

AI & Cognitive services

Föreläsningstidpunkter (jag uppmärksammat i video'r - ca tider)	1
Del 1 - 20220411_090142	1
Del 2 - 20220411_102703	1
Cosmos DB	1
Stream analytics	1
Modifiering av Device	2
Konfigurera Stream Analytics i Azure	2
CosmosDB-trigger (som en Azure function)	3
'scaffold' att bygga från i V.S.	3
Selektion av data	4
PUNKTER AV SPECIELL VIKT NÄR DET GÄLLER KOSTNADER	4
När du är klar för dagen	4
När du är klar med hela projektet	4

Föreläsningstidpunkter (jag uppmärksammat i video'r - ca tider)

Del 1 - 20220411_090142

12:20	Sammanfattning av tjänster som ingår i projektet
16:00	Cosmos DB

Del 2 - 20220411_102703

07:00	Stream Analytics
1:07:00	CosmosDB-trigger

Cosmos DB

- Create Cosmos DB
- Core (SQL)
- Western EU
- I övrigt inga ändringar från default - Skapa DB
- Vänta på att den blir depoy'ad

Stream analytics

Förutsättning: IoT-device(s). IoT Hub & CosmosDB skapade och körbara

Modifiering av Device

- Koda om tempsimulering så att den gör avvikande skutt. t.ex. som jag gjort i consoledevice:

```
while (await timer.WaitForNextTickAsync())
{
    lastTemp = temperature;
    temperature = rnd.Next(20, 30);
    if (temperature > lastTemp ) temperature=++lastTemp;
    else if (temperature < lastTemp) temperature=--lastTemp;
    int highlow = rnd.Next(0, 100);
    if (highlow > 97) { temperature = temperature + 5; Console.Write("rnd High : "); }
    if (highlow < 3) { temperature = temperature - 5; Console.Write("rnd Low : "); }
    int humidity = rnd.Next(30, 40);
    bool temperatureAlert = (temperature >= 27);

    var data = new { temperature = temperature, humidiity = humidity };
    var message = new Message(Encoding.UTF8.GetBytes(JsonConvert.SerializeObject(data)));
    await deviceClient.SendEventAsync(message);
}
```

Konfigurera Stream Analytics i Azure

- Skapa "Stream analytics job" i Azure, typ spikes & dips.

Stream analytics kostar löpande drygt 1 kr/h per stream. Glöm inte att stoppa tjänsten när den inte behövs.

- Default är 3 streams, men ändra set till 1 stream för att hålla ner kostnaden.
- tryck på [Create]

Tre delar att konf'a: Input, Query & Output:

- Input är default consumer group. Inte önskvärt - gå till IoT Hub
- Välj [built in endpoints]
- Lägg till egen consumergroup "stream"
- Gå tillbaka till stream analytics och välj ovanstående som [input].
- Välj Cosmosdb som [output]
- Skapa nya: Db analyticsdb och collection anomalydetection
- Välj [Query]
- Lägg till "SQLlike"-kod för att styra hur data bearbetas hämtat från Hans genomgång (mån 11 April 44 min in i videon):

```
WITH AnomalyDetectionCode AS
(
    SELECT
        IoTHub.ConnectionDeviceId AS Id,
        EventEnqueuedUtcTime As time,
        temperature,
        humidity,
        AnomalyDetection_SpikeAndDip(temperature, 95, 120, 'spikesanddips' )

        OVER (LIMIT DURATION(second, 120) ) AS SpikeAndDipScore

    FROM input
)
SELECT
```

```

IoTHub.ConnectionDeviceId AS Id,
time,
temperature,
humidity,
CAST(GetRecordPropertyValue(SpikeAndDipScore, 'Score') AS float) AS score,
CAST(GetRecordPropertyValue(SpikeAndDipScore, 'IsAnomaly') AS bigint) AS anomalyDetected
INTO
    output
FROM
    AnomalyDetectionCode

```

- [Save query]
- Kör [Test query] och verifiera att data kommer in
- Stäng query-fönstret
- Starta AnomalyDetection
- Kontrollerar att inga felmeddelanden (Varningstrianglar etc.) dyker upp
- Vänta 120 s på att den läser in sig på datan (Output Watermark)
- Kontrollera ev fel igen
- Om allt fungerar ska vi nu få in data i vår CosmosDB
- Gå till CosmosDB
- Välj [Data Explorer]
- Öppna Data-stream analytics-anomalydetection-items
- Kontrollera att det kommit in JSONdata

Slamkrypare 1: För att se om det kommit in någon 'anomaly'

- Tryck på [Edit Filter] och skriv
WHERE c.anomalyDetected > 0

Slamkrypare 2: AnomalyDetection_SpikeAndDip & LIMIT DURATION ovan i SQLlike-koden har samma värde (120), vilket förutsätter ett värde per sekund (120/120s) - helt galet!
Det är antal värden i ett 'sliding window' och den tid som erfordras för att läsa in detta antal.

Ex: Om du får in ett värde var 15:e sek och vill ha 20 värden som underlag för anomaly-detection ska du alltså sätta 20 resp. 300 som värden

CosmosDB-trigger (som en Azure function)

Du kommer att behöva namnet på Databasen (StreamAnalytics) och Collection (AnomalyDetection) från din CosmosDB nedan. De hittar du i Azure -> CosmosDB -> DataExplorer. Dessutom behöver du Primary connections string som du hittar -> Keys (långt ner i vänstermenyn)

- New project - döp t.ex. till "AzureFunctions" (Eller så återanvänder du det projekt du redan har)
- Välj 'Empty'
- Högerklicka och välj Add - New Azure Function
- Döp den (t.ex till AnomalyDetection)
- Leta upp och välj CosmosDBtrigger
- sätt 'Connection string setting name' (ex. CosmosDB)
- 'Data base' (ex: StreamAnalytics)
- 'Collection' (ex: AnomalyDetection)

- [OK]

'scaffold' att bygga från i V.S.

- Lägg till connection string (Hans lägger den i local.setting.json, jag direkt i koden)
- Lägg till " , CreateLeaseCollectionIfNotExists = true " efter "lease" och före parenteserna:

```
databaseName: "StreamAnalytics",
collectionName: "AnomalyDetection",
ConnectionStringSetting =
"AccountEndpoint=https://embeddedcontrolcosmos.documents.azure.com:443/;AccountKey=CR2TuRvMZQ1B1yf0DA6No01i
x1GkHl8sRfQs6EF0s9DQ9JoIFw==;"
```

```
LeaseCollectionName = "leases"
, CreateLeaseCollectionIfNotExists = true)]IReadOnlyList<Document> input,
ILogger log)
```

Selektion av data

- Följande kod är Hans exempel på detta

```
if (input != null && input.Count > 0)
{
    log.LogInformation("Document Id " + input[0].Id) ;
    log.LogInformation("DeviceId " + input[0].GetProperty<string>("deviceId"));
    log.LogInformation("Temperature " + input[0].GetProperty<float>("temperature"));
    log.LogInformation("Humidity " + input[0].GetProperty<int>("humidity"));
    log.LogInformation("Anomaly Detected? " + input[0].GetProperty<int>("anomalyDetected"));
}
```

PUNKTER AV SPECIELL VIKT NÄR DET GÄLLER KOSTNADER

När du är klar för dagen

- Stäng av Anomaly Detection, så den inte står och drar pengar.

När du är klar med hela projektet

- Radera hela Anomaly Detection - ifall den kostar något även avstängd (troligen lite, men ändå)