```

13. If you are interested in creating some visualizations, then use your reporting tool of choice. I used Microsoft Power BI as it can connect to cosmos DB and show data in real-time. Please see the below dashboard for your reference.

Count of Sentiment Type by Sentiment Type



Count of Sentiment Type by Hour and Sentiment Type



Message	Sentiment Type	Sentiment Score	Message Arrival Time
Its nice weather outside	Neutral	0.50	3/21/2021 3:48:57 AM
USA is a country	Neutral	0.50	3/21/2021 3:48:57 AM
Time to go to bed	Neutral	0.80	3/21/2021 3:48:57 AM
Its fun doing this exercise!	Positive	0.94	3/21/2021 3:48:56 AM
I'm having fun with this exercise!	Positive	0.89	3/21/2021 3:48:56 AM
I like swimming	Positive	0.95	3/21/2021 3:48:56 AM

