

DP 200 - Implementing a Data Platform Solution

Lab 6 - Performing Real-Time Analytics with Stream Analytics

Estimated Time: 60 minutes

Pre-requisites: It is assumed that the case study for this lab has already been read. It is assumed that the content and lab for module 1: Azure for the Data Engineer has also been completed

Lab files: The files for this lab are located in the *Allfiles\Labfiles\Starter\DP-200.6* folder.

Lab overview

The students will be able to describe what data streams are and how event processing works and choose an appropriate data stream ingestion technology for the AdventureWorks case study. They will provision the chosen ingestion technology and integrate this with Stream Analytics to create a solution that works with streaming data.

Lab objectives

After completing this lab, you will be able to:

1. Explain data streams and event processing
2. Ingest data with Event Hubs
3. Initiate a data generation application
4. Process Data with a Stream Analytics Jobs

Scenario

As part of the digital transformation project, you have been tasked by the CIO to help the customer services departments identify fraudulent calls. Over the last few years the customer services departments have observed an increase in calls from fraudulent customer who are asking for support for bikes that are no longer in warranty, or bikes that have not even been purchased at AdventureWorks.

The department are currently relying on the experience of customer services agents to identify this. As a result, they would like to implement a system that can help the agents track in real-time who could be making a fraudulent claim.

At the end of this lab, you will have:

1. Explained data streams and event processing
2. Ingested data with Event Hubs
3. Initiated a data generation application
4. Processed Data with Stream Analytics Jobs

IMPORTANT: As you go through this lab, make a note of any issue(s) that you have encountered in any provisioning or configuration tasks and log it in the table in the document located at *\Labfiles\DP-200-Issues-Doc.docx*. Document the Lab number, note the technology, Describe the issue, and what was the resolution. Save this document as you will refer back to it in a later module.

Exercise 1: Explain data streams and event processing

Estimated Time: 15 minutes

Group exercise

The main task for this exercise are as follows:

1. From the case study and the scenario, identify the data stream ingestion technology for AdventureWorks, and the high-level tasks that you will conduct as a data engineer to complete the social media analysis requirements.
2. The instructor will discuss the findings with the group.

Task 1: Identify the data requirements and structures of AdventureWorks.

1. From the lab virtual machine, start **Microsoft Word**, and open up the file **DP-200-Lab06-Ex01.docx** from the **Allfiles\Labfiles\Starter\DP-200.6** folder.
2. As a group, spend **10 minutes** discussing and listing the data requirements and data structure that your group has identified within the case study document.

Task 2: Discuss the findings with the Instructor

1. The instructor will stop the group to discuss the findings.

Result: After you completed this exercise, you have created a Microsoft Word document that shows a table of data streaming ingestion and the high-level tasks that you will conduct as a data engineer to complete the social media analysis requirements .

Exercise 2: Data Ingestion with Event Hubs.

Estimated Time: 15 minutes

Individual exercise

The main tasks for this exercise are as follows:

1. Create and configure an Event Hub Namespace.
2. Create and configure an Event Hub.
3. Configure Event Hub security.

Task 1: Create and configure an Event Hub Namespace.

1. In the Azure portal, click on the **Home** hyperlink at the top left of the screen.
2. In the Azure portal, click on the **+ Create a resource** icon , type **Event Hubs**, and then select **Event Hubs** from the resulting search. In the Event Hubs screen, click **Create**.
3. In the Create Namespace blade, type out the following options:
 - **Name:** **xx-phoneanalysis-ehn**, where xx are your initials
 - **Pricing Tier:** **Standard**
 - **Subscription:** **Your subscription**
 - **Resource group:** **awrgstudxx**
 - **Location:** select the location closest to you
 - **Throughput Units:** **20**
 - Leave other options to their default settings

Home > New > Event Hubs > Create Namespace

Create Namespace

Event Hubs

Name *

cto-phoneanalysis-ehn ✓
.servicebus.windows.net

Pricing tier (View full pricing details) *

Standard (20 Consumer groups, 1000... ▼

☒ Enable Kafka ⓘ

☐ Make this namespace zone redundant ⓘ

Subscription *

chtestao ▼

Resource group *

awrgstudcto ▼
[Create new](#)

Location *

West US 2 ▼

Throughput Units *

20

☐ Enable Auto-Inflate ⓘ

4. Then click **Create**

Note: The creation of the Event Hub Namespace takes approximately 1 minute.

Task 2: Create and configure an Event Hub

1. In the Azure portal, click on the **Home** hyperlink at the top left of the screen.
2. In the Azure portal, in the blade, click **Resource groups**, and then click **awrgstudxx**, where **xx** are your initials
3. Click on **xx-phoneanalysis-ehn**, where **xx** are your initials.
4. In the **xx-phoneanalysis-ehn** screen, click on **+ Event Hubs**.
5. Provide the name **xx-phoneanalysis-eh**, leave the other settings to their default values and then select **Create**.

Home > Resource groups > awrgstudcto > cto-phoneanalysis-ehn > Create Event Hub

Create Event Hub

Event Hubs

Name * ⓘ
cto-phoneanalysis-eh ✓

Partition Count ⓘ
 1

Message Retention ⓘ
 1

Capture ⓘ

Note: You will receive a message stating that the Event Hub is created after about 10 seconds

Task 3: Configure Event Hub security

1. In the Azure portal, in the **xx-phoneanalysis-ehn** screen, where **xx** are your initials. Scroll to the bottom of the window, and click on **xx-phoneanalysis-eh** event hub.
2. To grant access to the event hub, in the blade on the left click **Shared access policies**.
3. Under the **xx-phoneanalysis-eh - Shared access policies** screen, create a policy with **Manage** permissions by selecting **+ Add**. Give the policy the name of **xx-phoneanalysis-eh-sap**, check **Manage**, and then click **Create**.

Add SAS Policy

Event Hubs

Policy name *
cto-phoneanalysis-eh-sap ✓

☒ Manage

☐ Send

☐ Listen

4. Click on your new policy **xx-phoneanalysis-eh-sap** after it has been created, and then select the copy button for the **CONNECTION STRING - PRIMARY KEY** and paste the CONNECTION STRING - PRIMARY KEY into Notepad, this is needed later in the exercise.

NOTE: The connection string looks as follows:

```
Endpoint=sb://<Your event hub namespace>.servicebus.windows.net/;SharedAccessKeyName=<Your shared access policy name>;Sh
```

Notice that the connection string contains multiple key-value pairs separated with semicolons: Endpoint, SharedAccessKeyName, SharedAccessKey, and EntityPath.

5. Close down the Event hub screens in the portal

Result: After you completed this exercise, you have created an Azure Event Hub within an Event Hub Namespace and set the security for the Event Hub that can be used to provide access to the service.

Exercise 3: Starting the telecom event generator application

Estimated Time: 15 minutes

Individual exercise

The main tasks for this exercise are as follows:

1. Updates the application connection string
2. Run the application

Task 1: Updates the application connection string.

1. Browse to the location `\Labfiles\Starter\DP-200.6\DataGenerator`
2. Open the `telcodatagen.exe.config` file in a text editor of your choice
3. Update the element in the config file with the following details:
 - Set the value of the `EventHubName` key to the value of the `EntityPath` in the connection string.
 - Set the value of the `Microsoft.ServiceBus.ConnectionString` key to the connection string **without the EntityPath value** (don't forget to remove the semicolon that precedes it).
4. Save the file.

Task 2: Run the application.

1. Click on **Start**, and type **CMD**
2. Right click **Command Prompt**, click **Run as Administrator**, and in the User Access Control screen, click **Yes**
3. In Command Prompt, browse to the location `\Labfiles\Starter\DP-200.6\DataGenerator`
4. Type in the following command:

```
telcodatagen.exe 1000 0.2 2
```

NOTE: This command takes the following parameters: Number of call data records per hour. Percentage of fraud probability, which is how often the app should simulate a fraudulent call. The value 0.2 means that about 20% of the call records will look fraudulent. Duration in hours, which is the number of hours that the app should run. You can also stop the app at any time by ending the process (Ctrl+C) at the command line.

```
Administrator: Command Prompt - telcodatagen.exe 1000 0.2 2
Microsoft Windows [Version 10.0.18362.535]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>cd C:\Users\Chris\Desktop\Labfiles\Data Generator\TelcoGenerator

C:\Users\Chris\Desktop\Labfiles\Data Generator\TelcoGenerator>telcodatagen.exe 1000 0.2 2
#Sets: 1,#FilesDump: 1,#CDRPerFile: 1000,%CallBack: 0.2, #DurationHours: 2
Time Increment Per Set: 2
20191219 214157
MO,d0,0,US,012323002,466923300507919,123454855,466922202546859,20191219,214157,3
,0,,b,0,,3,,443,886932428688,,20191219 214157
MO,d0,2,Germany,345633961,466923000464324,345688489,466923000886460,20191219,214
201,3,534,,S,1,,3,,411,886932429827,,20191219 214201
MO,d0,3,UK,234569616,466923200779222,234575856,466922201102759,20191219,214201,3
,0,,b,1,,3,,300,1417955696232,,20191219 214201
MO,d0,5,Germany,234578130,466921402416657,789063569,466921200135361,20191219,214
202,3,0,,U,1,,4,,424,886932428927,,20191219 214202
MO,d0,7,China,012380552,466923101048691,123481384,466921302209862,20191219,21420
2,2,0,,S,0,,0,,422,1416960750071,,20191219 214202
```

After a few seconds, the app starts displaying phone call records on the screen as it sends them to the event hub. The phone call data contains the following fields:

Record	Definition
CallrecTime	The timestamp for the call start time.
SwitchNum	The telephone switch used to connect the call. For this example, the switches are strings that represent the country/region of origin (US, China, UK, Germany, or Australia).
CallingNum	The phone number of the caller.
CallingIMSI	The International Mobile Subscriber Identity (IMSI). It's a unique identifier of the caller.
CalledNum	The phone number of the call recipient.
CalledIMSI	International Mobile Subscriber Identity (IMSI). It's a unique identifier of the call recipient.

- 1. Minimize the command prompt window.

Result: After you completed this exercise, you have configured an application to generate data to mimic phone calls recieved by a call center.

Exercise 4: Processing Data with Stream Analytics Jobs

Estimated Time: 15 minutes

Individual exercise

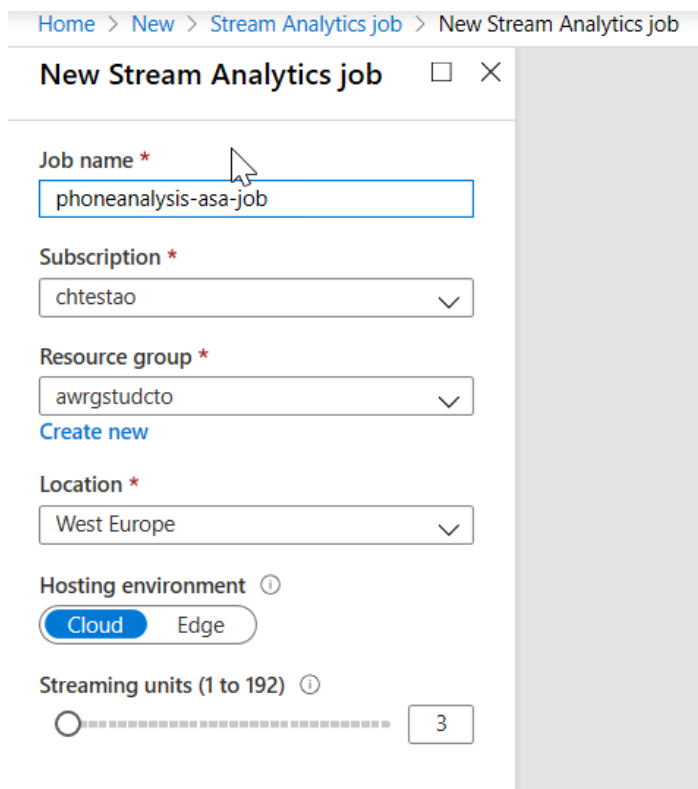
The main tasks for this exercise are as follows:

- 1. Provision a Stream Analytics job.
- 2. Specify the a Stream Analytics job input.
- 3. Specify the a Stream Analytics job output.
- 4. Defining a Stream Analytics query.
- 5. Start the Stream Analytics job.

6. Validate streaming data is collected

Task 1: Provision a Stream Analytics job.

1. Go back to the Azure portal, navigate and click on the + **Create a resource** icon, type **STREAM**, and then click the **Stream Analytics Job**, and then click **Create**.
2. In the **New Stream Analytics job** screen, fill out the following details and then click on **Create**:
 - **Job name**: phoneanalysis-asa-job.
 - **Subscription**: select your subscription
 - **Resource group**: awrgstudxx
 - **Location**: choose a location nearest to you.
 - Leave other options to their default settings



Home > New > Stream Analytics job > New Stream Analytics job

New Stream Analytics job

Job name *
phoneanalysis-asa-job

Subscription *
chtestao

Resource group *
awrgstudcto
[Create new](#)

Location *
West Europe

Hosting environment ⓘ
Cloud Edge

Streaming units (1 to 192) ⓘ
0 3

Note: You will receive a message stating that the Stream Analytics job is created after about 10 seconds. It may take a couple of minutes to update in the Azure portal.

Task 2: Specify the a Stream Analytics job input.

1. In the Azure portal, in the blade, click **Resource groups**, and then click **awrgstudxx**, where **xx** are your initials.
2. Click on **phoneanalysis-asa-job**.
3. In your **phoneanalysis-asa-job** Stream Analytics job window, in the left hand blade, under **Job topology**, click **Inputs**.
4. In the **Inputs** screen, click + **Add stream input**, and then click **Event Hubs**.
5. In the Event Hub screen, type in the following values and click the **Save** button.
 - **Input alias**: Enter a name for this job input as **PhoneStream**.
 - **Select Event Hub from your subscriptions**: checked
 - **Subscription**: Your subscription name
 - **Event Hub Namespace**: xx-phoneanalysis-ehn

- **Event Hub Name:** Use existing named xx-phoneanalysis-eh
- **Event Hub Policy Name:** xx-phoneanalysis-eh-sap
- Leave the rest of the entries as default values. Finally, click **Save***.

Event Hub

New input

✕

Input alias *

PhoneStream ✓

☐ Provide Event Hub settings manually
 ☒ Select Event Hub from your subscriptions

Subscription

chtestao ▾

Event Hub namespace * ⓘ

cto-phoneanalysis-ehn ▾

Event Hub name * ⓘ

☐ Create new
 ☒ Use existing

cto-phoneanalysis-eh ▾

Event Hub policy name * ⓘ

cto-phoneanalysis-eh-sap ▾

Event Hub policy key

.....

Event Hub consumer group ⓘ

Event serialization format * ⓘ

JSON ▾

You can implement a deserializer in C# that can read events in any format. You can try this out by [signing up for the preview program](#).

Encoding ⓘ

UTF-8 ▾

Event compression type ⓘ

None ▾

Save

ⓘ The selected resource and the stream analytics job are located in different regions. You will be billed to move data between regions.

6. Once completed, the **PhoneStream** Input job will appear under the input window. Close the input widow to return to the Resource Group Page

Task 3: Specify the a Stream Analytics job output.

1. Click on **phoneanalysis-asa-job**.
2. In your **phoneanalysis-asa-job** Stream Analytics job window, in the left hand blade, under **Job topology**, click **Outputs**.
3. In the **Outputs** screen, click + **Add**, and then click **Blob Storage**.
4. In the **Blob storage** window, type or select the following values in the pane:
 - **Output alias:** PhoneCallRefData
 - **Select Event Hub from your subscriptions:** checked
 - **Subscription:** Your subscription name
 - **Storage account:** :awsastudxx;, where xx is your initials
 - **Container:** Use existing and select **phonecalls**
 - Leave the rest of the entries as default values. Finally, click **Save**.

Blob storage/Data Lake Storage Gen2
New output

Output alias *
PhoneCallRefData

☐ Provide storage settings manually
☒ Select storage from your subscriptions

Subscription
chtestao

Storage account * ⓘ
awsastudcto

Storage account key
.....

Container *
☐ Create new
☒ Use existing
phonecalls

Path pattern ⓘ

Date format
YYYY/MM/DD

Time format
HH

Event serialization format * ⓘ
JSON

Encoding ⓘ
UTF-8

Format ⓘ
Line separated

Minimum rows ⓘ

Maximum time
Hours ⓘ
Minutes

Save

5. Close the output screen to return to the Resource Group page

Task 4: Defining a Stream Analytics query.

1. Click on **phoneanalysis-asa-job**.
2. In your **phoneanalysis-asa-job** window, in the **Query** screen in the middle of the window, click on **Edit query**
3. Replace the following query in the code editor:

```

SELECT
    *
INTO
    [YourOutputAlias]
FROM
    [YourInputAlias]

```

4. Replace with

```

SELECT System.Timestamp AS WindowEnd, COUNT(*) AS FraudulentCalls
INTO "PhoneCallRefData"
FROM "PhoneStream" CS1 TIMESTAMP BY CallRecTime
JOIN "PhoneStream" CS2 TIMESTAMP BY CallRecTime
ON CS1.CallingIMSI = CS2.CallingIMSI
AND DATEDIFF(ss, CS1, CS2) BETWEEN 1 AND 5
WHERE CS1.SwitchNum != CS2.SwitchNum
GROUP BY TumblingWindow(Duration(second, 1))

```

NOTE: This query performs a self-join on a 5-second interval of call data. To check for fraudulent calls, you can self-join the streaming data based on the CallRecTime value. You can then look for call records where the CallingIMSI value (the originating number) is the same, but the SwitchNum value (country/region of origin) is different. When you use a JOIN operation with streaming data, the join must provide some limits on how far the matching rows can be separated in time. Because the streaming data is endless, the time bounds for the relationship are specified within the ON clause of the join using the DATEDIFF function. This query is just like a normal SQL join except for the DATEDIFF function. The DATEDIFF function used in this query is specific to Stream Analytics, and it must appear within the ON...BETWEEN clause.

5. Select **Save Query**.

6. Close the Query window to return to the Stream Analytics job page.

Task 5: Start the Stream Analytics job

1. In your **phoneanalysis-asa-job** window, in the **Query** screen in the middle of the window, click on **Start**
2. In the **Start Job** dialog box that opens, click **Now**, and then click **Start**.

Note: In your **phoneanalysis-asa-job** window, a message appears after a minute that the job has started, and the started field changes to the time started

Note: Leave this running for 2 minutes so that data can be captured.

Task 6: Validate streaming data is collected

1. In the Azure portal, in the blade, click **Resource groups**, and then click **awrgstudxx**, and then click on **awsastudxx**, where **xx** are your initials.
2. In the Azure portal, click **Containers** box, and then click on the container named **phonecalls**.
3. Confirm that a JSON file appears, and note the size column.

Home > awrgstudcto > awsaastudcto - Containers > phonecalls

phonecalls

Container

Search (Ctrl+F)

Overview

Access Control (IAM)

Settings

Access policy

Properties

Metadata

Editor (preview)

Upload

Change access level

Refresh

Delete

Change tier

Acquire lease

Break lease

View snapshots

Create snapshot

Authentication method: Access key (Switch to Azure AD User Account)

Location: phonecalls

Search blobs by prefix (case-sensitive)

Show deleted blobs

Name	Modified	Access tier	Blob type	Size	Lease state	
<input type="checkbox"/> 0_fa8fc70223264e808d4052926ec9ca10_1.json	12/19/2019, 10:08:09 PM	Hot (Inferred)	Block blob	13.6 KiB	Available	...

4. Refresh Microsoft Edge, and when the screen has refreshed note the size of the file

Note: You could download the file to query the JSON data, you could also output the data to Power BI.

Result: After you completed this exercise, you have configured Azure Stream Analytics to collect streaming data into an JSON file store in Azure Blob. You have done this with streaming phone call data.

Close down

1. In the Azure portal, in the blade, click **Resource groups**, and then click **awrgstudxx**, and then click on **phoneanalysis-asa-job**.
2. In the **phoneanalysis-asa-job** screen, click on **Stop**. In the **Stop Streaming job** dialog box, click on **Yes**.
3. Close down the Command Prompt application.