DAT220x

Delivering a Data Warehouse in the Cloud

Lab 03 | Integrating and Ingesting Data

Overview

In this lab, you will load data and move data into your Azure SQL Data Warehouse, using Azure Blob Storage, Azure Data Factory, Azure Stream Analytics, and Polybase

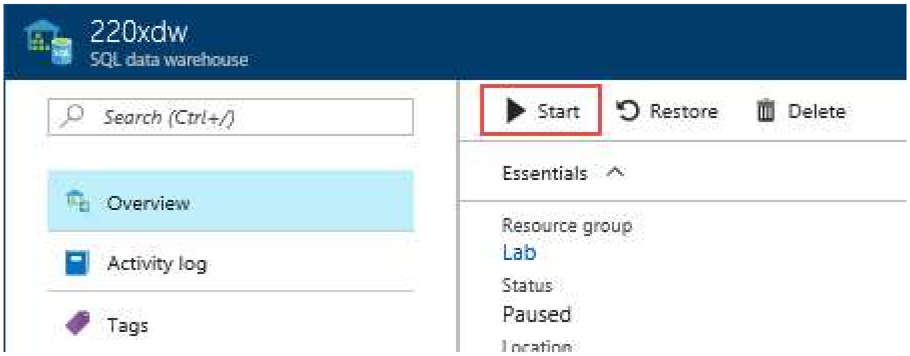
Note: The four labs in this course are cumulative. You cannot complete the following labs if this lab has not been successfully completed.

Exercise 1: Resume the Lab Environment

In this exercise, you will resume your SQL DW from its paused state, and restart your lab virtual machine.

1. Sign in to the Azure Portal by using your subscription. Navigate to the Lab resource group you created in the last lab, then click to open the blade for your SQL DW.

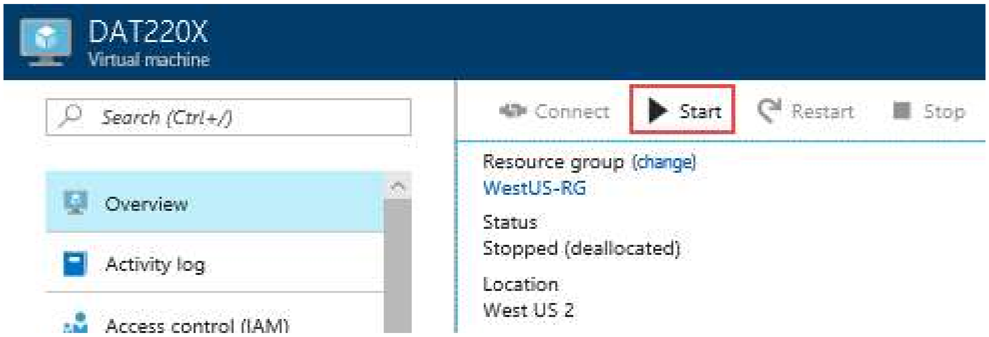
2. If necessary, resume the SQL DW from its paused state:



Resource group
Lab
Status
Paused
Location

3. Watch for the notification of the successful resume request.

4. Return to the Lab resource group blade, and click to open the virtual machine you created in your last lab. If necessary, start the VM



Resource group (change)
WestUS-RG

Status
Stopped (deallocated)

Location
West US 2

5. Once the Connect link becomes available, click Connect and open the Remote Desktop file, as you learned in Lab 1. (You may need to refresh the browser tab.)

6. Using the credentials you created in Lab 1, log in to the VM. Remember to use “use another account” when logging in.

7. Back in the Azure portal, navigate to the SQL DW created in the previous lab.

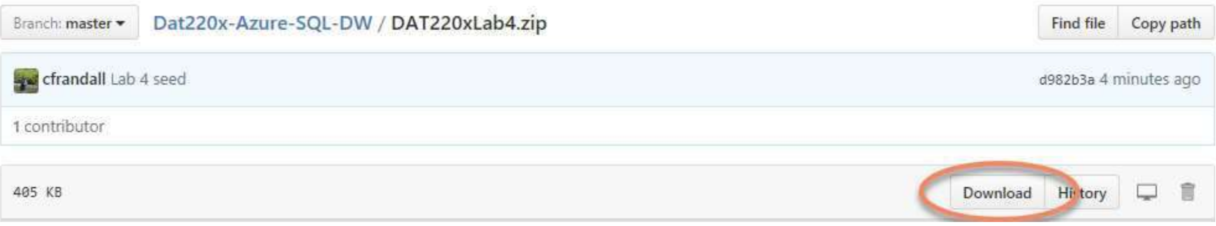
8. In the Essentials pane, click the Start button to resume your SQL Data Warehouse.

Exercise 2: Upload Data to Azure Blob Storage

In this exercise, you will upload a data file from your virtual machine to the Azure Storage account you created in Lab 1, Exercise 4.

1. In your virtual machine, open Microsoft Edge.

2. Navigate to the following location: https://github.com/MicrosoftLearning/Dat220x-Azure-SQL-DW/blob/master/DAT220xLab3.zip. Download the file.



3. Open a File Explorer window. Navigate to the downloaded file location.

4. Extract the zip file by right-clicking on it and, select Extract All…, then follow the prompts to extract the files.

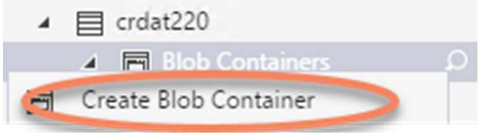
5. Open the extracted folder and locate the three files; BeachWeatherStationsAutomatedSensors.csv, 2016Orders.csv, and 2017Orders.csv. 2016Orders.csv and 2017Orders.csv contain random orders for the years 2016 and

2017, respectively, and BeachWeatherStationsAutomatedSensors.csv is a small CSV file that contains sensor data from various Lake Michigan beaches in Chicago, Illinois. This dataset was obtained from the city's open data portal (https://data.cityofchicago.org).

6. Using the taskbar shortcut you created in Lab 1, open Azure Storage Manager.

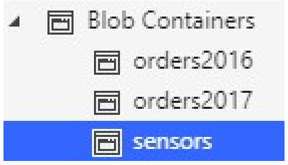
7. Navigate to the storage account you created in Lab 1.

8. Expand the container to view the Blob Containers node. Right-click, then click Create Blob Container:



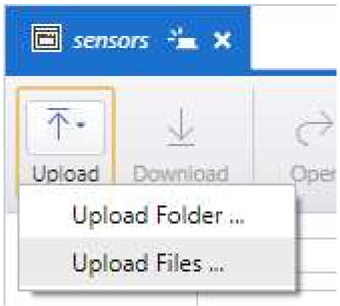
9. Name the new container sensors (case-sensitive).

10. Repeat steps 8 and 9 two more times to create two more blob containers; orders2016 and orders2017.



11. Select the sensors container.

12. In the pane on the right, click the Upload button, then click Upload Files:



13. Browse for and upload the BeachWeatherStationsAutomatedSensors.csv file that you downloaded for this lab. Leave the default blob type of Block Blob. When ready, click Upload.

14. The Activity Log will show your upload progress and will notify you when the upload has completed. If you click the Refresh button, the file should also appear in the storage container.

15. Select the orders2016 container.

16. Browse for and upload the Orders2016.csv file that you downloaded for this lab. Leave the default blob type of Block Blob. When ready, click Upload.

17. Do not upload the 2017Orders.csv file yet.

Exercise 3: Move Data with Azure Data Factory

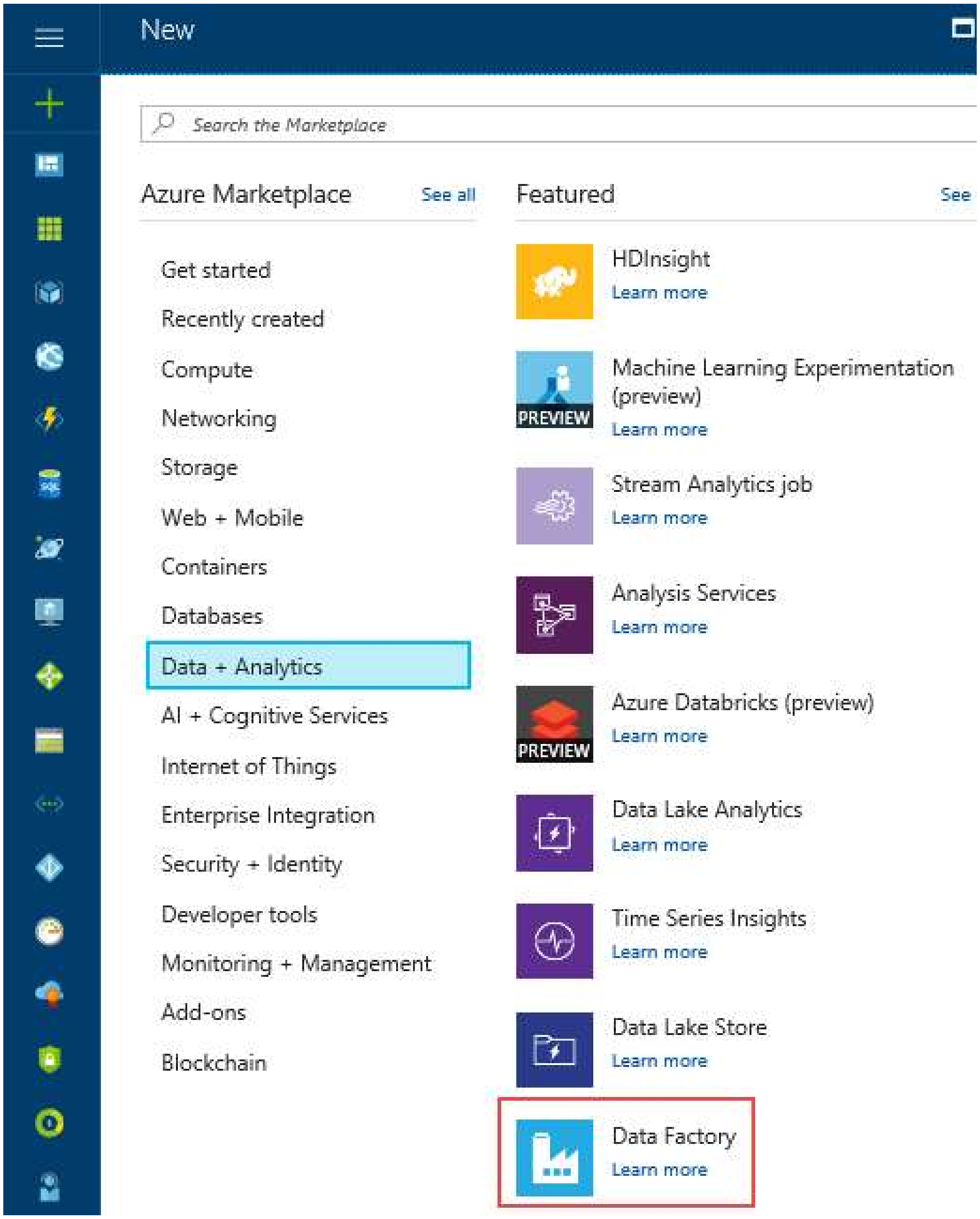
In this exercise, you will first create a table in the Azure SQL Data Warehouse sample database, then create an Azure Data Factory to integrate and move data from Azure blob storage to the table in Azure SQL Data Warehouse.

This example will use Azure Data Factory version 2, which at the time of this writing, is in Preview. It should be expected that the functionality, user-interface, and other items associated with Azure Data Factory version 2 will change, possibly affecting the steps in this exercise.

1. Switch to or open SQL Server Management Studio (SSMS).
2. Connect to your SQL DW.
3. On the SSMS toolbar, click the New Query button.
4. In the query window, enter the following T-SQL query and click the Execute button. This query creates a simple Orders table with three columns.

```
IF OBJECT_ID('dbo.Orders','U') IS NOT NULL DROP TABLE dbo.Orders;
CREATE TABLE [dbo].[Orders]
(
  [OrderID] [int] NOT NULL,
  [CustomerID] [int] NOT NULL,
  [OrderDate] [date] NOT NULL
)
WITH
(
  DISTRIBUTION = ROUND_ROBIN,
  CLUSTERED COLUMNSTORE INDEX
)
GO
```

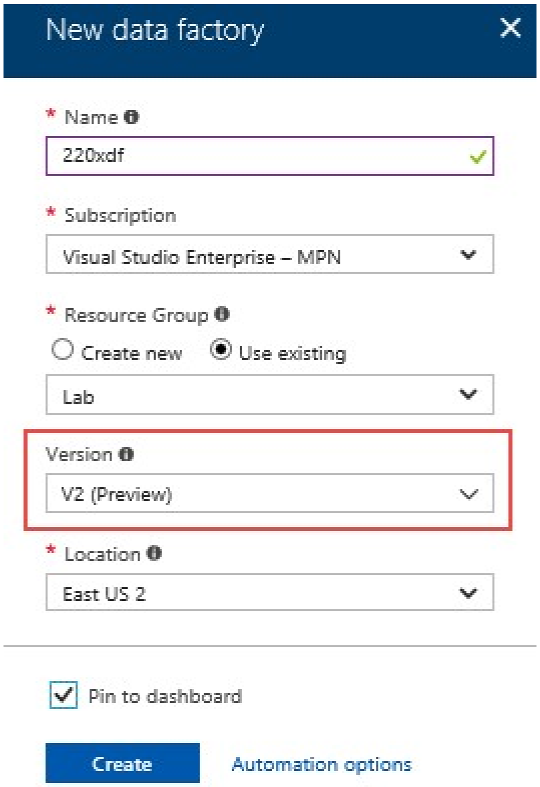
5. Close the query window.
6. In the portal, click the New button.
7. Select Data + Analytics.
8. Select Data Factory.



9. In the New data factory blade, provide a data factory name, select the Lab resource group, and select Data Factory version 2.

10. Select the Pin to dashboard checkbox.

11. Click Create.


New data factory

* Name: 220xdf

* Subscription: Visual Studio Enterprise – MPN

* Resource Group: ○ Create new ● Use existing

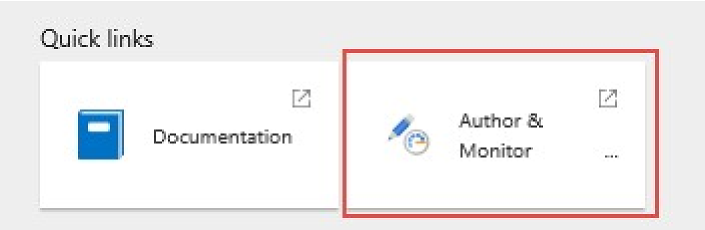
Lab

Version: V2 (Preview)

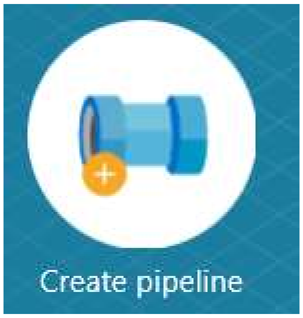
* Location: East US 2

☑ Pin to dashboard

12. Once the data factory is created, review the Essentials pane.
13. In the Quick Links section of the Essentials pane, click the Author & Manage tile.

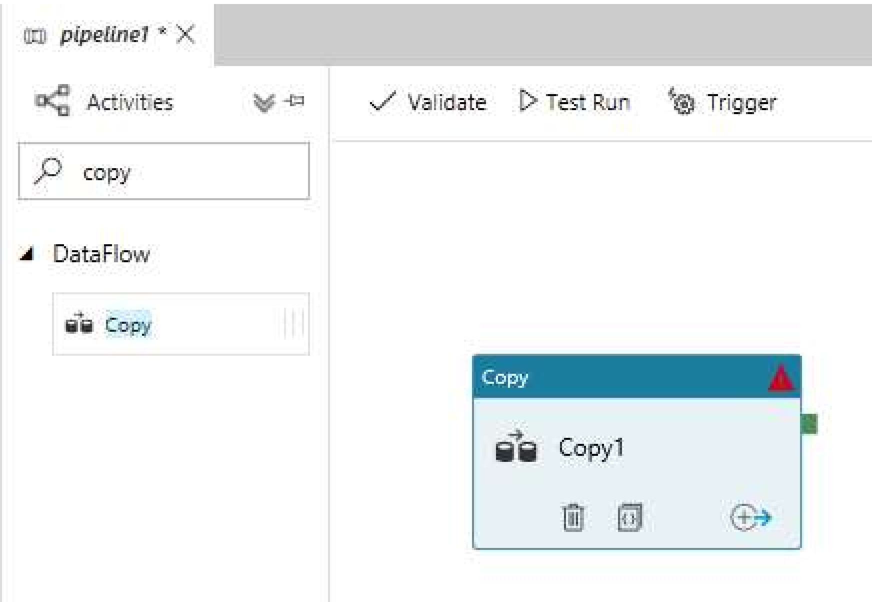


14. A new tab will open in your browser displaying the Let's get started page.
15. Click the Create Pipeline icon.



16. In the new pipeline page, review the different types of activities by expanding each activity group.

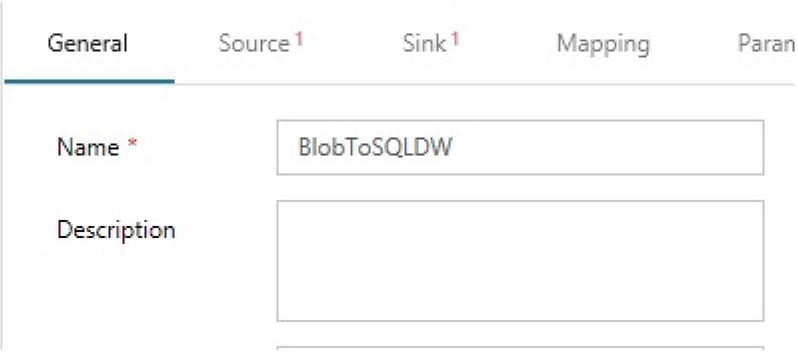
17. In the Search Activities search box, type Copy, and drag the Copy activity onto the pipeline designer surface.



18. The Copy activity displays a Red exclamation point, signifying that the activity has not been configured.

19. Ensure that the Copy activity is select in the pipeline designer surface.

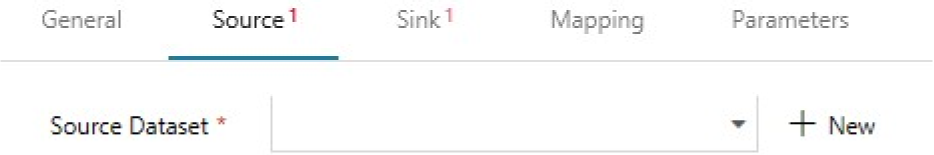
20. On the General tab for the Copy activity, name the Pipeline BlobToSQLDW.



Name * BlobToSQLDW

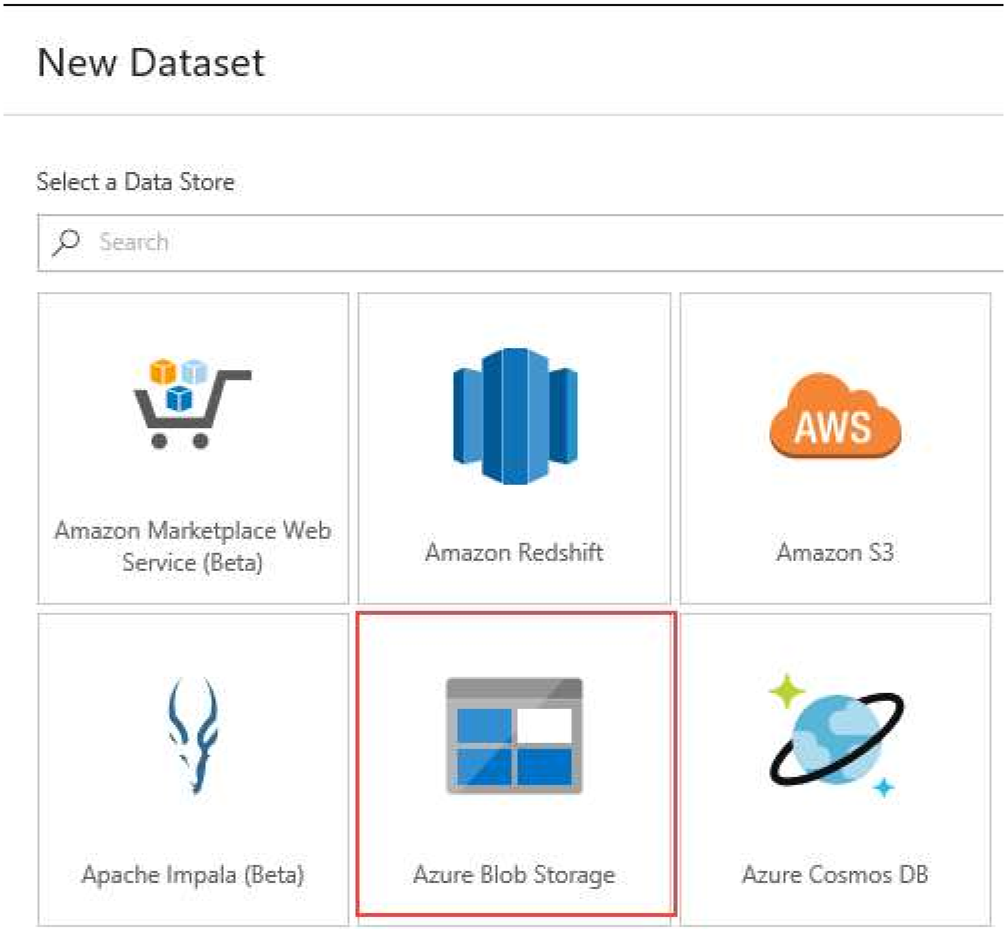
Description

21. Select the Source tab.

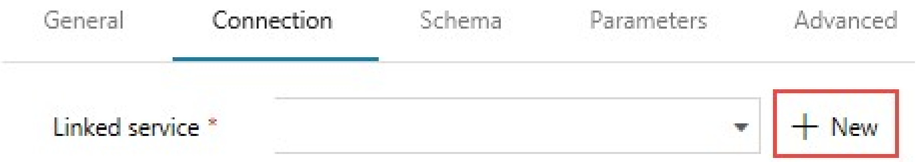


Source Dataset *

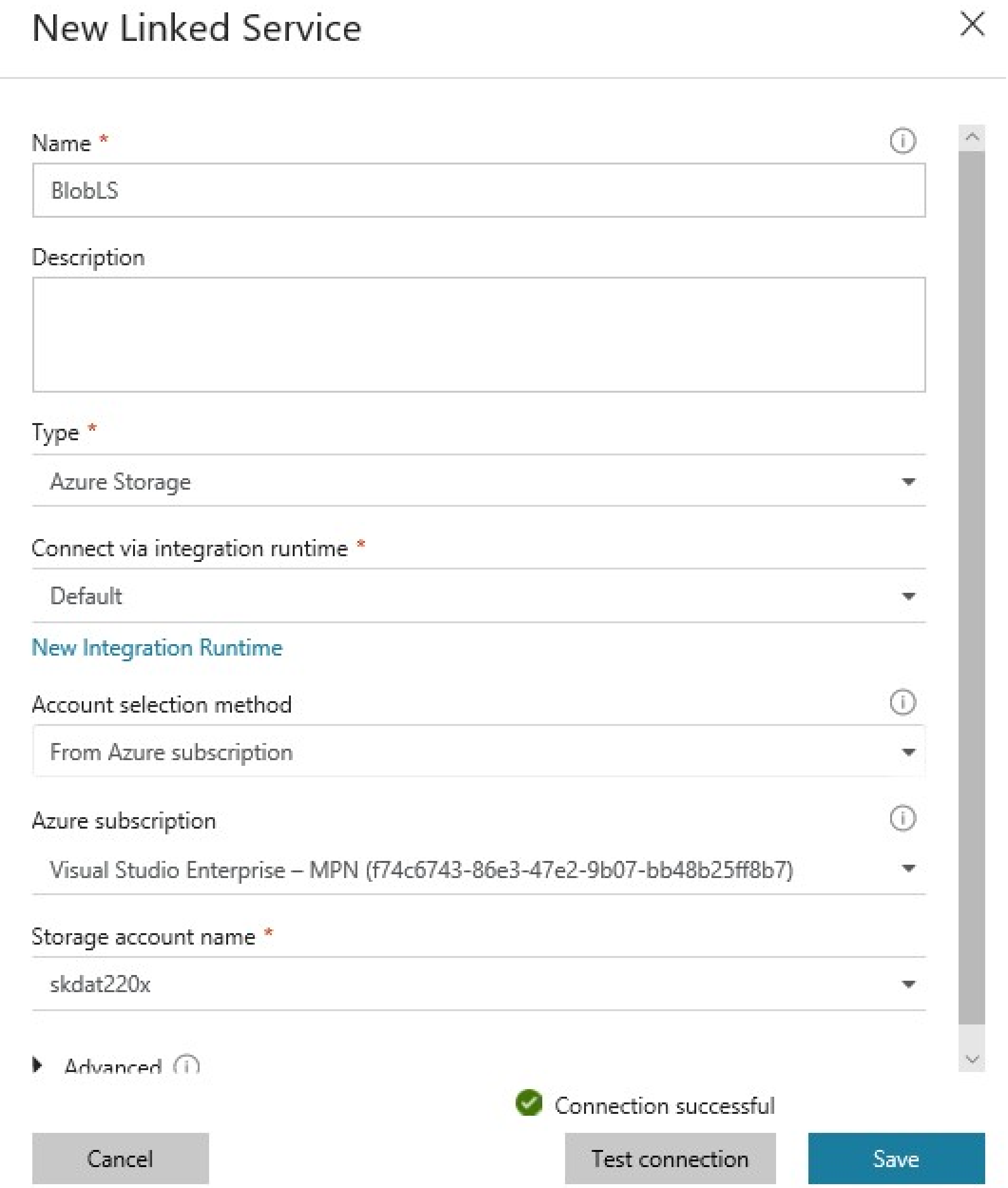
22. Click the + New button next to the Source Dataset.
23. In the New Dataset pane, review all the different data sources which integrate with Azure Data Factory, including Amazon and Teradata.
24. Select Azure Blob Storage.



25. Click Finish.
26. In the General tab for the Azure Blob Storage data set, name the dataset BlobDS.
27. Click the Connection tab.
28. Click the + New button to add a new Linked Service.



29. In the New Linked Service pane, name the Linked Service BlobLS.
30. Select the Azure subscription and the Storage Account name.
31. Leave the other fields with the default values.
32. Click Test Connection.



New Linked Service

Name *

BlobLS

Description

Type *

Azure Storage

Connect via integration runtime *

Default

New Integration Runtime

Account selection method

From Azure subscription

Azure subscription

Visual Studio Enterprise – MPN (f74c6743-86e3-47e2-9b07-bb48b25ff8b7)

Storage account name *

skdat220x

Advanced

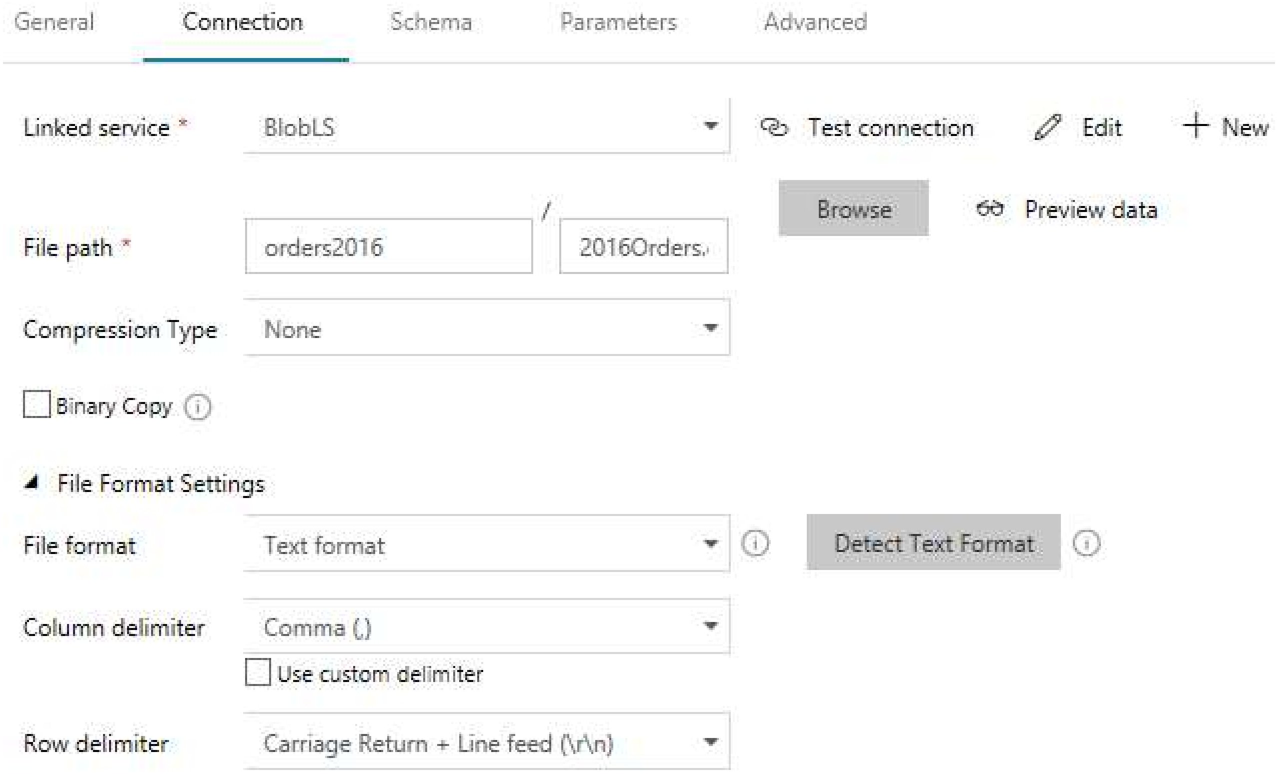
Connection successful

33. When the connection tests successful, click Save.

34. In the Connection tab for the Dataset, enter orders2016 for the first part of the file path, then enter 2016Orders.csv for the file name.

35. Click the Preview Data link to ensure that the connection and file path are configured correctly and to review the data.

36. Ensure that the File format is set to Text format, and that the column delimiter is set to Comma.



Linked service * BlobLS

File path * orders2016 / 2016Orders.

Compression Type None

□ Binary Copy

File Format Settings

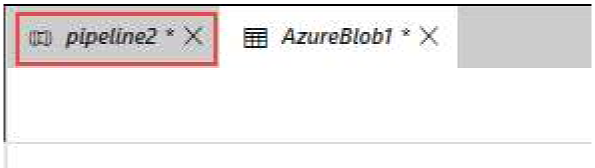
File format Text format

Column delimiter Comma (,)

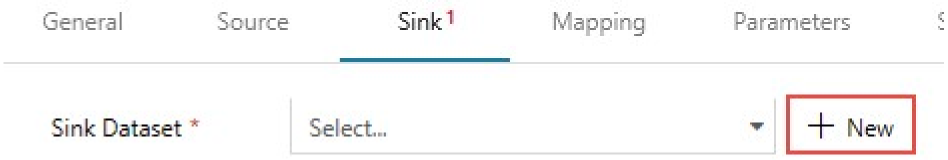
□ Use custom delimiter

Row delimiter Carriage Return + Line feed (\r\n)

37. Click on the pipeline tab.



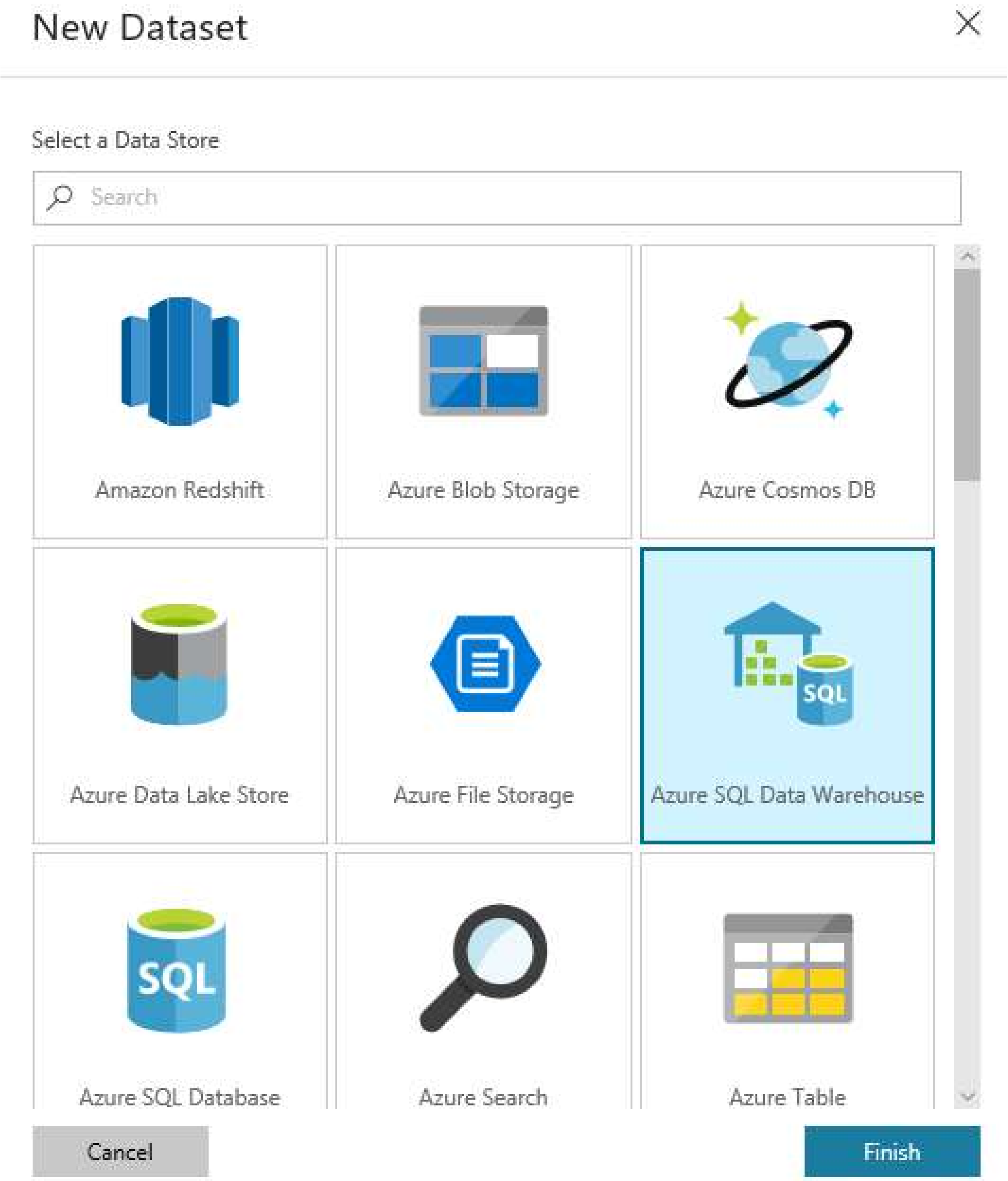
38. Click on the Sink tab for the Copy activity.



Sink Dataset * Select...

39. Click the + New button next to the Sink Dataset.

40. In the New Dataset pane, select Azure SQL Data Warehouse.



New Dataset

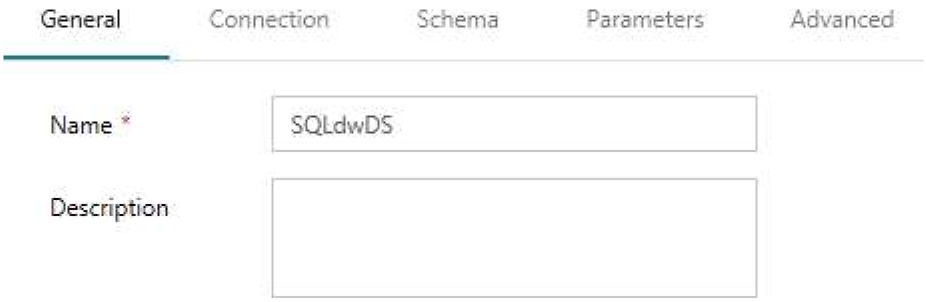
Select a Data Store

41. Click Finish.

42. Name the Dataset SQLdwDS.

Azure SQL Data Warehouse

SQLdwDS

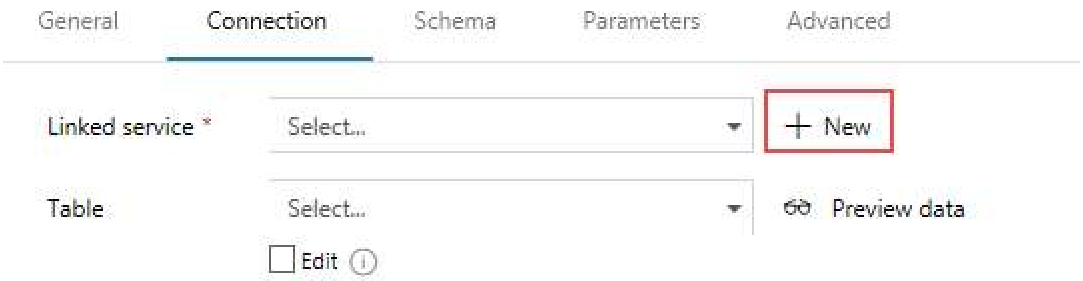


Name * SQLdwDS

Description

43. Click the Connection tab for the dataset.

44. Click the + New button next to the Linked Service.

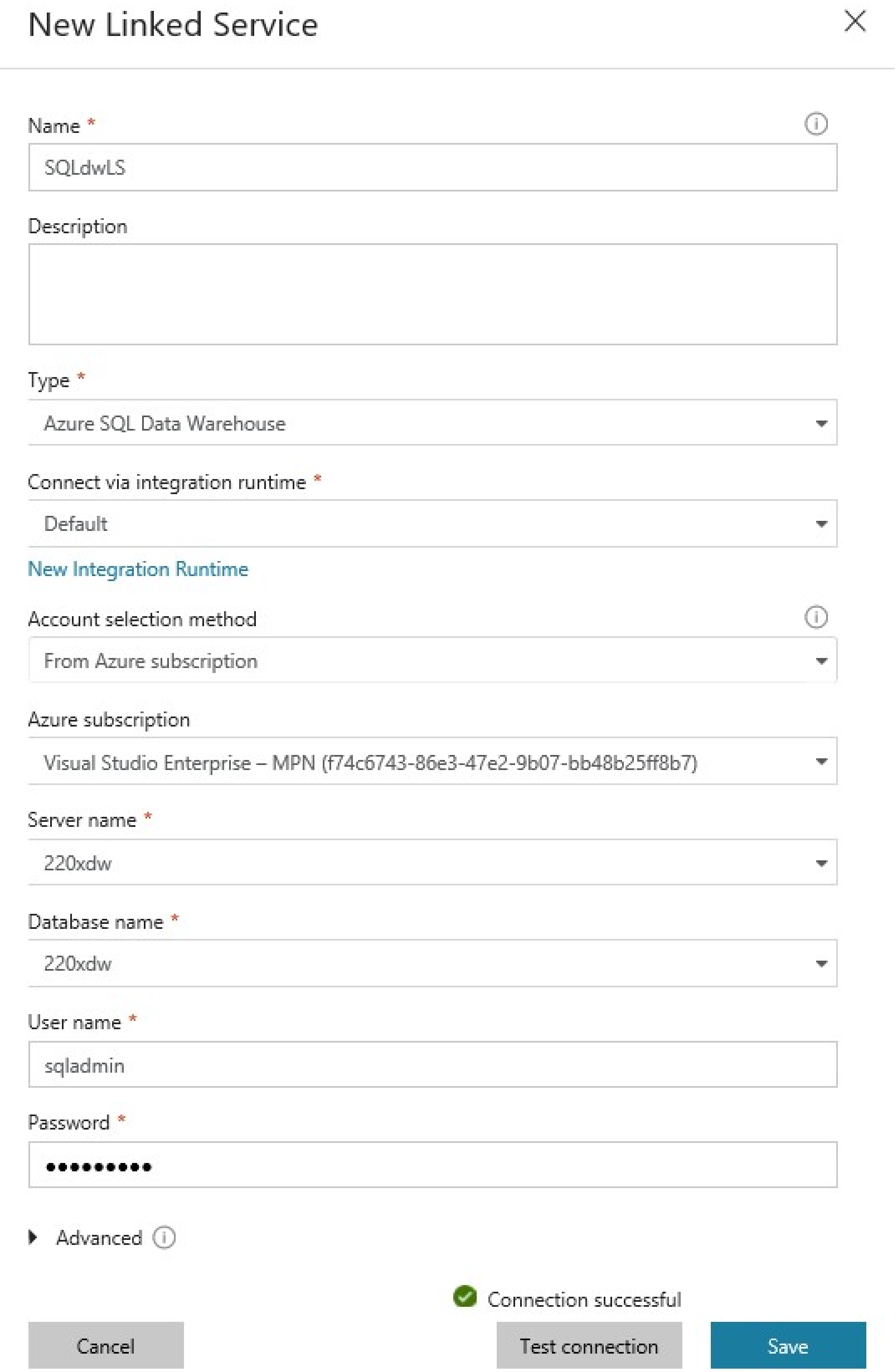


Linked service * Select...

Table Select...

□ Edit

45. In the New Linked Service pane, enter the name of SQLdwLS.
46. Select the Azure subscription.
47. Enter the server name and database name for your SQL Data Warehouse.
48. Enter the username and password for your SQL Data Warehouse.
49. Click Test Connection.



New Linked Service

Name * SQLdwLS

Description

Type * Azure SQL Data Warehouse

Connect via integration runtime * Default

New Integration Runtime

Account selection method From Azure subscription

Azure subscription Visual Studio Enterprise – MPN (f74c6743-86e3-47e2-9b07-bb48b25ff8b7)

Server name * 220xdw

Database name * 220xdw

User name * sqladmin

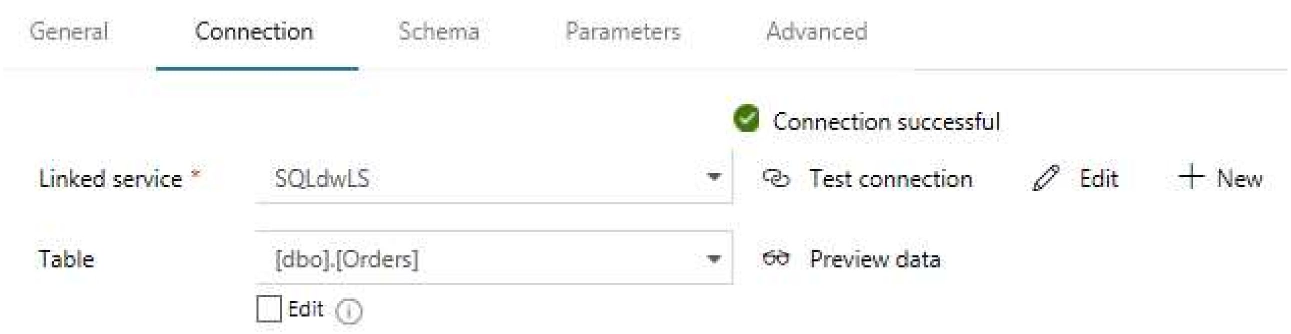
Password * ••••••••••

Advanced

Connection successful

50. When the connection tests successful, click Save.

51. On the Connection tab for the Dataset, select the Orders table from the list of tables.



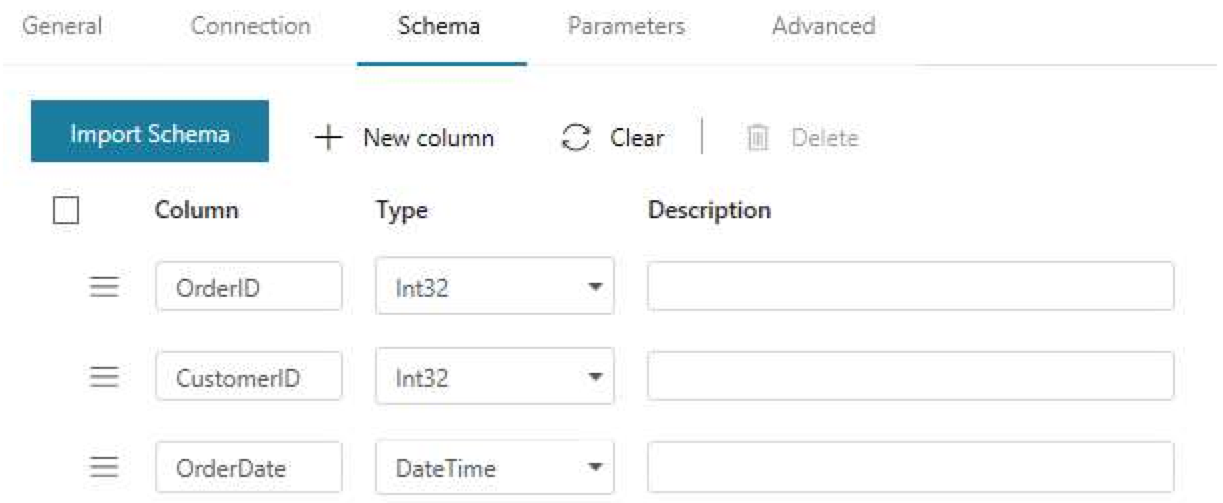
Linked service * SQLdwLS

Table [dbo].[Orders]

□ Edit

52. Click the Schema tab for the dataset.

53. Click Import Schema. Ensure that the OrderID, CustomerID, and OrderDate columns appear with their appropriate data types.

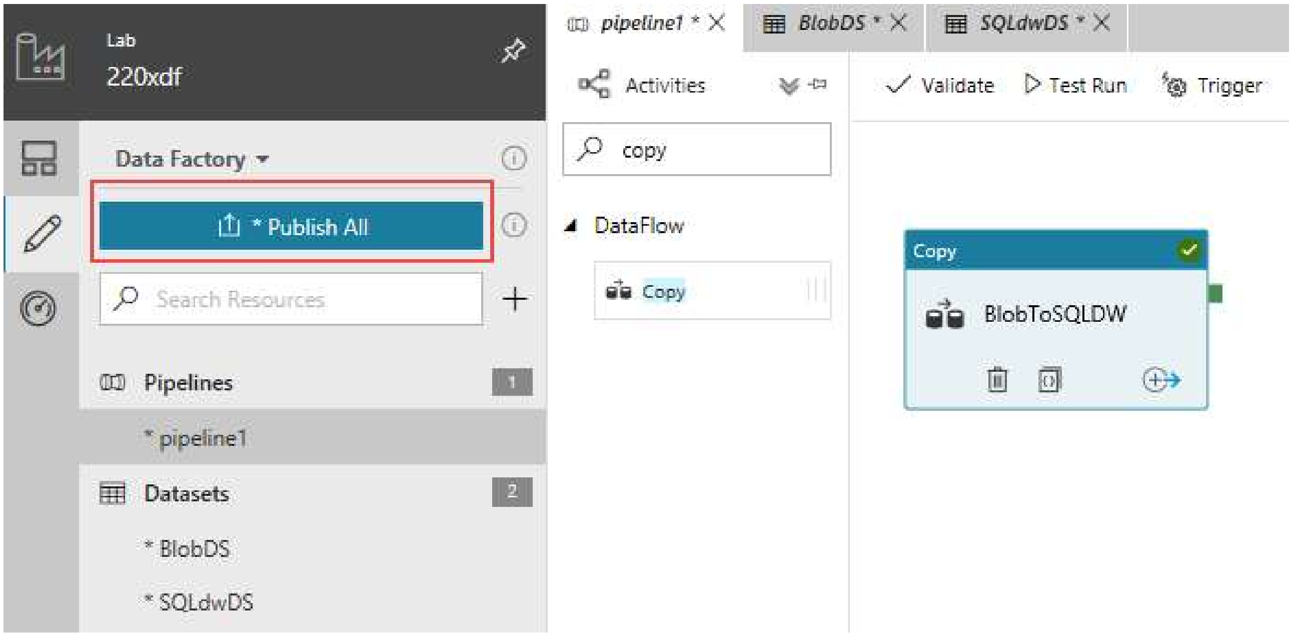


□	Column	Type	Description
	OrderID	Int32	
	CustomerID	Int32	
	OrderDate	DateTime	

54. Click on the pipeline tab.

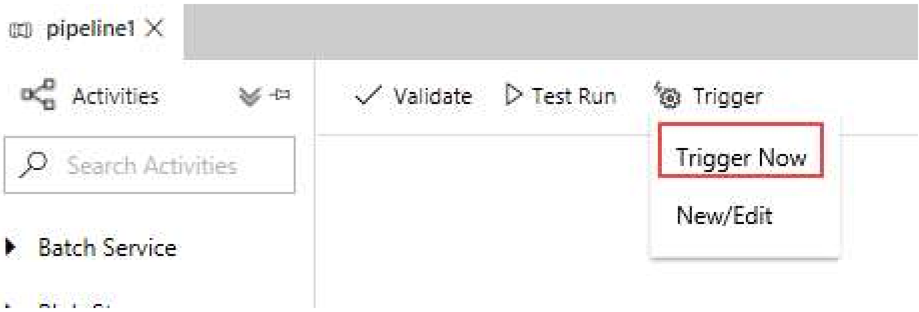
55. The Copy activity now shows a green check mark, indicating that the activity has been properly configured.

56. Click Publish All.

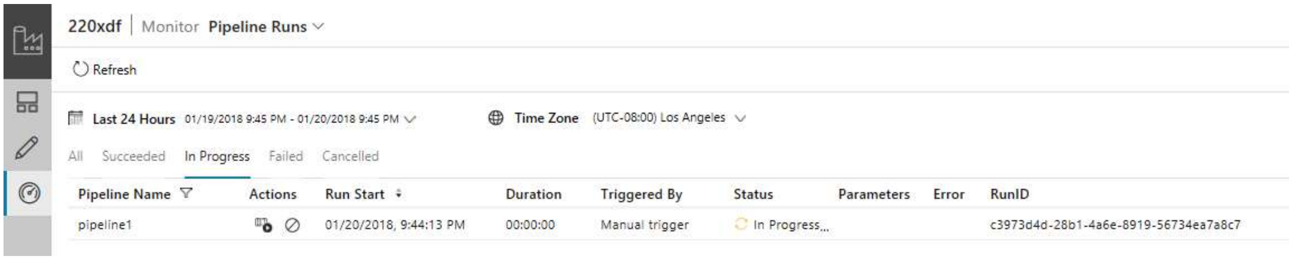


57. At this point the pipeline is configured and published, and ready to be run.

58. Click the Trigger icon on the pipeline toolbar and select Trigger Now.



59. Select the Monitor icon on the left toolbar and select the In Progress tab.

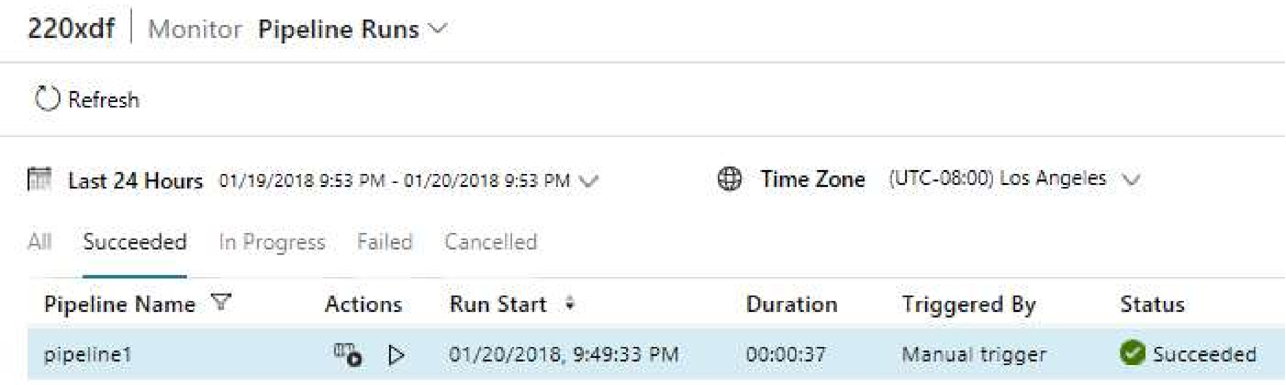

220xdf | Monitor Pipeline Runs
Refresh
Last 24 Hours 01/19/2018 9:45 PM - 01/20/2018 9:45 PM
Time Zone (UTC-08:00) Los Angeles
All
Succeeded
In Progress
Failed
Cancelled

Pipeline Name	Actions	Run Start	Duration	Triggered By	Status	Parameters	Error	RunID
pipeline1		01/20/2018, 9:44:13 PM	00:00:00	Manual trigger	In Progress...			c3973d4d-28b1-4a6e-8919-56734ea7a8c7

60. The Monitor tab allows you to monitor pipeline activity and look at pipeline history. The Succeeded tab shows all pipelines that have run successfully in the past. The In Progress tab shows all pipelines

currently executing and their status. The Failed tab shows all executed pipelines that have failed execution and provides insight into why they failed.

61. Once the pipelines has executed successfully, the record will disappear from the In Progress tab and a new record will appear in the Succeeded tab.



Last 24 Hours 01/19/2018 9:53 PM - 01/20/2018 9:53 PM

Time Zone (UTC-08:00) Los Angeles

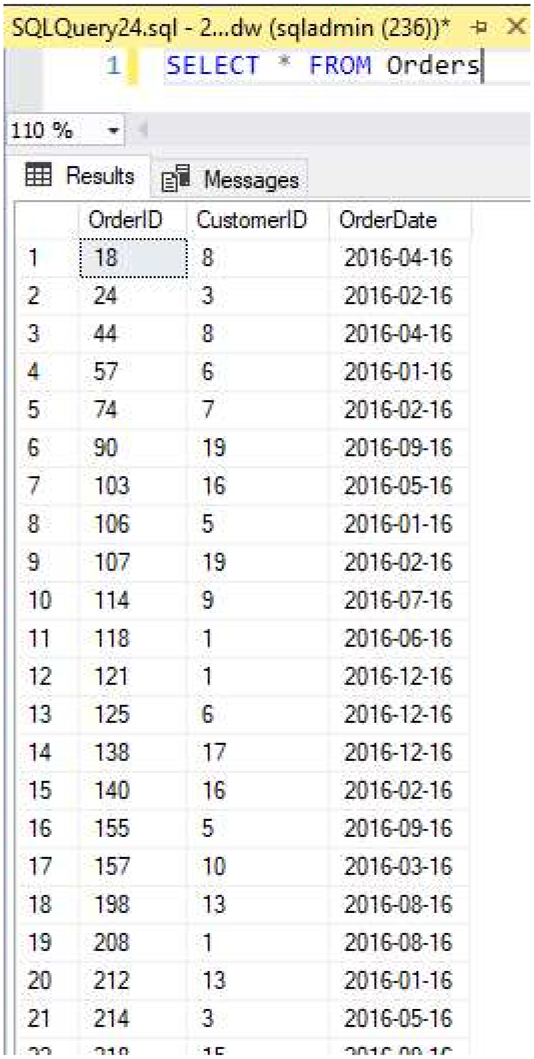
Pipeline Name	Actions	Run Start	Duration	Triggered By	Status
pipeline1		01/20/2018, 9:49:33 PM	00:00:37	Manual trigger	✓ Succeeded

62. In SQL Server Management Studio, select your data warehouse and create a new query window.

63. In the query window, type and execute the following:

```
SELECT * FROM Orders
```

64. The query results will show the rows imported from the text file in blob storage.



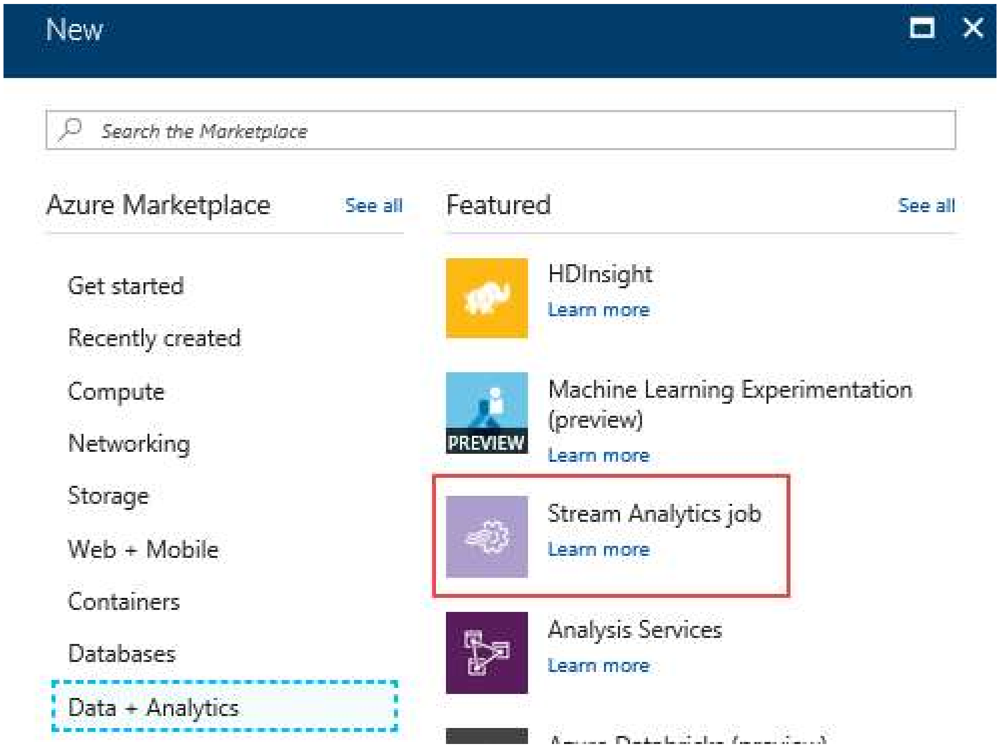
```
SELECT * FROM Orders
```

	OrderID	CustomerID	OrderDate
1	18	8	2016-04-16
2	24	3	2016-02-16
3	44	8	2016-04-16
4	57	6	2016-01-16
5	74	7	2016-02-16
6	90	19	2016-09-16
7	103	16	2016-05-16
8	106	5	2016-01-16
9	107	19	2016-02-16
10	114	9	2016-07-16
11	118	1	2016-06-16
12	121	1	2016-12-16
13	125	6	2016-12-16
14	138	17	2016-12-16
15	140	16	2016-02-16
16	155	5	2016-09-16
17	157	10	2016-03-16
18	198	13	2016-08-16
19	208	1	2016-08-16
20	212	13	2016-01-16
21	214	3	2016-05-16

Exercise 4: Move Data with Azure Stream Analytics

In this exercise, you will create and Azure Stream Analytics job to move data from Azure Blob Storage to a table in Azure SQL Data Warehouse.

1. In the portal, click the New button.
2. Select Data + Analytics.
3. Select Stream Analytics Job.

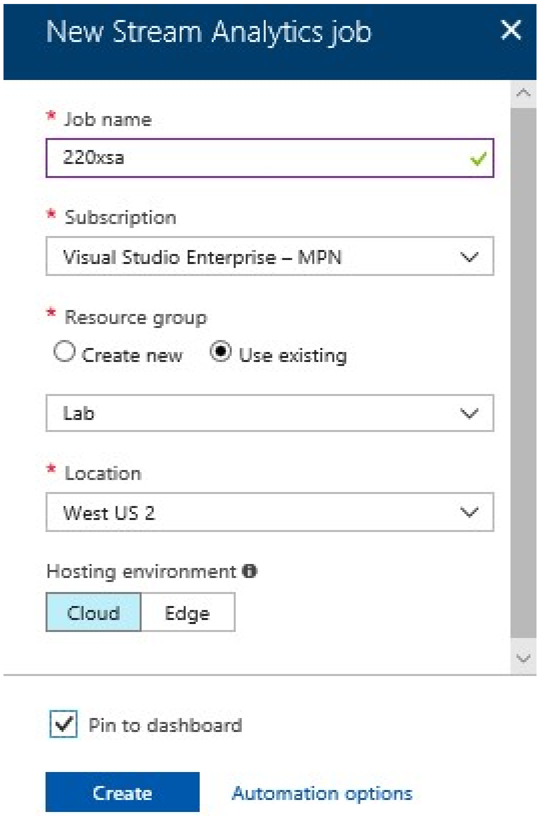


4. In the New Stream Analytics Job pane, enter the name 220xsa, select the Lab resource group.

5. Select the Location nearest you.

6. Check the Pin to dashboard checkbox.

7. Click Create.



New Stream Analytics job

* Job name: 220xsa

* Subscription: Visual Studio Enterprise – MPN

* Resource group: ○ Create new ● Use existing

Lab

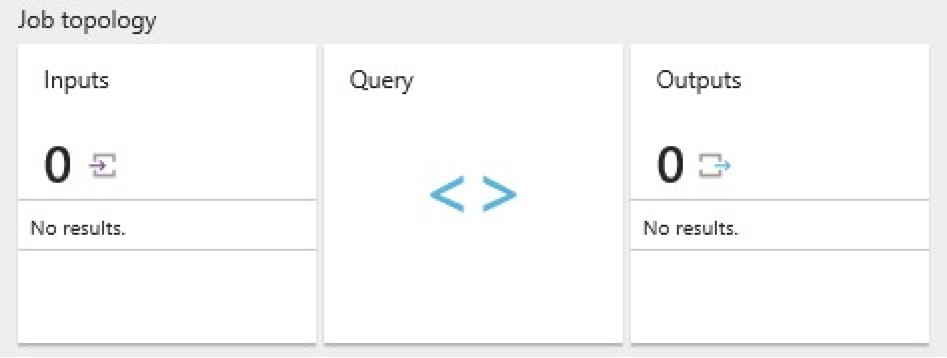
* Location: West US 2

Hosting environment: Cloud | Edge

☑ Pin to dashboard

8. Once the Stream Analytics job is created, the Essentials pane will display. Review the information on the Essentials pane.

9. The Job topology section of the Essentials pane indicates that at least one input and one output are needed to execute a Stream Analytics job.



Job topology

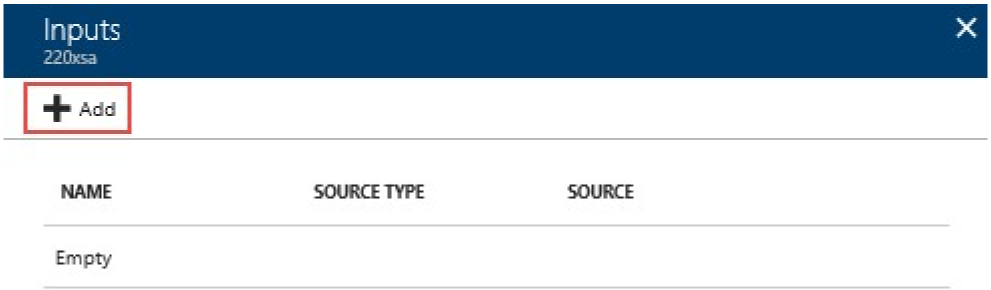
Inputs: 0 — No results.

Query

Outputs: 0 — No results.

10. Click the Inputs tile.

11. In the Inputs pane, click the + Add button.


Inputs
220xsa
Add

NAME	SOURCE TYPE	SOURCE
Empty		

12. Configure the input as follows:

Option	Value	Notes
Input alias	orderinput	
Source Type	Data stream	
Source	Blob storage	
Import Option	Select Blob storage from your subscription	
Storage account		Select the storage account created in Lab 1 and to which you uploaded the files in exercise 2.
Path pattern		Leave this field empty
Container	Orders2017	
Event Serialization format	CSV	
Delimiter	comma	

13. Leave all other options with their default value.

14. Click Create.

15. Close the Inputs pane.

16. In the Essentials pane in the Job topology section, click Outputs.

17. Click the + Add button on the Outputs pane.

18. Configure the output as follows:

Option	Value	Notes
Output alias	orderoutput	
Sink	SQL Database	
Import Option		Select SQL Database from your subscription

Database	220xdw	
Server name		This field will be populated for you.
Username		Enter the user name supplied when creating the SQL DW and server.
Password		Enter the password supplied when creating the SQL DW and server.
Table	Orders	

19. Click Create.

20. Click Close on the Outputs pane.

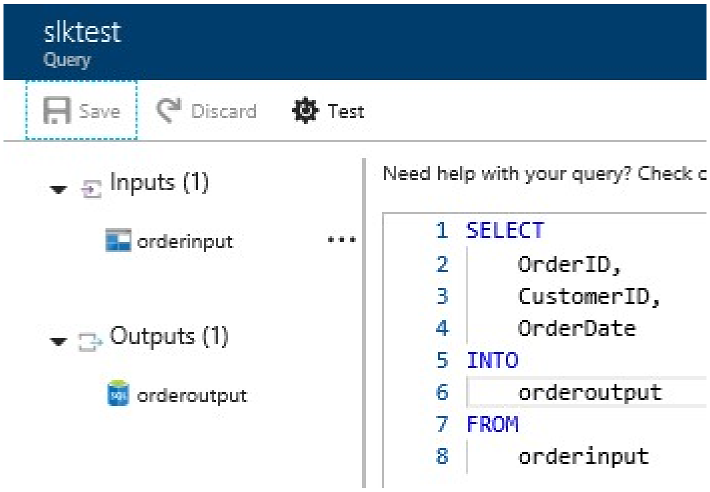
21. In the Essentials pane in the Job topology section, click Query.

22. In the query window, replace [YourOutputAlias] with the name of the output alias, orderoutput, and replace [YourInputAlias] with the name of the input alias, orderinput.

23. Replace the * with the names of the actual column names; OrderID, CustomerID, OrderDate.

24. Click Save.

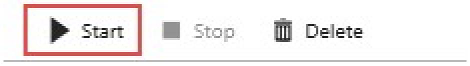
25. The query page should look like the following:



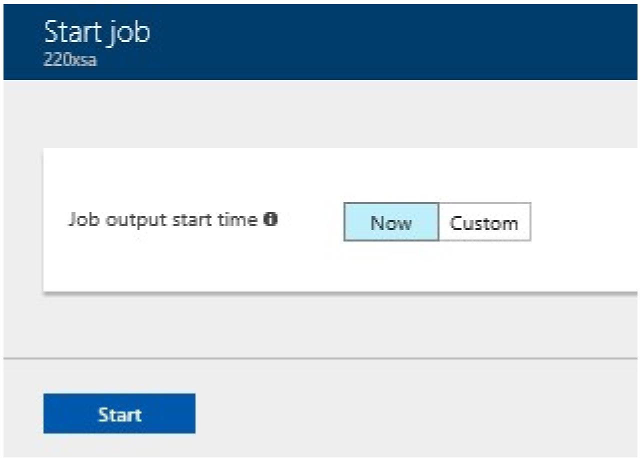
```
SELECT
    OrderID,
    CustomerID,
    OrderDate
INTO
    orderoutput
FROM
    orderinput
```

26. Close the query window.

27. In the Essentials pane for the Stream Analytics job, click the Start button.



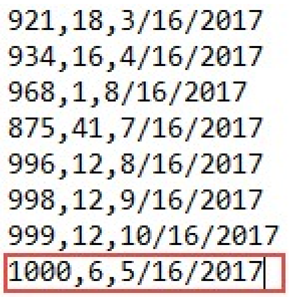
28. In the Start job pane, select the Now start time option and click Start.



29. While the job is starting, navigate to the directory where the sample files are stored, and right mouse click the 2017Orders.csv file and select Edit from the context menu.

30. Scroll to the end of the file and add a new entry for the year 2017:

    1000,6,5/16/2017



```
921,18,3/16/2017
934,16,4/16/2017
968,1,8/16/2017
875,41,7/16/2017
996,12,8/16/2017
998,12,9/16/2017
999,12,10/16/2017
1000,6,5/16/2017
```

31. Save and close the file.

32. In Azure Storage Explorer, select the orders2017 container.

33. In the pane on the right, click the Upload button, then click Upload Files.

34. Browse for and upload the 2017Orders.csv file. Leave the default blob type of Block Blob. When ready, click Upload.

35. Stream Analytics will recognize the arrival of the file in the orders container and process the file.

36. In the query window, type and execute the following:

```
SELECT * FROM Orders
```

37. The query results will show the rows imported using Azure Stream Analytics.

Exercise 5: Use T-SQL with Polybase to Load and Transform Data

In this exercise, you will create a table for the weather data in your SQL DW, using SQL Server Management Studio on your virtual machine. You will also create objects needed to map and import the data from the CSV file in Azure Storage.

1. Switch to or open SQL Server Management Studio (SSMS). Connect to your SQL DW.

2. Enter the following code into a new query window. Do not execute it yet. (The code is also contained in the file DAT20x_Lab_3_Scripts.sql from the .zip file you downloaded in Exercise 1.)

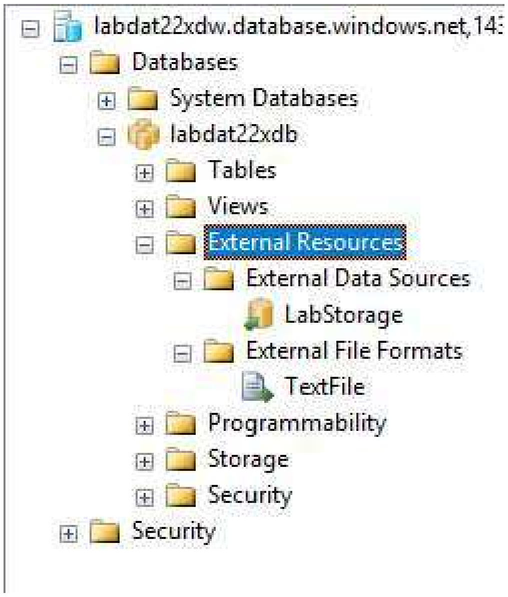
```
CREATE MASTER KEY;

CREATE DATABASE SCOPED CREDENTIAL LabStorageCredential WITH IDENTITY = 'user',
SECRET = '<StorageAccountKey1>';

CREATE EXTERNAL DATA SOURCE LabStorage WITH (
TYPE = HADOOP,
LOCATION =
'wasbs://<StorageContainer>@<StorageAccount>.blob.core.windows.net',
CREDENTIAL = LabStorageCredential
);

CREATE EXTERNAL FILE FORMAT TextFile WITH (
FORMAT_TYPE = DelimitedText, FORMAT_OPTIONS (FIELD_TERMINATOR = ',')
);
```

3. Modify the code to include your Azure storage account, container, and key. You should have this information from completing Lab 1, but if not, you can find them by accessing your storage account from the Azure Portal. Note: the account, container, and key fields are case-sensitive when querying. The values will look something like the following:

   Storage Account: skdat220
   Container: sensors
   Key: DCatHPwArPl2uZMoPd7f2QgC/TtQXfUyX9aeslbXLPszcb4QutzttaM4v1Zh49A==

4. Execute the queries.

5. In the Object Explorer pane, refresh the database, then expand External Resources. You should see your new objects:



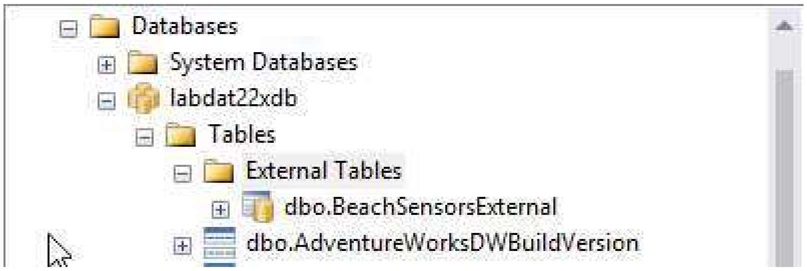
6. You can also verify the new objects by querying the catalog views sys.database_credentials, sys.external_data_sources, and sys_external_file_formats

7. To create the external table for the weather data, enter the following code into a query window that is connected to your SQL DW. (The code is also contained in the file DAT20x_Lab_3_Scripts.sql from the .zip file you downloaded in Exercise 1.)

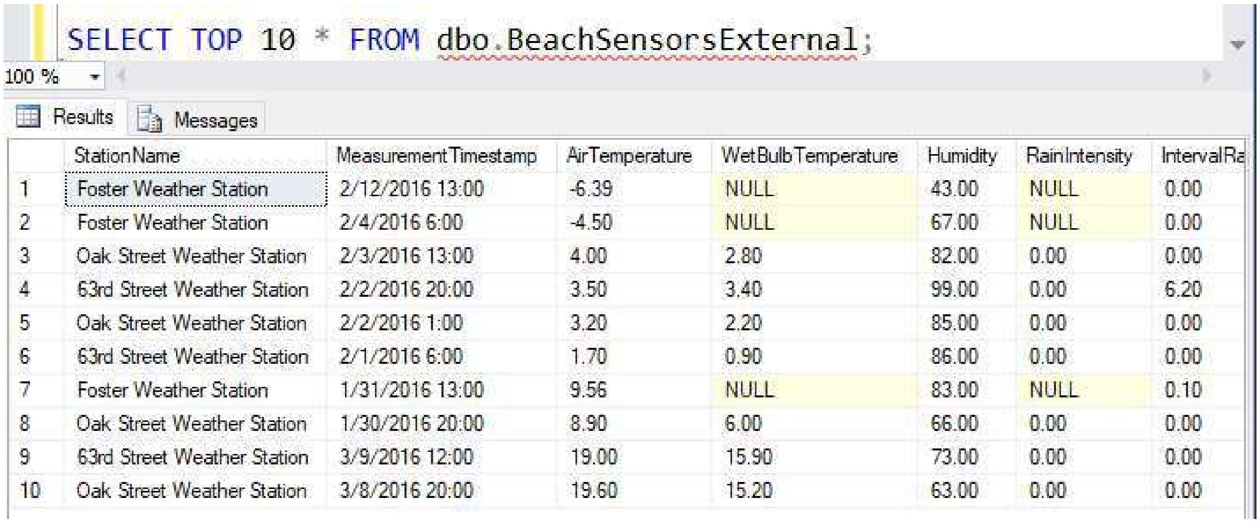
```
CREATE EXTERNAL TABLE dbo.BeachSensorsExternal (
  StationName VARCHAR(50) NOT NULL,
  MeasurementTimestamp VARCHAR(50) NOT NULL,
  AirTemperature DECIMAL(9,2) NULL,
  WetBulbTemperature DECIMAL(9,2) NULL,
  Humidity DECIMAL(9,2) NULL,
  RainIntensity DECIMAL(9,2) NULL,
  IntervalRain DECIMAL(9,2) NULL,
  TotalRain DECIMAL(9,2) NULL,
  PrecipitationType DECIMAL(9,2) NULL,
  WindDirection DECIMAL(9,2) NULL,
  WindSpeed DECIMAL(9,2) NULL,
  MaximumWindSpeed DECIMAL(9,2) NULL,
  BarometricPressure DECIMAL(9,2) NULL,
  SolarRadiation DECIMAL(9,2) NULL,
  Heading DECIMAL(9,2) NULL,
  BatteryLife DECIMAL(9,2) NULL,
  MeasurementTimestampLabel VARCHAR(50) NOT NULL,
  MeasurementID VARCHAR(100) NOT NULL
)
WITH (
  LOCATION='/',
  DATA_SOURCE=LabStorage, FILE_FORMAT=TextFile
);
```

8. In the Object Explorer pane, refresh the database, then expand Tables, then External

Resources. You should see your new objects:



9. Test the new table with a SELECT query:



```
SELECT TOP 10 * FROM dbo.BeachSensorsExternal;
```

	StationName	MeasurementTimestamp	AirTemperature	WetBulbTemperature	Humidity	RainIntensity	IntervalRa
1	Foster Weather Station	2/12/2016 13:00	-6.39	NULL	43.00	NULL	0.00
2	Foster Weather Station	2/4/2016 6:00	-4.50	NULL	67.00	NULL	0.00
3	Oak Street Weather Station	2/3/2016 13:00	4.00	2.80	82.00	0.00	0.00
4	63rd Street Weather Station	2/2/2016 20:00	3.50	3.40	99.00	0.00	6.20
5	Oak Street Weather Station	2/2/2016 1:00	3.20	2.20	85.00	0.00	0.00
6	63rd Street Weather Station	2/1/2016 6:00	1.70	0.90	86.00	0.00	0.00
7	Foster Weather Station	1/31/2016 13:00	9.56	NULL	83.00	NULL	0.10
8	Oak Street Weather Station	1/30/2016 20:00	8.90	6.00	66.00	0.00	0.00
9	63rd Street Weather Station	3/9/2016 12:00	19.00	15.90	73.00	0.00	0.00
10	Oak Street Weather Station	3/8/2016 20:00	19.60	15.20	63.00	0.00	0.00

10. Now that you have created the external table containing the raw data, you will create a second table in your SQL Data Warehouse, transform the data, and load it. Copy and paste the following code into SSDT, then execute it against your SQL DW database.

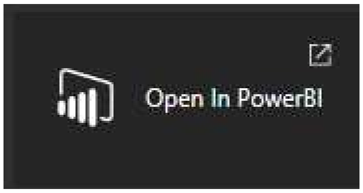
    In this code, you use T-SQL to create a new table called BeachSensor, then take a subset of columns from the raw table. After the BeachSensor table loads with the modified data from the external table, a query that displays the table's record count executes.

```
CREATE TABLE [dbo].[BeachSensor]
WITH (
  DISTRIBUTION = ROUND_ROBIN,
  CLUSTERED COLUMNSTORE INDEX

) AS
SELECT
  StationName,
  CAST(MeasurementTimestamp as DATETIME) AS MeasurementDateTime,
  AirTemperature,
  WetBulbTemperature,
  Humidity,
  RainIntensity,
  IntervalRain,
  TotalRain,
  PrecipitationType,
    WindDirection,
    WindSpeed,
    MaximumWindSpeed,
    BarometricPressure,
    SolarRadiation,
    Heading,
    BatteryLife
  FROM dbo.BeachSensorsExternal;

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

Exercise 6: Visualize Data with PowerBI

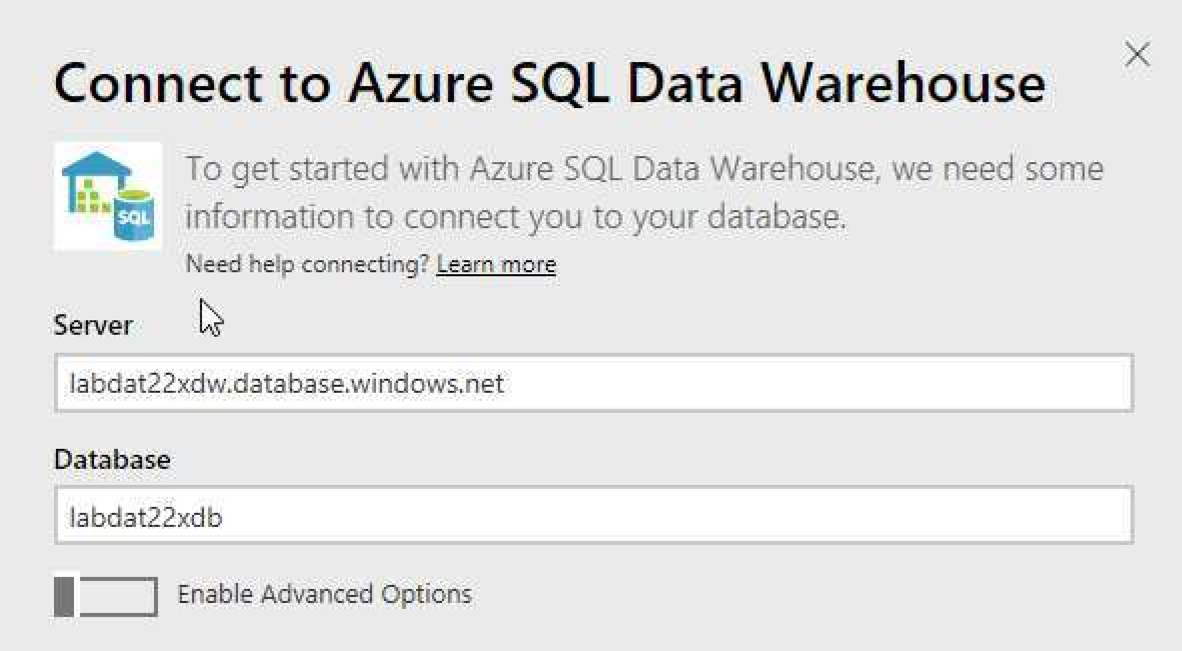
In this exercise you will explore the data that you imported in your data warehouse during the previous exercise by connecting to Power BI through the Azure Portal.

1. In the Azure Portal, open your SQL Data Warehouse database. At the top of the blade, click the Open in Power BI icon.



2. In the new browser tab that opens, follow the prompts to sign in to PowerBI.com, using the same credentials you use for the Azure Portal.

3. A Connect to Azure SQL Data Warehouse window should appear, with your Server and Database already populated. Click Next.



Connect to Azure SQL Data Warehouse

To get started with Azure SQL Data Warehouse, we need some information to connect you to your database.

Need help connecting? Learn more

Server

labdat22xdw.database.windows.net

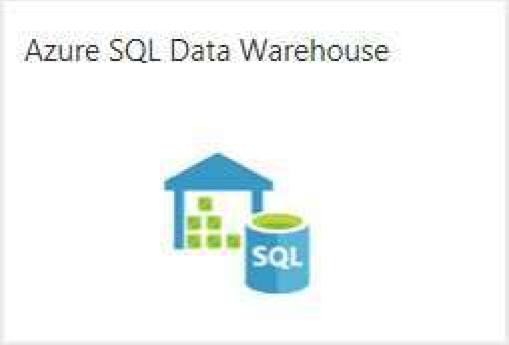
Database

labdat22xdb

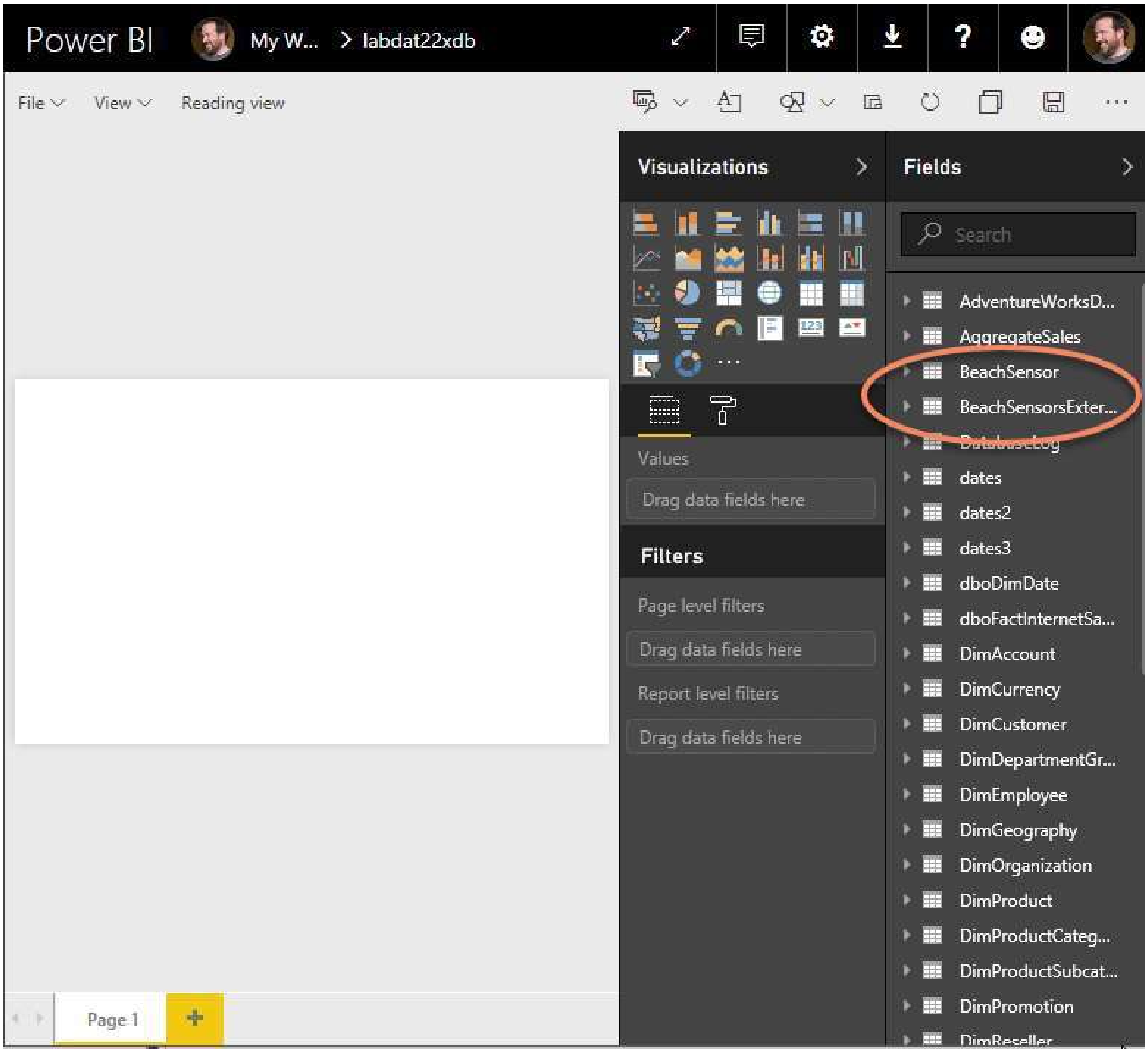
Enable Advanced Options

4. Enter your credentials for your SQL DW.

5. Once the schema has been imported into the PowerBI dashboard, a tile appears to represent your SQL DW:

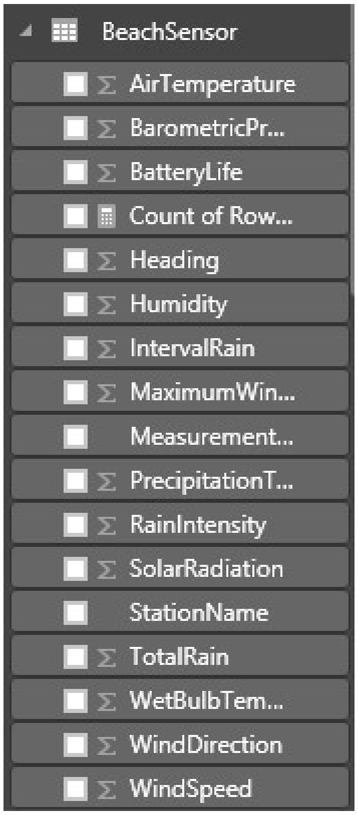


6. Click the tile to open the report editor:



7. Review the fields list. Note that all the sample database tables and views are listed, as well as those you created in this lab.

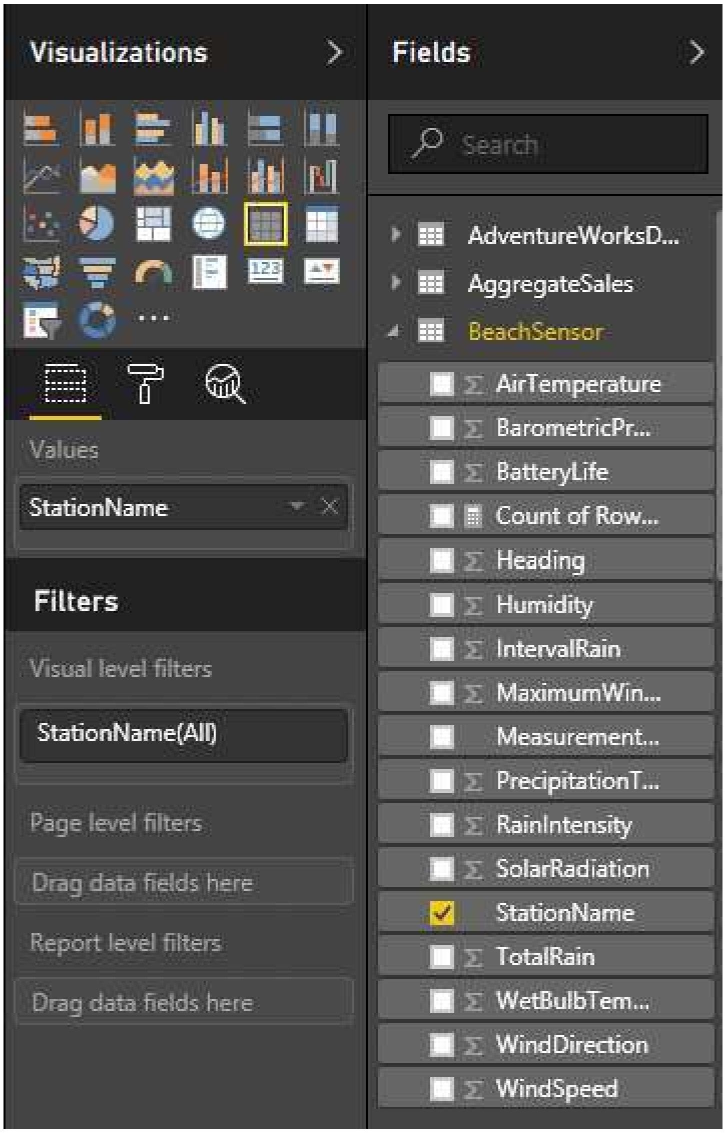
8. Expand the BeachSensor table by clicking on the arrow to the left of the table name.



BeachSensor

□ Σ AirTemperature
□ Σ BarometricPr...
□ Σ BatteryLife
□ Count of Row...
□ Σ Heading
□ Σ Humidity
□ Σ IntervalRain
□ Σ MaximumWin...
□ Measurement...
□ Σ PrecipitationT...
□ Σ RainIntensity
□ Σ SolarRadiation
□ StationName
□ Σ TotalRain
□ Σ WetBulbTem...
□ Σ WindDirection
□ Σ WindSpeed

9. Select StationName by checking the box to the left of the field name. StationName will appear in the Values section of the field well, and a table displaying the three weather station locations will appear on the report.



Values

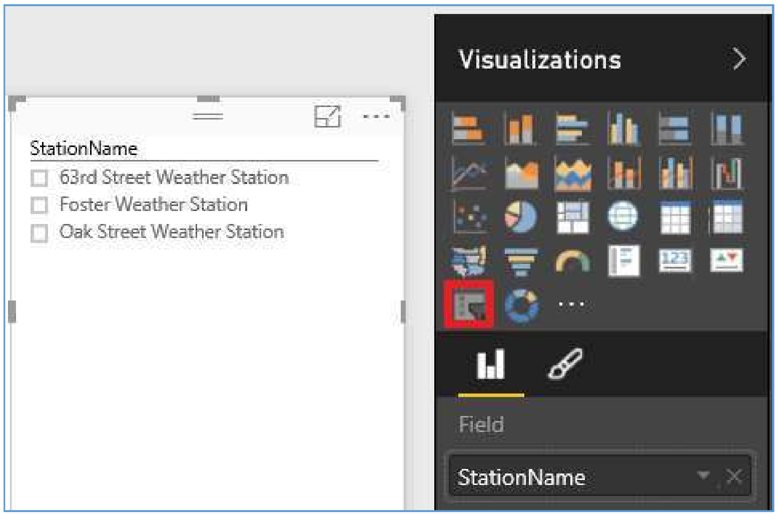
StationName

Visual level filters

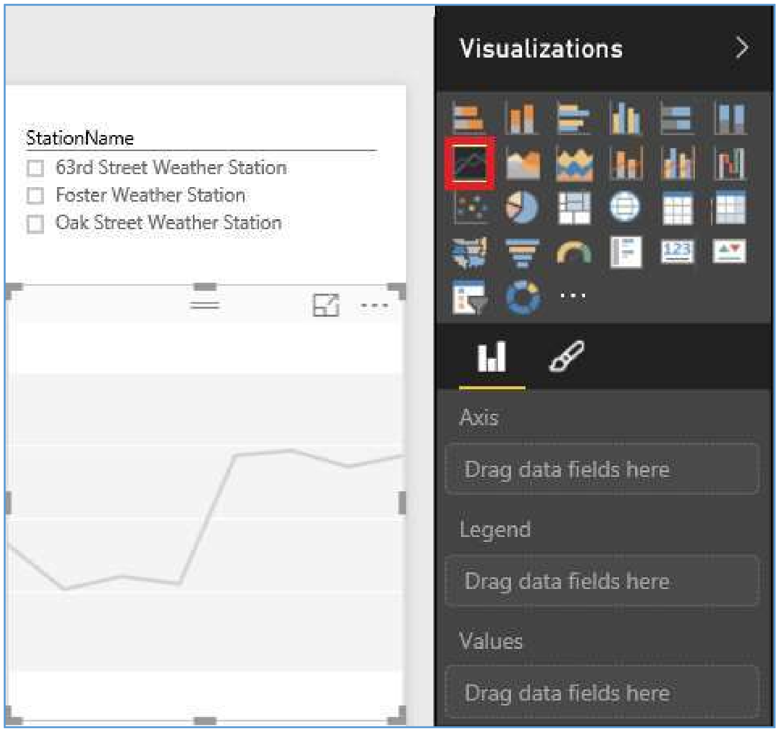
StationName(All)

☑ StationName

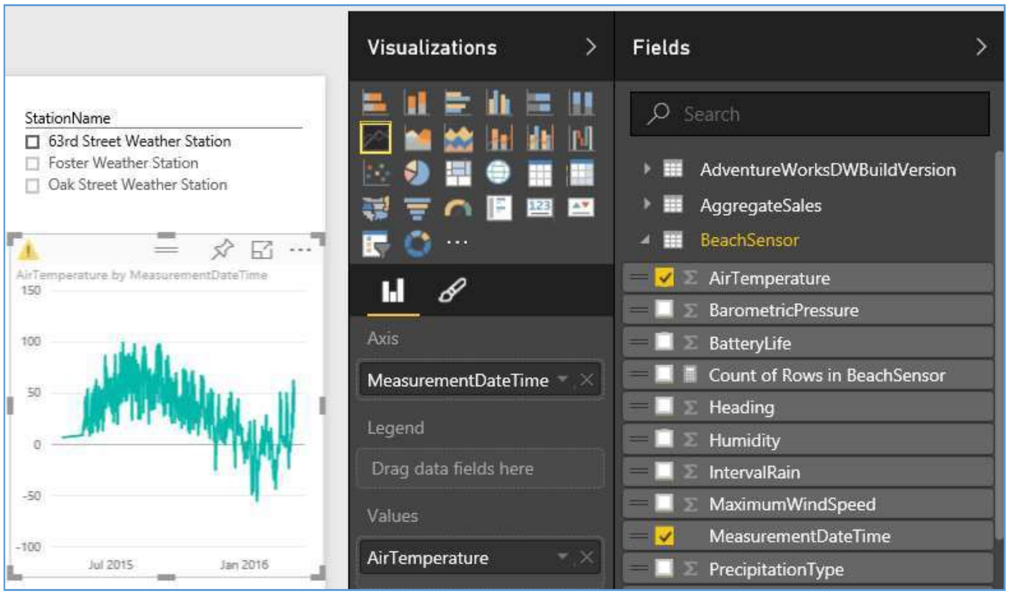
10. In the Visualizations pane, change the table to a slicer by clicking on the Slicer icon. On the report, the visual that previously displayed the three weather station names should now display checkboxes next to the station names.



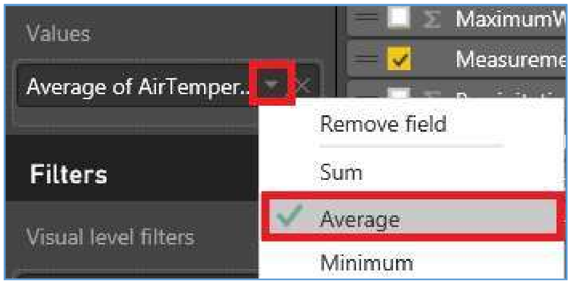
11. Click on whitespace in the report so that the slicer visual is no longer selected. In the Visualizations pane, select Line chart, and a new empty visual will display on the report.



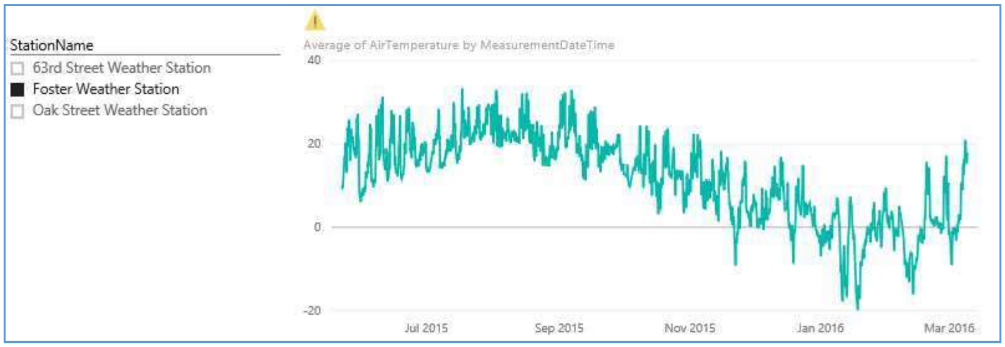
12. In the Fields pane, select MeasurementDateTime, which should appear under Axis in the field well. Then, select AirTemperature, which should appear under Values in the field well. The line chart should also display at this stage.



13. The air temperature (in degrees Celsius) will aggregate using Sum by default. Click the small arrow to the right of AirTemperature in Values, then select Average. The labels for the Y Axis on the line chart will adjust automatically.



14. Check the boxes in the StationName slicer to view the different air temperatures foreach location separately. Hover over the line chart to see a tooltip with exact values for each measurement.


StationName
☐ 63rd Street Weather Station
■ Foster Weather Station
☐ Oak Street Weather Station
Average of AirTemperature by MeasurementDateTime
40
20
0
-20
Jul 2015
Sep 2015
Nov 2015
Jan 2016
Mar 2016

15. As desired, continue to explore Power BI by adding new visuals and using different fields from the dataset. When you are done, close the Power BI browser tab.

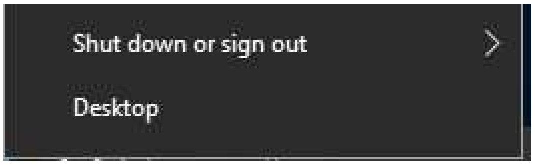
You have now completed the lab. Be sure to complete the Finishing Up exercise to shut down and stop the VM, and to pause the SQL DW.

Finishing Up

In this task, you will shut down and stop the VM, and pause the SQL DW. If you are immediately continuing to further labs in this course, you can skip this task. However, costs will continue to be incurred by the VM and SQL DW until they are deallocated or paused using the steps below.

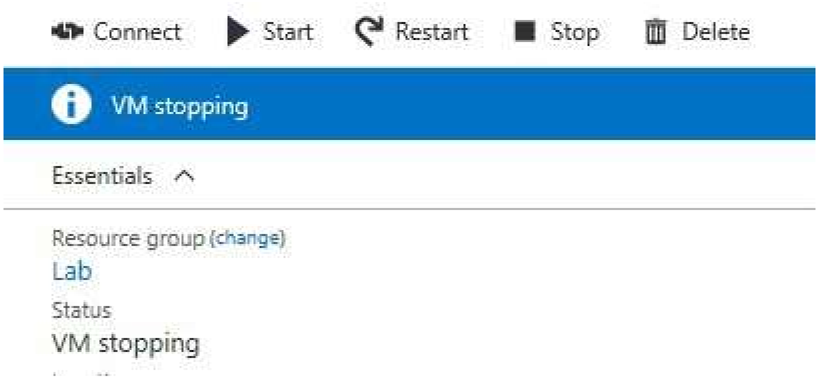
1. Close all open applications.

2. Right-click on the Windows button in the bottom left corner of your screen. Click on Shut down or sign out. Be sure this is the menu for you VM and not your desktop!



3. Click shut down.

4. In the Azure Portal Web browser page on your desktop, wait until the status of the VM updates.



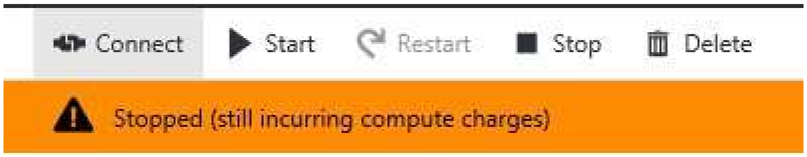
VM stopping

Resource group (change)
Lab

Status
VM stopping

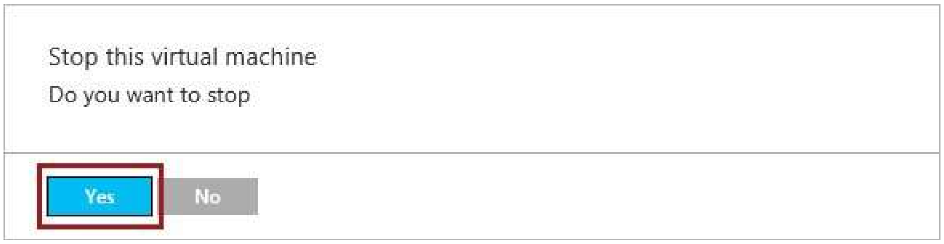
In this state, however, the VM is still billable.

5. To deallocate the VM, click Stop.



Stopped (still incurring compute charges)

6. When prompted to stop the VM, click Yes.

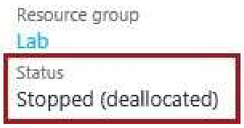


Stop this virtual machine

Do you want to stop

The deallocation can take several minutes to complete.

7. Verify that the VM status updates to Stopped (Deallocated).



Resource group

Lab

Status

Stopped (deallocated)

In this state, the VM is now not billable.

Note that a deallocated VM will likely acquire a different IP address the next time it is started.

8. Browse the Azure Portal to open the blade for your SQL DW.

9. Click the Pause button. Watch the status change, and confirm that the SQL DW has paused before you finish the lab.

10. Sign out of the Azure Portal.