

Big data and visualization hands-on lab step-by-step

Abstract and learning objectives

This hands-on lab is designed to provide exposure to many of Microsoft's transformative line of business applications built using Microsoft big data and advanced analytics.

By the end of the lab, you will be able to show an end-to-end solution, leveraging many of these technologies, but not necessarily doing work in every component possible.

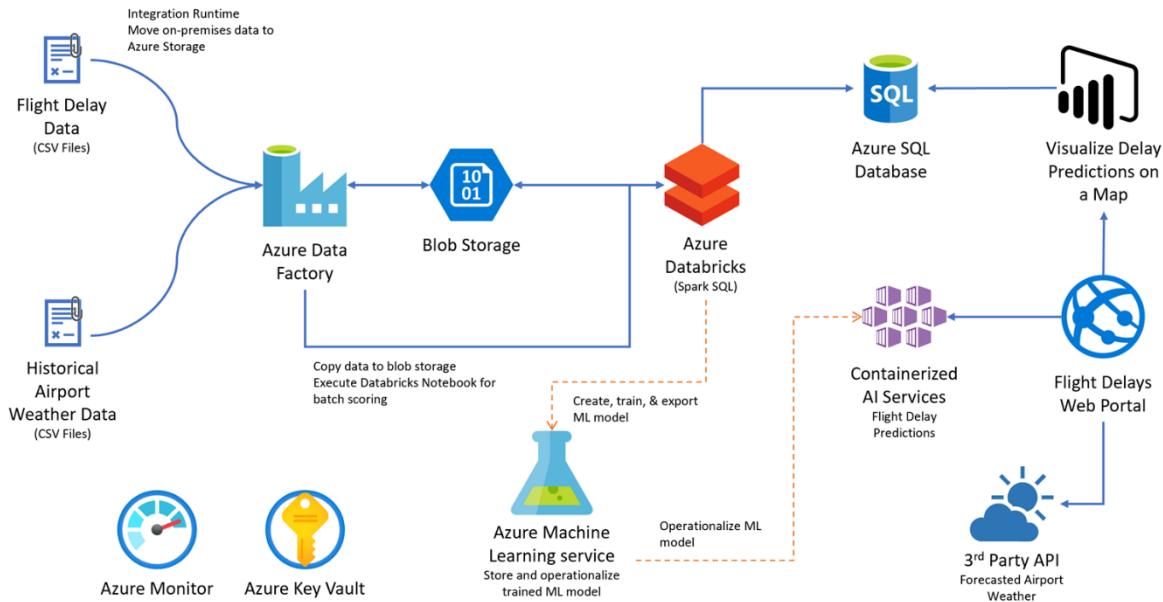
Overview

Margie's Travel (MT) provides concierge services for business travelers. In an increasingly crowded market, they are always looking for ways to differentiate themselves, and provide added value to their corporate customers.

They are looking to pilot a web app that their internal customer service agents can use to provide additional information useful to the traveler during the flight booking process. They want to enable their agents to enter in the flight information and produce a prediction as to whether the departing flight will encounter a 15-minute or longer delay, considering the weather forecasted for the departure hour.

Solution architecture

Below is a diagram of the solution architecture you will build in this lab. Please study this carefully so you understand the whole of the solution as you are working on the various components.



Requirements

1. Microsoft Azure subscription must be pay-as-you-go or MSDN.
 - o Trial subscriptions will not work.
2. If you are not a Service Administrator or Co-administrator for the Azure subscription, or if you are running the lab in a hosted environment, you will need to install [Visual Studio 2019 Community](#) with the **ASP.NET and web development** and **Azure development** workloads.
3. Follow all the steps provided in [Before the Hands-on Lab](#).

Exercise 1: Retrieve lab environment information and create Databricks cluster

In this exercise, you will retrieve your Azure Storage account name and access key and your Azure Subscription Id and record the values to use later within the lab. You will also create a new Azure Databricks cluster.

Task 1: Retrieve Azure Storage account information and Subscription Id

You will need to have the Azure Storage account name and access key when you create your Azure Databricks cluster during the lab. You will also need to create storage containers in which you will store your flight and weather data files.

- From the side menu in the Azure portal, choose **Resource groups**, then enter your resource group name into the filter box, and select it from the list.
- Next, select your lab Azure Storage account from the list.

<input type="checkbox"/> NAME ↑↓	<input type="checkbox"/> TYPE ↑↓
<input type="checkbox"/> BigDataLab	Azure Databricks Service
<input type="checkbox"/> big-data-lab	Machine Learning Experimenta...
<input type="checkbox"/> mcwailabexp (big-data-lab/mcwailabexp)	Microsoft.MachineLearningExp...
<input type="checkbox"/> bigdatalabexpstorage	Storage account
<input type="checkbox"/> BigDataLabFactory	Data factory (V2)
<input type="checkbox"/> big-data-lab-model-mgmt	Machine Learning Model Man...
<input type="checkbox"/> bigdatalabstore	Storage account

- On the left menu, select **Overview**, locate and copy your Azure **Subscription ID** and save to a text editor such as Notepad for later.

Resource group (change) :	hands-on-lab-bigdata
Status :	Primary: Available, Secondary: Available
Location :	West US, East US
Subscription (change) :	
Subscription ID :	-a031-1b21726acc1a
Tags (change) :	Click here to add tags

- Select **Access keys** (1) from the menu. Copy the **storage account name** (2) and the **key1** key (3) and copy the values to a text editor such as Notepad for later.

When you regenerate your access keys, you must update any Azure resources and applications that access this storage account to use the new keys. This action will not interrupt access to disks from your virtual machines. [Learn more](#)

Storage account name
bigdatalabstore

key1

Key
HD+91Y77b+TezEu1lh9QXXU2Va6Cjg9bu0RRpb/KtBj8IWQa6jwyA0OGTDmSNVFr8iSlkytFONEHLDl67Fgxg==

Connection string
DefaultEndpointsProtocol=https;AccountName=bigdatalabstore;AccountKey=HD+91Y77b+TezEu1lh9QXXU2Va6Cjg9b...

key2

Key

Task 2: Create an Azure Databricks cluster

You have provisioned an Azure Databricks workspace, and now you need to create a new cluster within the workspace. Part of the cluster configuration includes setting up an account access key to your Azure Storage account, using the Spark Config within the new cluster form. This will allow your cluster to access the lab files.

1. From the side menu in the Azure portal, select **Resource groups**, then enter your resource group name into the filter box, and select it from the list.
2. Next, select your Azure Databricks service from the list.

Name ↑↓	Type ↑↓
<input type="checkbox"/> BigDataLab	Azure Databricks Service
<input type="checkbox"/> BigDataLabFactory	Data factory (V2)
<input type="checkbox"/> bigdatalabstore	Storage account

3. In the Overview pane of the Azure Databricks service, select **Launch Workspace**.



[Launch Workspace](#)

Getting Started



Import Data from File



Azure Databricks will automatically log you in using Azure Active Directory Single Sign On.



Azure Databricks

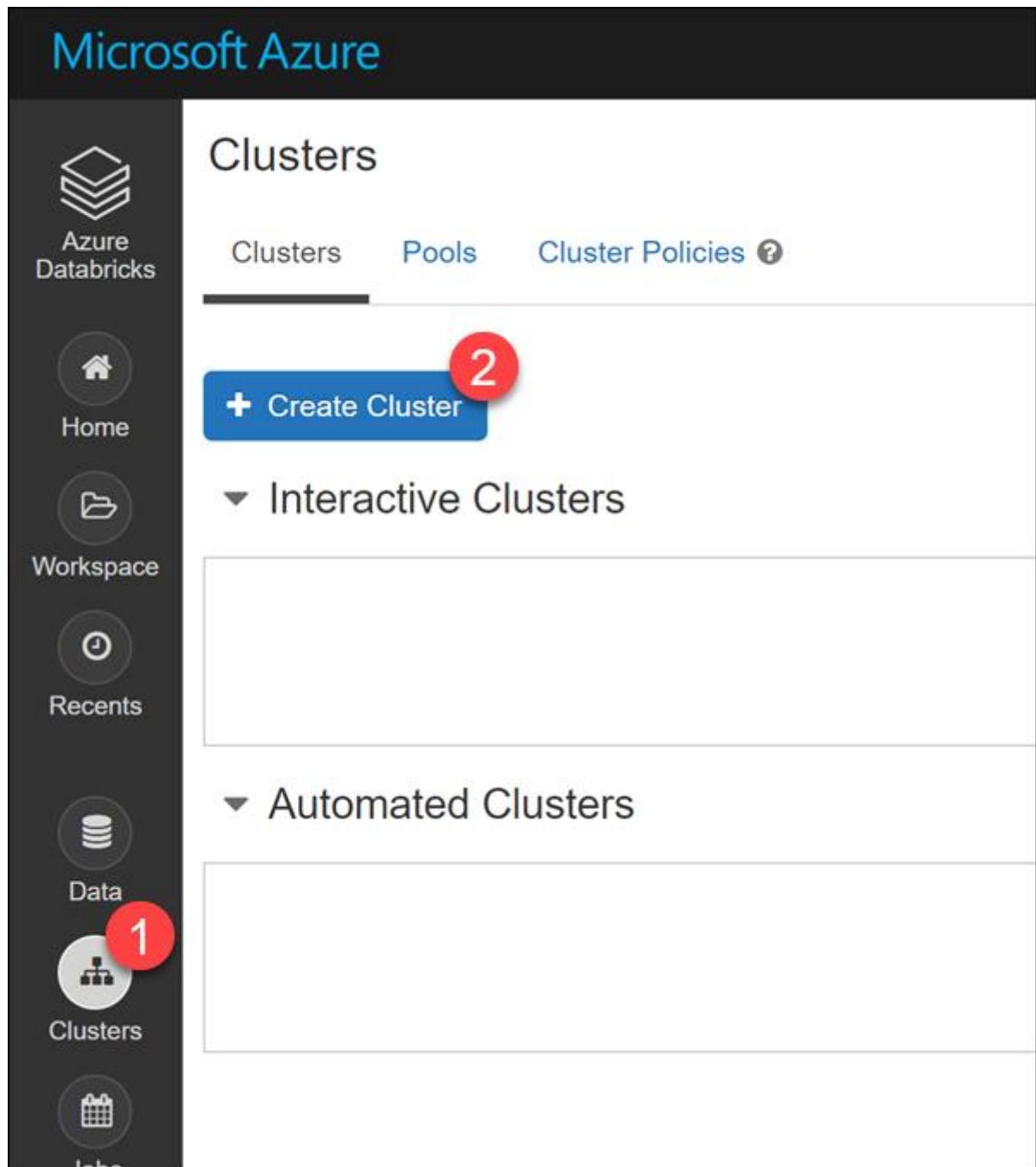
Sign In to Databricks

Sign in using Azure Active Directory Single Sign On.

 Signing you in

Contact your site administrator to request access.

4. Select **Clusters** (1) from the menu, then select + **Create Cluster** (2).



5. On the New Cluster form, provide the following:
 - **Cluster Name:** lab
 - **Cluster Mode:** Standard
 - **Pool:** Select None
 - **Databricks Runtime Version:** Runtime: 9.1 LTS ML (Scala 2.12, Spark 3.1.2) (**Note:** the runtime version CANNOT be > 6.6, due to compatibility issues with the supplied notebooks.)

- **Enable Autoscaling:** **Uncheck** this option.
- **Terminate after:** **Check** the box and enter 120
- **Worker Type:** **Standard_F4s**
- **Driver Type:** **Same as worker**
- **Workers:** 1
- **Spark Config:** Expand Advanced Options and edit the Spark Config by entering the connection information for your Azure Storage account that you copied above in Task 1. This will allow your cluster to access the lab files. Enter the following:

`spark.hadoop.fs.azure.account.key.<STORAGE_ACCOUNT_NAME>.blob.core.windows.net <ACCESS_KEY>`, where <STORAGE_ACCOUNT_NAME> is your Azure Storage account name, and <ACCESS_KEY> is your storage access key.

6. **Example:** `spark.hadoop.fs.azure.account.key.bigdatalabstore.blob.core.windows.net HD+91Y77b+TezEu1lh9QXXU2Va6Cjg9bu0RRpb/KtBj8lWQa6jwyA00GTDmSNVFr8iS1kytIFON EHLd167Fgxg==`

Create Cluster

New Cluster

[Cancel](#)[Create Cluster](#)DBU / hour: 1 [?](#)1 Workers: 8 GB Memory, 4 Cores
1 Driver: 8 GB Memory, 4 Cores

Cluster name

lab

Cluster mode [?](#)

Standard

Databricks runtime version [?](#)[Learn more](#)

Runtime: 9.1 LTS ML (Scala 2.12, Spark 3.1.2)

i 50% promotional discount applied to Photon during preview [?](#) X

Autopilot options

 Enable autoscaling [?](#) Terminate after minutes of inactivity [?](#)Worker type [?](#)

Standard_F4

8 GB Memory, 4 Cores

Workers

01

[?](#) Spot instances [?](#)**New** Configure separate pools for workers and drivers for flexibility. [Learn more](#)

Driver type

Same as worker

8 GB Memory, 4 Cores

DBU / hour: 1 [?](#)

Standard_F4

▼ Advanced options

Azure Data Lake Storage credential passthrough [?](#) Available on Azure Databricks premium [Learn more](#) Enable credential passthrough for user-level data access[Spark](#) [Tags](#) [Logging](#) [Init Scripts](#)

Spark config [?](#)

```
spark.hadoop.fs.azure.account.key.bigdatalabstore.blob.core.windows.net
HD+91Y77b+TezEu1lh9QXXU2Va6Cjg9bu0RRpb/KtBj8IWQa6jwyA0OGTDmSNVFr8iSlkytIFON
EHLdl67Fgxg==
```

7.

8. Select **Create Cluster**.

Exercise 2: Load Sample Data and Databricks Notebooks

In this exercise, you will implement a classification experiment. You will load the training data from your local machine into a dataset. Then, you will explore the data to identify the primary components you should use for prediction, and use two different algorithms for predicting the classification. You will then evaluate the performance of both algorithms and choose the algorithm that performs best. The model selected will be exposed as a web service that is integrated with the optional sample web app at the end.

Task 1: Upload the Sample Datasets

1. Before you begin working with machine learning services, there are three datasets you need to load.
2. Select the file provided to you.
3. Extract the ZIP and verify you have the following files:
 - FlightDelaysWithAirportCodes.csv
 - FlightWeatherWithAirportCodes.csv
 - AirportCodeLocationLookupClean.csv
4. Open your Azure Databricks workspace. Before continuing to the next step, verify that your new cluster is running. Do this by navigating to **Clusters** on the left-hand menu and ensuring that the state of your cluster is **Running**.

Clusters

Clusters Pools Cluster Policies ?

+ Create Cluster

▼ Interactive Clusters

Name	State
lab	Running

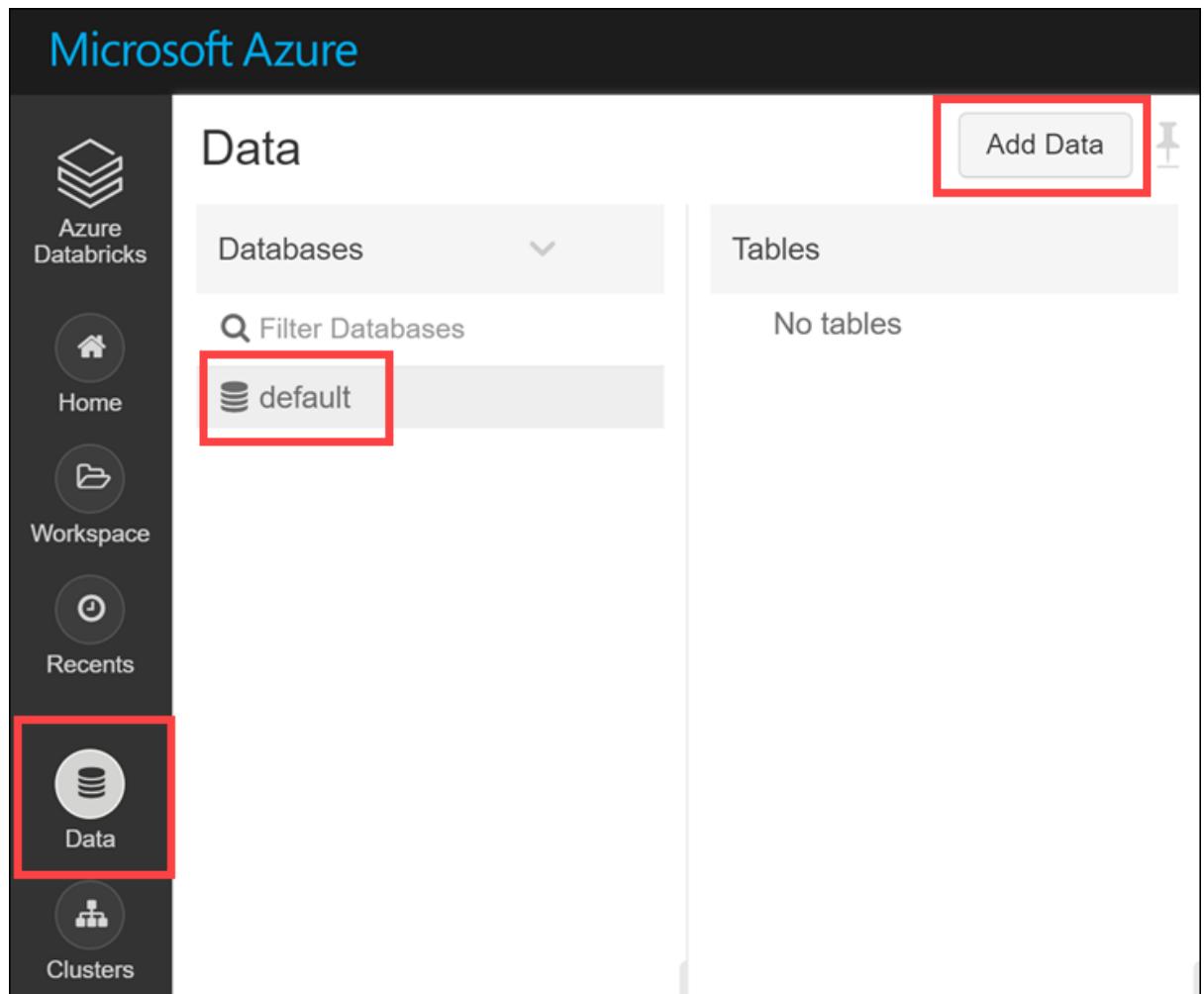
▼ Automated Clusters

--

Azure Databricks

Home Workspace Recents Data Clusters

5. Select **Data** from the menu. Next, select **default** under Databases (if this does not appear, start your cluster). Finally, select **Add Data** above the Tables header.



6. Select **Upload File** under Create New Table, and then select either select or drag-and-drop the FlightDelaysWithAirportCodes.csv file into the file area. Select **Create Table with UI**.

Create New Table

Data source [?](#)

[Upload File](#) DBFS Other Data Sources

Upload to DBFS [?](#)

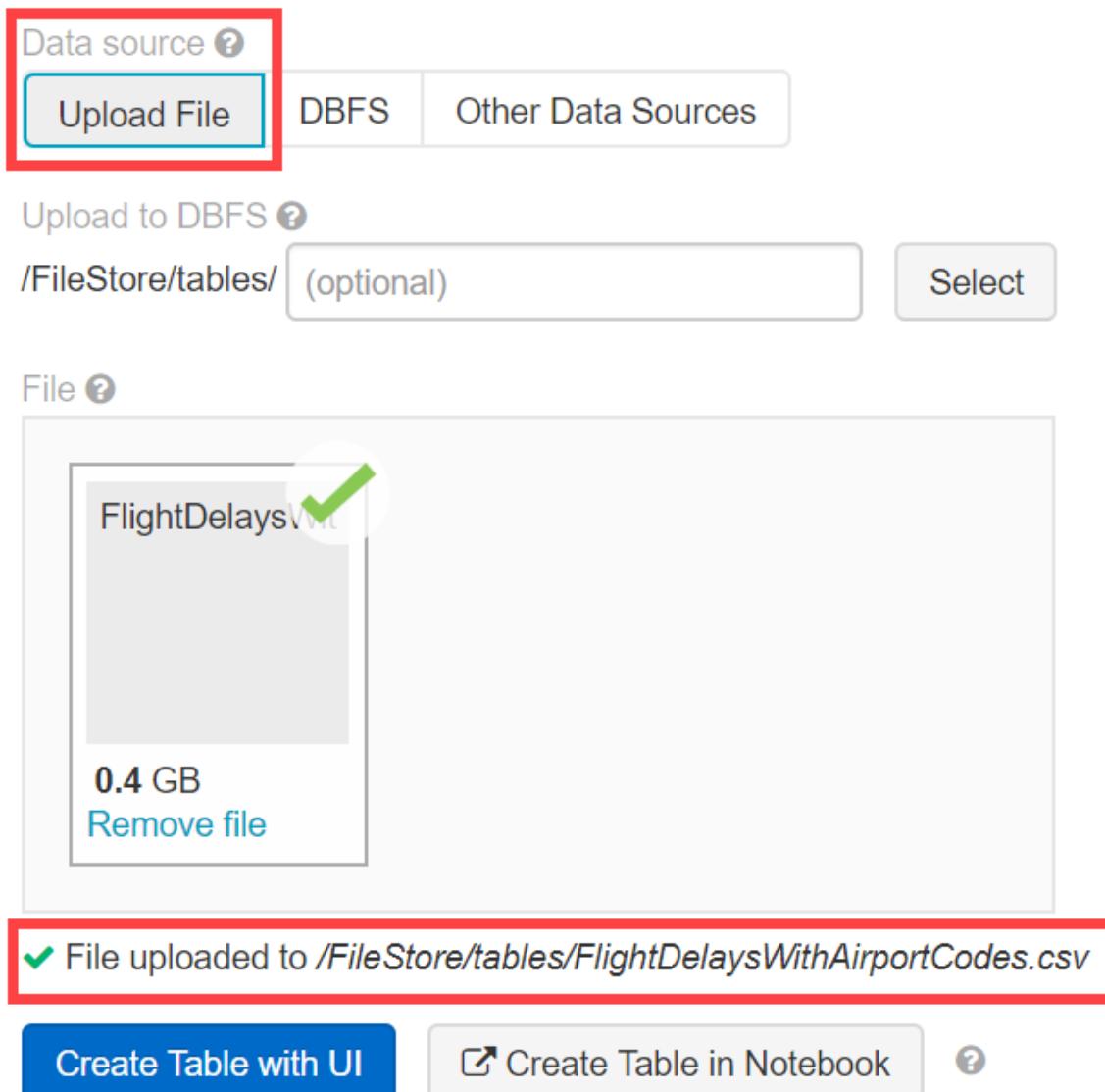
/FileStore/tables/ (optional) [Select](#)

File [?](#)

FlightDelaysWithAirportCodes.csv
0.4 GB [Remove file](#)

✓ File uploaded to /FileStore/tables/FlightDelaysWithAirportCodes.csv

[Create Table with UI](#) [Create Table in Notebook](#) [?](#)



7. Select your cluster to preview the table, then select **Preview Table**.
8. Change the Table Name to `flight_delays_with_airport_codes` and select the checkmark for **First row is header**. Select **Create Table**.

Specify Table Attributes

Specify the Table Name, Database and Schema to add this to the data UI for other users to access

Table Name ?	Table Preview		
flight_delays_with_airport_c	Year	Month	DayofMonth
	STRING	STRING	STRING
	2013	4	19
	2013	4	19
	2013	4	19
	2013	4	19
	2013	4	19
	2013	4	19
	2013	4	19

Create in Database [?](#)

default

File Type [?](#)

CSV

Column Delimiter [?](#)

,

First row is header [?](#)

Infer schema [?](#)

Multi-line [?](#)

Create Table

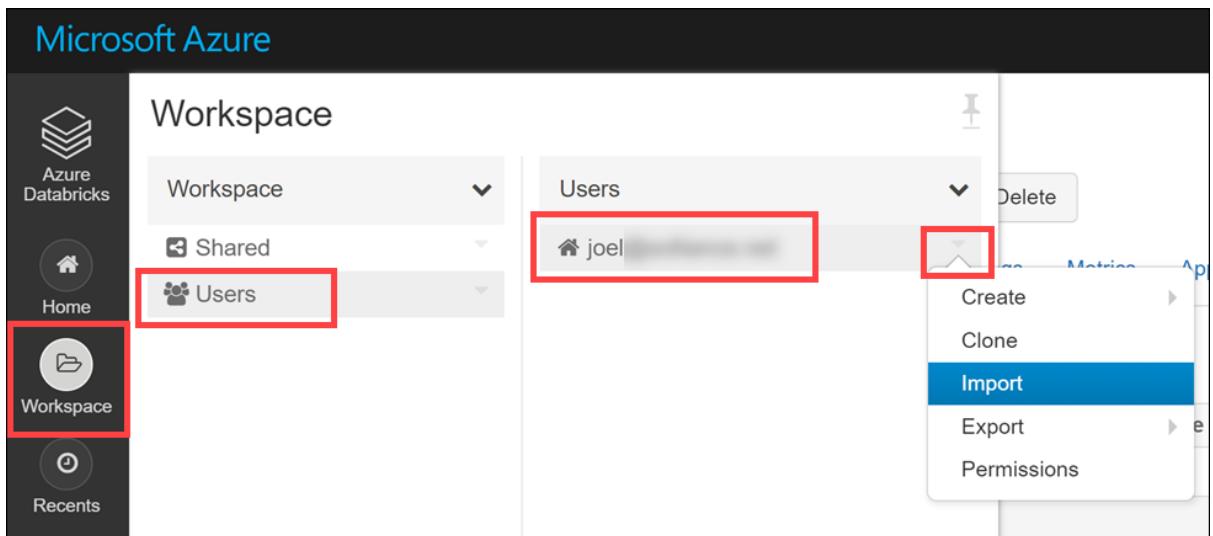
Create Table in

9. Repeat steps 5 through 8 for the FlightWeatherWithAirportCode.csv and AirportCodeLocationsClean.csv files, setting the name for each dataset in a similar fashion:
 - o flightweatherwithairportcode_csv renamed to **flight_weather_with_airport_code**
 - o airportcodelocationlookupclean_csv renamed to **airport_code_location_lookup_clean**

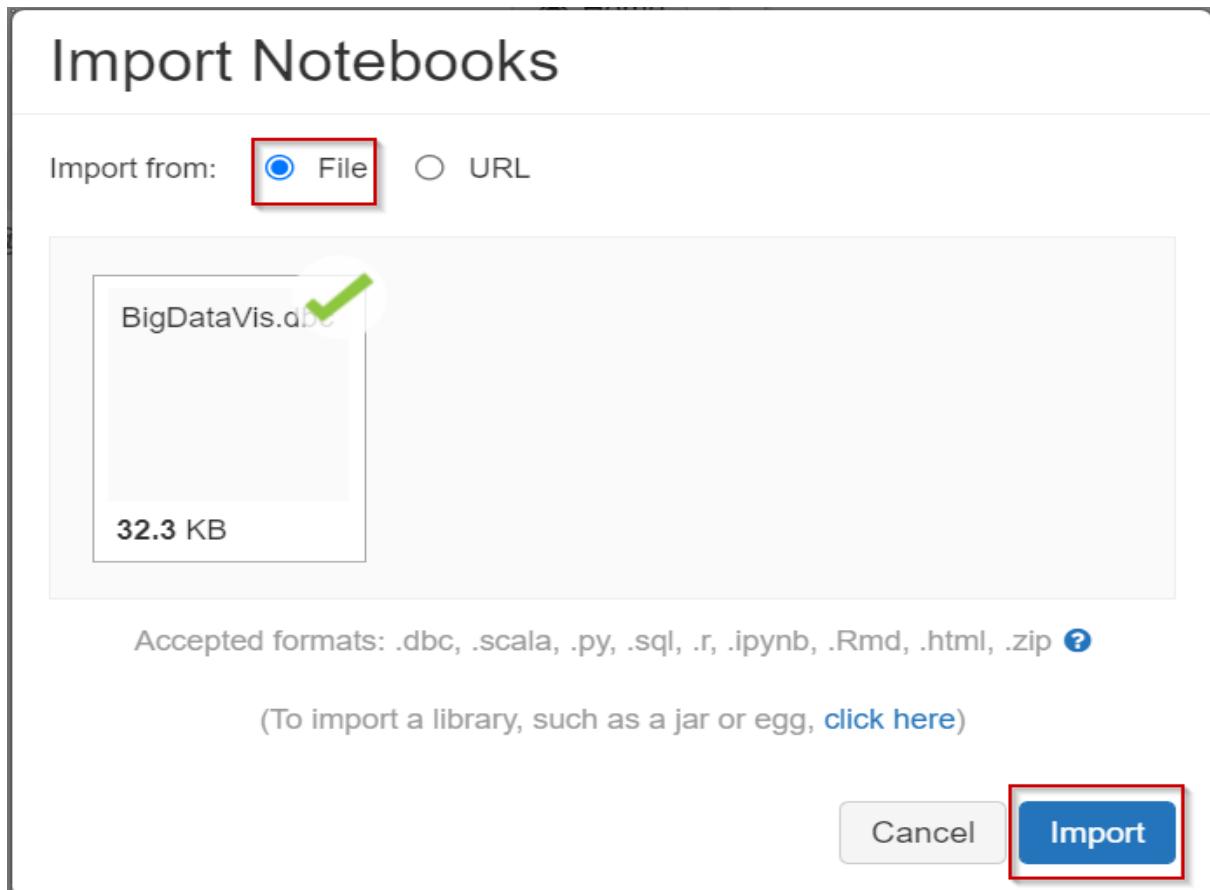
Databases	Tables
Filter Databases	Filter Tables
default	 airport_code_location_lookup_clean
	 flight_delays_with_airport_codes
	 flight_weather_with_airport_code

Task 2: Open Azure Databricks and complete lab notebooks

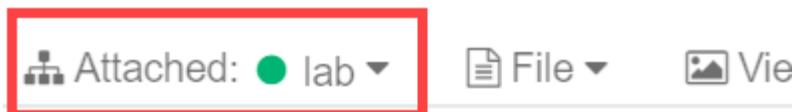
1. Within Azure Databricks, select **Workspace** on the menu, then **Users**, then select the down arrow next to your user name. Select **Import**.



2. Within the Import Notebooks dialog, select Import from file. Select the dbc file provided to you.



3. After importing, expand the new **BigDataVis** folder.
4. Before you begin, make sure you attach your cluster to the notebooks, using the dropdown. You will need to do this for each notebook you open. There are 5 notebooks included in the BigDataVisdbc



5. Run each cell of the notebooks located in the **Exercise 2** folder (01, 02 and 03) individually by selecting within the cell, then entering **Ctrl+Enter** on your keyboard. Pay close attention to the instructions within the notebook so you understand each step of the data preparation process.

A screenshot of the Azure Databricks workspace interface. The left sidebar shows "Azure Databricks", "Home", and "Workspace". The main content area has two dropdown menus: "BigDataVis" and "Exercise 2". The "Exercise 2" menu is expanded, showing three items: "01 Data Preparation", "02 Train and Evaluate Mo...", and "03 Deploy as Web Service". Both the "Exercise 2" menu and its items are highlighted with red boxes.

6. Do NOT run the `clean` up part of Notebook 3 (i.e. this command: `service.delete()`). You will need the URL of your Machine Learning Model exposed later in **Exercise 8: Deploy intelligent web app (Optional Lab)**.
Note: you could get this URL by updating your Notebook and adding this line `print(service.scoring_uri)`, or by going to your Azure Machine Learning service workspace via the Azure portal and then to the "Deployments" blade.
7. Do NOT run any notebooks within the Exercise 5 or 6 folders. They will be discussed later in the lab.

Exercise 3: Setup Azure Data Factory

In this exercise, you will create a baseline environment for Azure Data Factory development for further operationalization of data movement and processing. You will create a Data Factory service, and then install the Data Management Gateway which is the agent that facilitates data movement from on-premises to Microsoft Azure.

Task 1: Download and stage data to be processed

1. Open a web browser.

2. Select the AdventureWorks zip file provided to you.
3. Extract it to a new folder called **C:\Data**.

Task 2: Install and configure Azure Data Factory Integration Runtime on your machine

1. To download the latest version of Azure Data Factory Integration Runtime, go to <https://www.microsoft.com/en-us/download/details.aspx?id=39717>.

The screenshot shows a download interface. On the left, there is a list of files with checkboxes. The first file, 'IntegrationRuntime_5.16.8096.1.msi', has a checked checkbox and is highlighted with a red border. On the right, there is a summary box titled 'Download Summary' which lists the selected file. At the bottom right of the summary box is a blue 'Next' button with a red border.

File Name	Size
<input checked="" type="checkbox"/> IntegrationRuntime_5.16.8096.1.msi	1.0 GB
<input type="checkbox"/> IntegrationRuntime_5.14.8067.1.msi	1.0 GB
<input type="checkbox"/> IntegrationRuntime_5.15.8092.1.msi	999.0 MB
<input type="checkbox"/> Release Notes.doc	211 KB

Choose the download you want

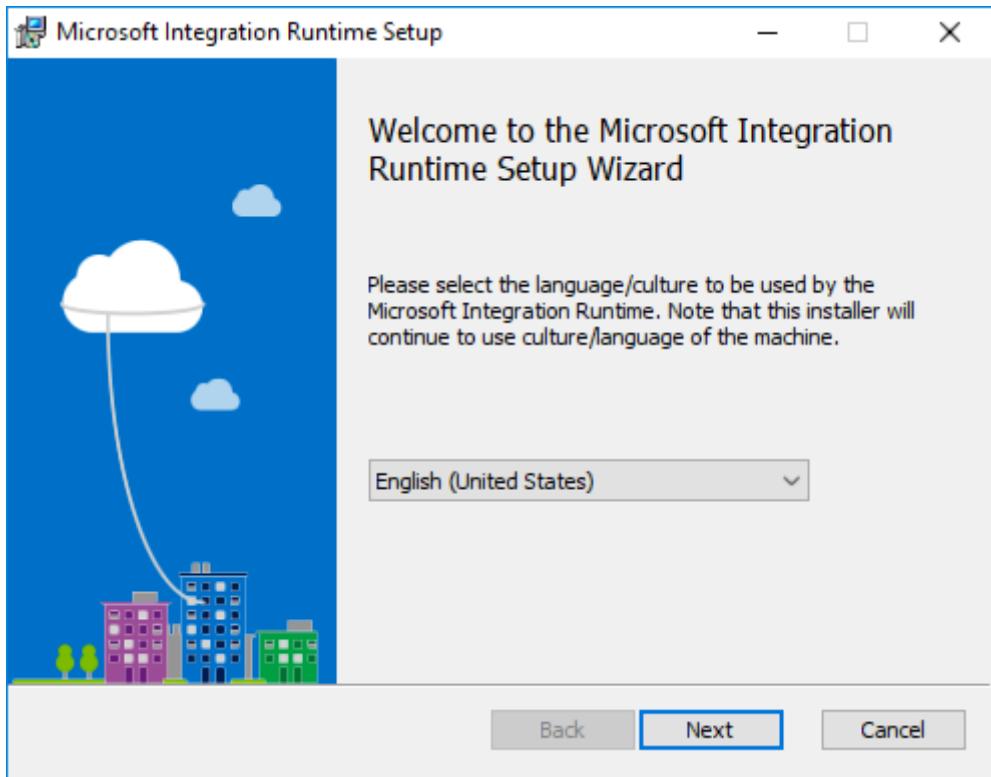
Download Summary:
KBMBGB

1. IntegrationRuntime_5.16.8096.1.msi

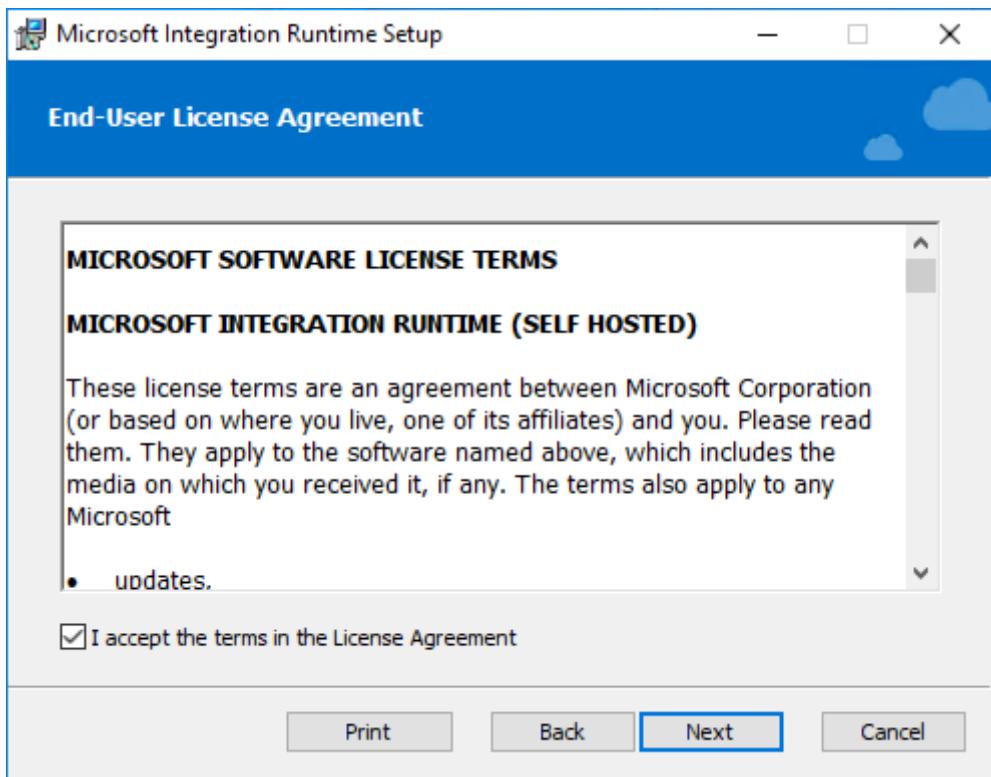
Total Size: 1.0 GB

Next

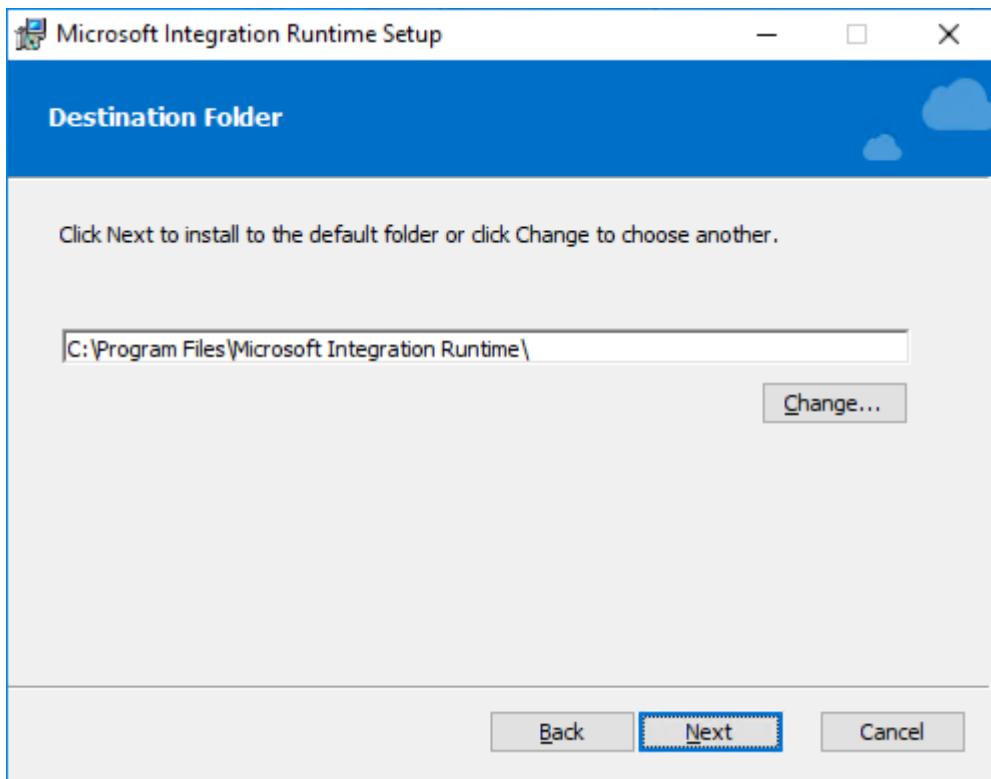
2. Select Download, then choose the download you want from the next screen.
3. Run the installer, once downloaded.
4. When you see the following screen, select Next.



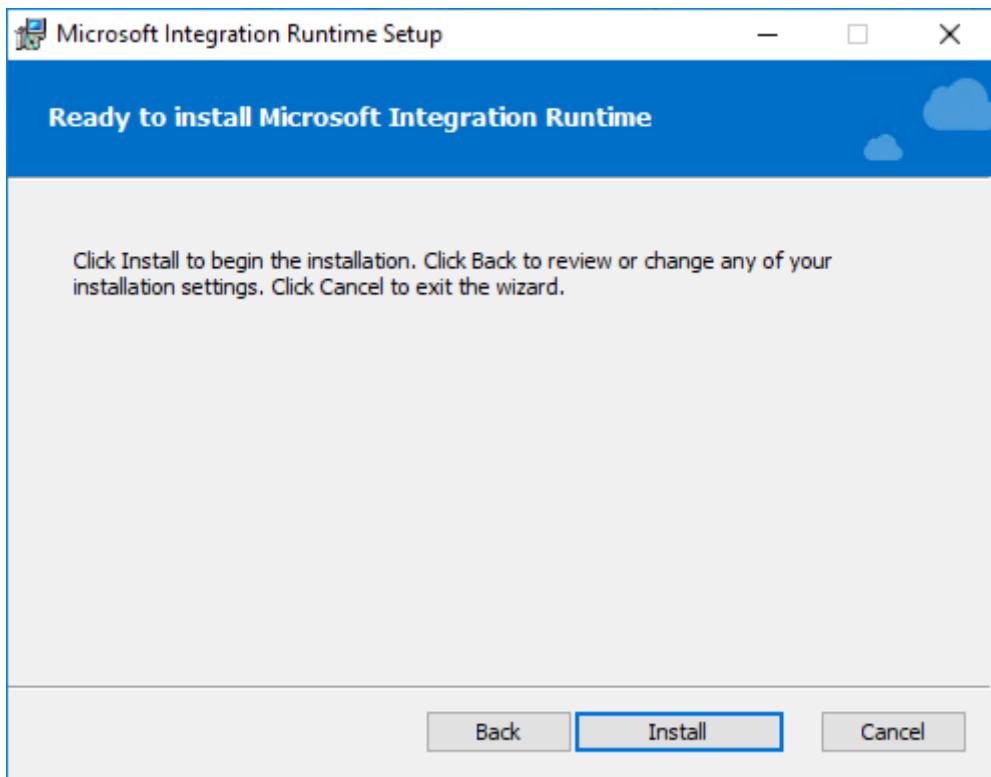
5. Check the box to accept the terms and select Next.



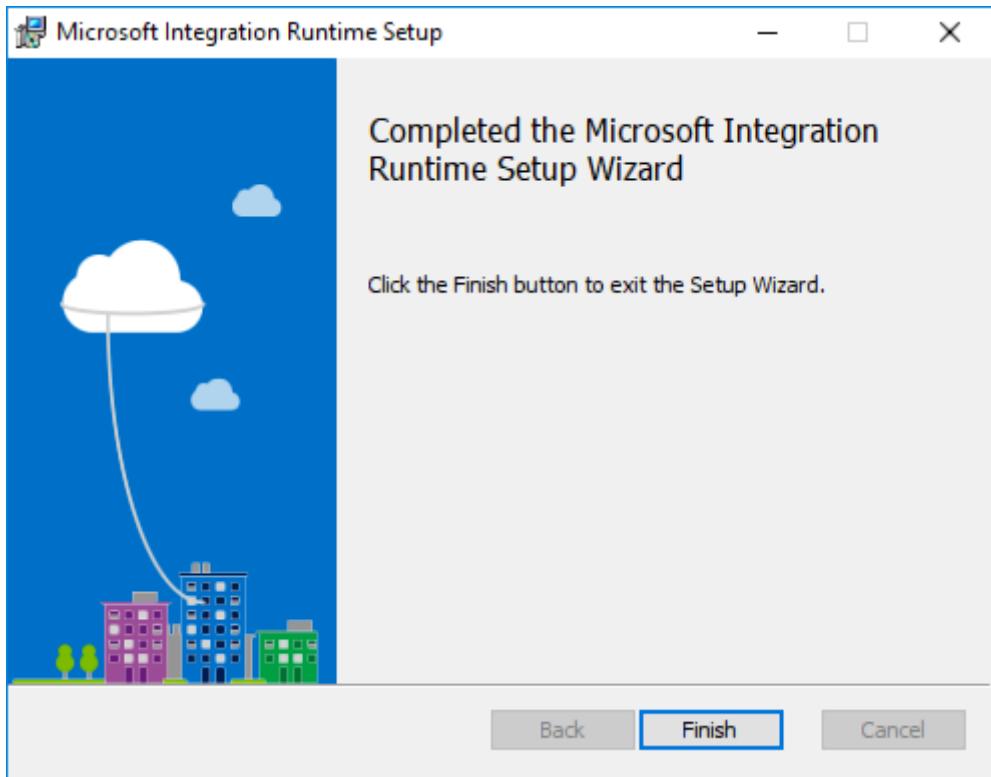
6. Accept the default Destination Folder, and select Next.



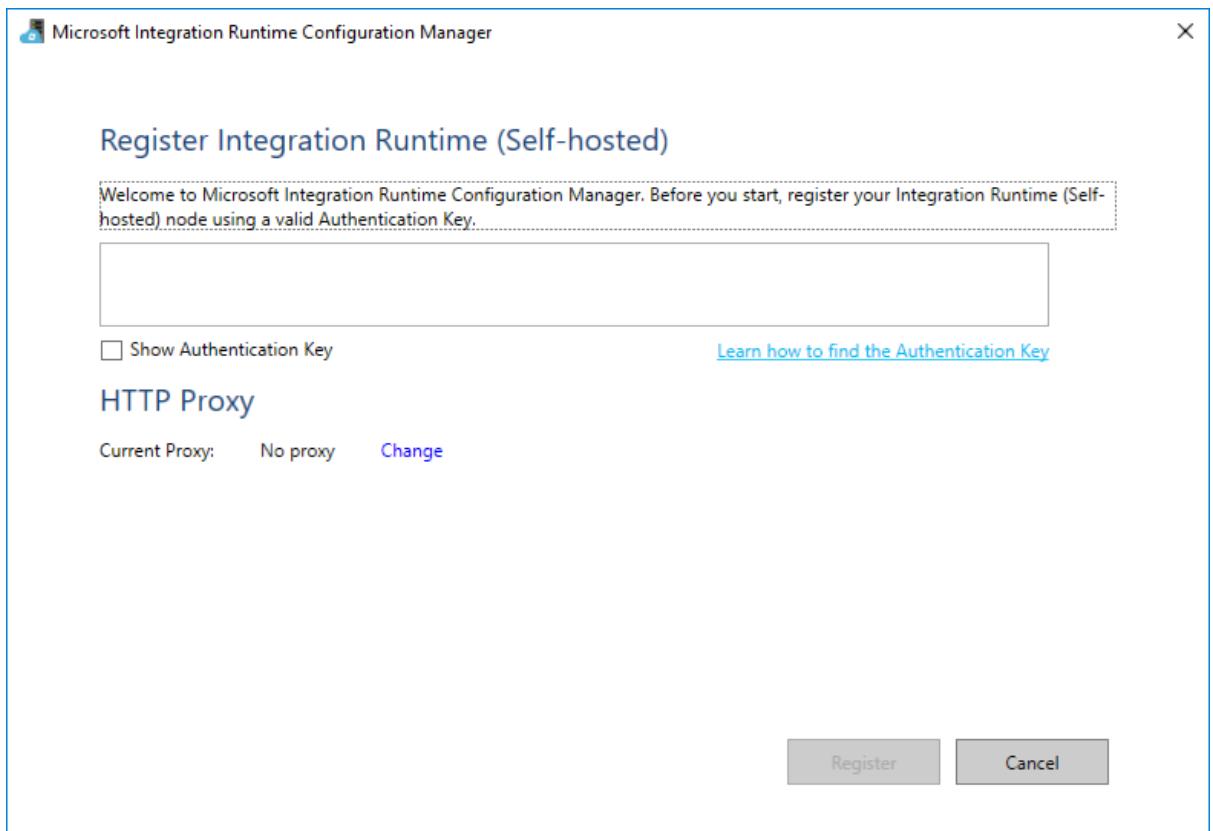
7. Choose Install to complete the installation.



8. Select Finish once the installation has completed.



9. After selecting Finish, the following screen will appear. Keep it open for now. You will come back to this screen once the Data Factory in Azure has been provisioned, and obtain the gateway key so we can connect Data Factory to this "on-premises" server.

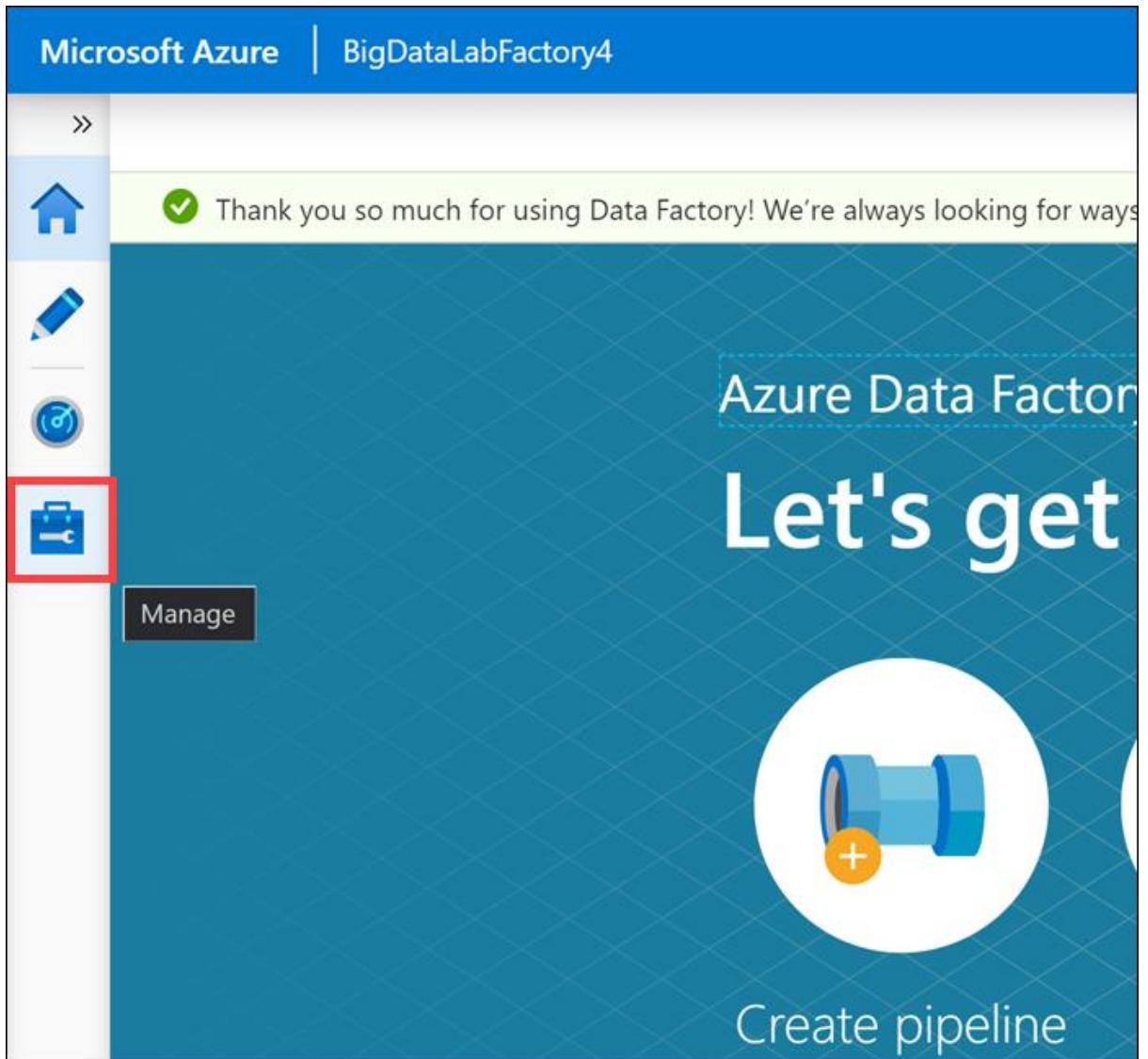


Task 3: Configure Azure Data Factory

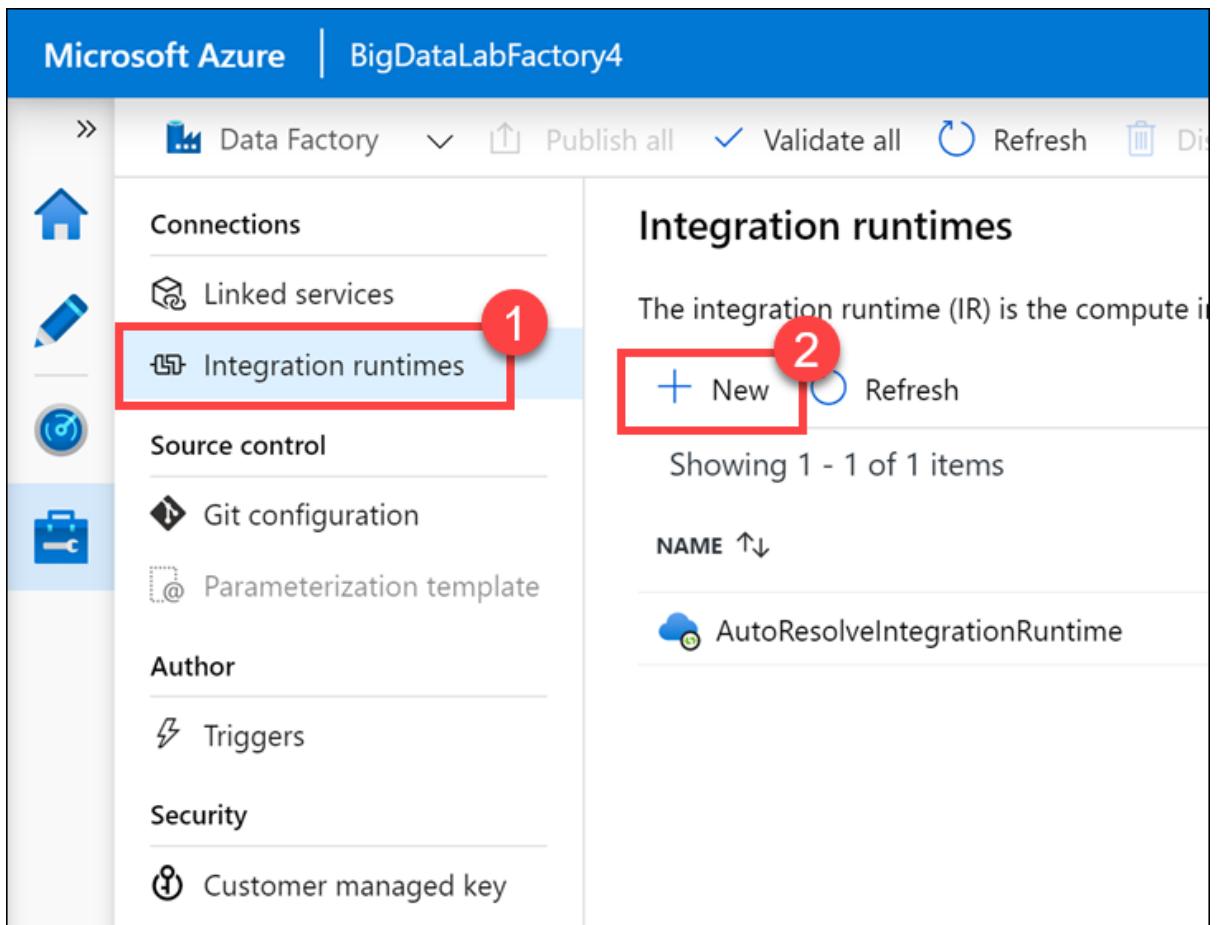
1. Launch a new browser window, and navigate to the Azure portal (<https://portal.azure.com>). Once prompted, log in with your Microsoft Azure credentials. If prompted, choose whether your account is an organization account or a Microsoft account. This will be based on which account was used to provision your Azure subscription that is being used for this lab.
2. From the side menu in the Azure portal, choose **Resource groups**, then enter your resource group name into the filter box, and select it from the list.
3. Next, select your Azure Data Factory service from the list.
4. On the Data Factory Overview screen, select **Author & Monitor**.

The screenshot shows the Azure Data Factory Overview page. The left sidebar has a 'Overview' tab highlighted with a red box. The main content area displays basic information about the data factory, including its status as 'Succeeded', location as 'East US', and subscription details. Below this, there are sections for 'Documentation' and 'Monitoring'. The 'Author & Monitor' button, located in the 'Monitoring' section, is also highlighted with a red box.

5. A new page will open in another tab or new window. Within the Azure Data Factory site, select **Manage** on the menu.



6. Now, select **Integration runtimes** in the menu beneath Connections (1), then select **+ New** (2).



7. In the Integration Runtime Setup blade that appears, select **Azure, Self-Hosted**, then select **Continue**.

Integration runtime setup

Integration Runtime is the native compute used to execute or dispatch activities. Choose what integration runtime to create based on required capabilities. [Learn more](#)



Azure, Self-Hosted

Perform data flows, data movement and dispatch activities to external compute.



Azure-SSIS

Lift-and-shift existing SSIS packages to execute in Azure.

Continue

Cancel

8. Select **Self-Hosted** then select **Continue**.

Integration runtime setup

Network environment:

Choose the network environment of the data source / destination or external compute to which the integration runtime will connect to for data flows, data movement or dispatch activities:



Azure

Use this for running data flows, data movement, external and pipeline activities in a fully managed, serverless compute in Azure.



Self-Hosted

Use this for running activities in an on-premise / private network

[View more ▾](#)

External Resources:

You can use an existing self-hosted integration runtime that exists in another resource. This way you can reuse your existing infrastructure where self-hosted integration runtime is setup.



Linked Self-Hosted

[Learn more ▾](#)

[Continue](#)

[Back](#)

[Cancel](#)

9. Enter a **Name**, such as bigdatagateway-[initials], and select **Create**.

Integration runtime setup

Private network support is realized by installing integration runtime to machines in the same on-premises network/VNET as the resource the integration runtime is connecting to. Follow below steps to register and install integration runtime on your self-hosted machines.

Name *

bigdatagateway-jdh



Description

Enter description here...

Type

Self-Hosted

10. Under Option 2: Manual setup, copy the Key1 authentication key value by selecting the Copy button, then select **Close**.

Integration runtime setup

[Settings](#) [Nodes](#) [Auto update](#) [Sharing](#)

Install integration runtime on Windows machine or add further nodes using the Authentication Key.

Name [\(i\)](#)
bigdatagateway-jdh

Option 1: Express setup
[Click here to launch the express setup for this computer](#)

Option 2: Manual setup

Step 1: [Download and install integration runtime](#)

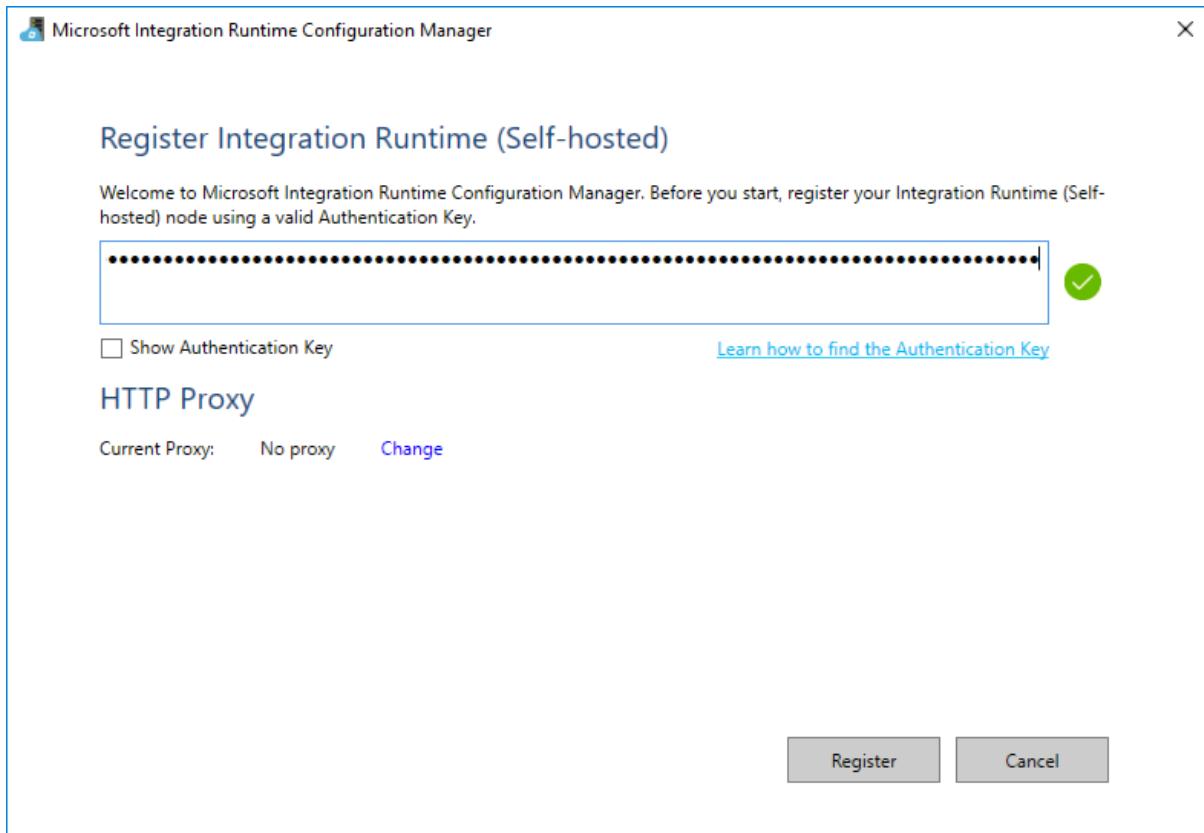
Step 2: Use this key to register your integration runtime

NAME	AUTHENTICATION KEY	Actions
Key1	IR@1ae98a6b-b644-42ee-b27c-c3a8e1e7268c@BigDataLabFactory4@eu2	 
Key2	IR@1ae98a6b-b644-42ee-b27c-c3a8e1e7268c@BigDataLabFactory4@eu2	 

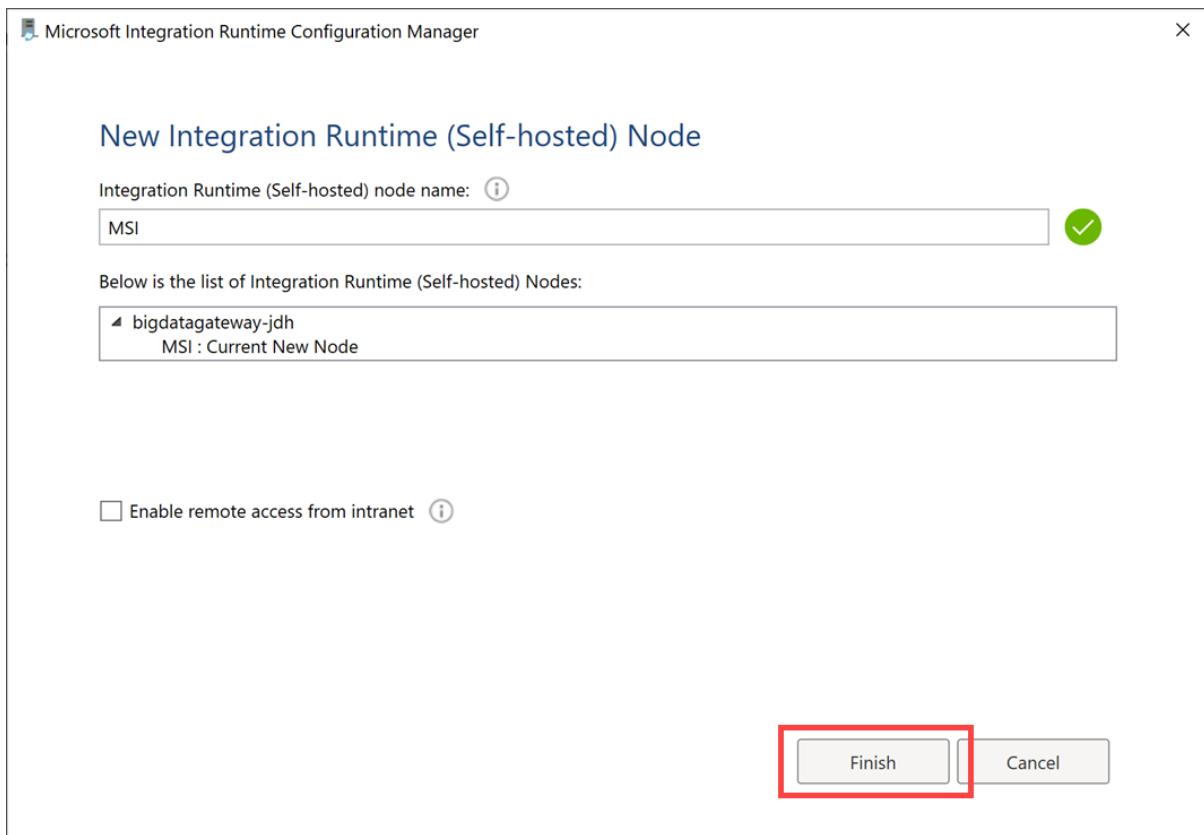
[Close](#)

11. *Don't close the current screen or browser session.*

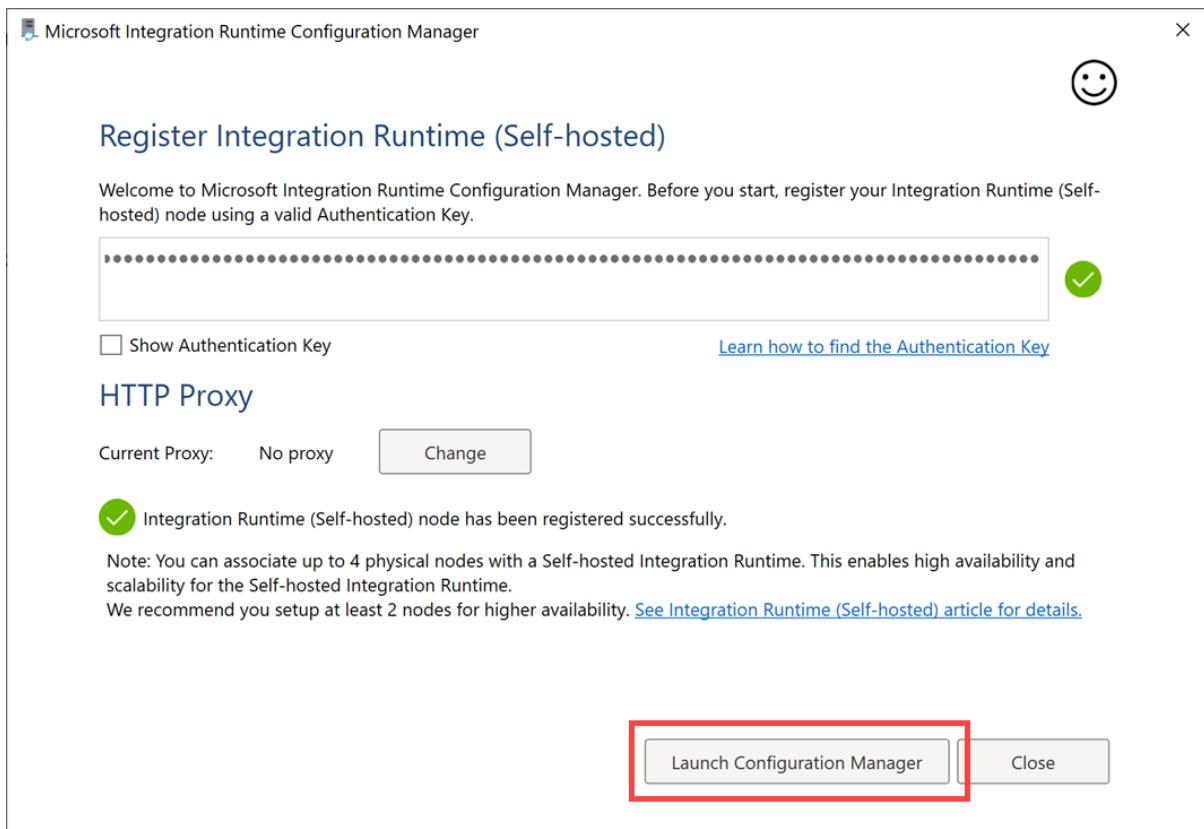
12. Paste the **Key1** value into the box in the middle of the Microsoft Integration Runtime Configuration Manager screen.



13. Select **Register**.
14. It can take up to a minute or two to register. If it takes more than a couple of minutes, and the screen does not respond or returns an error message, close the screen by selecting the **Cancel** button.
15. The next screen will be New Integration Runtime (Self-hosted) Node. Select Finish.



16. You will then get a screen with a confirmation message. Select the **Launch Configuration Manager** button to view the connection details.



The screenshot shows the Microsoft Integration Runtime Configuration Manager interface. At the top, there's a header bar with the title 'Microsoft Integration Runtime Configuration Manager' and navigation tabs: Home, Settings, Diagnostics, Update, and Help. The 'Home' tab is selected.

The main content area displays a green checkmark icon followed by the text 'Self-hosted node is connected to the cloud service'. Below this, it shows the 'Integration Runtime' as 'bigdatagateway-jdh' and the 'Node' as 'MSI'. There is a button labeled 'Stop Service'.

Below this section, there's a 'Data Source Credential' section with an 'i' icon. It shows the 'Credential store' as 'On-premises', 'Credential status' as 'In sync', and 'Last backup time' as 'N/A'. It includes two buttons: 'Generate Backup' and 'Import Backup'.

At the bottom of the screen, there's a status bar with a green checkmark icon and the text 'Connected to the cloud service (Data Factory V2)'. On the right side of the status bar is a refresh icon.

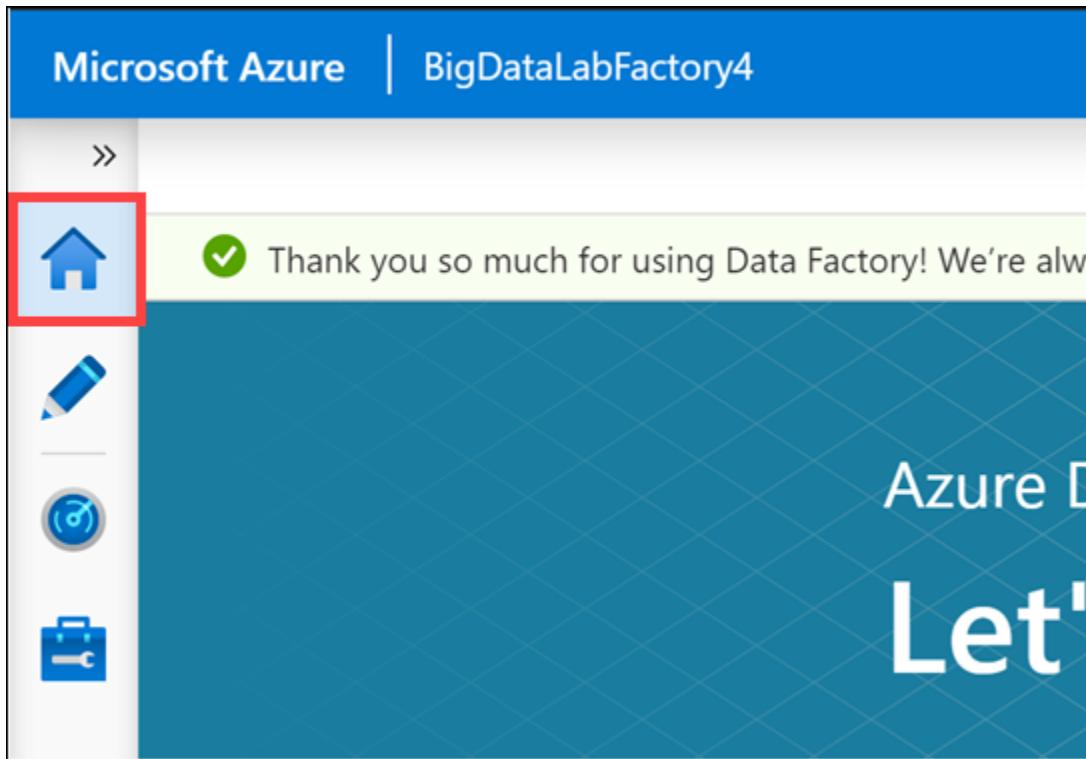
17. You can now return to the Azure Data Factory page, and view the Integration Runtime you just configured. You may need to select **Refresh** to view the Running status for the IR.

The screenshot shows the Azure Data Factory 'Integration runtimes' page. The left sidebar has navigation items: Data Factory, Publish all, Validate all, Refresh, Discard all, Data flow debug, ARM template, and a dropdown. The 'Connections' item is expanded, showing 'Linked services', 'Integration runtimes' (which is selected), 'Source control', 'Git configuration', and 'Parameterization template'. The 'Author' and 'Security' sections are also listed.

The main content area is titled 'Integration runtimes'. It says 'The integration runtime (IR) is the compute infrastructure to provide the following data integration capabilities across different network environment.' It includes a 'New' button and a 'Refresh' button. Below this, it says 'Showing 1 - 2 of 2 items'.

NAME ↑↓	TYPE ↑↓	SUB-TYPE ↑↓	STATUS ↑↓
AutoResolveIntegrationRuntime	Azure	Public	✓ Running
bigdatagateway-jdh	Self-Hosted	---	✓ Running

18. Select the Azure Data Factory Overview button on the menu. Leave this open for the next exercise.



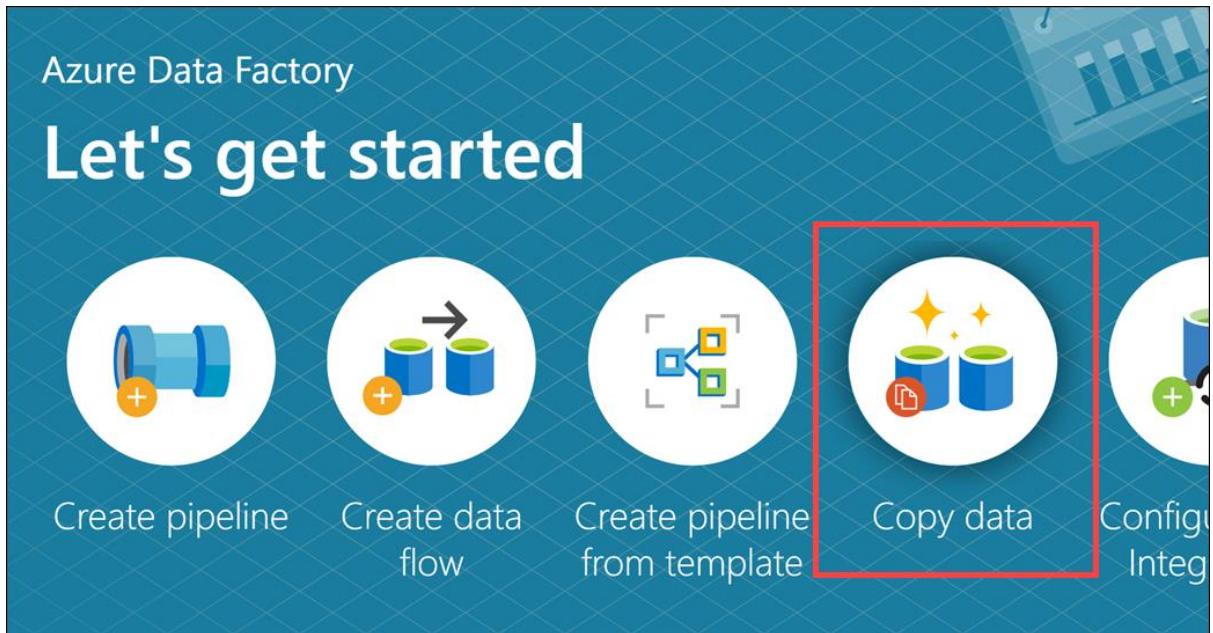
Exercise 4: Develop a data factory pipeline for data movement

Duration: 20 minutes

In this exercise, you will create an Azure Data Factory pipeline to copy data (.CSV files) from an on-premises server (your machine) to Azure Blob Storage. The goal of the exercise is to demonstrate data movement from an on-premises location to Azure Storage (via the Integration Runtime).

Task 1: Create copy pipeline using the Copy Data Wizard

1. Within the Azure Data Factory overview page, select **Copy Data**.



2. In the Copy Data properties, enter the following:

- **Task name:** CopyOnPrem2AzurePipeline
- **Task description:** (Optional) This pipeline copies time-sliced CSV files from on-premises C:\\Data to Azure Blob Storage as a continuous job.
- **Task cadence or Task schedule:** Select **Run regularly on schedule**
- **Trigger type:** **Select Schedule**
- **Start date time (UTC):** Enter **03/01/2018 12:00 AM**
- **Recurrence:** Every 1, and select **Month(s)**
- Under the **Advanced recurrence options**, make sure you have a value of 0 in the textboxes for **Hours (UTC)** and **Minutes (UTC)**, otherwise it will fail later during Publishing.
- **End:** **No End**

Properties

Enter name and description for the copy data task.

Task name *

CopyOnPrem2AzurePipeline

Task description

This pipeline copies time-sliced CSV files from on-premises C:\\Data to Azure Blob Storage as a continuous job.



Task cadence or task schedule

Run once now Run regularly on schedule

Trigger type *

Schedule Tumbling window

Start Date (UTC) *

03/01/2018 12:00 AM



Recurrence *

Every

Month(s)



Advanced recurrence options

Month days Week days

Select day(s) of the month to execute

1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	Last			

Execute at these times



Hours (UTC)

0

Minutes (UTC)

0

Schedule execution times (UTC)

00:00

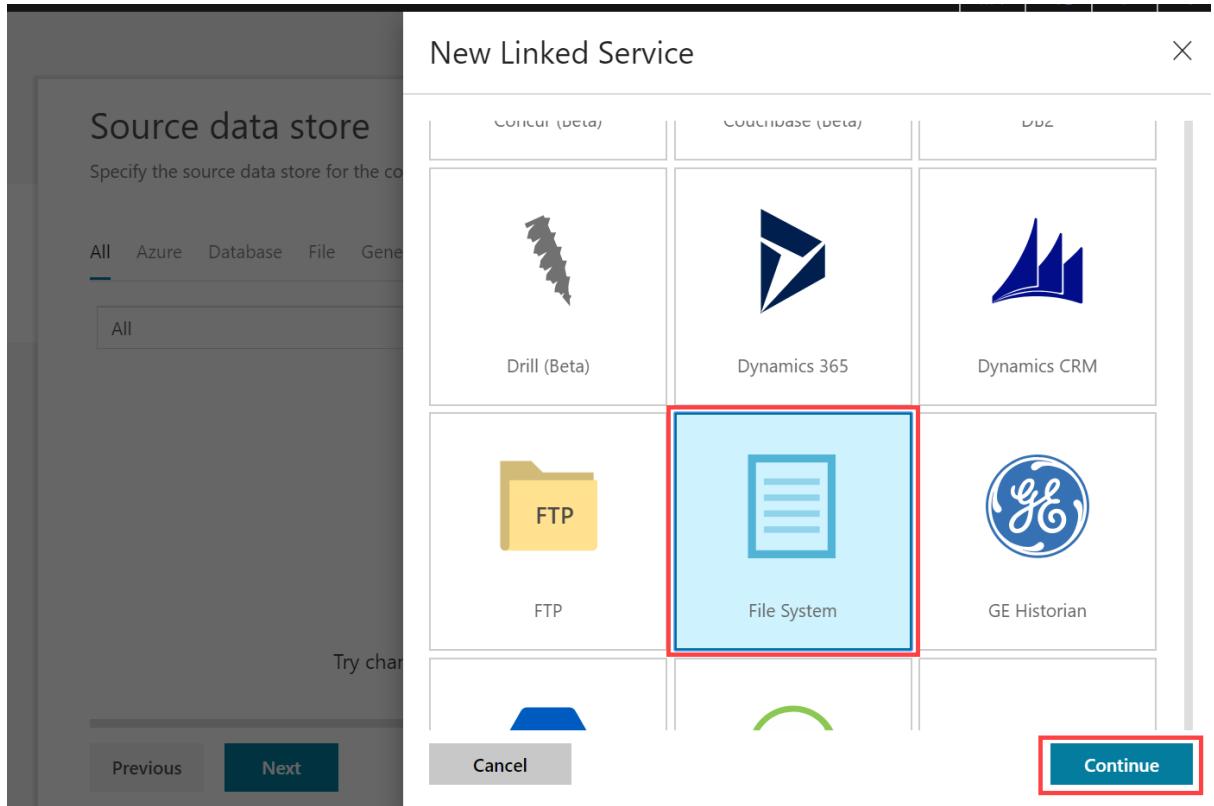
End *

No End On Date

Previous

Next

3. Select **Next**.
4. On the Source data store screen, select **+ Create new connection**.
5. Scroll through the options and select **File System**, then select **Continue**.



6. In the New Linked Service form, enter the following:
 - **Name:** OnPremServer
 - **Connect via integration runtime:** Select the Integration runtime created previously in this exercise.
 - **Host:** C:\Data
 - **User name:** Use your machine's login username.
 - **Password:** Use your machine's login password.
7. Select **Test connection** to verify you correctly entered the values. Finally, select **Create**.

New linked service (File System)

Name *
OnPremServer

Description

Connect via integration runtime *
bigdatagateway-jdh

Host *
C:\Data

User name *

Password Azure Key Vault
Password *
.....

Annotations
+ New

Advanced ⓘ

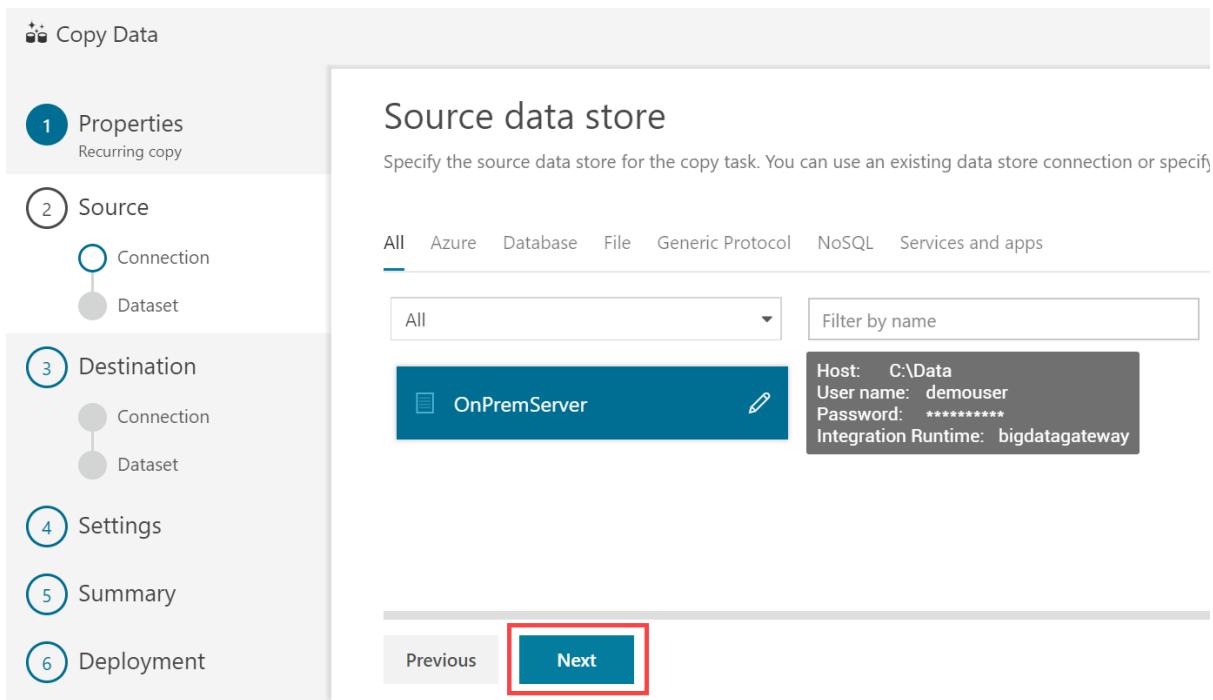
Create Back

✓ Connection successful

🔗 Test connection

Cancel

8. On the Source data store page, select **Next**.



9. On the **Choose the input file or folder** screen, select **Browse**, then select the **FlightsAndWeather** folder. Next, select **Load all files** under file loading behavior, check **Recursively**, then select **Next**.

Choose the input file or folder

Select a source file or folder to be copied to the destination data store.

The screenshot shows the 'Choose the input file or folder' screen. It has a 'File or folder' input field containing 'FlightsAndWeather/' and a 'Browse' button to its right, which is highlighted with a red box. Below this, there's a 'File loading behavior' dropdown set to 'Load all files'. Underneath, there are two options: 'Binary copy' with an unchecked checkbox and 'Recursively' with a checked checkbox. Further down is a 'Max concurrent connections' input field. At the bottom, there are 'Previous' and 'Next' buttons, with 'Next' being the one highlighted with a red box.

10. On the File format settings page, select the following options:

- o **File format: Text format**
- o **Column delimiter: Comma (,)**

- **Row delimiter:** Auto detect (\r, \n, or \r\n)
- **Skip line count:** 0
- **First row as header:** Checked

File format settings

File format
Text format i Detect text format

Column delimiter
Comma (,) ▼ Edit

Row delimiter
Auto detect (\r,\n, or \r\n) ▼ Edit

Skip line count
0 i

First row as header i

► Advanced

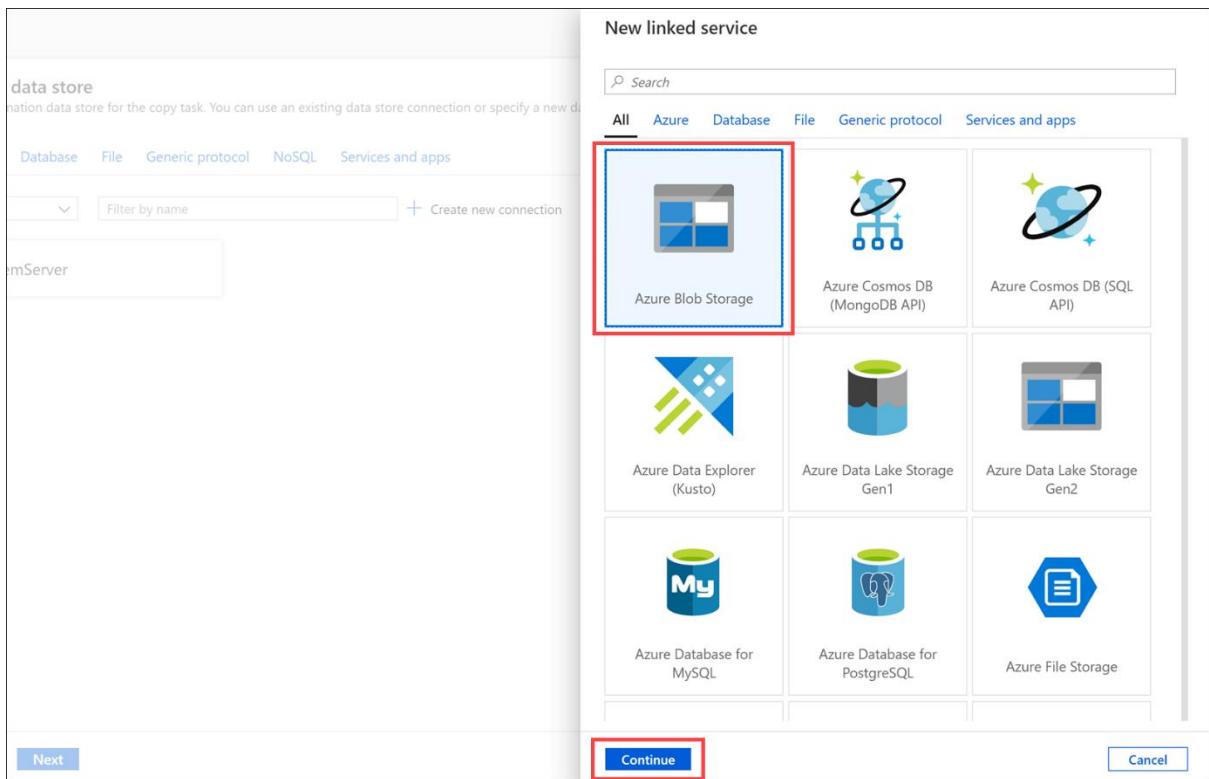
Preview	Schema
OriginAirportCode Month DayofMonth CRSDepHour DayOfWeek Carrier Dest	
ORD 3 9 9 3 UA LAS	
MDW 3 5 8 6 WN CLE	
... - - - -	

Previous
Next

11. Select **Next**.

12. On the Destination data store screen, select + **Create new connection**.

13. Select **Azure Blob Storage** within the New Linked Service blade, then select **Continue**.



14. On the New Linked Service (Azure Blob Storage) account screen, enter the following, test your connection, and then select **Create**.

- **Name:** BlobStorageOutput
- **Connect via integration runtime:** Select your Integration Runtime.
- **Authentication method:** Select **Account key**
- **Account selection method:** **From Azure subscription**
- **Storage account name:** Select the blob storage account you provisioned in the before-the-lab section.

New linked service (Azure Blob Storage)

Name *
BlobStorageOutput

Description

Connect via integration runtime *
bigdatagateway-jdh

Authentication method
Account key

Connection string Azure Key Vault

Account selection method
 From Azure subscription Enter manually

Azure subscription

Storage account name *
bigdatalabstore2

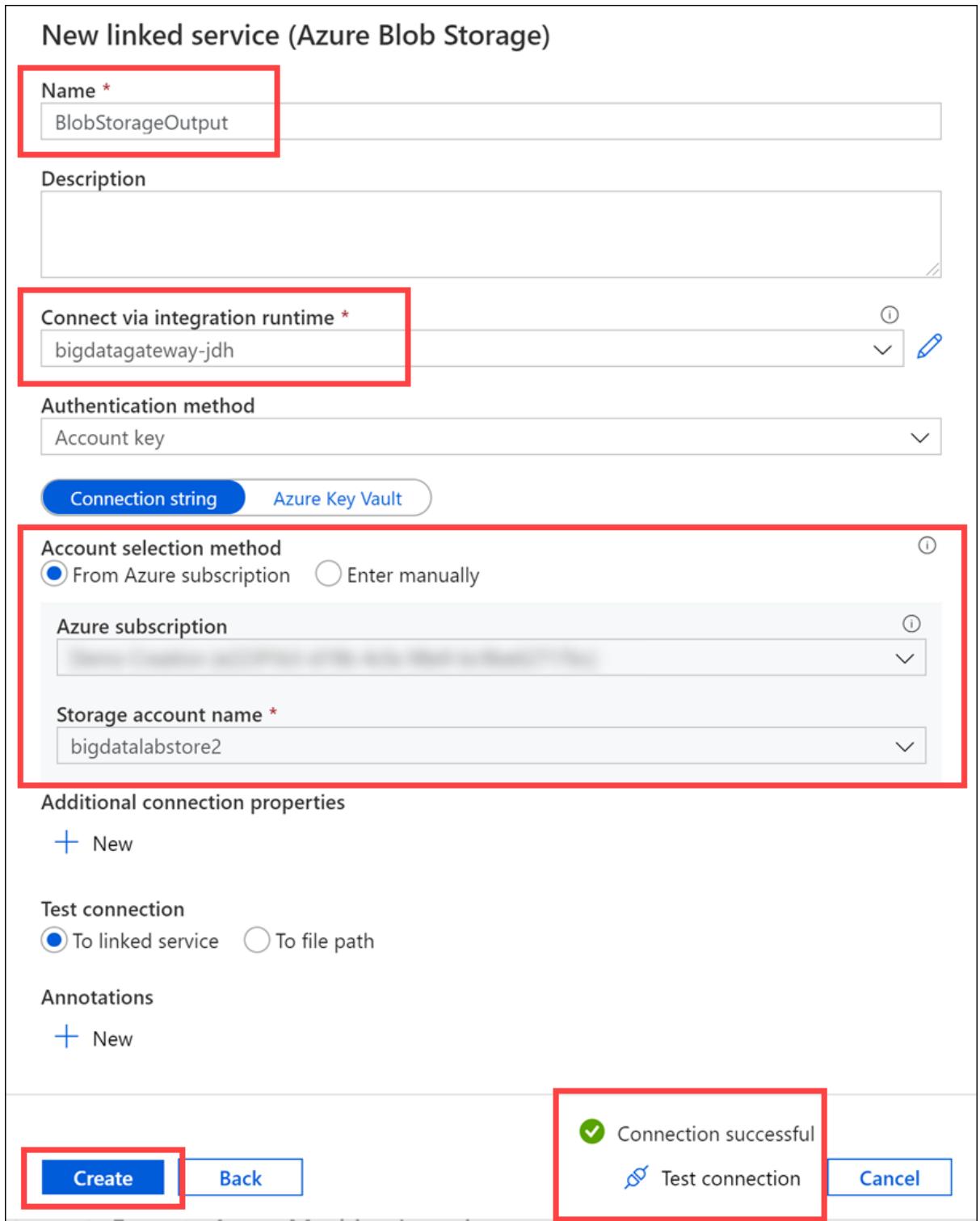
Additional connection properties
+ New

Test connection
 To linked service To file path

Annotations
+ New

Create Back

Connection successful
Test connection Cancel

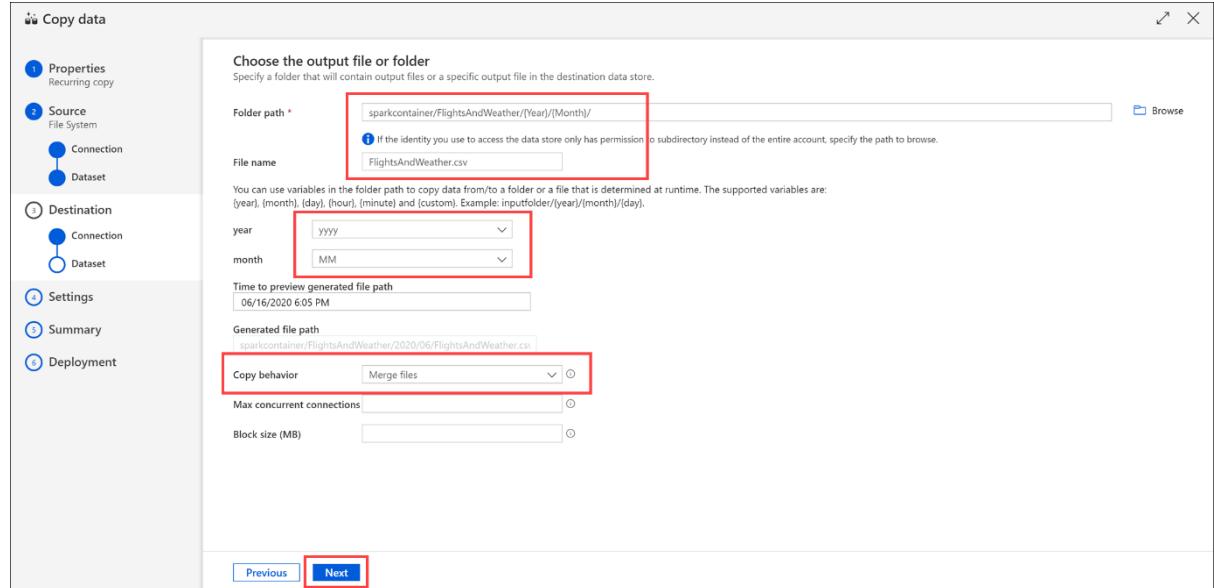


15. On the Destination data store page, select **Next**.

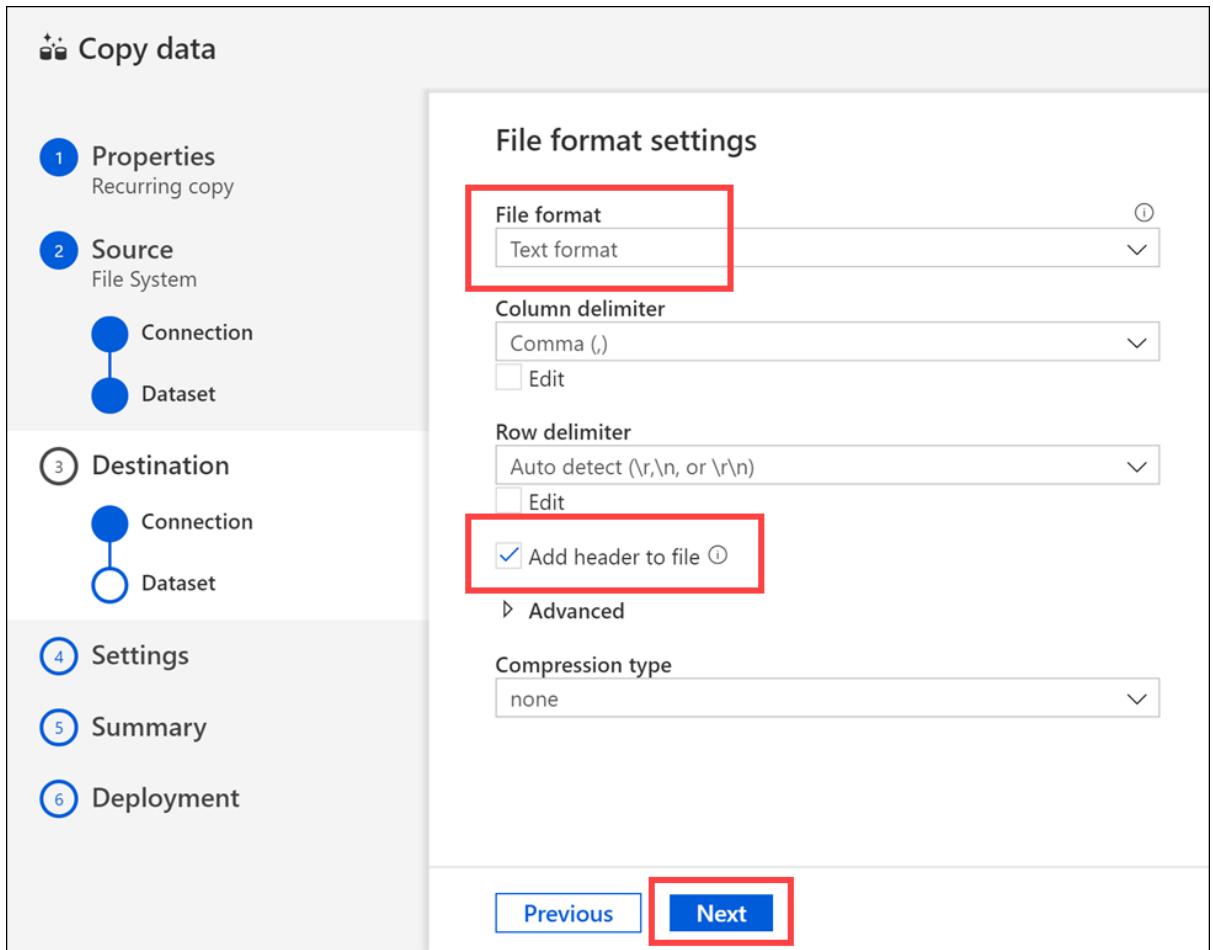
16. From the **Choose the output file or folder** tab, enter the following:

- o **Folder path:** sparkcontainer/FlightsAndWeather/{Year}/{Month}/
- o **Filename:** FlightsAndWeather.csv
- o **Year:** yyyy

- **Month: MM**
- **Copy behavior: Merge files**
- Select **Next**.



17. On the File format settings screen, select the **Text format** file format, and check the **Add header to file** checkbox, then select **Next**.



18. On the **Settings** screen, select **Skip incompatible rows** under Fault tolerance, and uncheck **Enable logging**. Expand Advanced settings and set Degree of copy parallelism to 10, then select **Next**.

Settings
More options for data movement

Fault tolerance: Skip incompatible rows

Enable logging:

Performance settings:

- Enable staging:
- Advanced settings:

Data integration unit: Auto

You will be charged # of used DIUs * copy duration * \$0.25/DIU-hour. Local currency and separate discounting may apply per subscription type. [Learn more](#)

Degree of copy parallelism: 10

Previous Next

19. Review settings on the **Summary** tab, but **DO NOT choose Next**.

Summary
You are running pipeline to copy data from File System to Azure Blob Storage.

Properties

Task name: CopyOnPrem2AzurePipeline

Task description: This pipeline copies time-sliced CSV files from on-premises C:\\Data to Azure Blob Storage as a continuous job.

Source

Connection name: OnPremServer

Dataset name: SourceDataset_i0e

Column delimiter: ,

Escape character: \\

Quote char: "

First row as header: true

Folder path: FlightsAndWeather

Destination

Previous Next

20. Scroll down on the summary page until you see the **Copy Settings** section. Select **Edit** next to **Copy Settings**.

Summary
You are running pipeline to copy data from File System to Azure Blob Storage.

The diagram illustrates a data flow from 'File System' to 'Azure Blob Storage'. On the left, there is a blue icon representing a file system. A horizontal arrow points from this icon to a second blue icon representing Azure Blob Storage. Below the icons, the text 'File System' and 'Azure Blob Storage' are labeled respectively.

Column delimiter	,
Escape character	\
Quote char	"
First row as header	true
Copy settings	
Timeout	7.00:00:00
Retry	0
Retry interval	30
Secure output	false
Secure input	false
Trigger	
Name	Trigger i0e

Edit (highlighted with a red box)

Edit (highlighted with a red box)

21. Change the following Copy setting:

- **Retry:** 3
- Select **Save**.

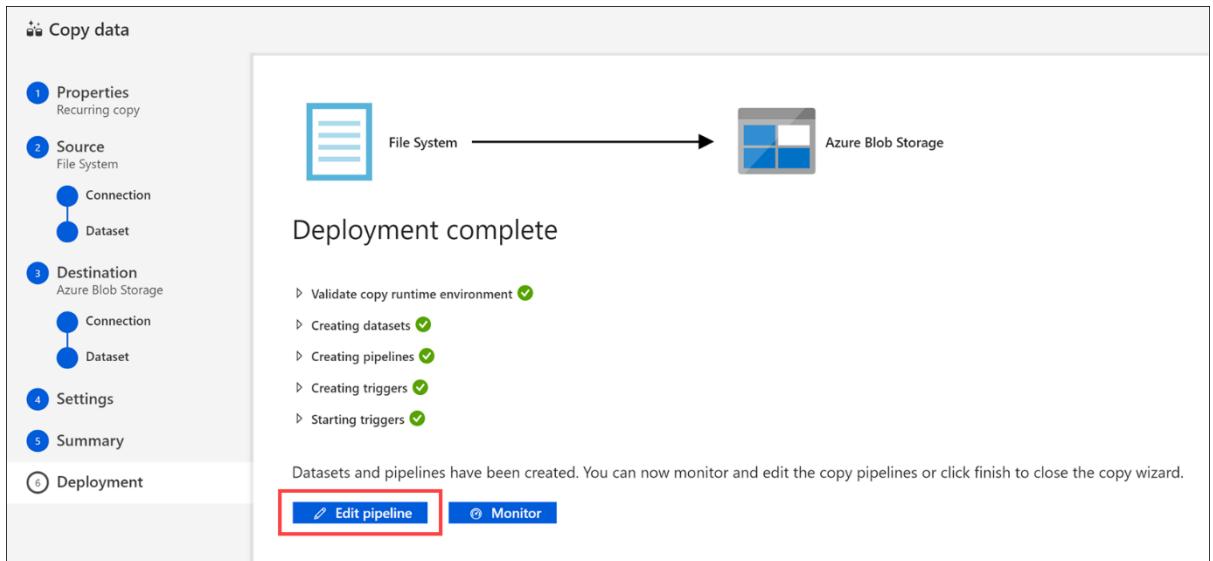
Copy settings

Timeout	7.00:00:00
Retry	<input type="text" value="3"/> (highlighted with a red box)
Retry interval	30
Secure output	<input type="checkbox"/>
Secure input	<input type="checkbox"/>

Save | **Cancel** (highlighted with a red box)

22. After saving the Copy settings, select **Next** on the Summary tab.

23. On the **Deployment** screen you will see a message that the deployment is in progress, and after a minute or two that the deployment completed. Select **Edit Pipeline** to close out of the wizard and navigate to the pipeline editing blade.



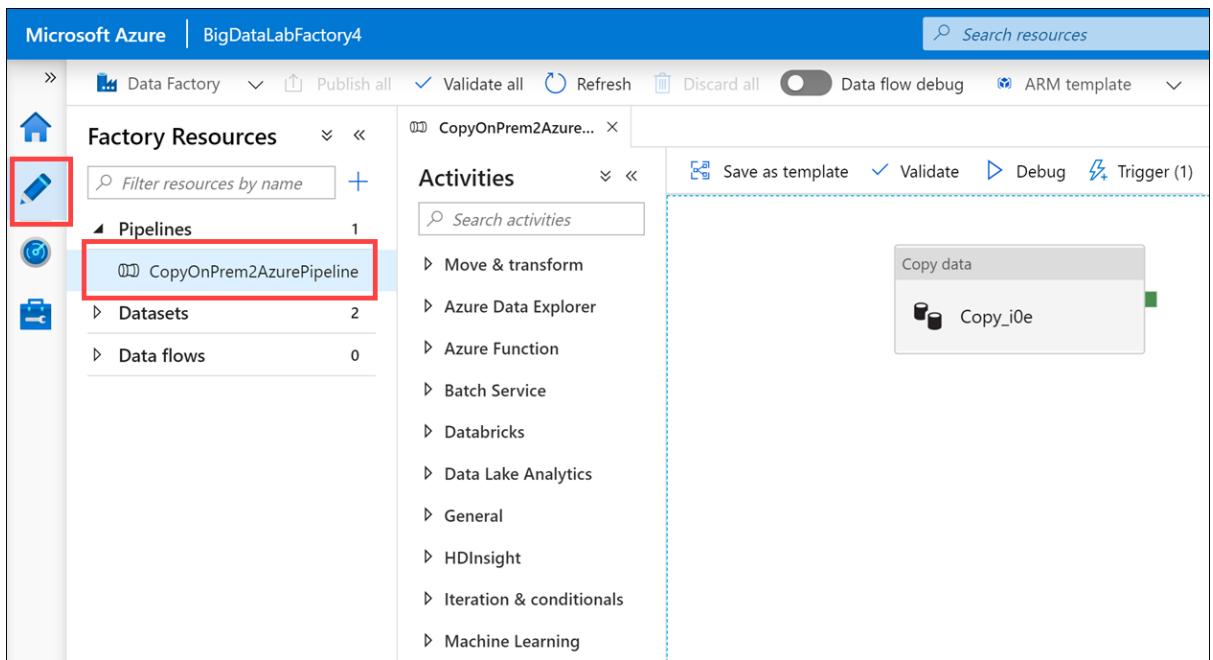
Exercise 5: Operationalize ML scoring with Azure Databricks and Data Factory

Duration: 20 minutes

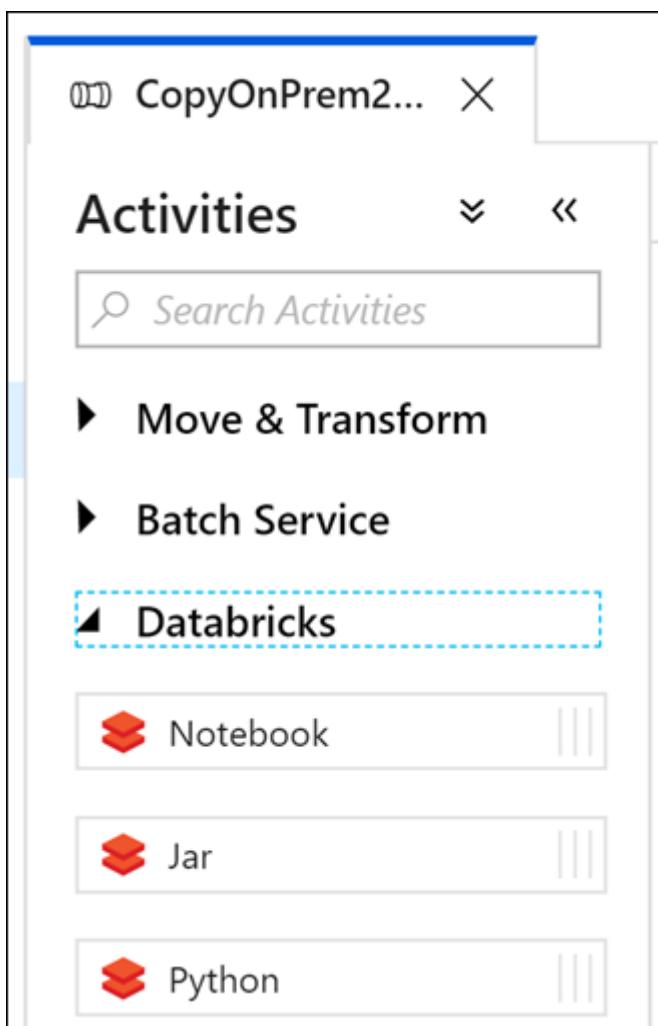
In this exercise, you will extend the Data Factory to operationalize the scoring of data using the previously created machine learning model within an Azure Databricks notebook.

Task 1: Create Azure Databricks Linked Service

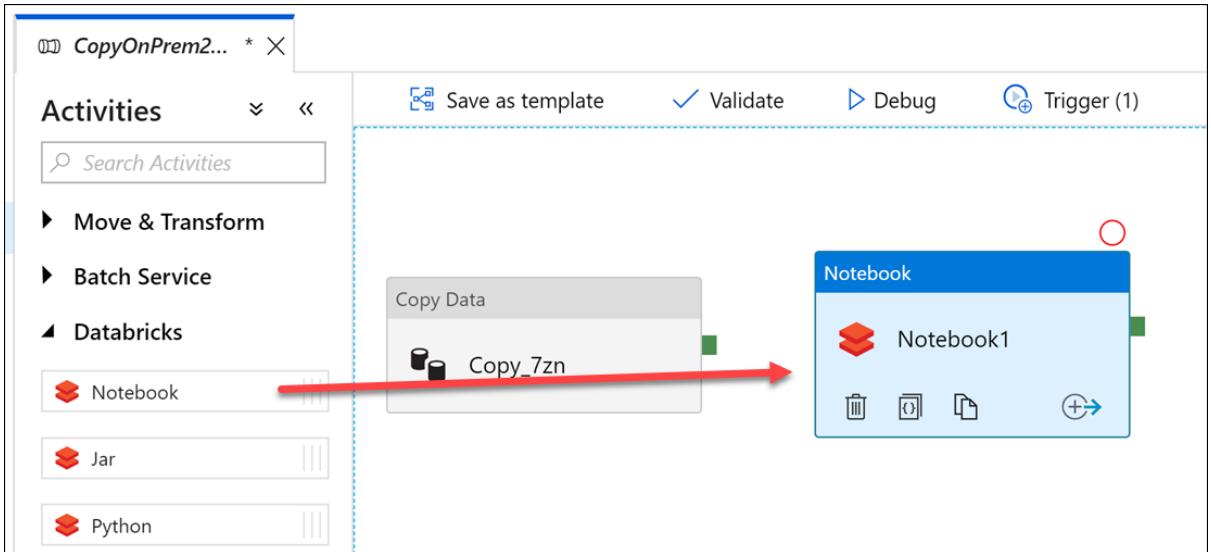
1. Return to, or reopen, the Author & Monitor page for your Azure Data Factory in a web browser, navigate to the Author view, and select the pipeline.



- Once there, expand Databricks under Activities.



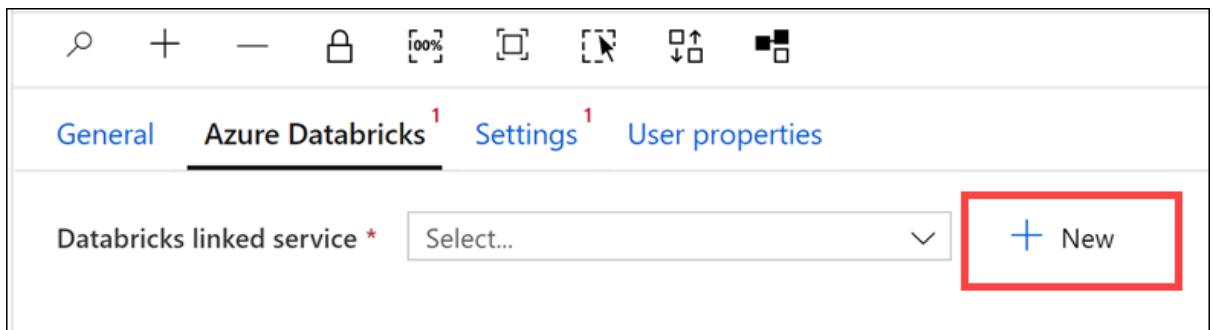
3. Drag the Notebook activity onto the design surface to the side of the Copy activity.



4. Select the Notebook activity on the design surface to display tabs containing its properties and settings at the bottom of the screen. On the **General** tab, enter BatchScore into the Name field.

General		Azure Databricks 1	Settings 1	User Properties
Name *	BatchScore			
Description				
Timeout	7.00:00:00			
Retry	0			
Retry interval	30			
Secure output	<input type="checkbox"/> ⓘ			

5. Select the **Azure Databricks** tab, and select **+ New** next to the Databricks Linked service drop down. Here, you will configure a new linked service which will serve as the connection to your Databricks cluster.



6. On the New Linked Service dialog, enter the following:

- **Name:** AzureDatabricks
- **Connect via integration runtime:** Leave set to Default.
- **Account selection method: From Azure subscription**
- **Azure subscription:** Choose your Azure Subscription.
- **Databricks workspace:** Pick your Databricks workspace to populate the Domain automatically.
- **Select cluster: Existing interactive cluster**

New linked service (Azure Databricks)

Name *
AzureDatabricks

Description

Connect via integration runtime *
AutoResolveIntegrationRuntime

Account selection method *
From Azure subscription

Azure subscription *

Databricks workspace *
BigDataLab

Select cluster

New job cluster Existing interactive cluster Existing instance pool

Databrick Workspace URL *
https://adb-678483592075405.5.azuredatabricks.net

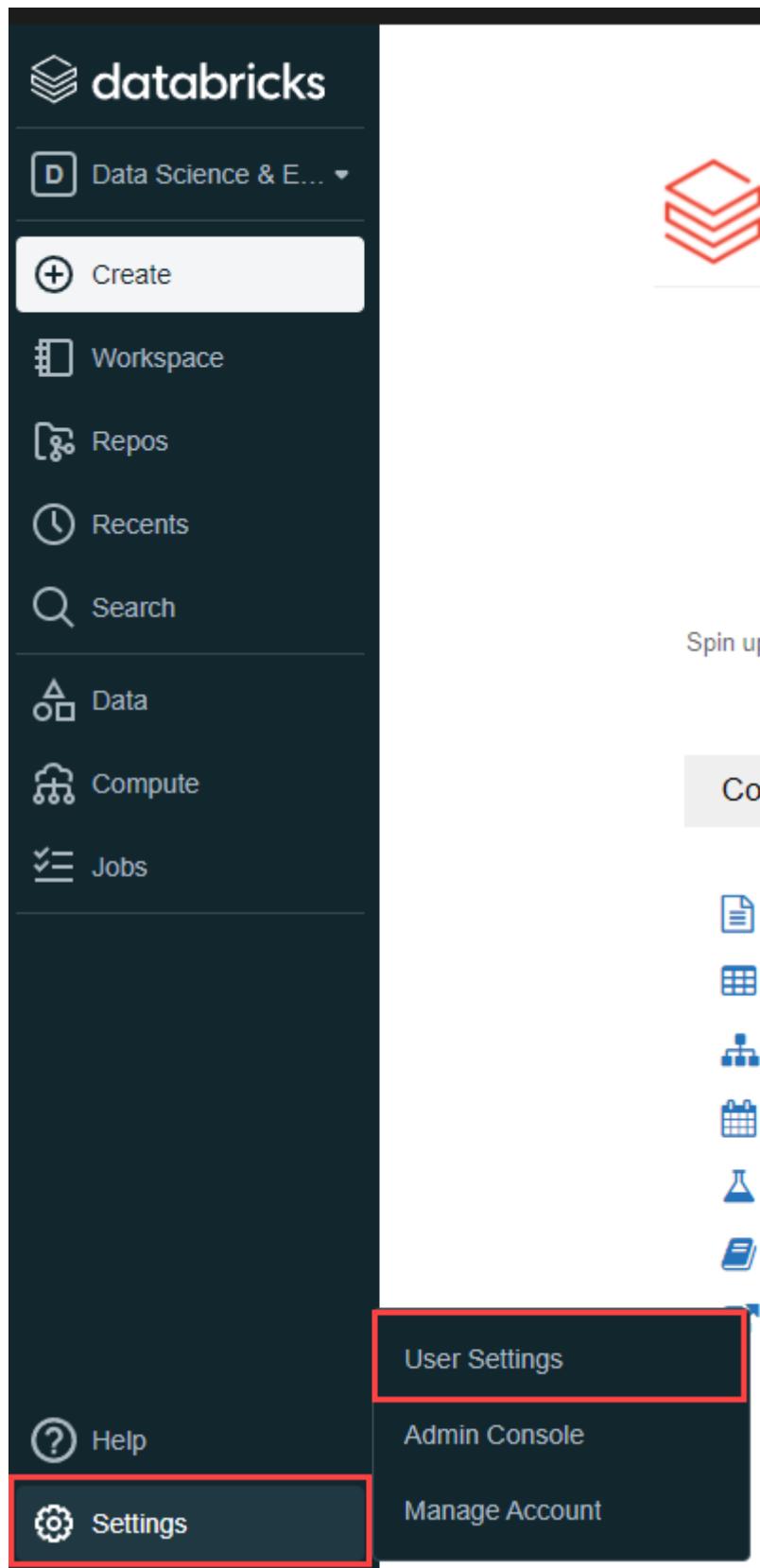
Access token **Azure Key Vault**

Access token *

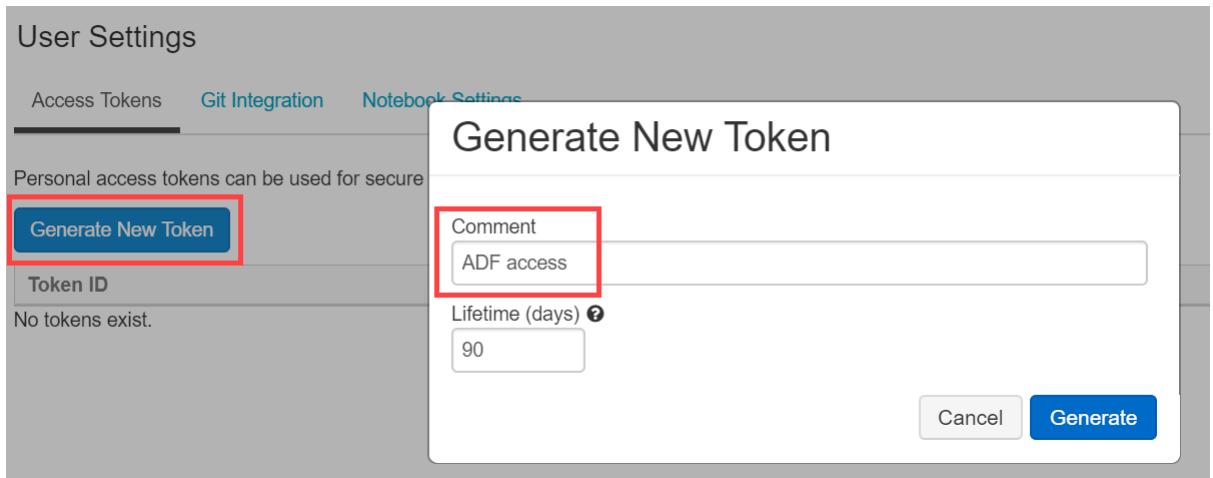
Existing cluster ID *
Add region and token to list options

Annotations
+ New

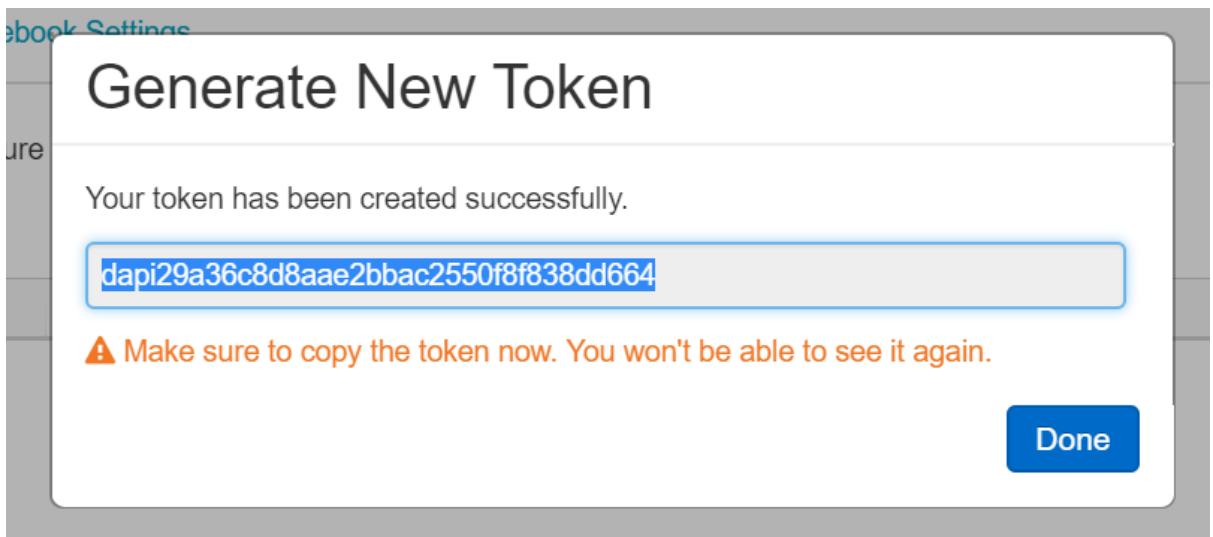
7. Leave the form open and open your Azure Databricks workspace in another browser tab. You will generate and retrieve the Access token here.
8. In Azure Databricks, select the Account icon in the top corner of the window, then select **User Settings**.



9. Select **Generate New Token** under the Access Tokens tab. Enter **ADF access** for the comment and leave the lifetime at 90 days. Select **Generate**.



10. **Copy** the generated token and **paste it into a text editor** such as Notepad for a later step.



11. Switch back to your Azure Data Factory screen and paste the generated token into the **Access token** field within the form. After a moment, select your cluster underneath **Choose from existing clusters**. Select **Create**.

Select cluster

New job cluster Existing interactive cluster Existing instance pool

Domain/Region *

<https://westus.azuredatabricks.net>

Access token	Azure Key Vault
Access token *	<input type="text"/>
Choose from existing clusters *	<input type="text"/> lab

Annotations

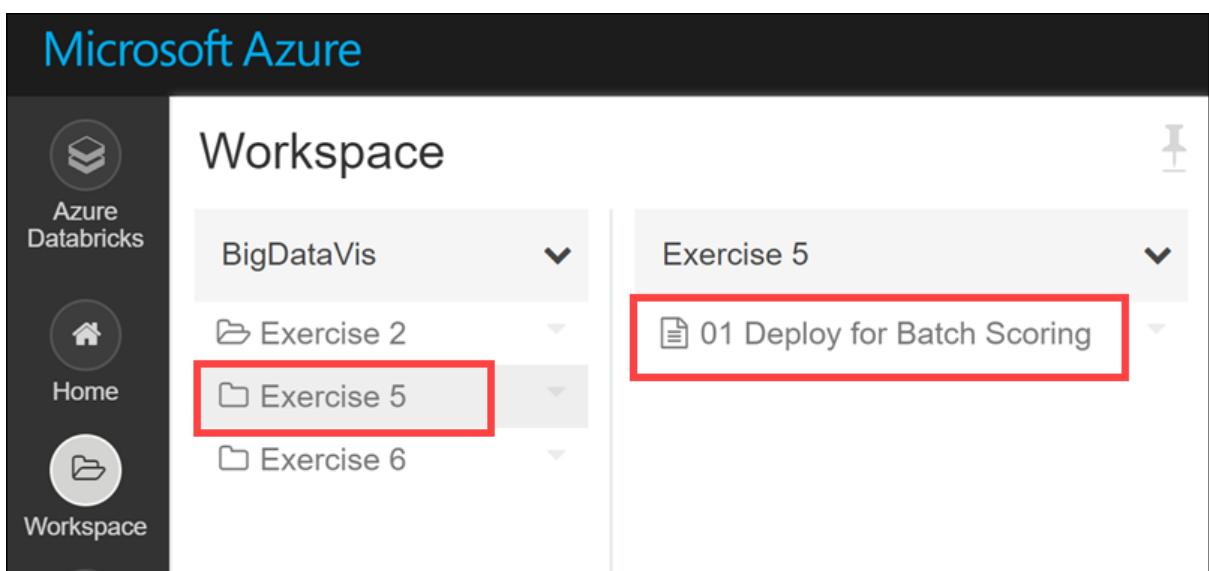
+ New

► Parameters

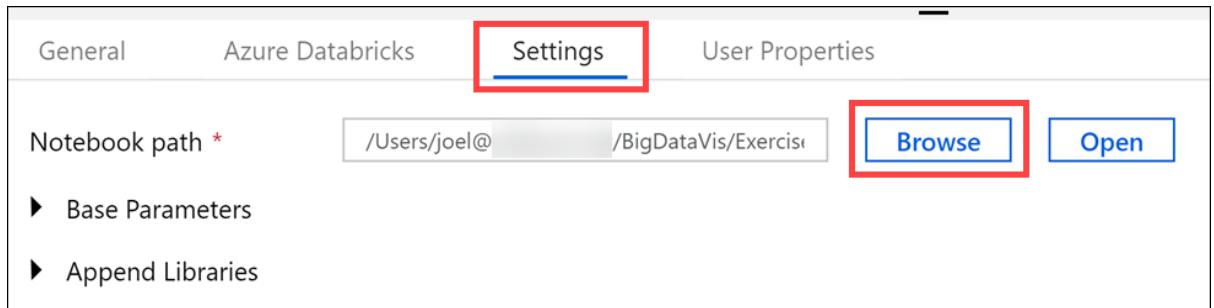
► Advanced ⓘ

Create Test connection Cancel

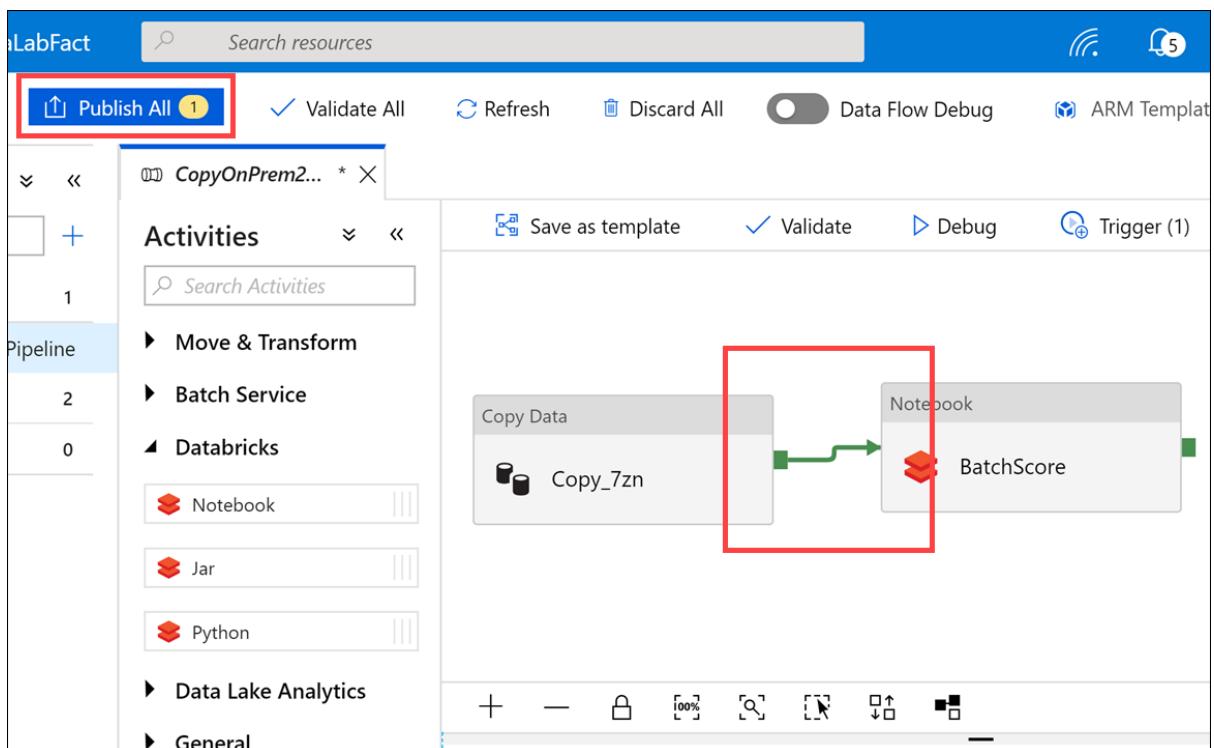
12. Switch back to Azure Databricks. Select **Workspace** in the menu. Select the **Exercise 5** folder then open notebook **01 Deploy for Batch Scoring**. Examine the content but *don't run any of the cells yet*. You need to **replace STORAGE-ACCOUNT-NAME** with the name of the blob storage account you copied in Exercise 1 into Cmd 4.



13. Switch back to your Azure Data Factory screen. Select the **Settings** tab, then browse to your **Exercise 5/01 Deploy for Batch Score** notebook into the Notebook path field.



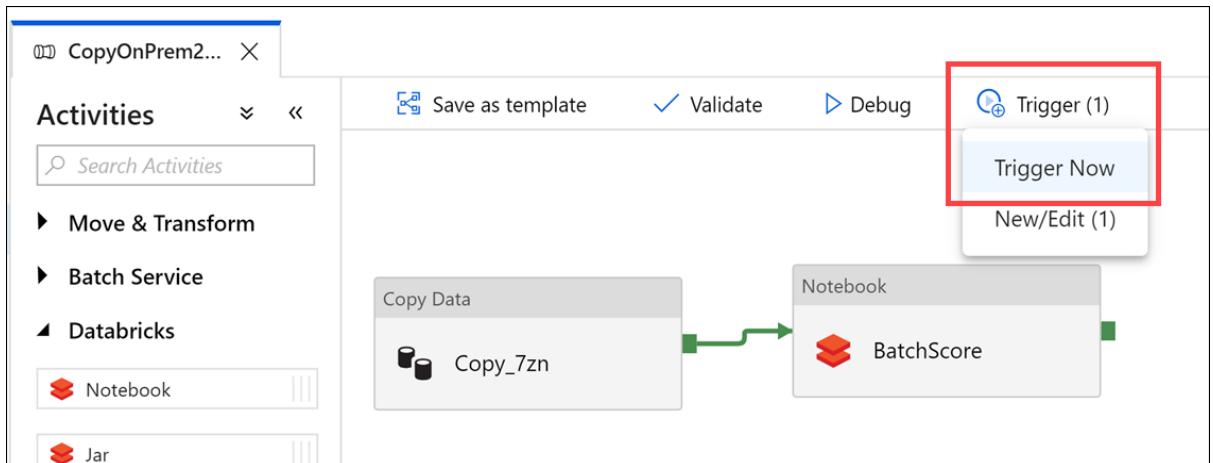
14. The final step is to connect the Copy activities with the Notebook activity. Select the small green box on the side of the copy activity, and drag the arrow onto the Notebook activity on the design surface. What this means is that the copy activity has to complete processing and generate its files in your storage account before the Notebook activity runs, ensuring the files required by the BatchScore notebook are in place at the time of execution. Select **Publish All**, then **Publish**, after making the connection.



Task 2: Trigger workflow

1. Switch back to Azure Data Factory. Select your pipeline if it is not already opened.

2. Select **Trigger**, then **Trigger Now** located above the pipeline design surface.



3. Enter 3/1/2017 into the **windowStart** parameter, then select **OK**.

The screenshot shows the 'Pipeline run' dialog box. At the top, there is a warning message: '⚠ Trigger pipeline now using last published configuration.' Below it, there is a section titled 'Parameters' with a table:

NAME	TYPE	VALUE
windowStart	String	3/1/2017

At the bottom, there are 'OK' and 'Cancel' buttons. The '3/1/2017' input field is highlighted with a red rectangle.

4. Select **Monitor** in the menu. You will be able to see your pipeline activity in progress as well as the status of past runs.

The screenshot shows the Microsoft Azure portal's 'Pipeline runs' page. The left sidebar has a 'Filters' section with a red box around the 'Trigger runs' button. The main area displays a table of pipeline runs:

Pipeline Name	Run Start	Duration	Triggered By	Status	Parameters	Annotat
CopyOnPrem2AzurePipeline	6/16/20, 2:53:17 PM	00:00:06	Manual trigger	In progress		

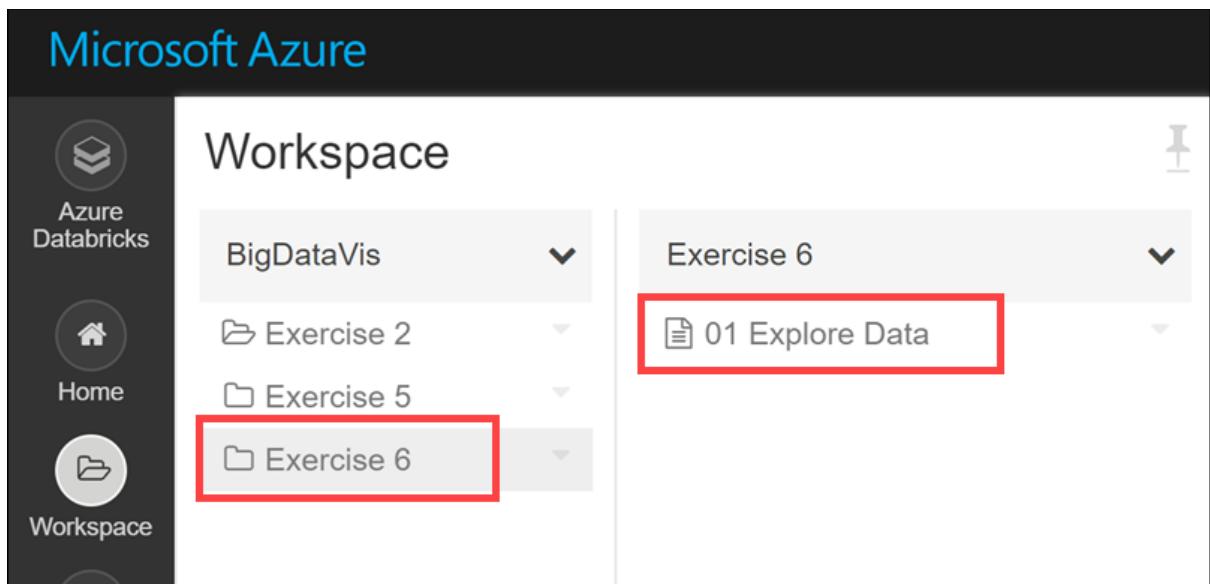
Note: You may need to restart your Azure Databricks cluster if it has automatically terminated due to inactivity.

Exercise 6: Summarize data using Azure Databricks

In this exercise, you will prepare a summary of flight delay data using Spark SQL.

Task 1: Summarize delays by airport

1. Open your Azure Databricks workspace, expand the **Exercise 6** folder and open the final notebook called **01 Explore Data**.



2. Execute each cell and follow the instructions in the notebook that explains each step.

Exercise 7: Visualizing in Power BI Desktop

In this exercise, you will create visualizations in Power BI Desktop.

Task 1: Obtain the JDBC connection string to your Azure Databricks cluster

Before you begin, you must first obtain the JDBC connection string to your Azure Databricks cluster.

1. In Azure Databricks, go to Clusters and select your cluster.
2. On the cluster edit page, scroll down to the bottom of the page, expand **Advanced Options**, then select the **JDBC/ODBC** tab.

▼ Advanced Options

Azure Data Lake Storage Credential Passthrough ?

Enable credential passthrough for user-level data access

Spark Tags Logging Init Scripts JDBC/ODBC Permissions

Server Hostname
adb-6784833592075405.5.azuredatabricks.net

Port
443

Protocol
HTTPS

HTTP Path
sql/protocolv1/o/6784833592075405/0615-225254-need937

JDBC URL ?

```
jdbc:spark://adb-6784833592075405.5.azuredatabricks.net:443/default;transportMode=http;ssl=1;httpPath=https/protocolv1/o/6784833592075405/0615-225254-need937;AuthMech=3;UID=token;PWD=<personal-access-token>
```

3. On the **JDBC/ODBC** tab, copy and save the first JDBC URL.

- Construct the JDBC server address that you will use when you set up your Spark cluster connection in Power BI Desktop.
- Take the JDBC URL and do the following:
 - Replace `jdbc:spark` with `https`.
 - Remove everything in the path between the port number and `sql`, retaining the components indicated by the boxes in the image below. Also remove `;AuthMech=3;UID=token;PWD=<personal-access-token>` from the end of the string.

▼ Advanced Options

Azure Data Lake Storage Credential Passthrough ?

Enable credential passthrough for user-level data access

Spark Tags Logging Init Scripts JDBC/ODBC **Permissions**

Server Hostname
adb-6784833592075405.5.azuredatabricks.net

Port
443

Protocol
HTTPS

HTTP Path
sql/protocolv1/o/6784833592075405/0615-225254-need937

JDBC URL ?

```
jdbc:spark://adb-
6784833592075405.5.azuredatabricks.net:443/default;transportMode=http;ssl=1;httpP
ath=sql/protocolv1/o/6784833592075405/0615-225254-
need937;AuthMech=3;UID=token;PWD=<personal-access-token>
```

- In our example, the server address would be:

<https://adb-6784833592075405.5.azuredatabricks.net:443/sql/protocolv1/o/6784833592075405/0615-225254-need937>

Task 2: Connect to Azure Databricks using Power BI Desktop

1. If you did not already do so during the before the hands-on lab setup, download Power BI Desktop from <https://powerbi.microsoft.com/en-us/desktop/>.
2. When Power BI Desktop starts, you will need to enter your personal information, or Sign in if you already have an account.

Welcome to Power BI Desktop

Where can we send you the latest tips and tricks for Power BI?

First Name *

Last Name *

Email Address *

Enter your phone number *

Country/region *

Company name *

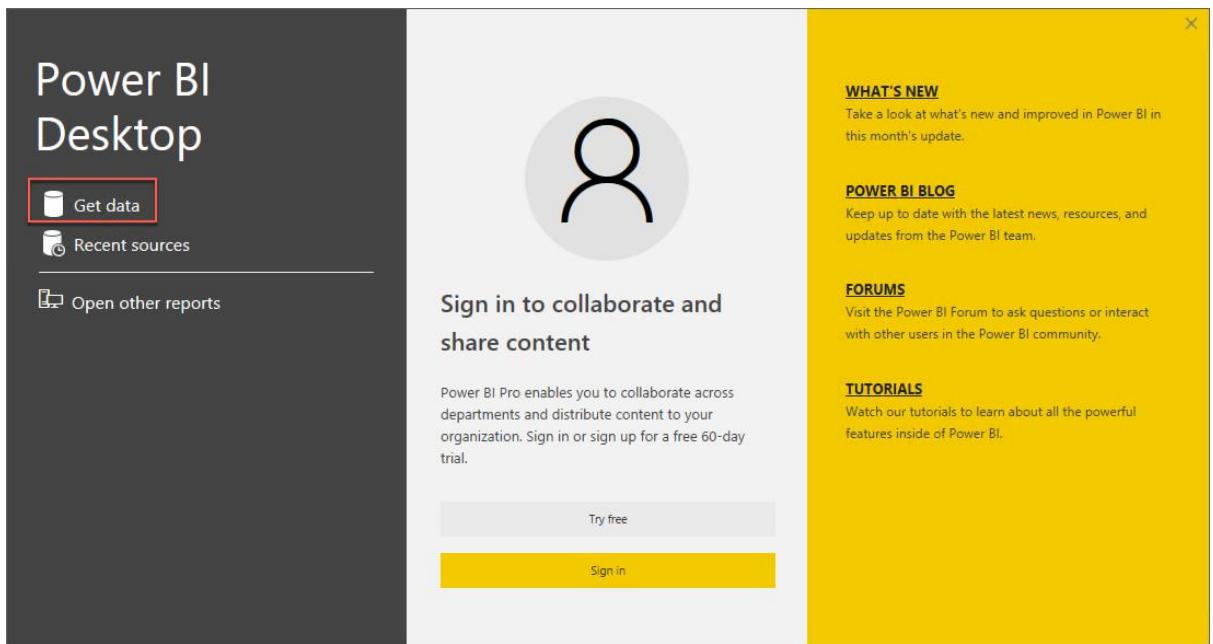
Job Role*

Microsoft may use your contact information to provide updates and special offers about Business Intelligence and other Microsoft products and services. You can unsubscribe at any time. To learn more you can read the [privacy statement](#).

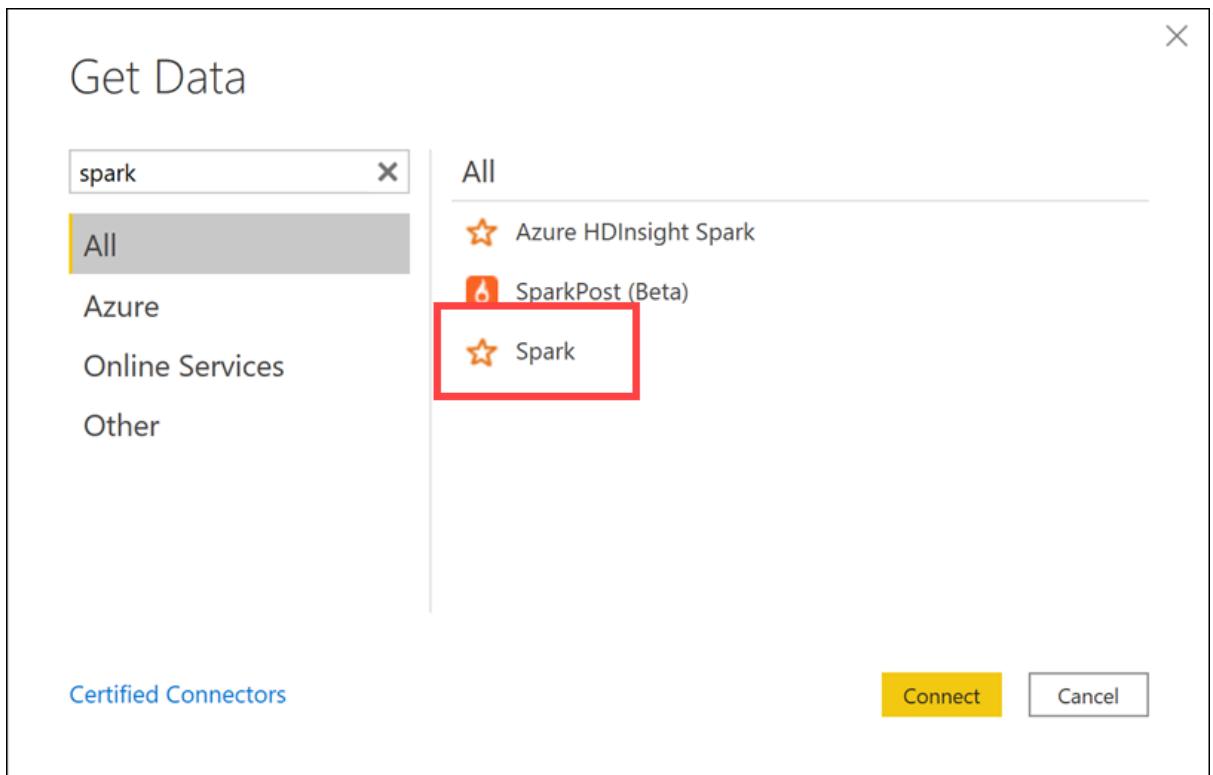
Done

[Already have a Power BI account? Sign in](#)

3. Select Get data on the screen that is displayed next.

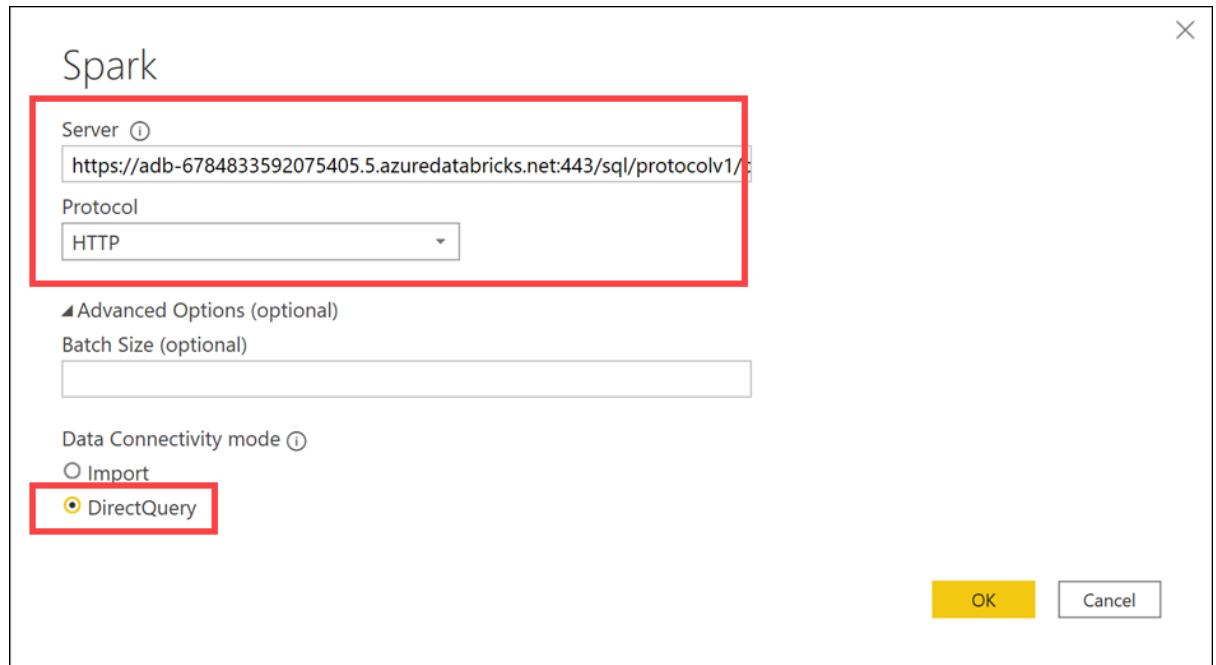


4. Select **Spark** from the list of available data sources. You may enter Spark into the search field to find it faster.



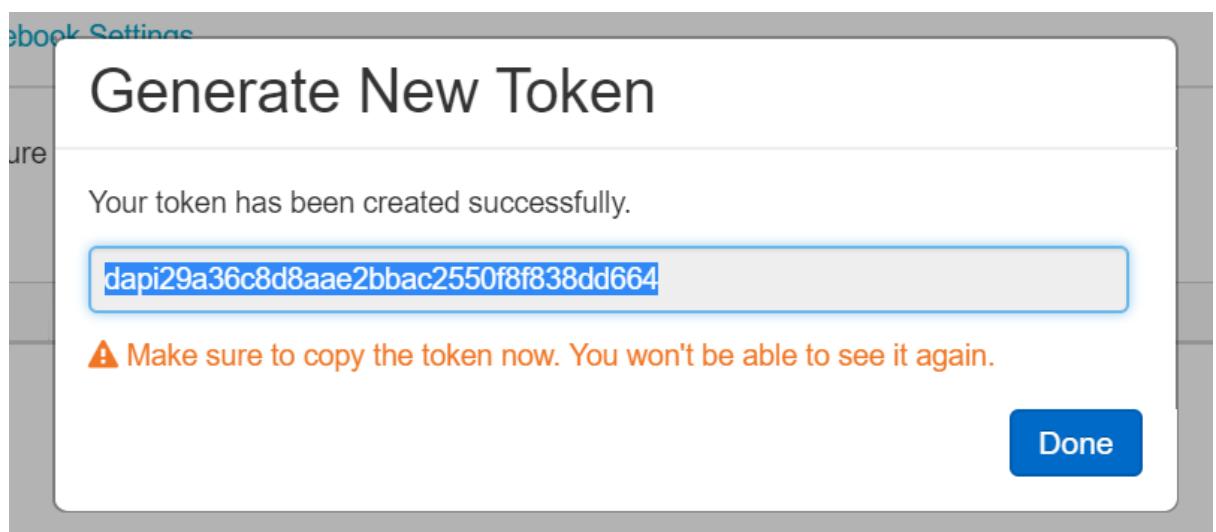
5. Select **Connect**.
6. On the next screen, you will be prompted for your Spark cluster information.
7. Paste the JDBC connection string you constructed into the **Server** field.
8. Select the **HTTP** protocol.

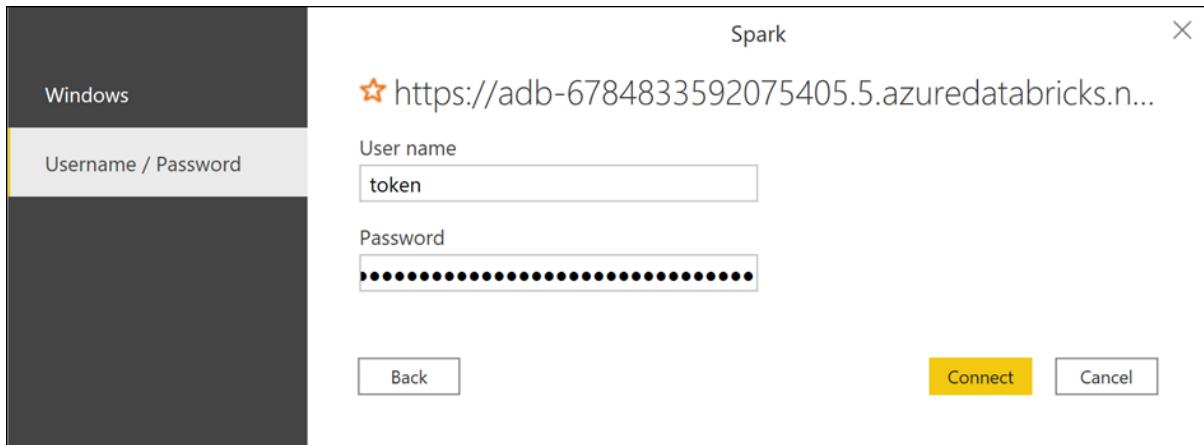
9. Select **DirectQuery** for the Data Connectivity mode, and select **OK**. This option will offload query tasks to the Azure Databricks Spark cluster, providing near-real time querying.



10. Enter your credentials on the next screen as follows:

- **User name:** token
- **Password:** Remember that ADF Access token we generated for the Azure Data Factory notebook activity? Paste the same value here for the password.





11. Select **Connect**.

12. In the Navigator dialog, check the box next to **flight_delays_summary**, and select **Load**.

OriginAirportCode	Month	DayofMonth	CRSDepHour	NumDelays	Origin
MEM	4	10	18	2 35.	
OGG	4	14	20	2 20.	
SDF	4	18	13	2 38.	
HNL	4	9	21	2 21.	
BNA	4	3	16	2 36.	
RSW	4	24	15	2 26.	
AUS	4	18	20	2 30.	
MKE	4	21	17	2 42.	
PDX	4	26	19	2 45.	
TPA	4	2	19	2 27.	
BWI	4	10	12	2 39.	
STL	4	5	18	2 38.	
SEA	4	1	16	2 47.	
SJC	4	3	14	2 37.	
JAX	4	7	18	2 30.	
SJC	4	5	16	2 37.	
OAK	4	14	21	2 37.	
HNL	4	21	16	2 21.	
LAS	4	17	22	2 36.	
DCA	4	14	20	2 38.	
DAL	4	14	15	2 32.	
SNA	4	16	20	2 33.	
OAK	4	27	17	2 37.	

Task 3: Create Power BI report

- Once the data finishes loading, you will see the fields appear on the far side of the Power BI Desktop client window.

Fields

Search

flight_delays_sum...

- Σ CRSDepHour
- Σ DayofMonth
- Σ Month
- Σ NumDelays
- OriginAirpo...
- OriginLatLo...

2. From the Visualizations area, next to Fields, select the Globe icon to add a Map visualization to the report design surface.

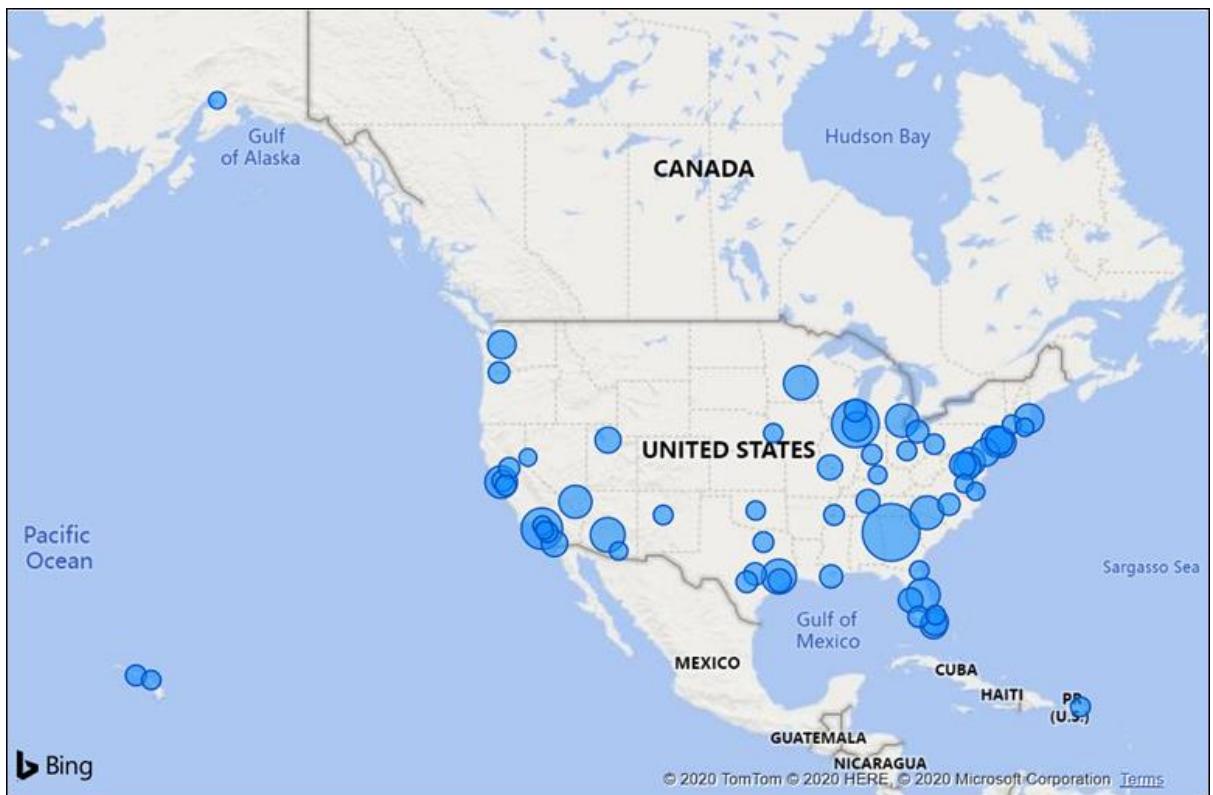
Visualizations

The image shows a grid of icons representing various data visualizations. The globe icon is located in the second row, third column from the left, and is highlighted with a red square.

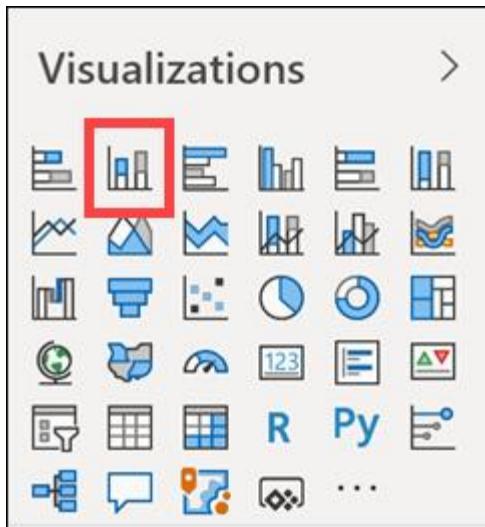
3. With the Map visualization still selected, drag the **OriginLatLong** field to the **Location** field under Visualizations. Then Next, drag the **NumDelays** field to the **Size** field under Visualizations.

The screenshot shows the Power BI interface with the 'Visualizations' and 'Fields' panes open. In the 'Fields' pane, the 'flight_delays_su...' table is selected. The 'NumDelays' field is checked and highlighted with a yellow circle. The 'OriginLatLong' field is also present in the list. In the 'Visualizations' pane, under the 'Location' section, 'OriginLatLong' is selected. In the 'Size' section, 'NumDelays' is selected. Red arrows point from the 'OriginLatLong' and 'NumDelays' fields in the 'Fields' pane to their respective sections in the 'Visualizations' pane.

4. You should now see a map that looks similar to the following (resize and zoom on your map if necessary):



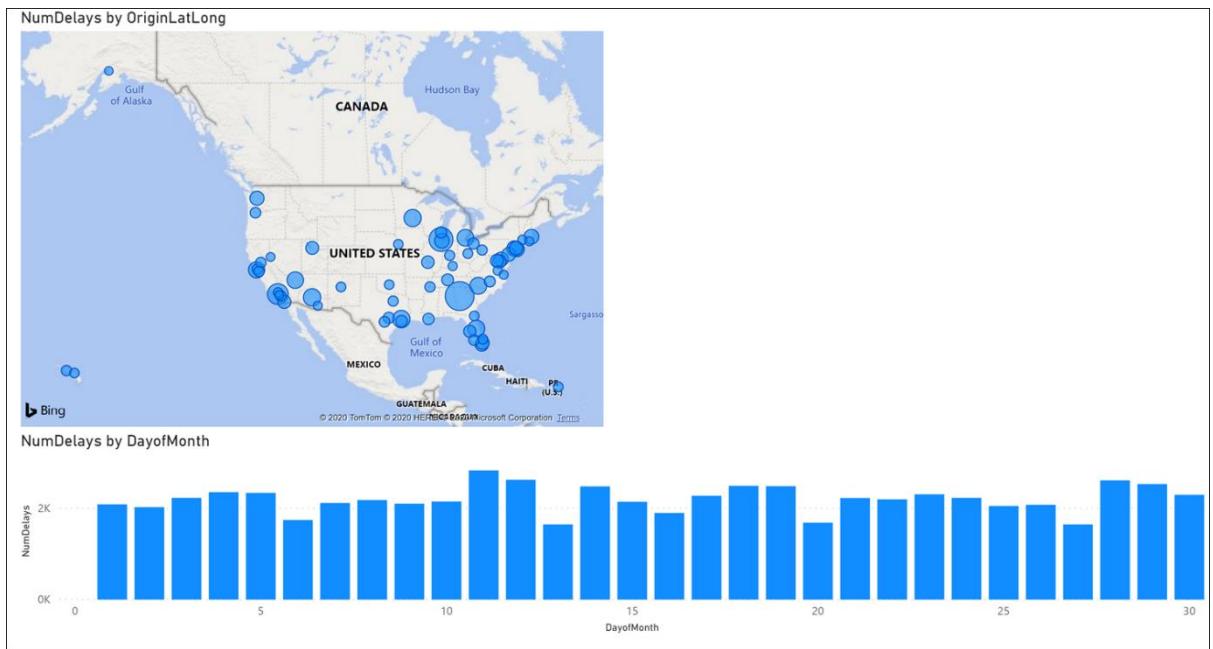
5. Unselect the Map visualization by selecting the white space next to the map in the report area.
6. From the Visualizations area, select the **Stacked Column Chart** icon to add a bar chart visual to the report's design surface.



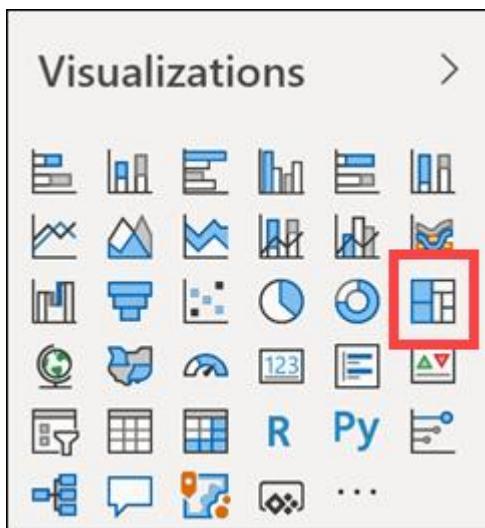
7. With the Stacked Column Chart still selected, drag the **DayofMonth** field and drop it into the **Axis** field located under Visualizations.
8. Next, drag the **NumDelays** field over, and drop it into the **Value** field.

The screenshot shows the Power BI Visualizations pane on the left and the Fields pane on the right. In the Visualizations pane, under the 'Axis' section, 'DayofMonth' is selected. In the Values section, 'NumDelays' is listed. A red arrow points from the 'DayofMonth' selection in the Axis section to the 'NumDelays' field in the Values section. The Fields pane on the right lists various fields: 'flight_delays_su...' (selected), 'CRSDepHour', 'DayofMonth' (selected), 'Month', 'NumDelays' (selected), 'OriginAirpo...', and 'OriginLatLo...'. A search bar at the top of the Fields pane contains the text 'Search'.

9. Grab the corner of the new Stacked Column Chart visual on the report design surface, and drag it out to make it as wide as the bottom of your report design surface. It should look something like the following.



10. Unselect the Stacked Column Chart visual by selecting on the white space next to the map on the design surface.
11. From the Visualizations area, select the Treemap icon to add this visualization to the report.



12. With the Treemap visualization selected, drag the **OriginAirportCode** field into the **Group** field under Visualizations.
13. Next, drag the **NumDelays** field over, and drop it into the **Values** field.

The screenshot shows the Power BI interface with two main panes: 'Visualizations' on the left and 'Fields' on the right.

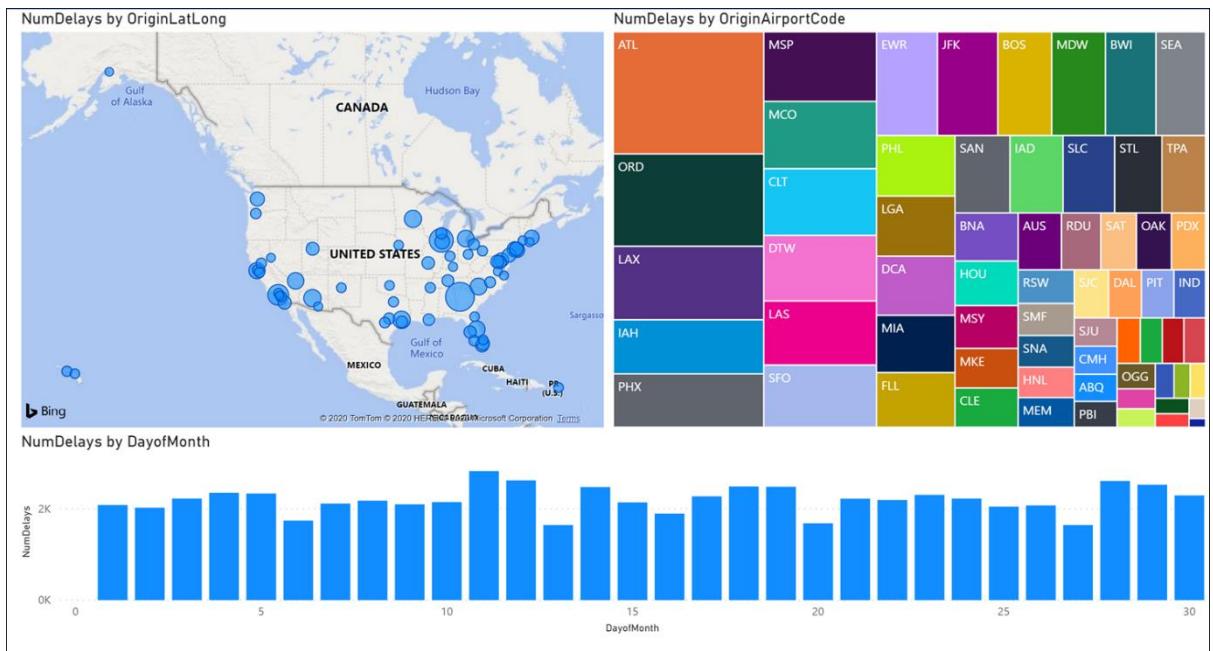
Visualizations pane:

- Shows various visualization icons including Bar, Line, Map, and Treemap.
- A red arrow points from the 'OriginAirportCode' field in the Fields pane down to the 'Values' section in the Visualizations pane.
- The 'Group' icon is highlighted with a yellow bar below it.
- The 'Treemap' icon is selected.

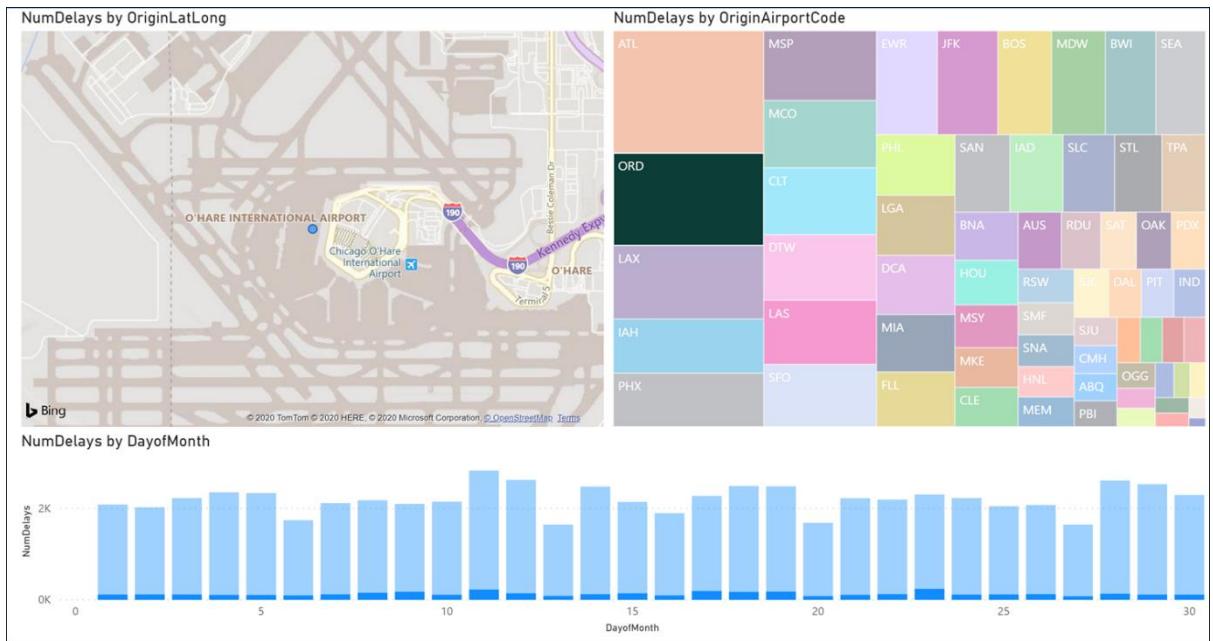
Fields pane:

- A search bar at the top with the placeholder 'Search'.
- An expandable tree view under 'flight_delays_su...' with several fields listed:
 - Σ CRSDepHour
 - Σ DayofMonth
 - Σ Month
 - Σ NumDelays
 - OriginAirpo...
 - OriginLatLo...

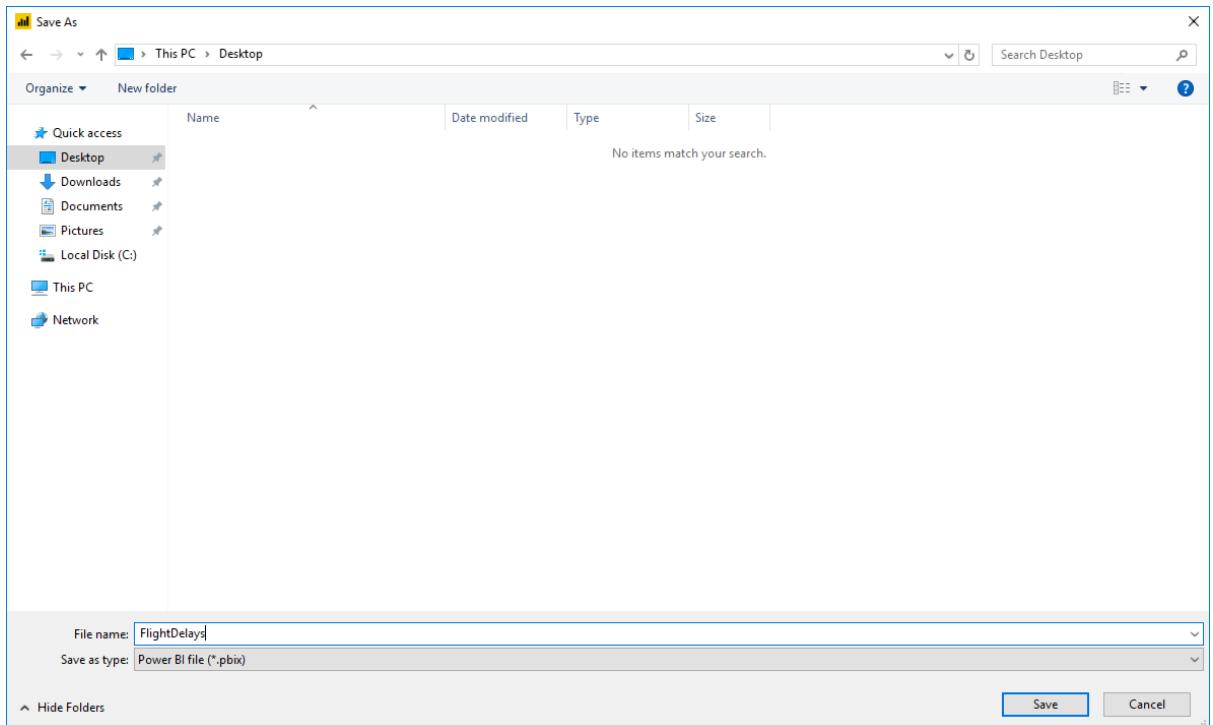
14. Grab the corner of the Treemap visual on the report design surface, and expand it to fill the area between the map and the side edge of the design surface. The report should now look similar to the following.



15. You can cross filter any of the visualizations on the report by selecting one of the other visuals within the report, as shown below (This may take a few seconds to change, as the data is loaded).



16. You can save the report, by choosing Save from the File menu, and entering a name and location for the file.



Deploy web app from GitHub(optional)

1. Navigate to <https://github.com/Microsoft/MCW-Big-data-analytics-and-visualization/blob/main/Hands-on%20lab/lab-files/BigDataTravel/README.md> in your browser of choice, but where you are already authenticated to the Azure portal.
2. Read through the README information on the GitHub page.
3. Select **Deploy to Azure** button in the Readme.md file.



4. On the following page, ensure the fields are populated correctly.
 - Ensure the correct Directory and Subscription are selected.
 - Select the Resource Group that you have been using throughout this lab.
 - Either keep the default Site name, or provide one that is globally unique, and then choose a Site Location.
 - Enter the **OpenWeather API Key**.

- Finally, enter the **ML URL**. We got this from Azure databricks Notebook [#3](#) in the Exercise 2 folder.

The screenshot shows the Microsoft Azure portal's "Custom deployment" interface. At the top, there's a navigation bar with "Microsoft Azure" and a search bar. Below it, the page title is "Custom deployment" with a "Deploy from a custom template" link. The "Basics" tab is selected. Under "Template", there's a "Customized template" section showing "2 resources". To the right are buttons for "Edit template", "Edit parameters", and "Visualize".

Project details:

- Subscription: A dropdown menu with one item highlighted.
- Resource group: A dropdown menu showing "hands-on-lab-bigdata" and a "Create new" option, both highlighted with a red box.

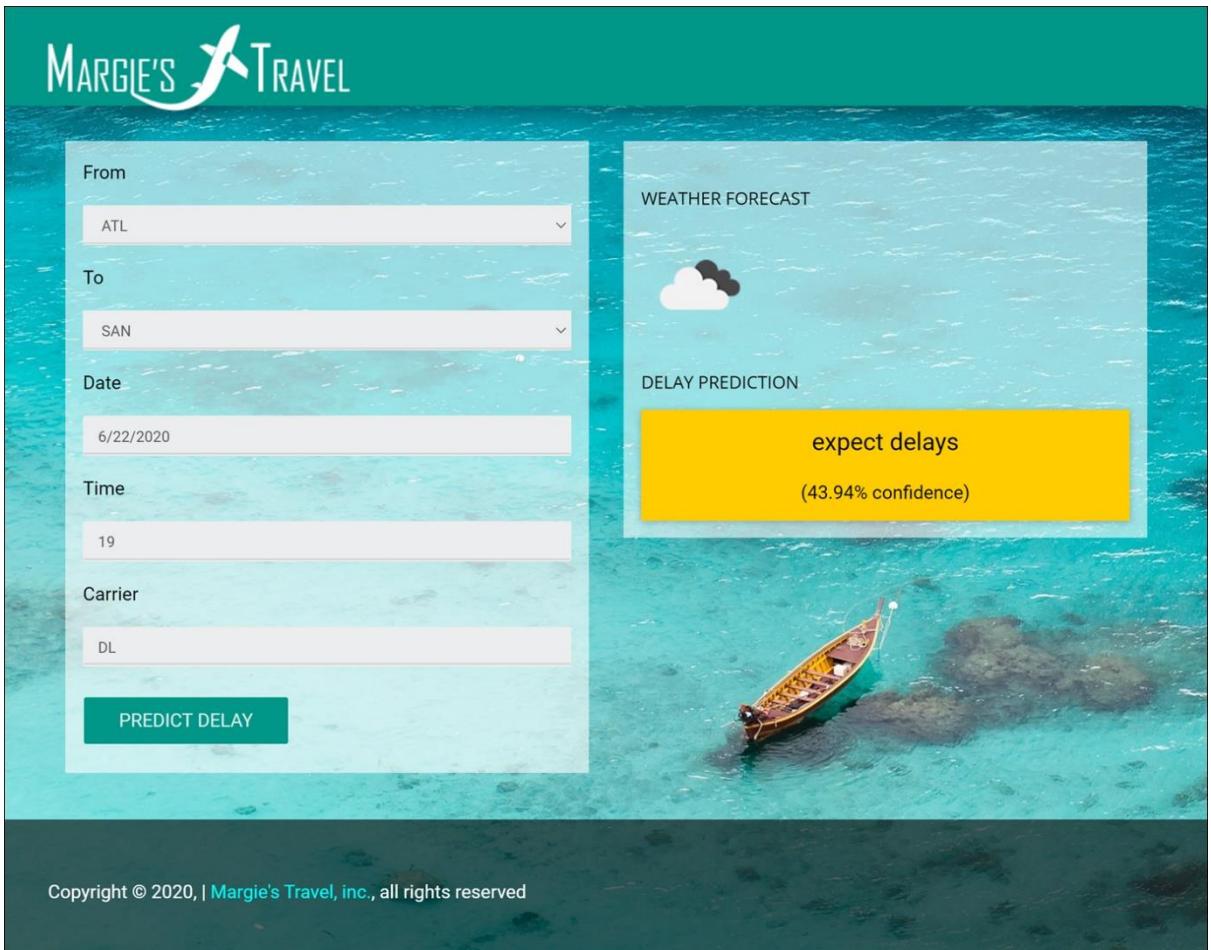
Instance details:

- Region: "(US) West US 2"
- Site Name: "[concat('webapp-', newGuid())]"
- Sku: "F1"
- ML Url: "https://adb-359520185019386.6.azuredatabricks.net/model/Delay%20..."
- Weather Api Key: "63b" (highlighted with a red box)

At the bottom, there are navigation buttons: "Review + create" (highlighted with a red box), "< Previous", and "Next : Review + create >".

- Select **Review + create (4)**, and on the following screen, select **Create**.
 - The page should begin deploying your application while showing you a status of what is currently happening.
- Note:** If you run into errors during the deployment that indicate a bad request or unauthorized, verify that the user you are logged into the portal with is either a Service Administrator or a Co-Administrator. You won't have permissions to deploy the website otherwise.
- After a short time, the deployment will complete, and you will be able to access the web site.

- Try a few different combinations of origin, destination, date, and time in the application. The information you are shown is the result of both the ML API you published, as well as information retrieved from the OpenWeather API.



- Congratulations! You have built and deployed an intelligent system to Azure.

After the hands-on lab

In this exercise, attendees will deprovision any Azure resources that were created in support of the lab.

Task 1: Delete resource group

- Using the Azure portal, navigate to the Resource group you used throughout this hands-on lab by selecting **Resource groups** in the menu.
- Search for the name of your research group and select it from the list.
- Select **Delete** in the command bar and confirm the deletion by re-typing the Resource group name and selecting **Delete**.

You should follow all steps provided *after* attending the Hands-on lab.