Big Data and Visualization

Friday, August 24, 2018 11:57 AM

Analyze weather data using Azure Machine Learning, build a data pipeline using Azure Data Factory, summarize it in HDInsight Spark, and visualize it using Power BI.

In this workshop, attendees will build an end-to-end solution to predict flight delays taking into account the weather forecast using Power BI, Azure HDInsight Spark, an Azure Machine Learning.

What You Will Learn

- Azure Data Factory
- Azure HDInsight Spark
- Azure Machine Learning
- Power BI Desktop
- Advanced Analytics

Ideal Audience

- CIOs
- VPs and Directors of Business Intelligence
- IT Managers
- Data Architects and DBAs
- Data Analysts and Data Scientists

Overview

Abstract and learning objectives

In this workshop, you will build a complete Azure Machine Learning (ML) model for predicting if an upcoming flight will experience delays, based on flight data and weather conditions. In addition, you will learn to:

- Develop a data factory pipeline for data movement
- Analyze data using Spark on HDInsight
- Build and operationalize a Machine Learning model for predictions
- Visualize Big Data and predictions using Power BI Desktop

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. The goal is to show an end-to-end solution, leveraging many of these technologies, but not necessarily doing work in every component possible. The lab architecture is below and includes:

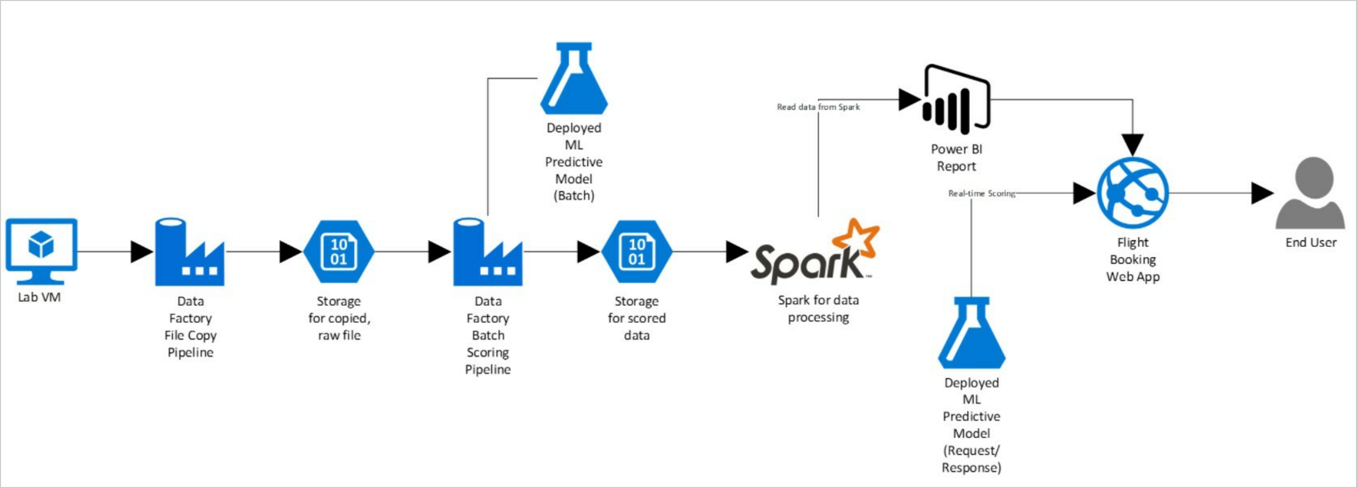
- Azure Machine Learning (Azure ML)
- Azure Data Factory (ADF)
- Azure Storage
- HDInsight Spark
- Power BI Desktop
- Azure App Service

Overview

AdventureWorks Travel (AWT) 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. In this hands-on lab, attendees will build an end-to-end solution to predict flight delays, accounting for the weather forecast.

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.


Deployed ML Predictive Model (Batch)
Read data from Spark
Power BI Report
Real-time Scoring
Lab VM
Data Factory File Copy Pipeline
Storage for copied, raw file
Data Factory Batch Scoring Pipeline
Storage for scored data
Spark for data processing
Flight Booking Web App
End User
Deployed ML Predictive Model (Request/ Response)

The solution begins with loading their historical data into blob storage using Azure Data Factory (ADF). By setting up a pipeline containing a copy activity configured to copy time partitioned source data, they could pull all their historical information, as well as ingest any future data, into Azure blob storage through a scheduled, and continuously running pipeline. Because their historical data is stored on-premises, AWT would need to install and configure an Azure Data Factory Integration Runtime (formerly known as a Data Management Gateway). Azure Machine Learning (Azure ML) would be used to develop a two-class classification machine learning model, which would then be operationalized as a Predictive Web Service using ML Studio. After operationalizing the ML model, a second ADF pipeline, using a Linked Service pointing to Azure ML's Batch Execution API and an AzureMLBatchExecution activity, would be used to apply the operational model to data as it is moved to the proper location in Azure storage. The scored data in Azure storage can be explored and prepared using Spark SQL on HDInsight, and the results visualized using a map visualization in Power BI.

Time Estimate: 5.0 hours

Requirements

Setup Requirements

- A corporate email address (e.g., your @microsoft.com email)
- Microsoft Azure Subscription must be pay-as-you-go or MSDN

Additional Requirements

You will need a subscription to Microsoft Azure. Please see the next page for how to create a trial subscription.

Azure Registration

Azure

We need an active Azure subscription in order to perform this workshop. There are a few ways to accomplish this. If you already have an active Azure subscription, you can skip the remainder of this page. Otherwise, you'll either need to use an Azure Pass or create a trial account. The instructions for both are below.

Azure Pass

If you've been provided with a voucher, formally known as an Azure Pass, then you can use that to create a subscription. In order to use the Azure Pass, direct your browser to https://www.microsoftazurepass.com and, following the prompts, use the code provided to create your subscription.

Trial Subscription

Direct your browser to https://azure.microsoft.com/en-us/free/ and begin by clicking on the green button that reads **Start free**.

1. In the first section, complete the form in its entirety. Make sure you use your *real* email address for the important notifications.

2. In the second section, enter a *real* mobile phone number to receive a text verification number. Click send message and re-type the received code.

3. Enter a valid credit card number. **NOTE:** You will *not* be charged. This is for verification of identity only in order to comply with federal regulations. Your account statement may see a temporary hold of $1.00 from Microsoft, but, again, this is for verification only and will "fall off" your account within 2-3 banking days.

4. Agree to Microsoft's Terms and Conditions and click **Sign Up**.

This may take a minute or two, but you should see a welcome screen informing you that your subscription is ready. The Azure subscription is good for up to $200 of resources for 30 days. After 30 days, your subscription (and resources) will be suspended unless you convert your trial subscription to a paid one. And, should you choose to do so, you can elect to use a different credit card than the one you just entered.

Congratulations! You've now created an Azure tenant and subscription!

Setup

Exercise 0: Before the workshop

Duration: 45 mins

Synopsis: In this exercise, you will set up your environment for use in the rest of the hands-on lab.

You should follow all the steps provided in this section to prepare your environment before attending the hands-on lab.

Task 1: Deploy HDInsight cluster, Azure ML, and Storage Accounts to Azure

1. Click the **Deploy to Azure** link below, and you will be taken to the Azure portal, and presented with a form for a new custom deployment (which uses an Azure Resource Management (ARM) template from a GitHub repository). You will be presented with a blade to provide some custom parameters as show in the screenshot below.

   Deploy to Azure

2. In the Custom deployment blade that appears, enter the following values:

   - Subscription: Select your subscription

   - Resource group: Use and existing Resource group, or create a new one by entering a unique name, such as "bigdatalab-[your intials or first name]".

   - Location: Select a location for the Resource group. Recommend using East US, East US 2, West Central US, or West US 2, as some resources, such as Data Factory, are only available in those regions.

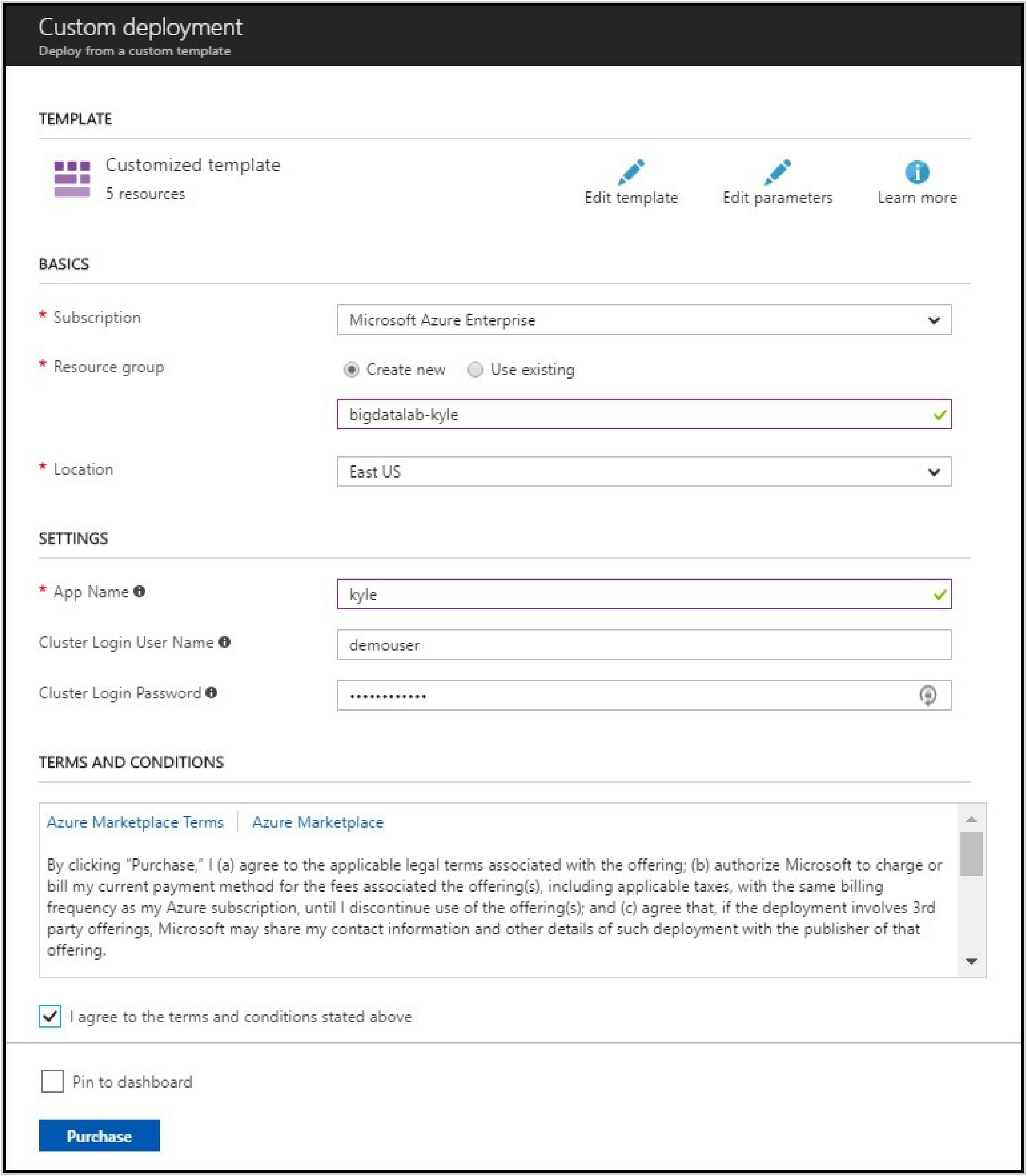
   - App name: Enter a unique name, such as your initials or first name. This value must be between 3 and 10 characters long, and should not contain any special characters. Note the name, as you will need to use it in your Lab VM deployment in Task 3 as well.

   - Cluster Login User Name: Enter a name, or accept the default. Note all references to this in the lab use the default user name, demouser, so if you change it, please note it for future reference throughout the lab.

- Cluster Login Password: Enter a password, or accept the default. Note all references to this in the lab use the default password, **Password.1!!**, so if you change it, please note it for future reference throughout the lab.

- Check the box to agree to the terms.

- Select Purchase.



Custom deployment
Deploy from a custom template

TEMPLATE

Customized template
5 resources

BASICS

* Subscription	Microsoft Azure Enterprise
* Resource group	● Create new ○ Use existing bigdatalab-kyle ✓
* Location	East US

SETTINGS

* App Name	kyle ✓
Cluster Login User Name	demouser
Cluster Login Password	••••••••••••

TERMS AND CONDITIONS

Azure Marketplace Terms | Azure Marketplace

By clicking "Purchase," I (a) agree to the applicable legal terms associated with the offering; (b) authorize Microsoft to charge or bill my current payment method for the fees associated the offering(s), including applicable taxes, with the same billing frequency as my Azure subscription, until I discontinue use of the offering(s); and (c) agree that, if the deployment involves 3rd party offerings, Microsoft may share my contact information and other details of such deployment with the publisher of that offering.

☑ I agree to the terms and conditions stated above

□ Pin to dashboard

3. The deployment will take about 15 minutes to complete.

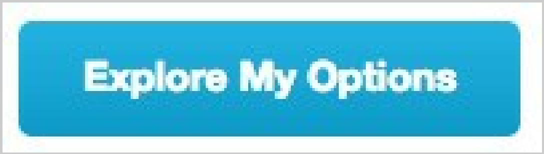
4. Wait for the deployment to complete before attempting to deploy the Lab Virtual Machine in Task 3, as it depends on the Virtual Network created by this deployment. In the meantime, you can move on to the next task, Task 2, while this deployment is ongoing.

Task 2: Register for a trial API account at WeatherUnderground.com

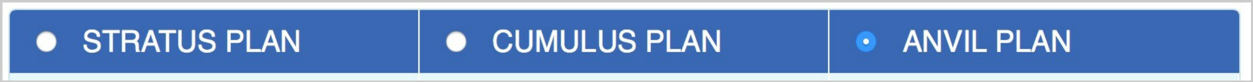
To retrieve the 10-day hourly weather forecast, you will use an API from WeatherUnderground.com. There is a free developer version that provides you access to the API you need for this hands-on lab.

1. Navigate to http://www.wunderground.com/weather/api/.

2. Select Explore My Options.

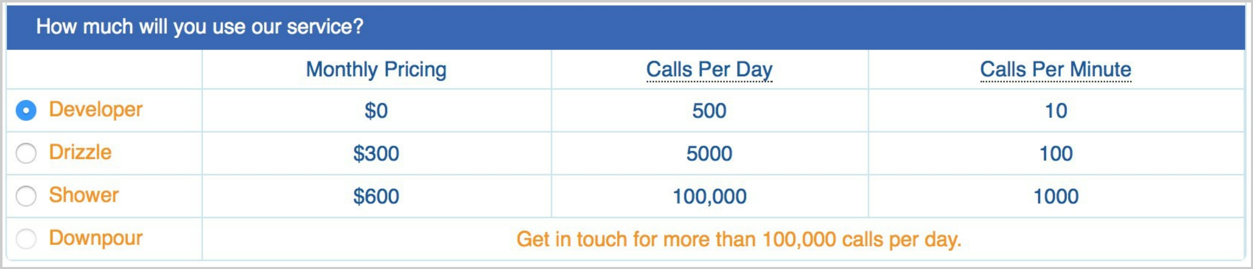


3. On the Get Your API Key page, select **Anvil Plan**.



○ STRATUS PLAN ○ CUMULUS PLAN ● ANVIL PLAN

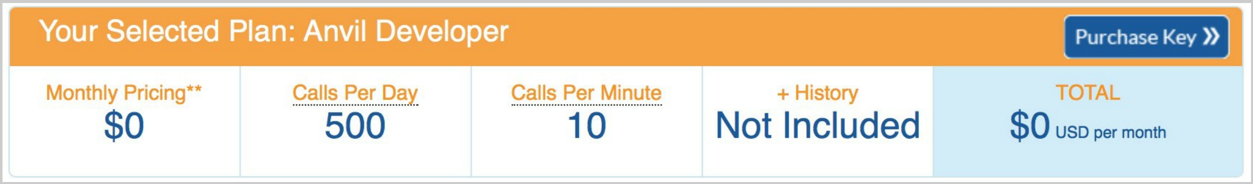
4. Scroll down until you see the area titled How much will you use our service? Ensure **Developer** is selected.



How much will you use our service?

	Monthly Pricing	Calls Per Day	Calls Per Minute
● Developer	$0	500	10
○ Drizzle	$300	5000	100
○ Shower	$600	100,000	1000
○ Downpour	Get in touch for more than 100,000 calls per day.		

5. Select **Purchase Key**.

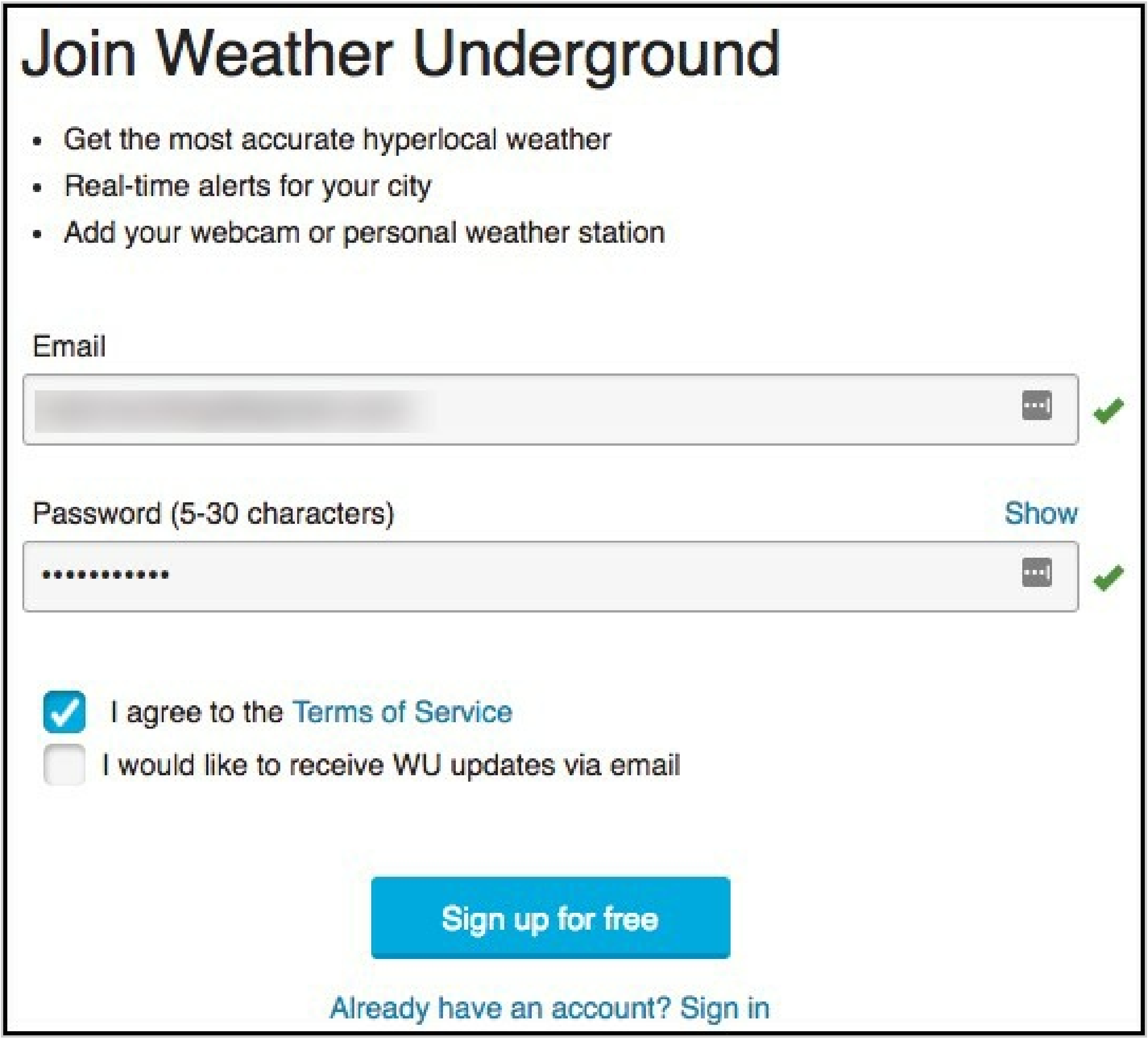


Your Selected Plan: Anvil Developer

Monthly Pricing**	Calls Per Day	Calls Per Minute	+ History	TOTAL
$0	500	10	Not Included	$0 USD per month

6. Complete the Create an Account form by providing your email address and a password, and agreeing to the terms.

7. Select Sign up for free.



Join Weather Underground

- Get the most accurate hyperlocal weather
- Real-time alerts for your city
- Add your webcam or personal weather station

Email

[redacted] ✓

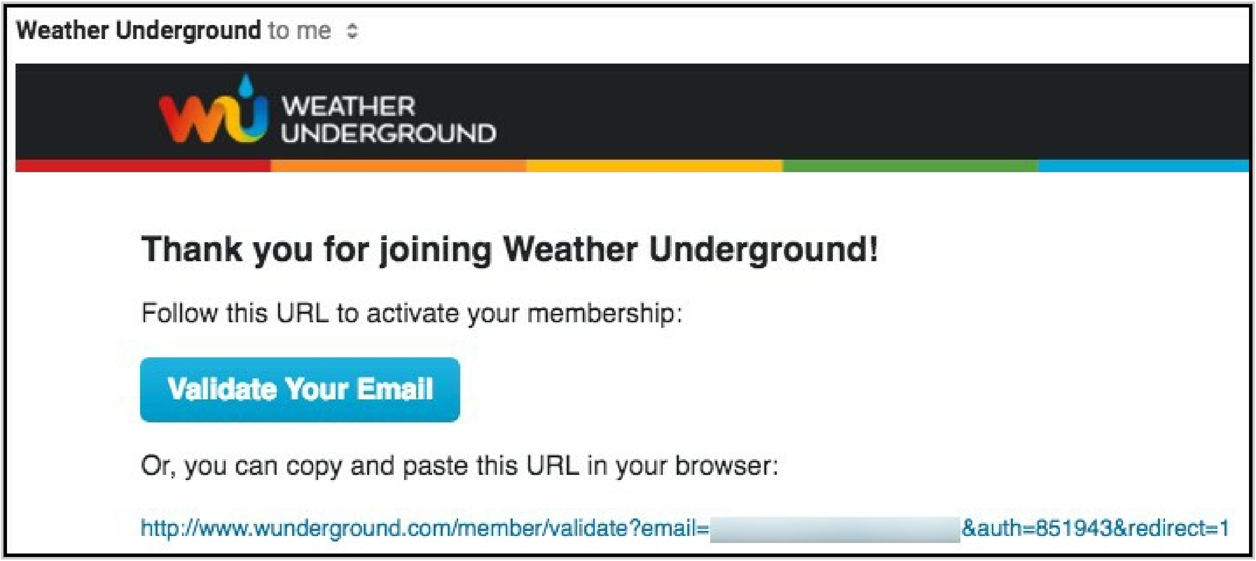
Password (5-30 characters) Show

•••••••••••• ✓

☑ I agree to the Terms of Service

☐ I would like to receive WU updates via email

8. In a few moments you should receive a confirmation email at the email address you provided. Select the **Validate Your Email** link found within the email.



Weather Underground to me

Thank you for joining Weather Underground!

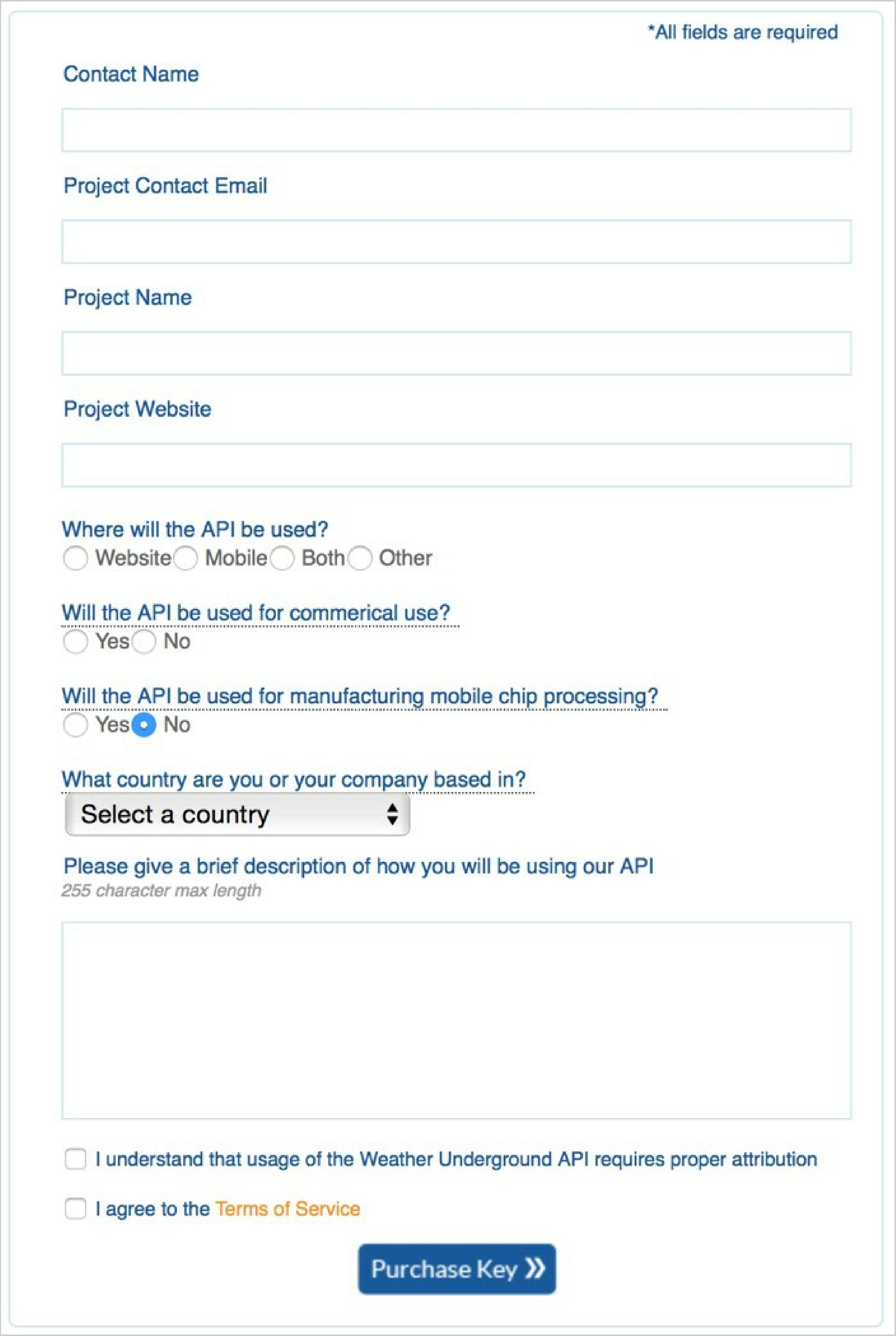
Follow this URL to activate your membership:

Or, you can copy and paste this URL in your browser:

http://www.wunderground.com/member/validate?email=[redacted]&auth=851943&redirect=1

9. Once you have validated your email, go back to the Get Your API Key page, re-select Anvil and select Purchase Key.

10. Complete the brief contact form. When answering where will the API be used, select **Website**. For Will the API be used for commercial use, select **No**. Select **Purchase Key**.



*All fields are required

Contact Name

Project Contact Email

Project Name

Project Website

Where will the API be used?
○ Website ○ Mobile ○ Both ○ Other

Will the API be used for commerical use?
○ Yes ○ No

Will the API be used for manufacturing mobile chip processing?
○ Yes ● No

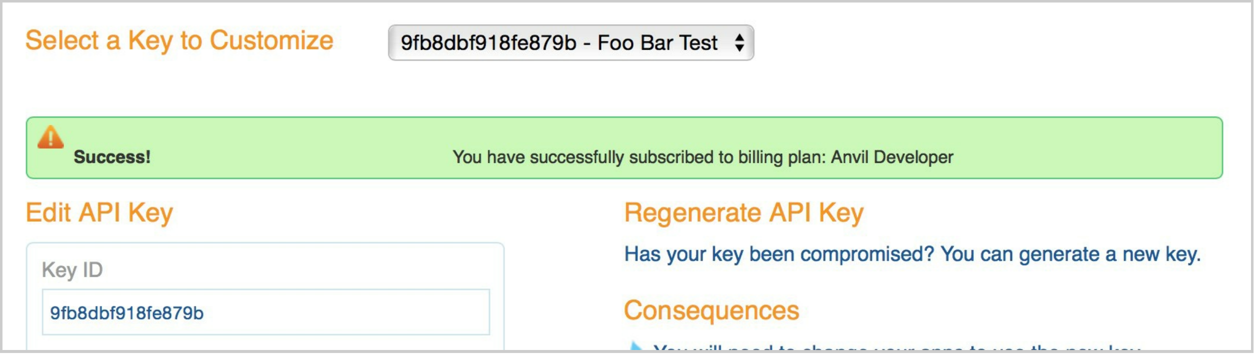
What country are you or your company based in?
Select a country

Please give a brief description of how you will be using our API
255 character max length

□ I understand that usage of the Weather Underground API requires proper attribution

□ I agree to the Terms of Service

11. You should be taken to a page that displays your key, similar to the following:



Select a Key to Customize

9fb8dbf918fe879b - Foo Bar Test

Success! You have successfully subscribed to billing plan: Anvil Developer

Edit API Key

Key ID

9fb8dbf918fe879b

Regenerate API Key

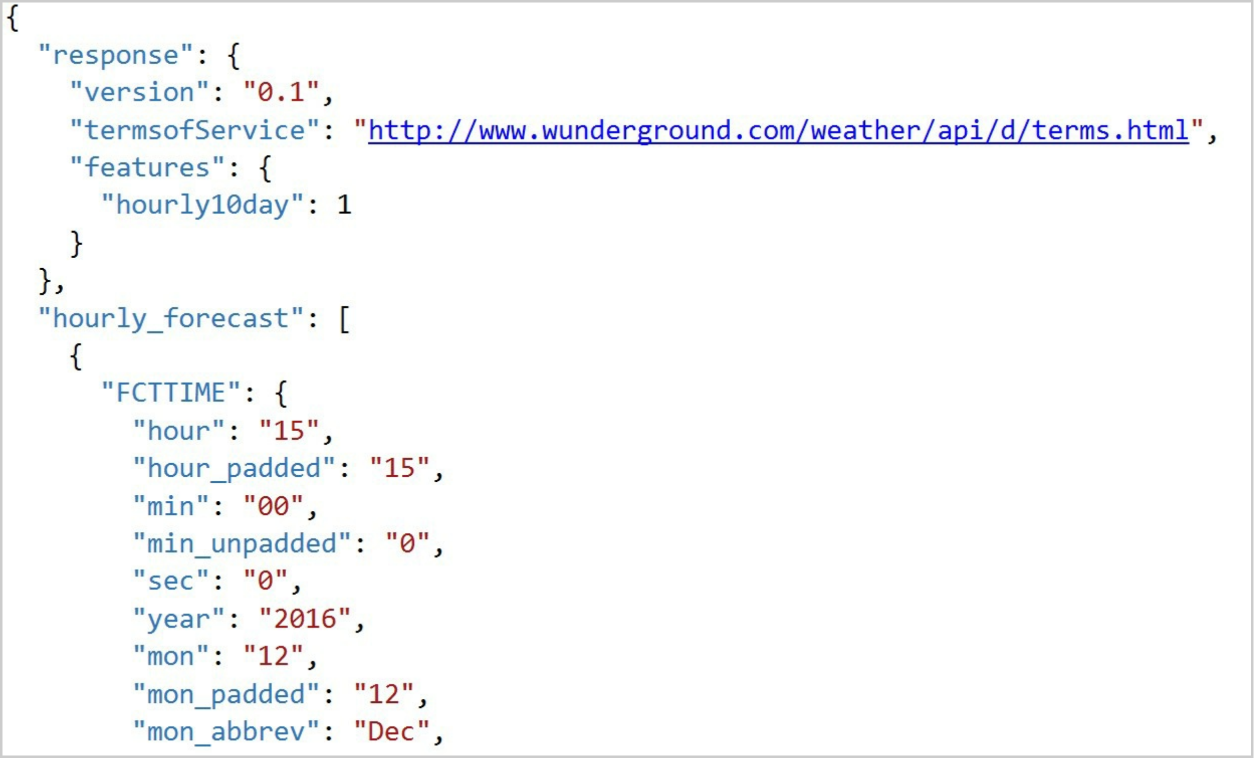
Has your key been compromised? You can generate a new key.

Consequences

12. Take note of your API Key. It is available from the text box labeled **Key ID**.

13. To verify that your API Key is working, **modify the following URL** to include your API Key: **http://api.wunderground.com/api//hourly10day/q/SEATAC.json**.

14. Open your modified link in a browser, you should get a JSON result showing the 10-day, hourly weather forecast for the Seattle-Tacoma International Airport



```
{
  "response": {
    "version": "0.1",
    "termsofService": "http://www.wunderground.com/weather/api/d/terms.html",
    "features": {
      "hourly10day": 1
    }
  },
  "hourly_forecast": [
    {
      "FCTTIME": {
        "hour": "15",
        "hour_padded": "15",
        "min": "00",
        "min_unpadded": "0",
        "sec": "0",
        "year": "2016",
        "mon": "12",
        "mon_padded": "12",
        "mon_abbrev": "Dec",
```

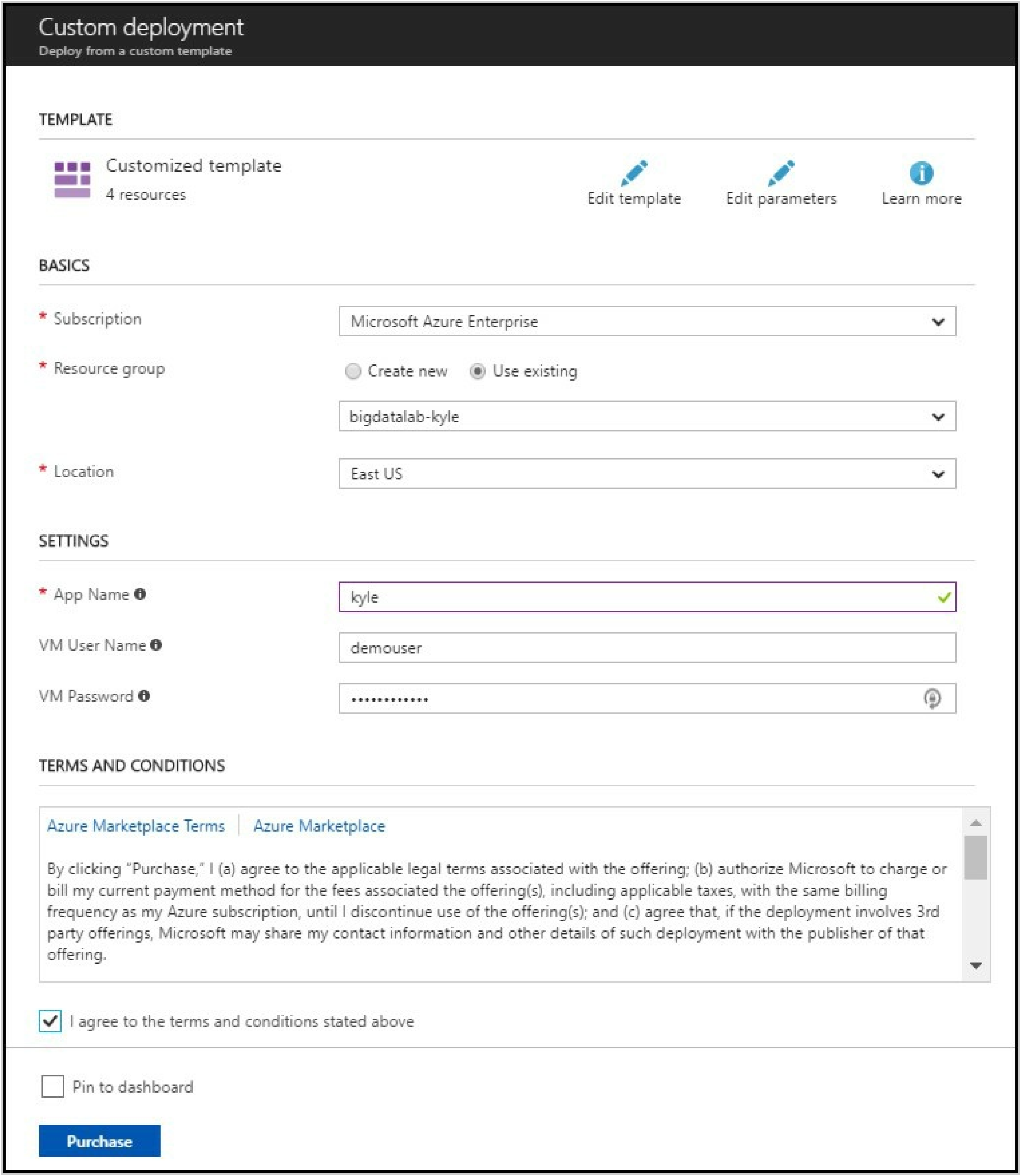
Task 3: Deploy Lab Virtual Machine (Lab VM) to Azure

1. Click the **Deploy to Azure** link below, and you will be taken to the Azure portal, and presented with a form for a new custom deployment (which uses an ARM template from a GitHub repository). You will be presented with a blade to provide some custom parameters as show in the screenshot below.

   Deploy to Azure

2. In the Custom deployment blade that appears, enter the following values:

    - Subscription: Select your subscription
    - Resource group: Choose **Use Existing**, and select the same resource group you used when deploying your HDInsight cluster and Azure ML workspace, above.
    - Location: The location should be automatically selected to be the same as your Resource Group.
    - App name: **IMPORTANT:** You must enter the **same App name** you used in the deployment above in Task 1.
    - VM User Name: Enter a name, or accept the default. Note all references to this in the lab use the default user name, **demouser**, so if you change it, please note it for future reference throughout the lab.
    - VM Password: Enter a password, or accept the default. Note all references to this in the lab use the default password, **Password.1!!**, so if you change it, please not it for future reference throughout the lab.
    - Check the box to agree to the terms.
    - Select **Purchase**.



Custom deployment
Deploy from a custom template

TEMPLATE

Customized template
4 resources

BASICS

* Subscription: Microsoft Azure Enterprise

* Resource group: ○ Create new ● Use existing

bigdatalab-kyle

* Location: East US

SETTINGS

* App Name: kyle

VM User Name: demouser

VM Password: ••••••••••••

TERMS AND CONDITIONS

Azure Marketplace Terms | Azure Marketplace

By clicking "Purchase," I (a) agree to the applicable legal terms associated with the offering; (b) authorize Microsoft to charge or bill my current payment method for the fees associated the offering(s), including applicable taxes, with the same billing frequency as my Azure subscription, until I discontinue use of the offering(s); and (c) agree that, if the deployment involves 3rd party offerings, Microsoft may share my contact information and other details of such deployment with the publisher of that offering.

☑ I agree to the terms and conditions stated above

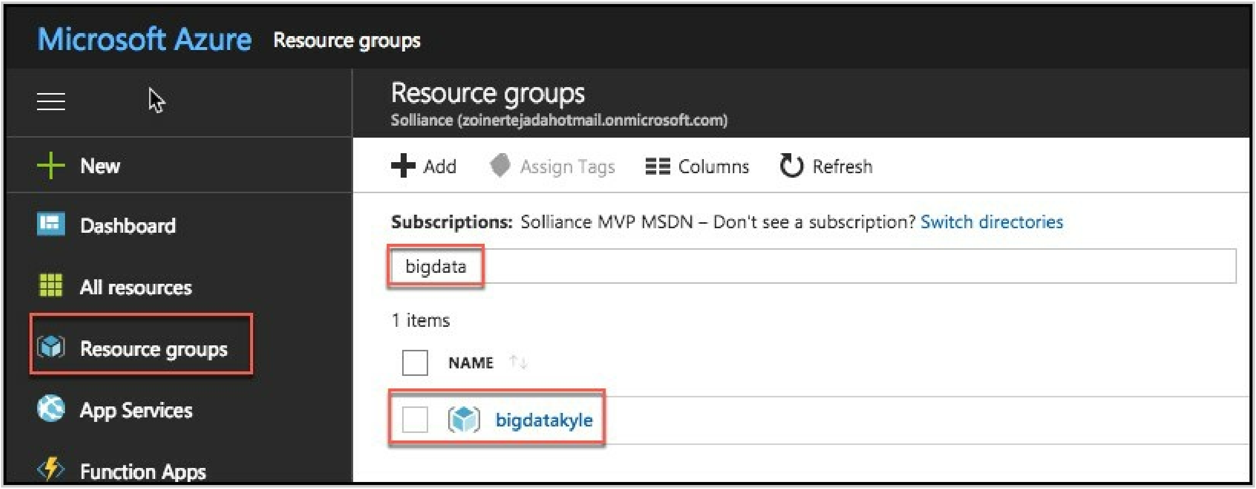
□ Pin to dashboard

3. The deployment will take about 10 minutes to complete.

Task 4: Install Power BI Desktop on the Lab VM

1. Connect to the Lab VM. (If you are already connected to your Lab VM, skip to Step 7.

2. From the left side menu in the Azure portal, click on **Resource groups**, then enter your resource group name into the filter box, and select it from the list.



Resource groups
Solliance (zoinertejadahotmail.onmicrosoft.com)

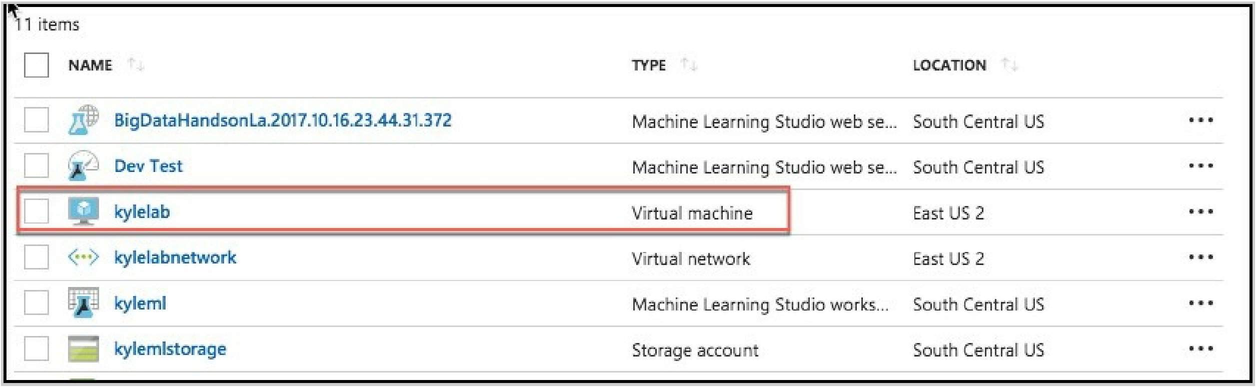
Subscriptions: Solliance MVP MSDN – Don't see a subscription? Switch directories

bigdata

1 items

NAME
bigdatakyle

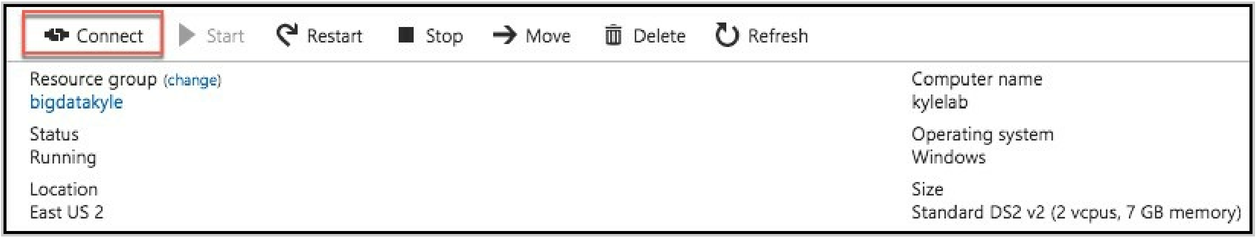
3. Next, select your lab virtual machine from the list.



11 items

NAME	TYPE	LOCATION	
BigDataHandsonLa.2017.10.16.23.44.31.372	Machine Learning Studio web se...	South Central US	•••
Dev Test	Machine Learning Studio web se...	South Central US	•••
kylelab	Virtual machine	East US 2	•••
kylelabnetwork	Virtual network	East US 2	•••
kyleml	Machine Learning Studio works...	South Central US	•••
kylemlstorage	Storage account	South Central US	•••

4. On your Lab VM blade, select Connect from the top menu.



Resource group (change)
bigdatakyle

Status
Running

Location
East US 2

Computer name
kylelab

Operating system
Windows

Size
Standard DS2 v2 (2 vcpus, 7 GB memory)

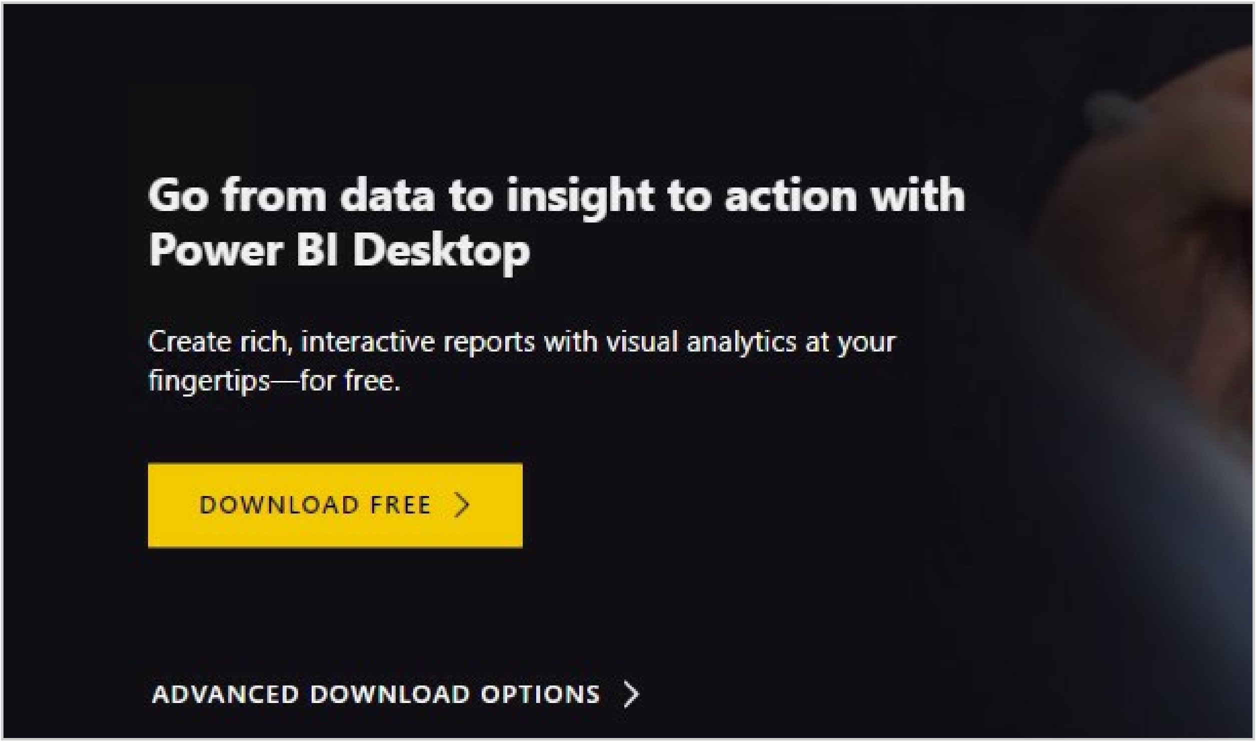
5. Download and open the RDP file.

6. Select **Connect**, and enter the following credentials (or the non-default credentials if you changed them):

- User name: demouser
- Password: Password.1!!

7. In a web browser on the Lab VM navigate to the Power BI Desktop download page (https://powerbi.microsoft.com/en-us/desktop/).

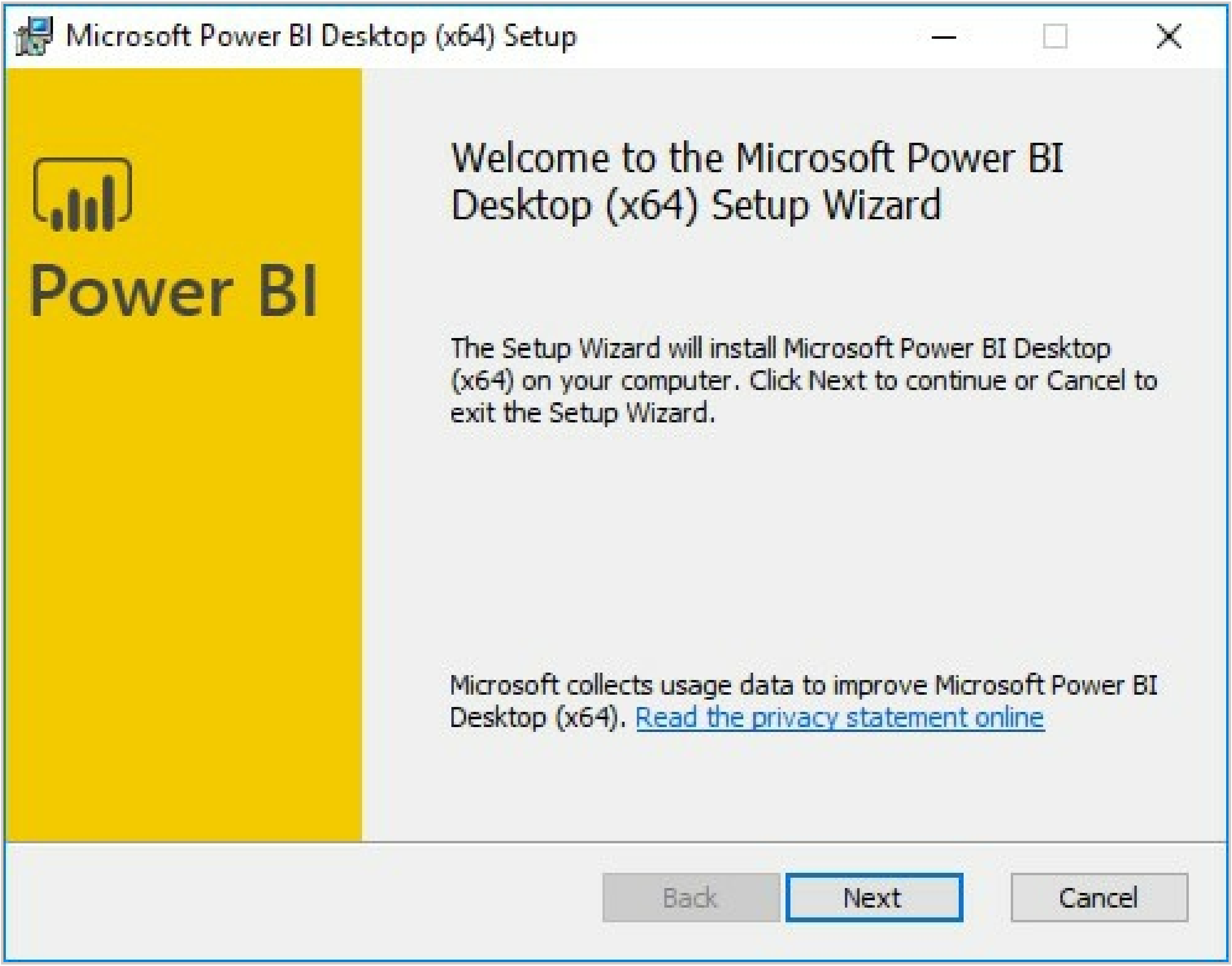
8. Select the **Download Free** link in the middle of the page



Go from data to insight to action with Power BI Desktop

Create rich, interactive reports with visual analytics at your fingertips—for free.

9. Run the installer.
10. Select Next on the welcome screen.

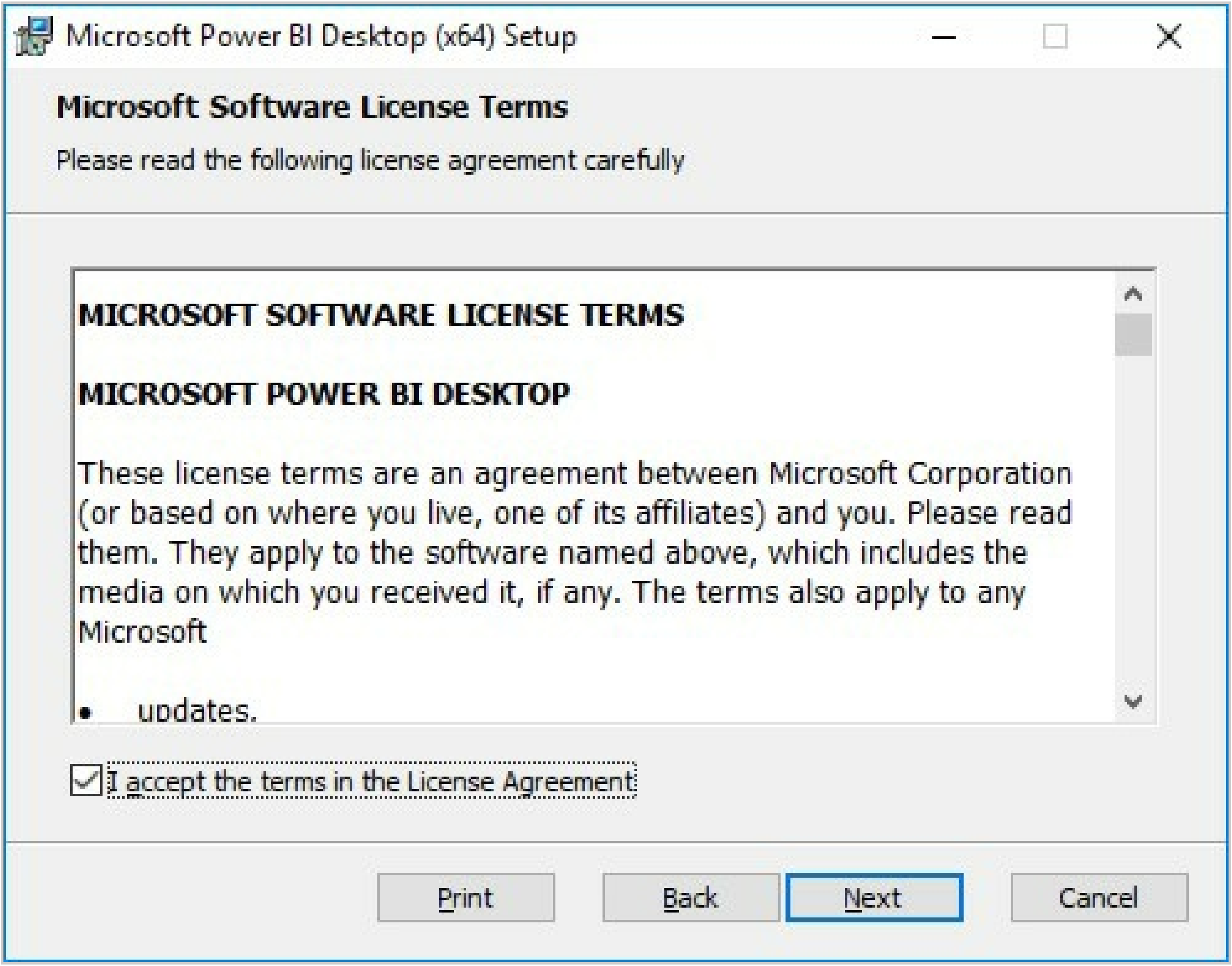


Welcome to the Microsoft Power BI Desktop (x64) Setup Wizard

The Setup Wizard will install Microsoft Power BI Desktop (x64) on your computer. Click Next to continue or Cancel to exit the Setup Wizard.

Microsoft collects usage data to improve Microsoft Power BI Desktop (x64). Read the privacy statement online

11. Accept the license agreement, and select **Next**.



Microsoft Software License Terms

Please read the following license agreement carefully

MICROSOFT SOFTWARE LICENSE TERMS

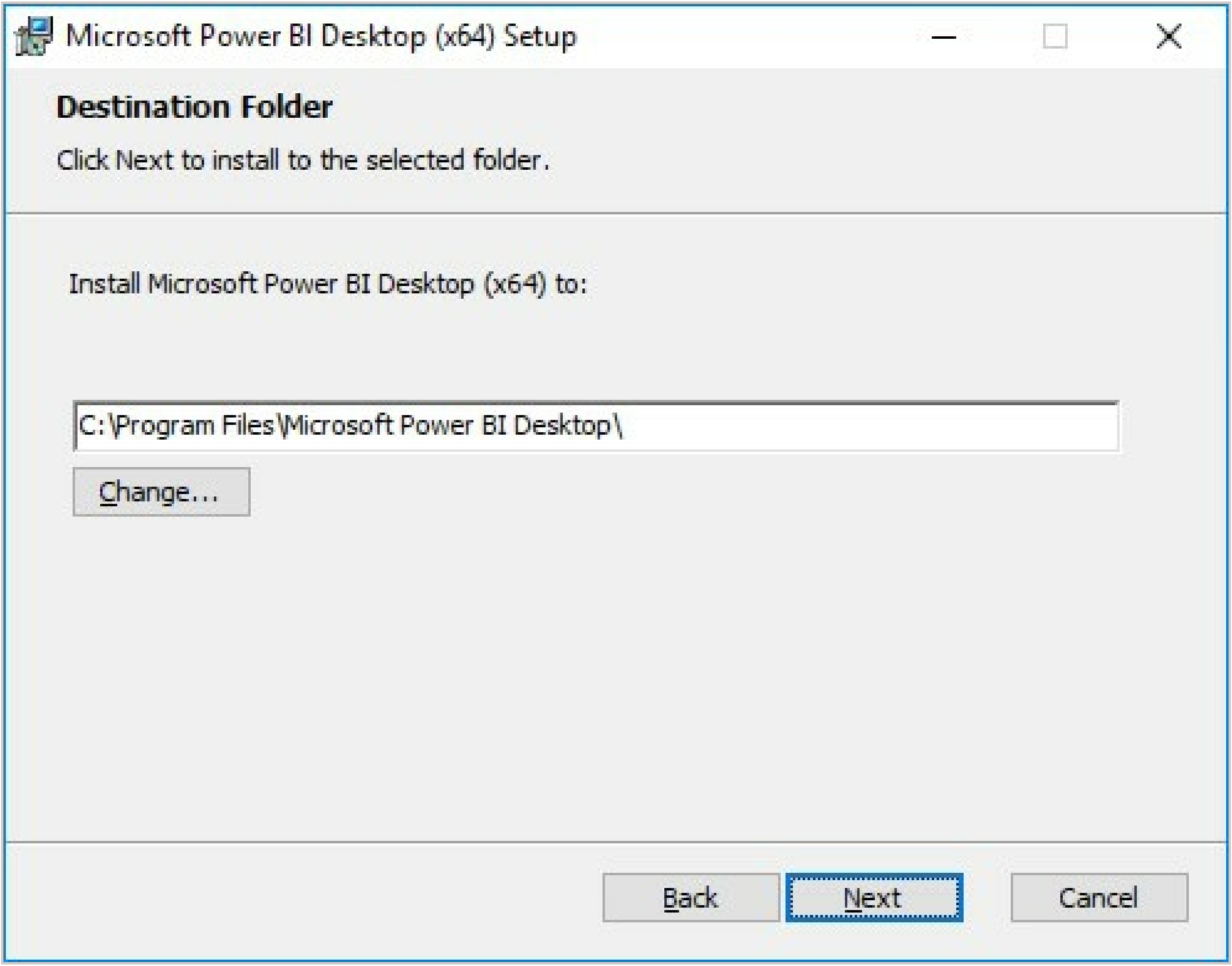
MICROSOFT POWER BI DESKTOP

These license terms are an agreement between Microsoft Corporation (or based on where you live, one of its affiliates) and you. Please read them. They apply to the software named above, which includes the media on which you received it, if any. The terms also apply to any Microsoft

- updates,

☑ I accept the terms in the License Agreement

12. Leave the default destination folder, and select **Next**.



Microsoft Power BI Desktop (x64) Setup

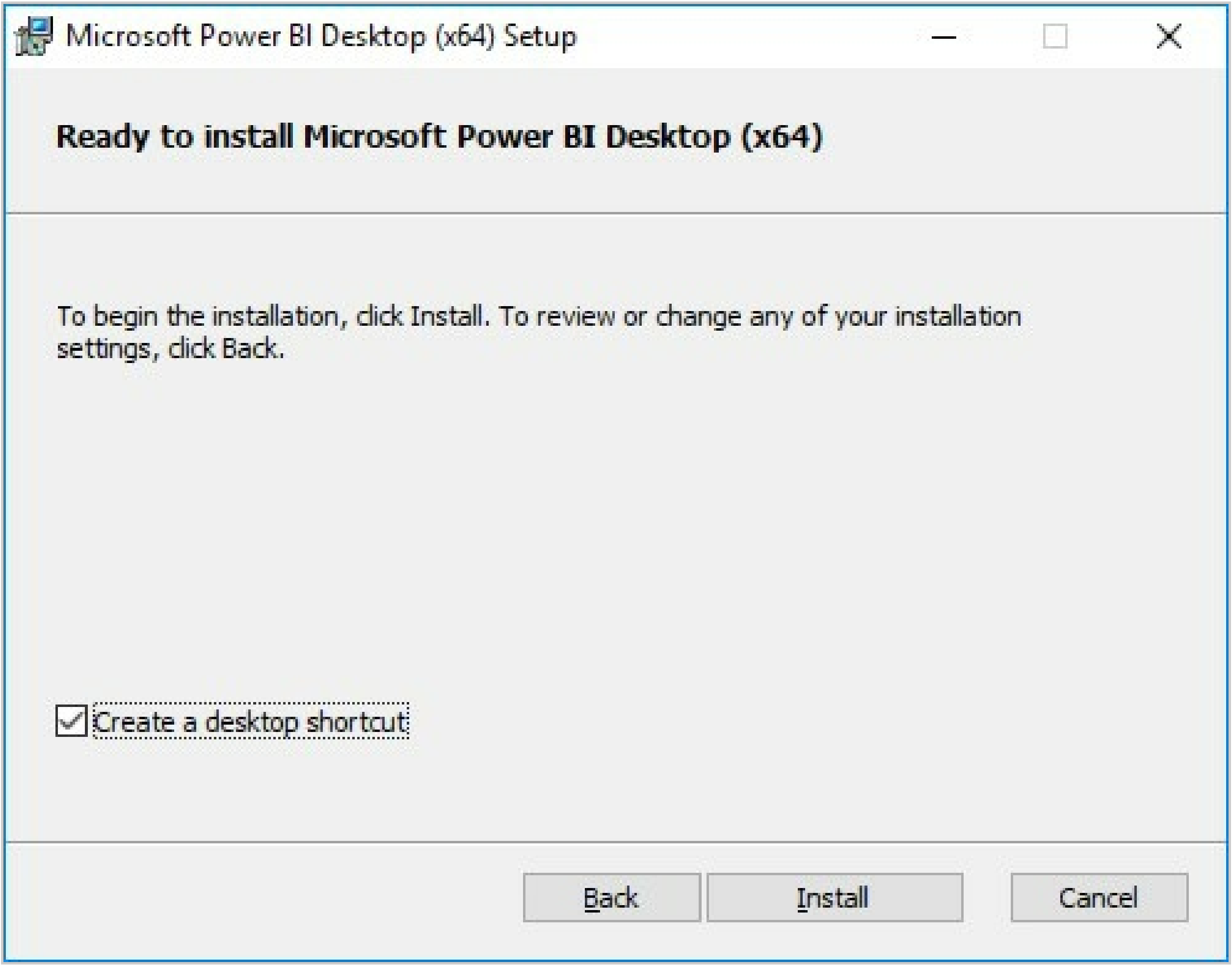
Destination Folder

Click Next to install to the selected folder.

Install Microsoft Power BI Desktop (x64) to:

C:\Program Files\Microsoft Power BI Desktop\

13. Make sure the **Create a desktop shortcut** box is checked, and select **Install**.



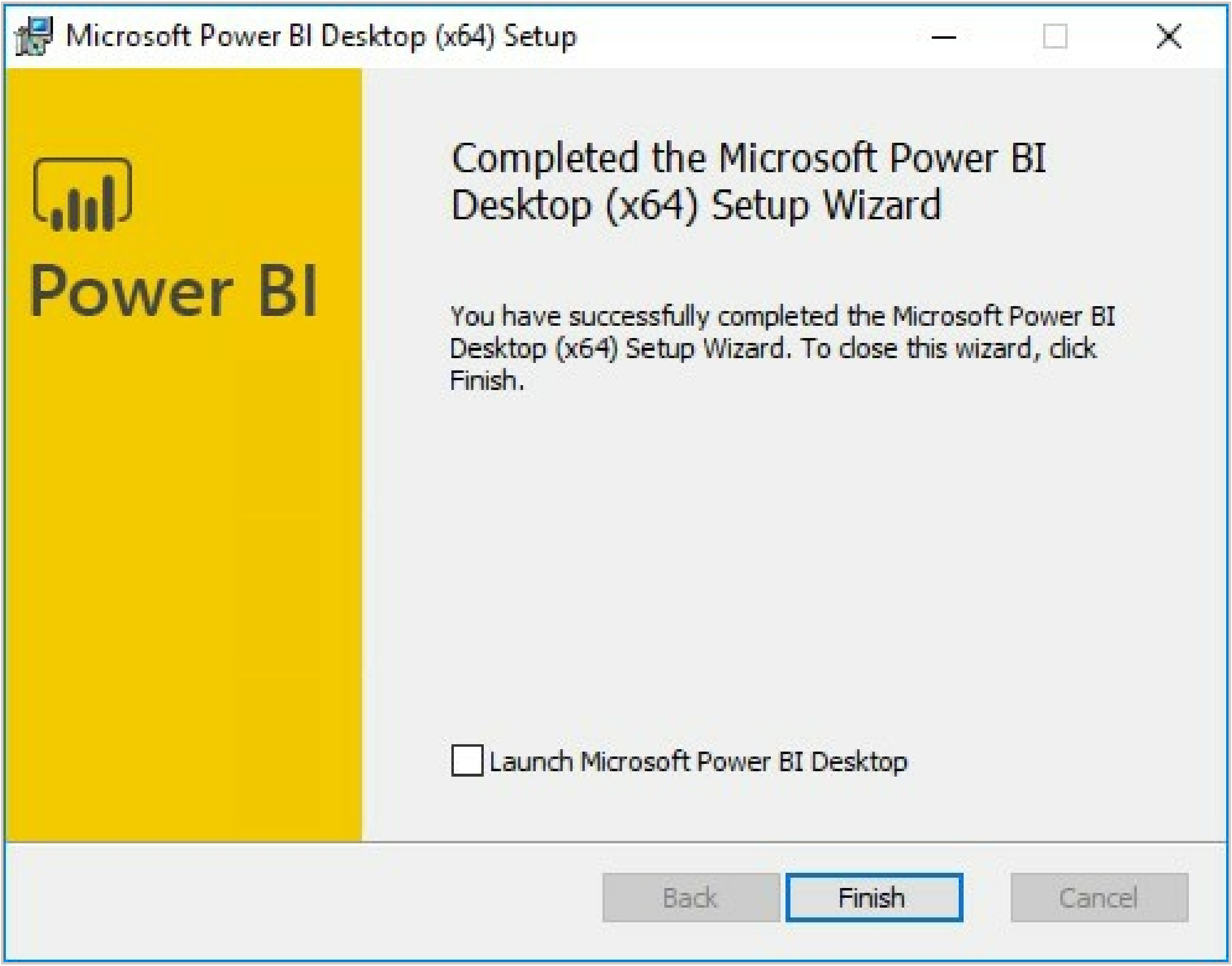
Microsoft Power BI Desktop (x64) Setup

Ready to install Microsoft Power BI Desktop (x64)

To begin the installation, click Install. To review or change any of your installation settings, click Back.

☑ Create a desktop shortcut

14. **Uncheck** Launch Microsoft Power BI Desktop, and select **Finish**.



Completed the Microsoft Power BI Desktop (x64) Setup Wizard

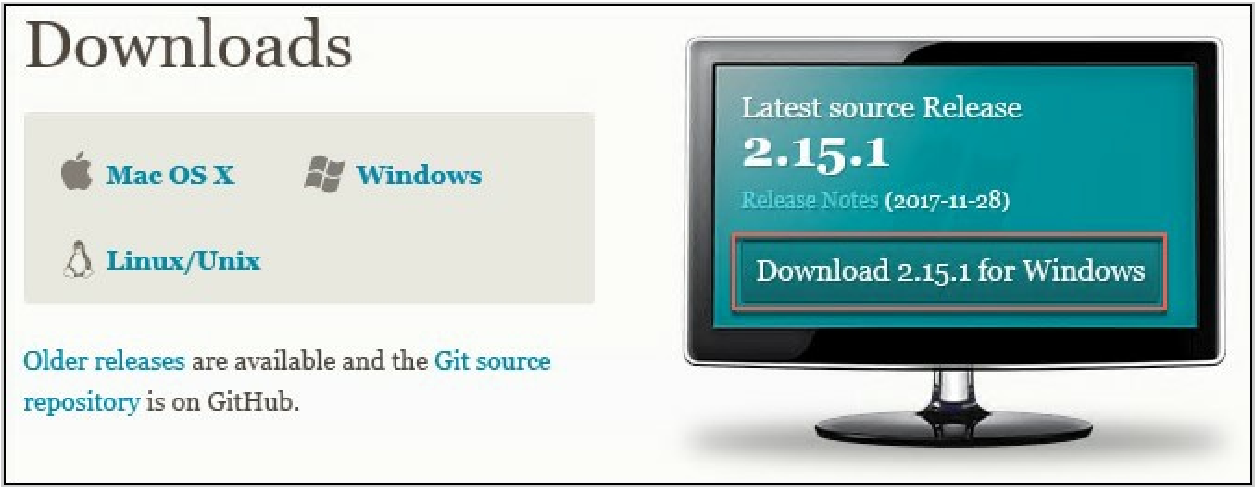
You have successfully completed the Microsoft Power BI Desktop (x64) Setup Wizard. To close this wizard, click Finish.

□ Launch Microsoft Power BI Desktop

Task 5: Install an SSH client

In this task, you will download, and install the Git Bash SSH client. This will be used to interact with the HDInsight cluster.

1. On your Lab VM, open a browser, and navigate to https://git-scm.com/downloads to download Git Bash.

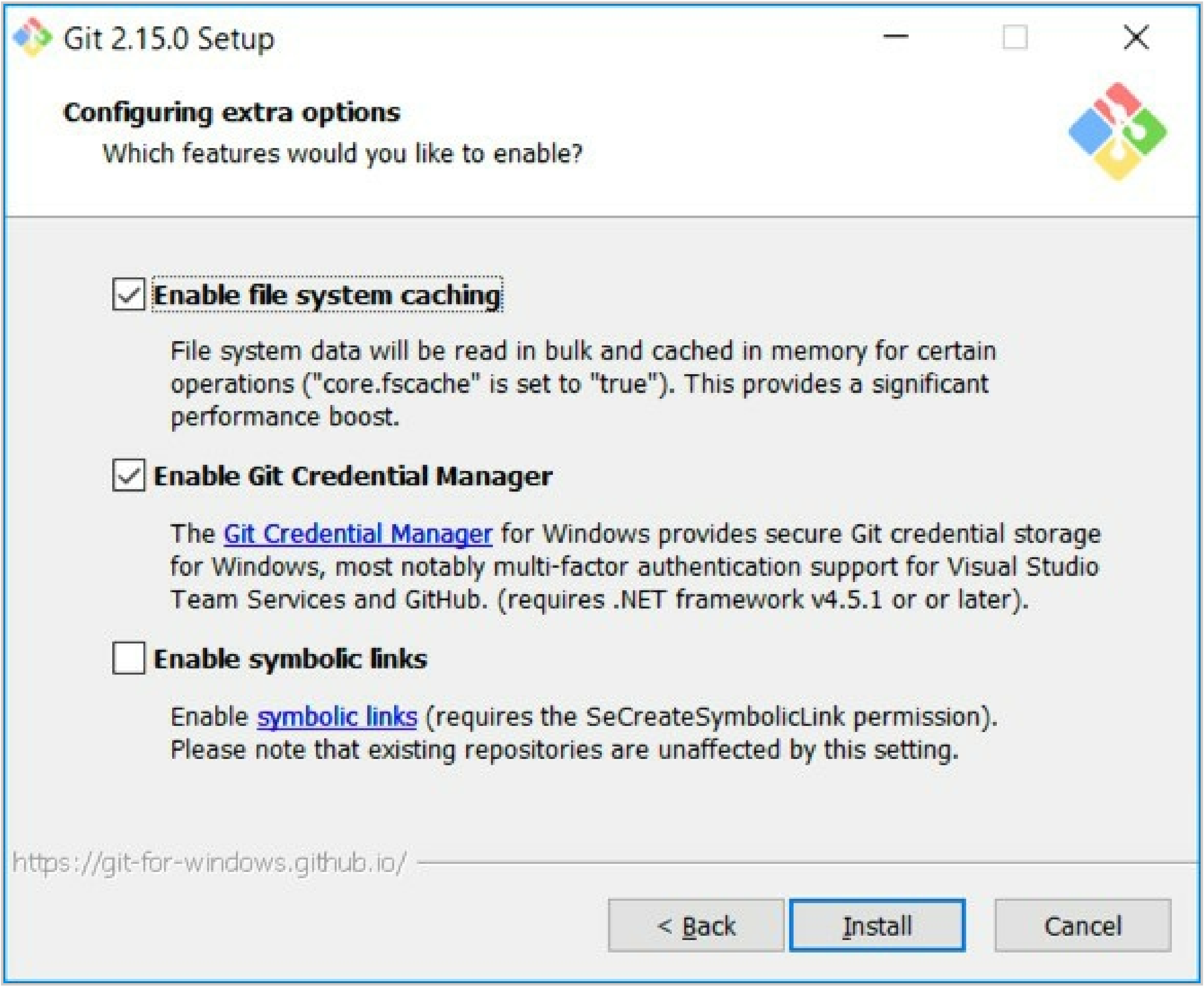


Older releases are available and the Git source repository is on GitHub.

Latest source Release 2.15.1 Release Notes (2017-11-28)

2. Select the **Download 2.xx.x for Windows** button.

3. Run the downloaded installer, selecting **Next** on each screen to accept the defaults.
4. On the last screen, select **Install** to complete the installation.



Git 2.15.0 Setup

Configuring extra options

Which features would you like to enable?

☑ **Enable file system caching**

File system data will be read in bulk and cached in memory for certain operations ("core.fscache" is set to "true"). This provides a significant performance boost.

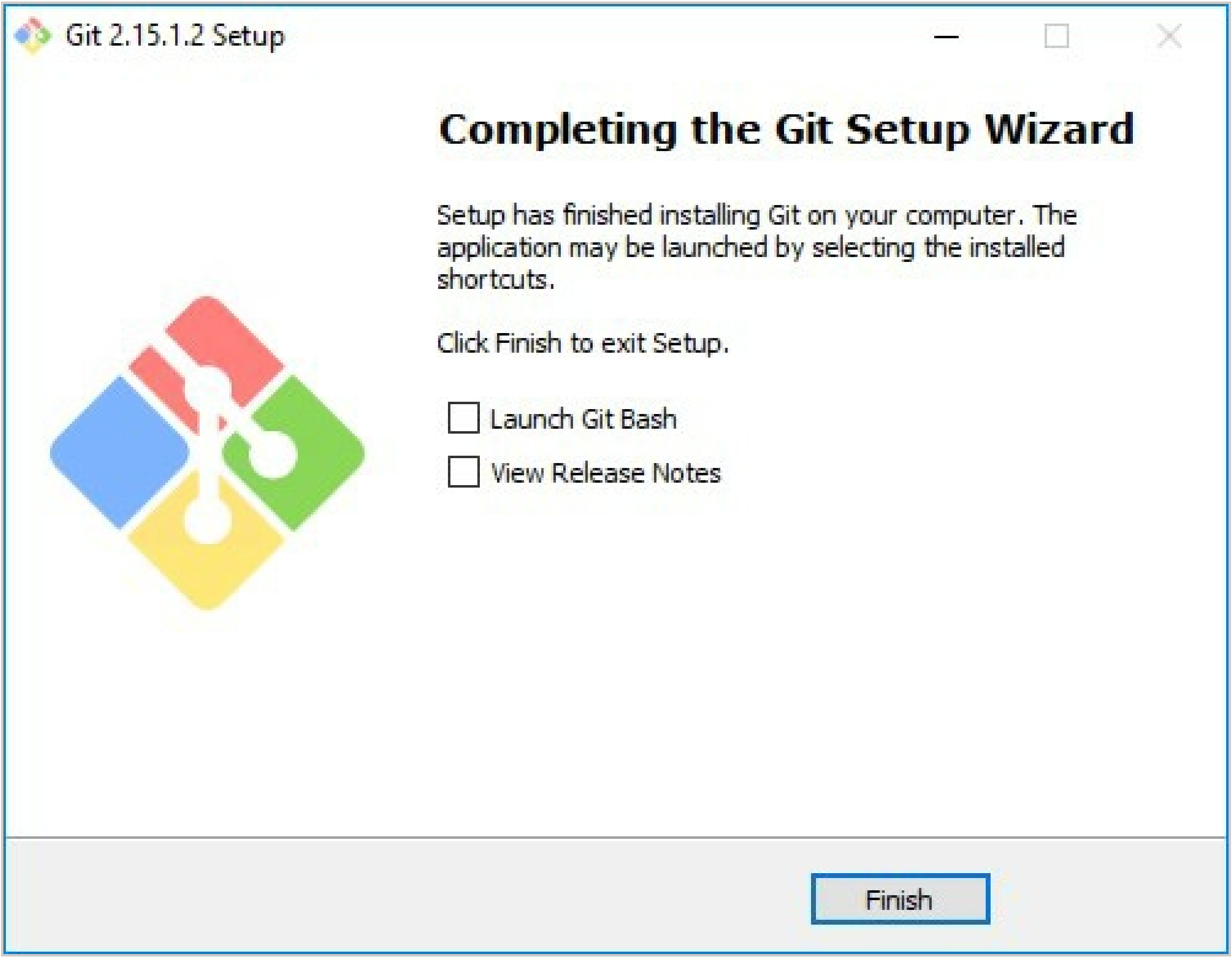
☑ **Enable Git Credential Manager**

The Git Credential Manager for Windows provides secure Git credential storage for Windows, most notably multi-factor authentication support for Visual Studio Team Services and GitHub. (requires .NET framework v4.5.1 or or later).

☐ **Enable symbolic links**

Enable symbolic links (requires the SeCreateSymbolicLink permission). Please note that existing repositories are unaffected by this setting.

5. When the install is complete, **uncheck View Release Notes**, and select **Finish**.



Completing the Git Setup Wizard

Setup has finished installing Git on your computer. The application may be launched by selecting the installed shortcuts.

Click Finish to exit Setup.

□ Launch Git Bash

□ View Release Notes

Build a ML Model

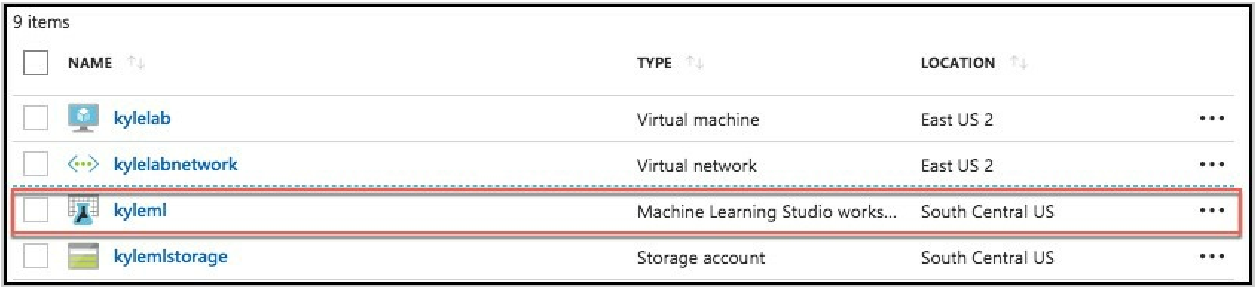
Exercise 1: Build a Machine Learning Model

Duration: 60 minutes

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

Task 1: Navigate to Machine Learning Studio

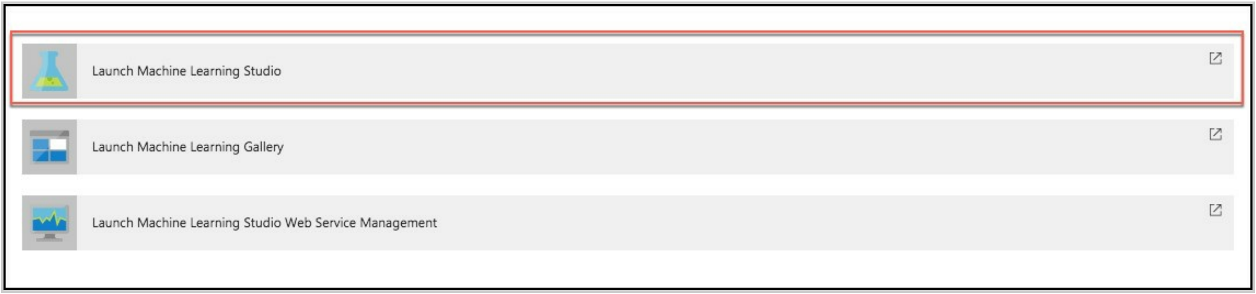
1. In a browser, go to the Azure portal (https://portal.azure.com), and navigate to your Machine Learning Studio workspace under the Resource Group you created when completing the prerequisites for this hands-on lab.



9 items

□	NAME	TYPE	LOCATION	
□	kylelab	Virtual machine	East US 2	•••
□	kylelabnetwork	Virtual network	East US 2	•••
□	kyleml	Machine Learning Studio works...	South Central US	•••
□	kylemlstorage	Storage account	South Central US	•••

2. On the Machine Learning Studio workspace blade, select **Launch Machine Learning Studio**.



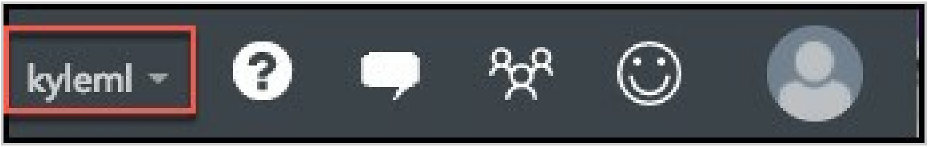
Launch Machine Learning Studio

Launch Machine Learning Gallery

Launch Machine Learning Studio Web Service Management

3. Sign in, if prompted.

4. If you have multiple Azure ML workspaces, choose the one you created for this hands-on lab from the drop-down menu near the top right of Azure Machine Learning Studio.



Task 2: Upload the Sample Datasets

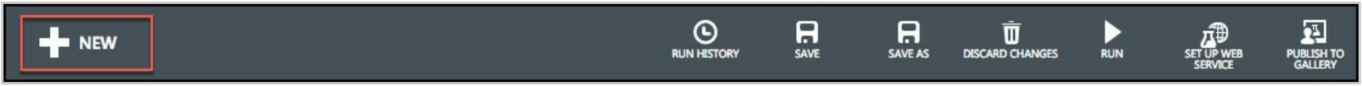
1. Before you begin creating a machine learning experiment, there are three datasets you need to load.

2. Download the three CSV sample datasets from here: http://bit.ly/2wGAqrl (If you get an error, or the page won’t open, try pasting the URL into a new browser window and verify the case sensitive URL is exactly as shown).

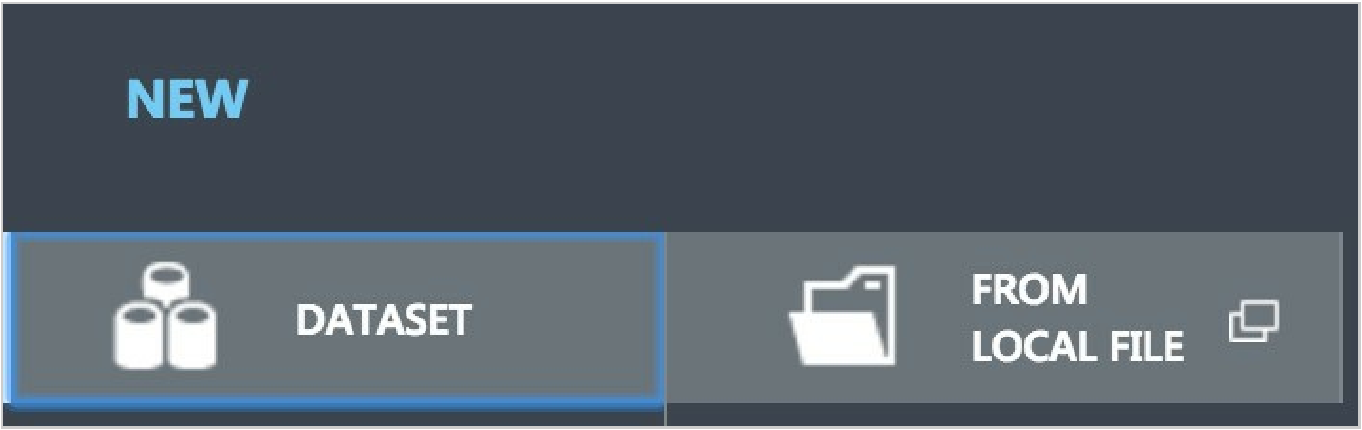
3. Extract the ZIP and verify you have the following files:

- FlightDelaysWithAirportCodes.csv
- FlightWeatherWithAirportCodes.csv
- AirportCodeLocationLookupClean.csv

4. In the Machine Learning Studio browser window, select **+ NEW** at the bottom left.

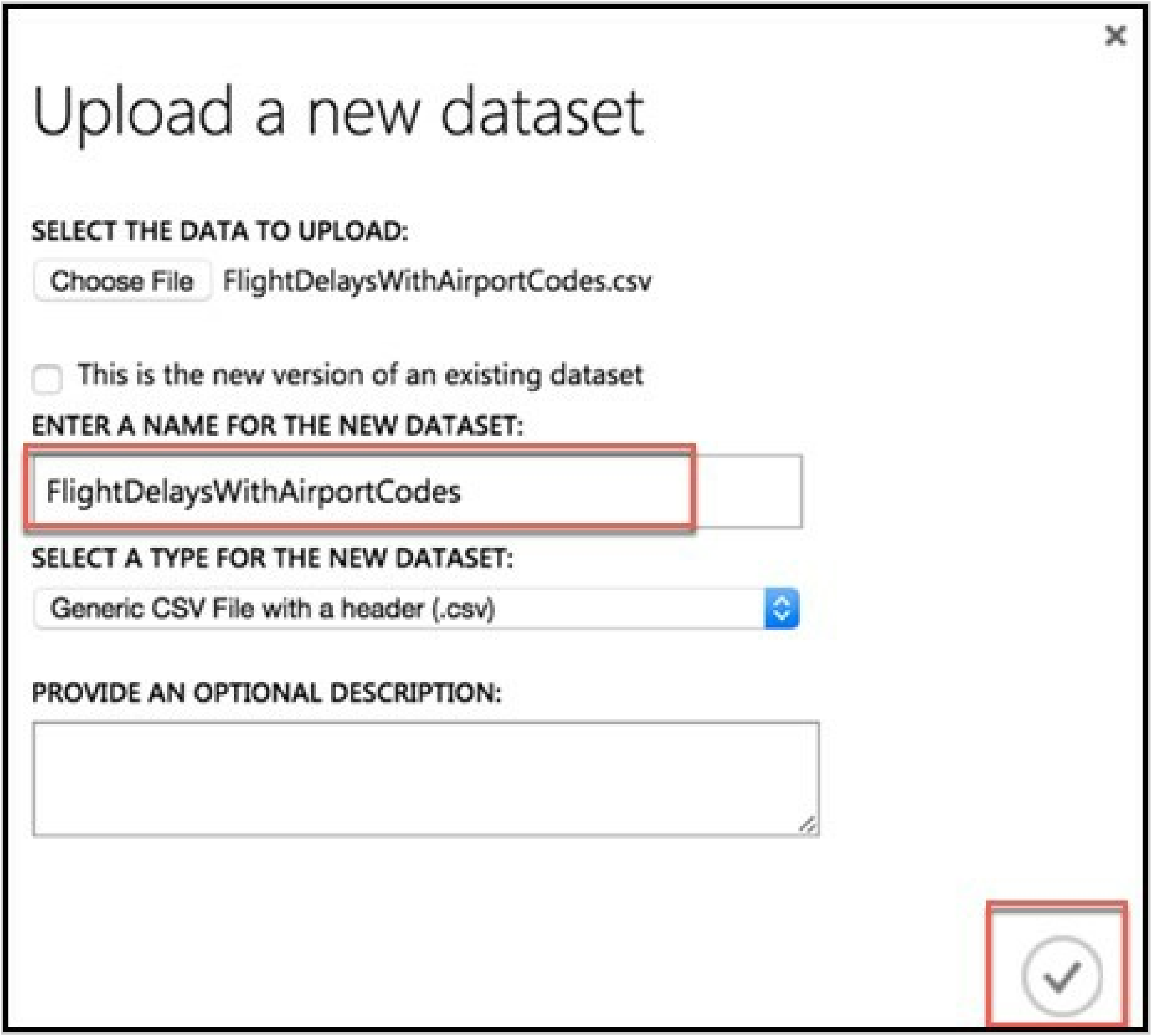


5. Select **Dataset** under New, and then select **From Local File**.



6. In the dialog that appears, select **Choose File**, browse to the FlightDelaysWithAirportCodes.csv file you downloaded in the previous step, and select **Open**.

7. **Change the name** of the dataset to "FlightDelaysWithAirportCodes," and **select the checkmark** to upload the data into a new dataset.



Upload a new dataset

SELECT THE DATA TO UPLOAD:

Choose File FlightDelaysWithAirportCodes.csv

□ This is the new version of an existing dataset

ENTER A NAME FOR THE NEW DATASET:

FlightDelaysWithAirportCodes

SELECT A TYPE FOR THE NEW DATASET:

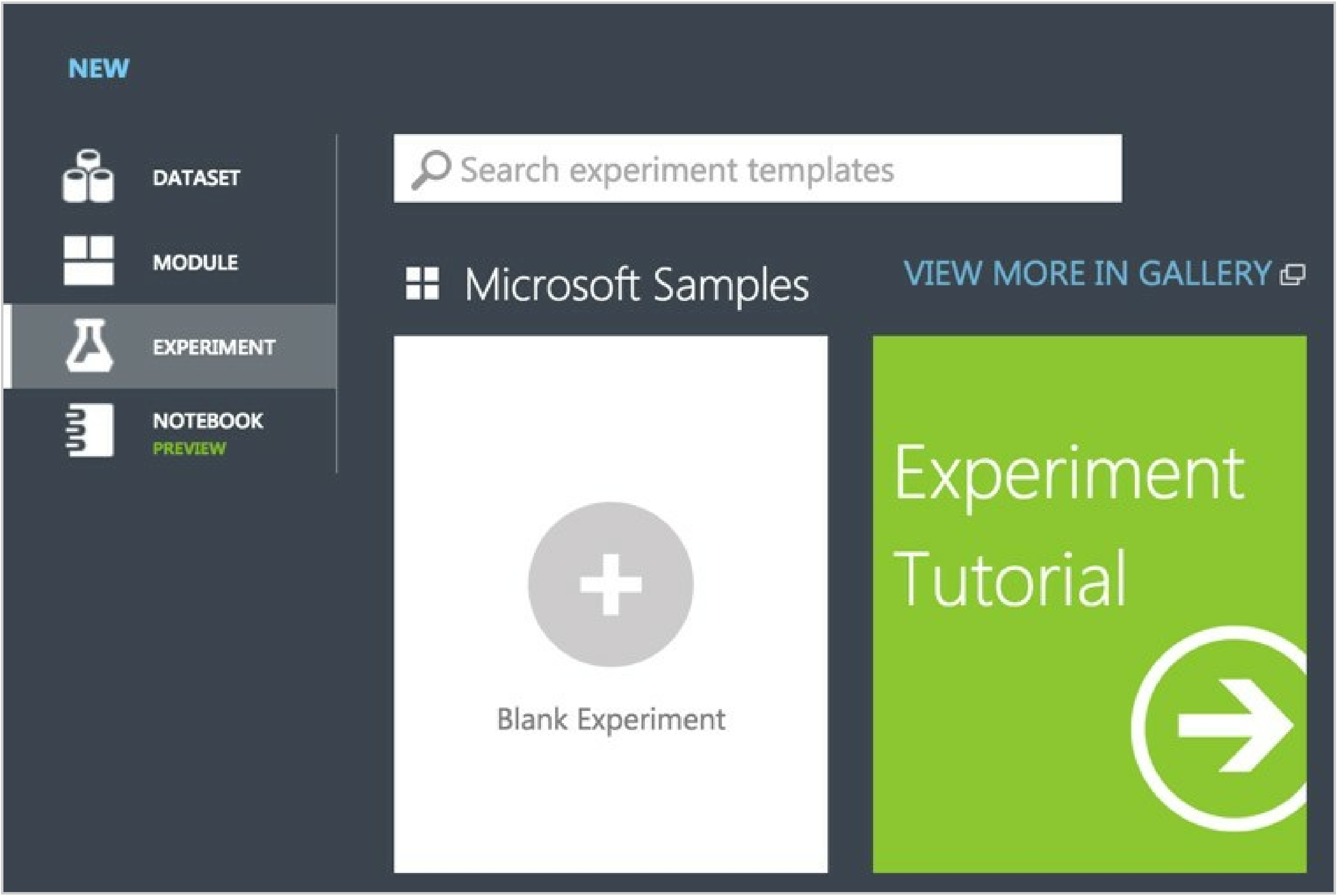
Generic CSV File with a header (.csv)

PROVIDE AN OPTIONAL DESCRIPTION:

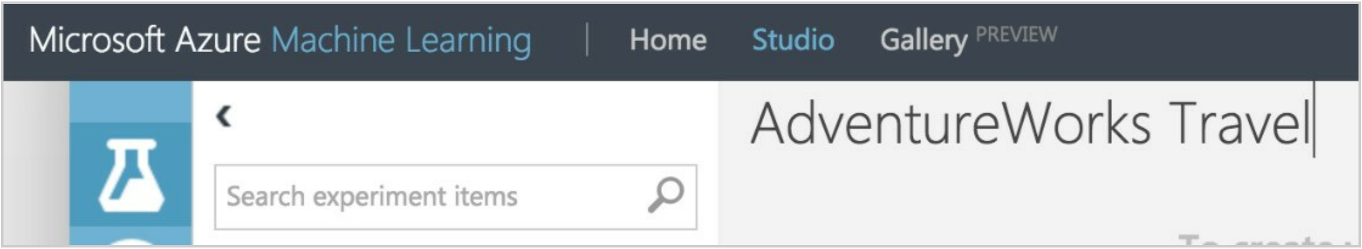
8. Repeat the previous step for the FlightWeatherWithAirportCode.csv and AirportCodeLocationsClean.csv files, setting the name for each dataset in a similar fashion.

Task 3: Start a new experiment

1. Select **+ NEW** in the command bar at the bottom left of the page, and select **Experiment**.
2. From the options that appear, select Blank Experiment.



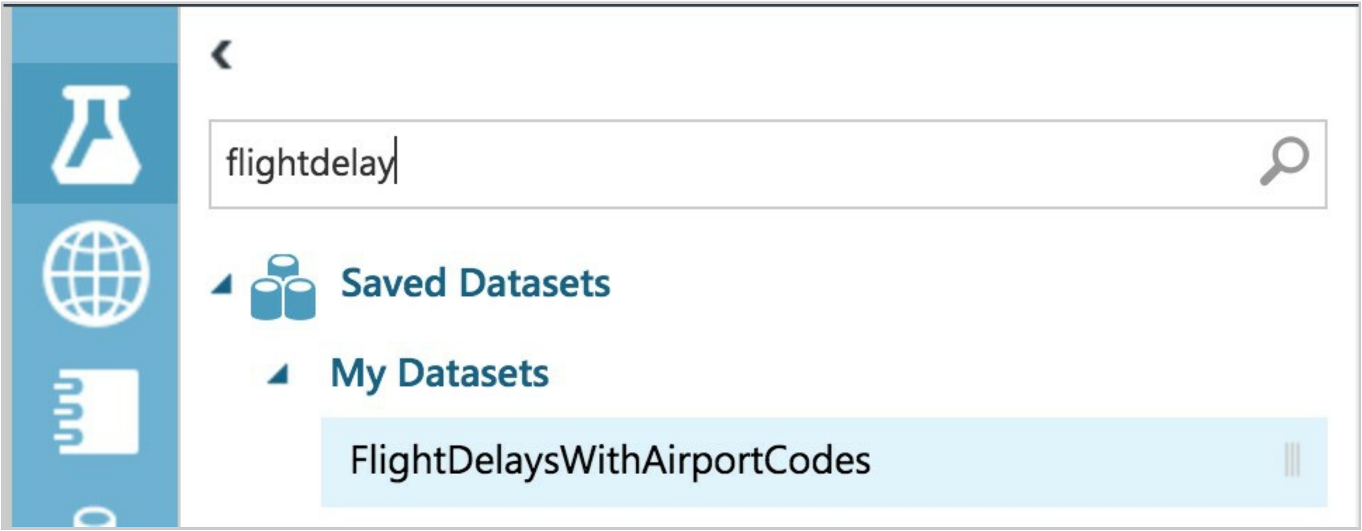
3. Give your new experiment a name, such as AdventureWorks Travel by editing the "Experiment created on ..." label near the top of the design surface.



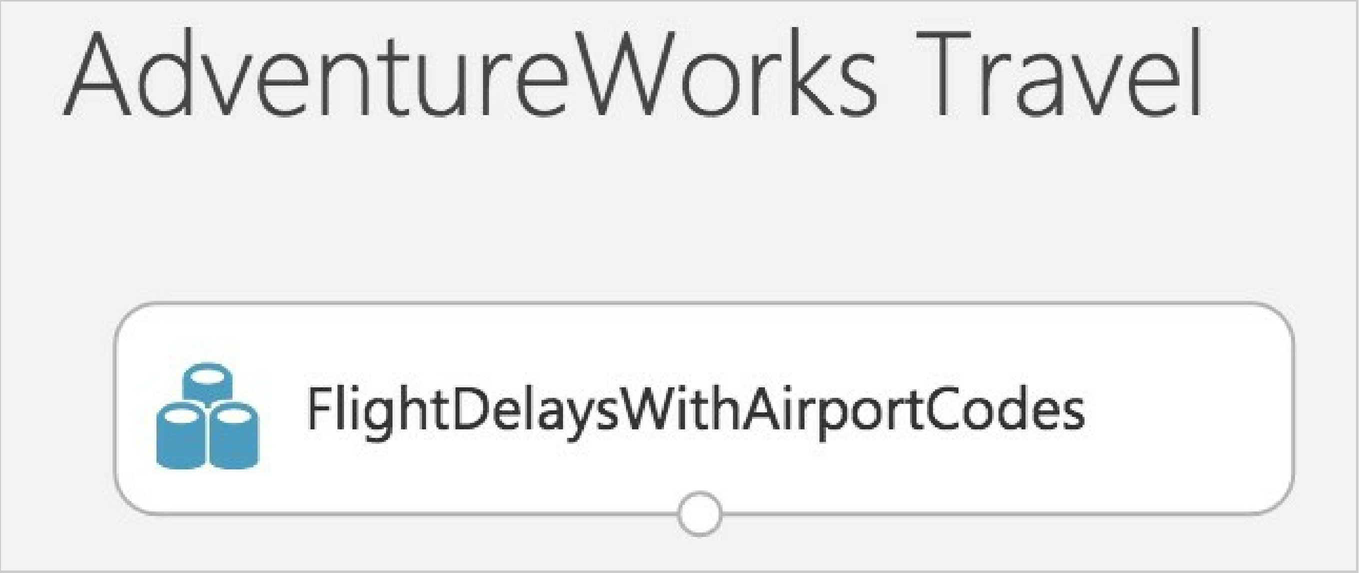
AdventureWorks Travel

Task 4: Prepare flight delay data

1. In the toolbar on the left, in the **Search experiment items** box, type the name of the dataset you created with flight delay data (FlightDelaysWithAirportCodes). You should see a component for it listed under Saved Datasets, My Datasets.

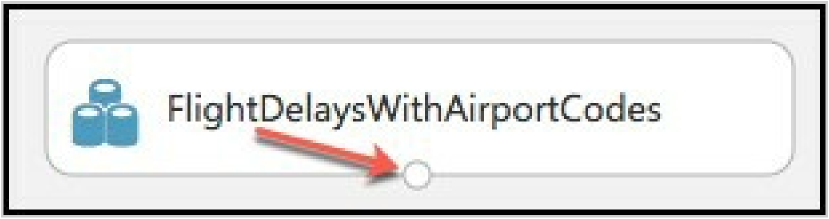


2. **Select and drag** the FlightDelaysWithAirportCodes module onto the design surface.

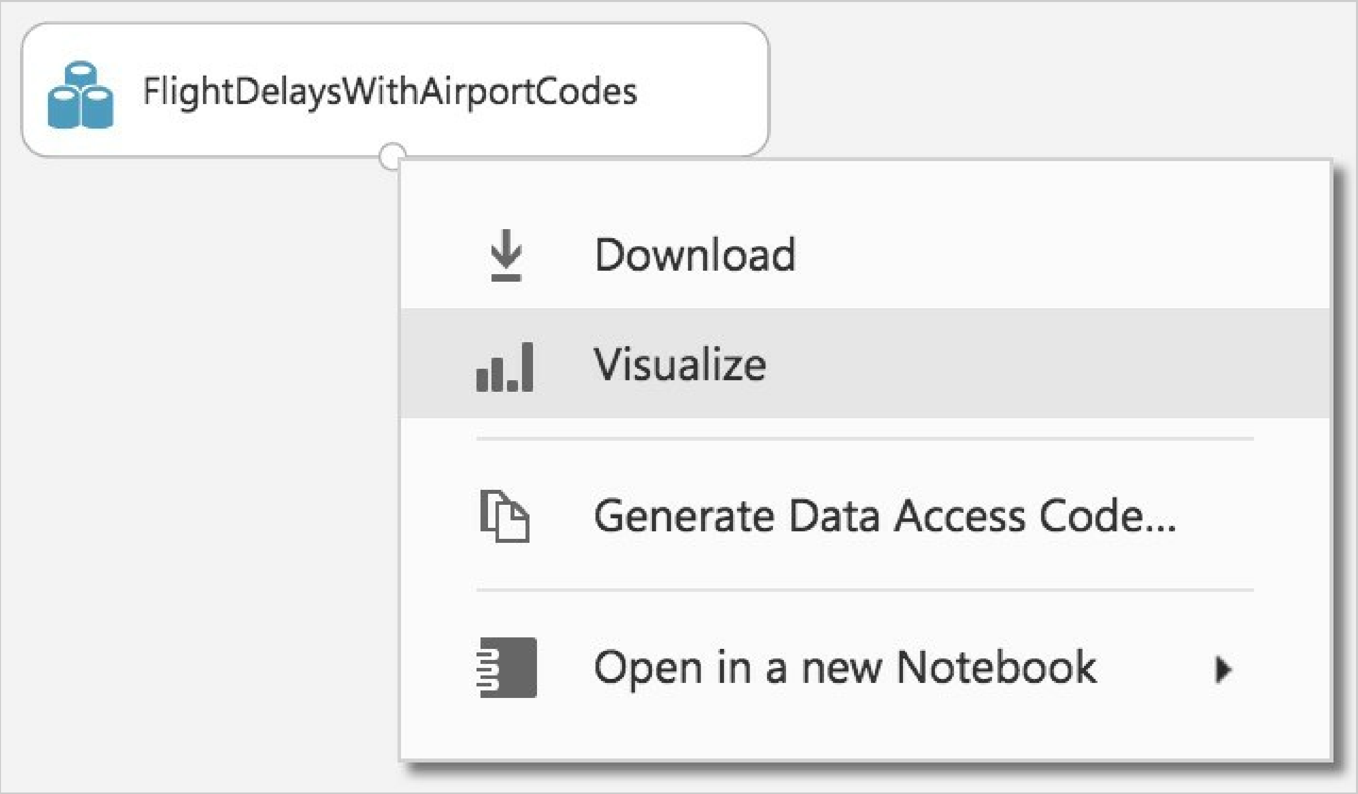


3. Next, you will explore the Flight delays datasets to understand what kind of cleanup (e.g., data munging) will be necessary.

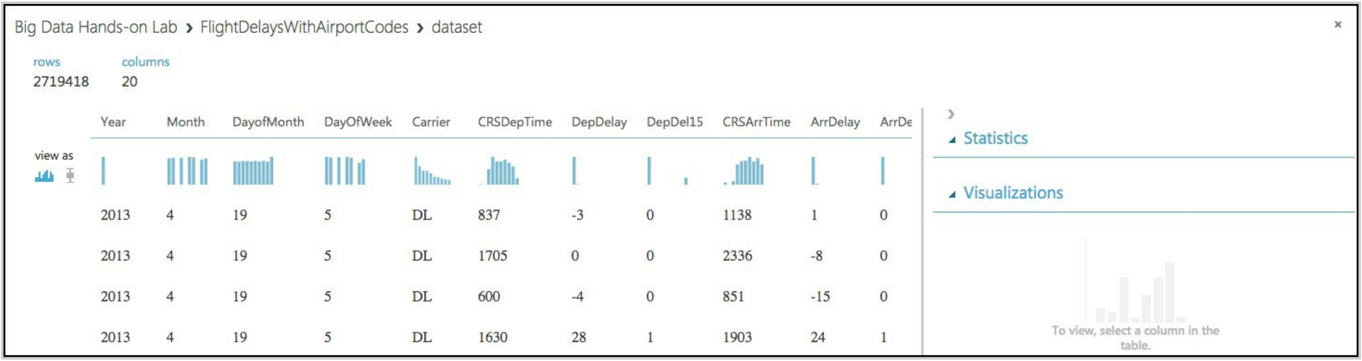
4. Hover over the output port of the **FlightDelaysWithAirportCodes** module.



5. Right-click on the port and select **Visualize**.



6. A new dialog will appear showing a maximum of 100 rows by 100 columns sample of the dataset. You can see at the top that the dataset has a total of 2,719,418 rows (also referred to as examples in Machine Learning literature) and has 20 columns (also referred to as features).

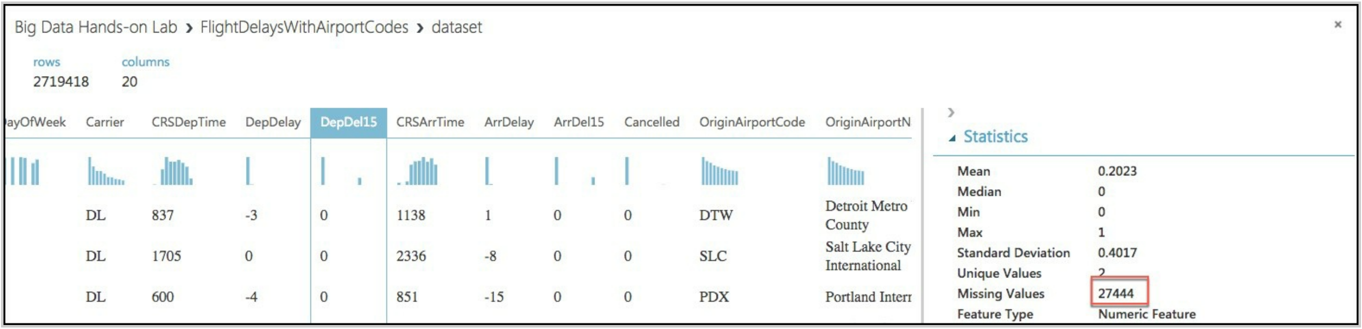


Big Data Hands-on Lab › FlightDelaysWithAirportCodes › dataset

rows	columns
2719418	20

Year	Month	DayofMonth	DayOfWeek	Carrier	CRSDepTime	DepDelay	DepDel15	CRSArrTime	ArrDelay	ArrDe
2013	4	19	5	DL	837	-3	0	1138	1	0
2013	4	19	5	DL	1705	0	0	2336	-8	0
2013	4	19	5	DL	600	-4	0	851	-15	0
2013	4	19	5	DL	1630	28	1	1903	24	1

7. Because all 20 columns are displayed, you can scroll the grid horizontally. Scroll until you see the **DepDel15** column, and select it to view statistics about the column. The DepDel15 column displays a 1 when the flight was delayed at least 15 minutes and 0 if there was no such delay. In the model you will construct, you will try to predict the value of this column for future data. Notice in the Statistics panel that a value of 27444 appears for Missing Values. This means that 27,444 rows do not have a value in this column. Since this value is very important to our model, we will need to eliminate any rows that do not have a value for this column.



Big Data Hands-on Lab › FlightDelaysWithAirportCodes › dataset

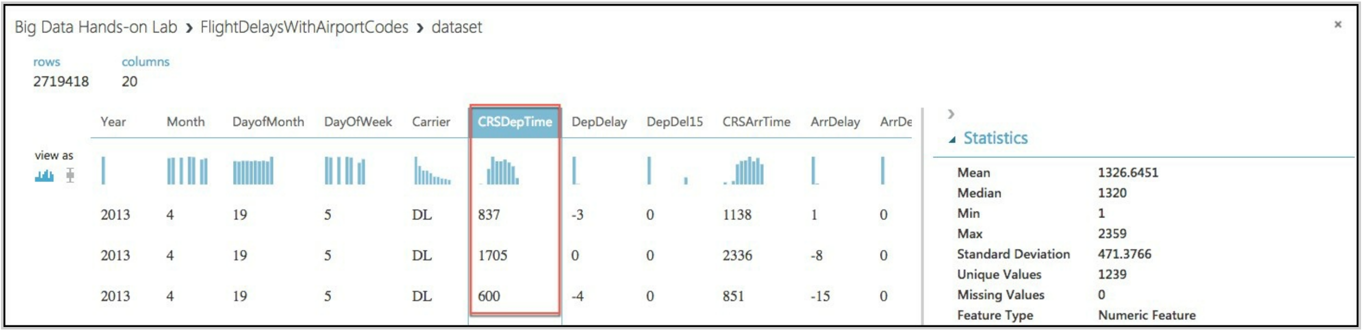
rows	columns
2719418	20

ayOfWeek	Carrier	CRSDepTime	DepDelay	DepDel15	CRSArrTime	ArrDelay	ArrDel15	Cancelled	OriginAirportCode	OriginAirportN
	DL	837	-3	0	1138	1	0	0	DTW	Detroit Metro County
	DL	1705	0	0	2336	-8	0	0	SLC	Salt Lake City International
	DL	600	-4	0	851	-15	0	0	PDX	Portland Interr

Statistics

Mean	0.2023
Median	0
Min	0
Max	1
Standard Deviation	0.4017
Unique Values	2
Missing Values	27444
Feature Type	Numeric Feature

8. Next, select the **CRSDepTime** column. Our model will approximate departure times to the nearest hour, but departure time is captured as an integer. For example, 8:37 am is captured as 837. Therefore, we will need to process the CRSDepTime column, and round it down to the nearest hour. To perform this rounding will require two steps, first you will need to divide the value by 100 (so that 837 becomes 8.37). Second, you will round this value down to the nearest hour (so that 8.37 becomes 8.)



Big Data Hands-on Lab › FlightDelaysWithAirportCodes › dataset

rows	columns
2719418	20

Year	Month	DayofMonth	DayOfWeek	Carrier	CRSDepTime	DepDelay	DepDel15	CRSArrTime	ArrDelay	ArrDe
2013	4	19	5	DL	837	-3	0	1138	1	0
2013	4	19	5	DL	1705	0	0	2336	-8	0
2013	4	19	5	DL	600	-4	0	851	-15	0

Statistics

Mean	1326.6451
Median	1320
Min	1
Max	2359
Standard Deviation	471.3766
Unique Values	1239
Missing Values	0
Feature Type	Numeric Feature

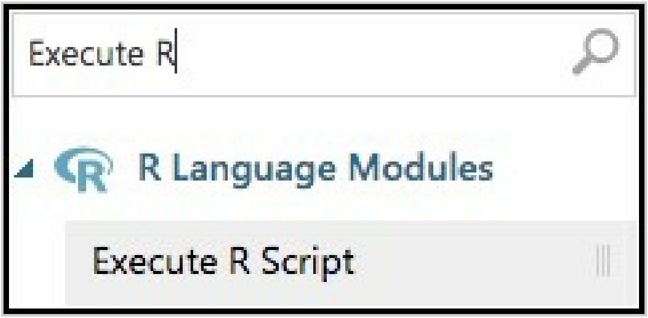
9. Finally, we do not need all 20 columns present in the FlightDelaysWithAirportCodes dataset, so we will need to pare down the columns in the dataset to the 12.

10. Close the Visualize dialog, and go back to the design surface.

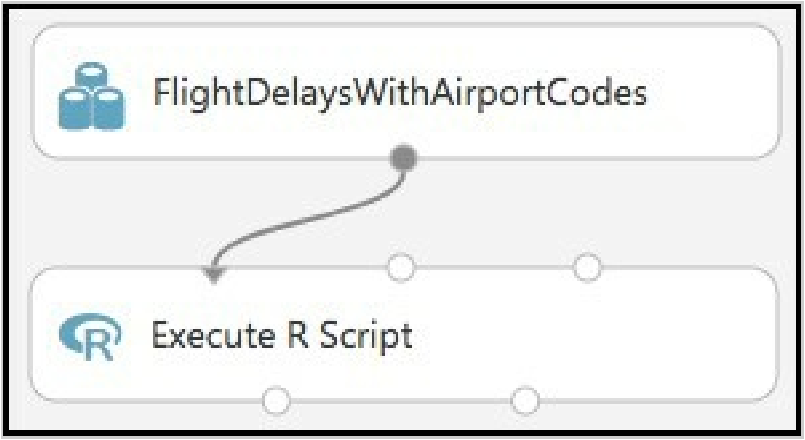
11. To perform our data munging, we have multiple options, but in this case, we’ve chosen to use an **Execute R Script** module, which will perform the following tasks:

- Remove rows with missing values
- Generate a new column, named “CRSDepHour,” which contains the rounded down value from CRSDepTime
- Pare down columns to only those needed for our model

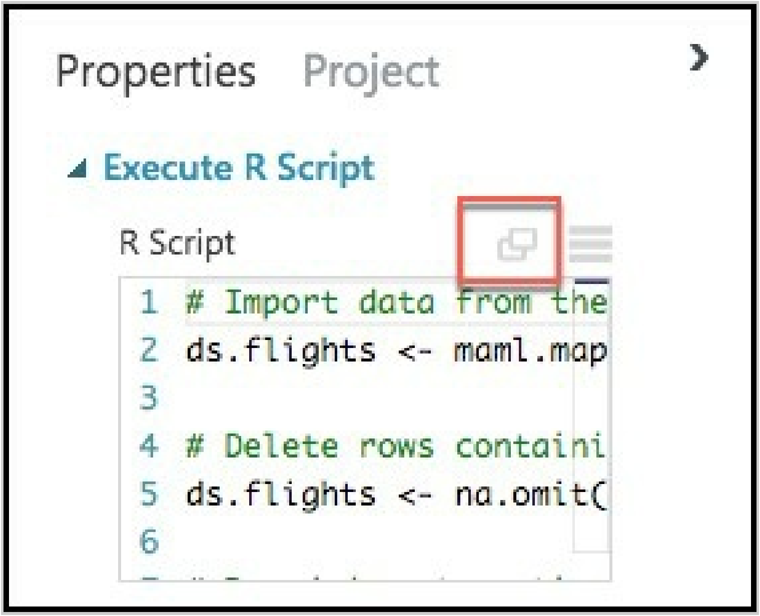
12. To add the module, search for **Execute R Script** by entering **“Execute R”** into the Search experiment items box.



13. **Drag this module** on to the design surface beneath your FlightDelaysWithAirportCodes dataset. Select the small circle at the bottom of the FlightDelaysWithAirportCodes dataset, drag and release when your mouse is over the circle found in the top left of the Execute R Script module. These circles are referred to as ports, and by taking this action you have connected the output port of the dataset with the input port of the Execute R Script module, meaning data from the dataset will flow along this path.



14. In the **Properties** panel for **Execute R Script** module, select the **Double Windows** icon to maximize the script editor.

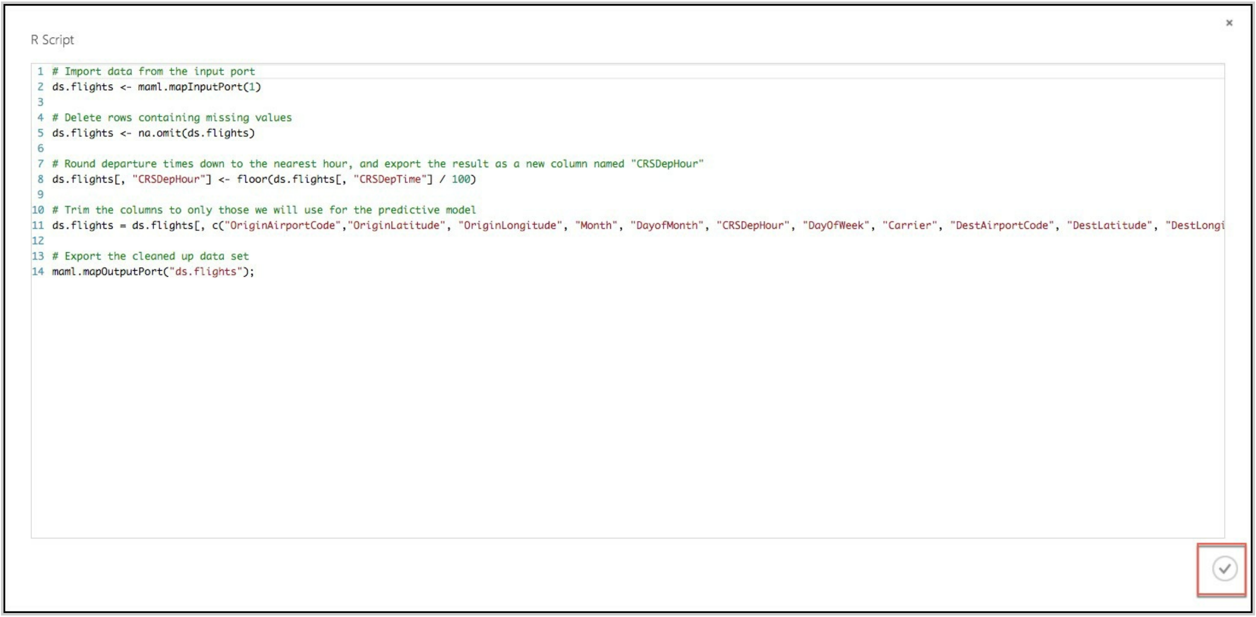


```
# Import data from the
ds.flights <- maml.map

# Delete rows containi
ds.flights <- na.omit(

```

15. Replace the script with the following (Press CTRL+A to select all then CTRL+V to paste)

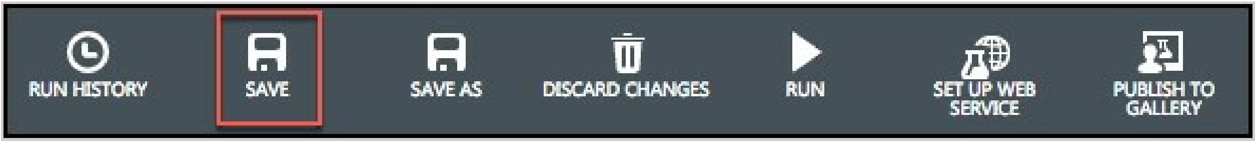
```
# Import data from the input port
ds.flights <- maml.mapInputPort(1)

# Delete rows containing missing values
ds.flights <- na.omit(ds.flights)

# Round departure times down to the nearest hour, and export the result as a new column named "CRSDepHour"
ds.flights[, "CRSDepHour"] <- floor(ds.flights[, "CRSDepTime"] / 100)

# Trim the columns to only those we will use for the predictive model
ds.flights = ds.flights[, c("OriginAirportCode","OriginLatitude", "OriginLongitude", "Month", "DayofMonth", "CRSDepHour", "DayOfWeek", "Carrier", "DestAirportCode", "DestLatitude", "DestLongitude", "DepDel15")]

# Export the cleaned up data set
maml.mapOutputPort("ds.flights");
```

16. **Select the check mark** in the bottom right to save the script (Do not worry if the formatting is off before hitting the check mark.)

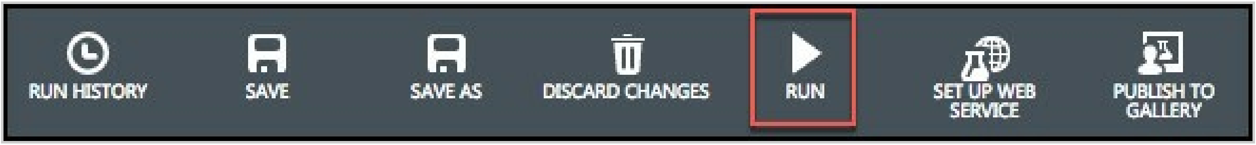


```
# Import data from the input port
ds.flights <- maml.mapInputPort(1)

# Delete rows containing missing values
ds.flights <- na.omit(ds.flights)

# Round departure times down to the nearest hour, and export the result as a new column named "CRSDepHour"
ds.flights[, "CRSDepHour"] <- floor(ds.flights[, "CRSDepTime"] / 100)

# Trim the columns to only those we will use for the predictive model
ds.flights = ds.flights[, c("OriginAirportCode","OriginLatitude", "OriginLongitude", "Month", "DayofMonth", "CRSDepHour", "DayOfWeek", "Carrier", "DestAirportCode", "DestLatitude", "DestLongi

# Export the cleaned up data set
maml.mapOutputPort("ds.flights");
```

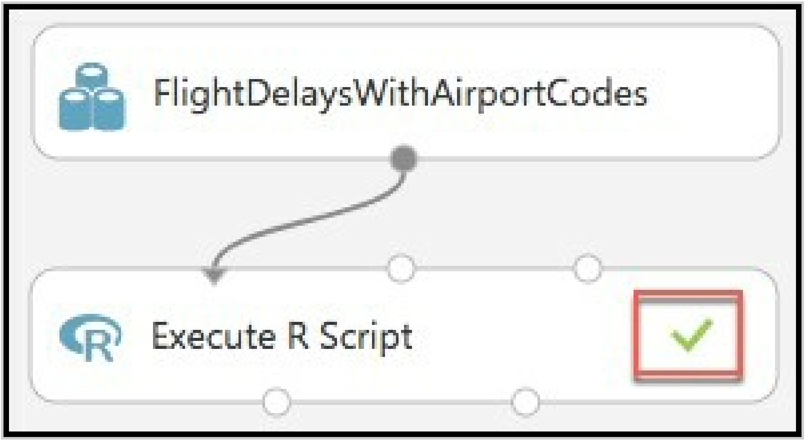
17. Select Save on the command bar at the bottom to save your in-progress experiment.



18. Select Run in the command bar at the bottom to run the experiment.

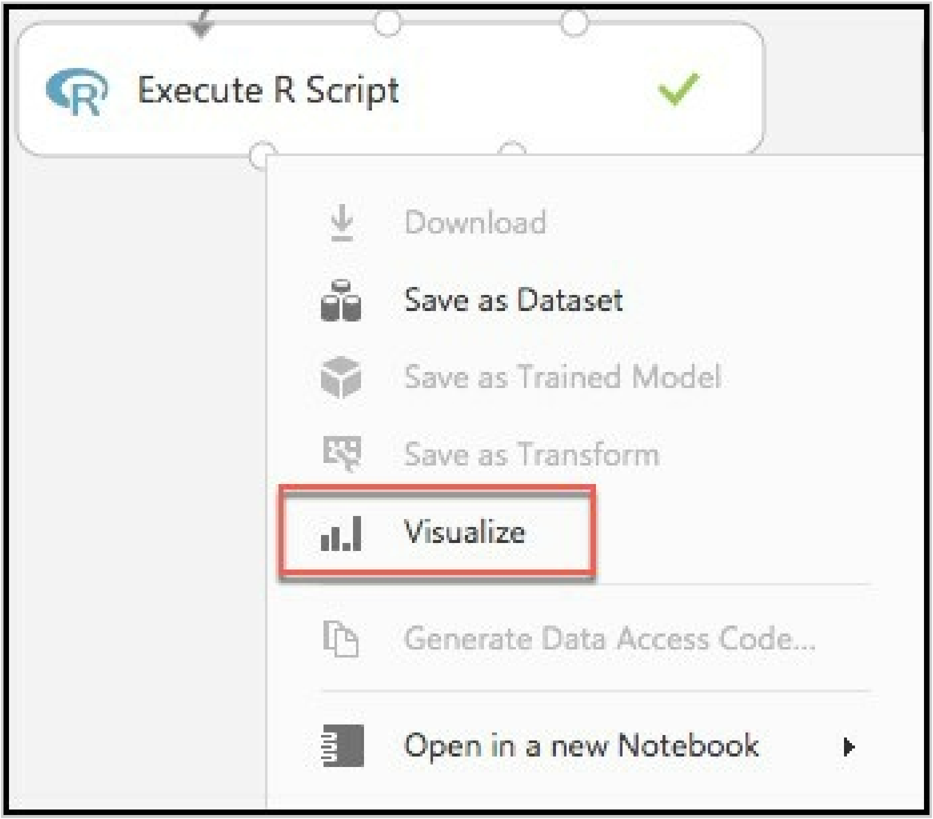


19. When the experiment is finished running, you will see a finished message in the top right corner of the design surface, and green check marks over all modules that ran.

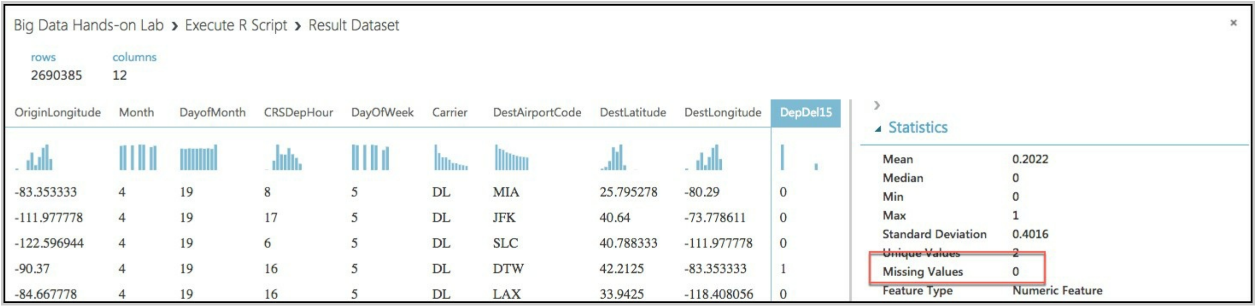


20. You should run your experiment whenever you need to update the metadata describing what data is flowing through the modules, so that newly added modules can be aware of the shape of your data (most modules have dialogs that can suggest columns, but before they can make suggestions you need to have run your experiment).

21. To verify the results of our R script, right-click the left output port (Result Dataset) of the Execute R Script module and select **Visualize**.



22. In the dialog that appears, scroll over to **DepDel15** and select the column. In the statistics you should see that Missing Values reads 0.



Big Data Hands-on Lab › Execute R Script › Result Dataset

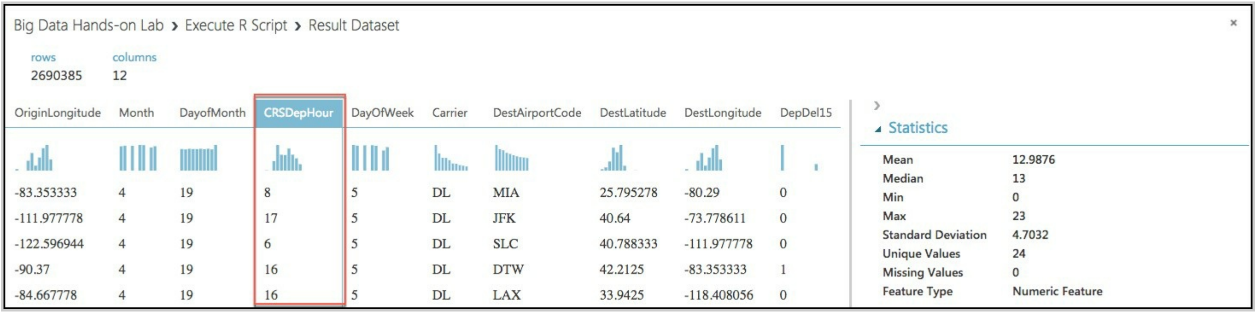
rows	columns
2690385	12

OriginLongitude	Month	DayofMonth	CRSDepHour	DayOfWeek	Carrier	DestAirportCode	DestLatitude	DestLongitude	DepDel15
-83.353333	4	19	8	5	DL	MIA	25.795278	-80.29	0
-111.977778	4	19	17	5	DL	JFK	40.64	-73.778611	0
-122.596944	4	19	6	5	DL	SLC	40.788333	-111.977778	0
-90.37	4	19	16	5	DL	DTW	42.2125	-83.353333	1
-84.667778	4	19	16	5	DL	LAX	33.9425	-118.408056	0

Statistics

Mean	0.2022
Median	0
Min	0
Max	1
Standard Deviation	0.4016
Unique Values	2
Missing Values	0
Feature Type	Numeric Feature

23. Now, select the **CRSDepHour** column, and verify that our new column contains the rounded hour values from our CRSDepTime column.



Big Data Hands-on Lab › Execute R Script › Result Dataset

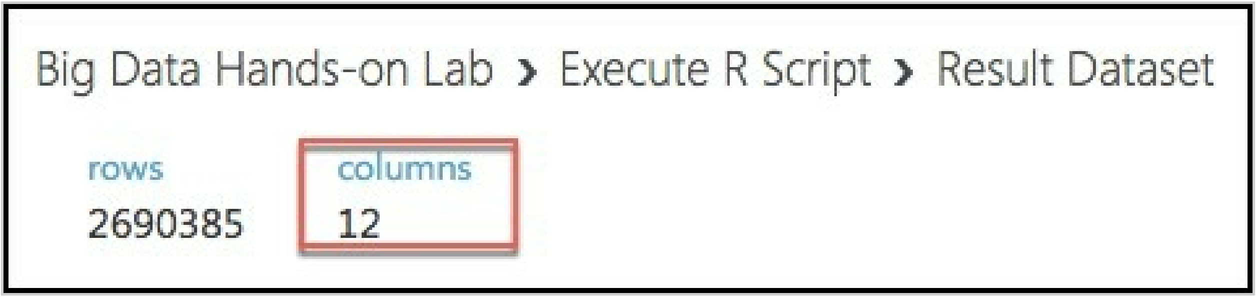
rows	columns
2690385	12

OriginLongitude	Month	DayofMonth	CRSDepHour	DayOfWeek	Carrier	DestAirportCode	DestLatitude	DestLongitude	DepDel15
-83.353333	4	19	8	5	DL	MIA	25.795278	-80.29	0
-111.977778	4	19	17	5	DL	JFK	40.64	-73.778611	0
-122.596944	4	19	6	5	DL	SLC	40.788333	-111.977778	0
-90.37	4	19	16	5	DL	DTW	42.2125	-83.353333	1
-84.667778	4	19	16	5	DL	LAX	33.9425	-118.408056	0

Statistics

Mean	12.9876
Median	13
Min	0
Max	23
Standard Deviation	4.7032
Unique Values	24
Missing Values	0
Feature Type	Numeric Feature

24. Finally, observe that we have reduced the number of columns from 20 to 12. Close the dialog.



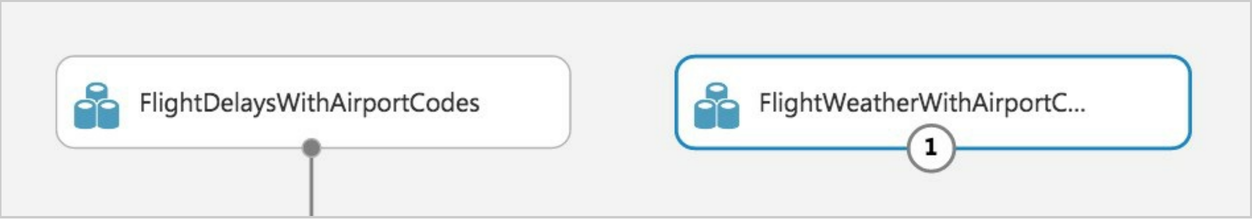
Big Data Hands-on Lab › Execute R Script › Result Dataset

rows	columns
2690385	12

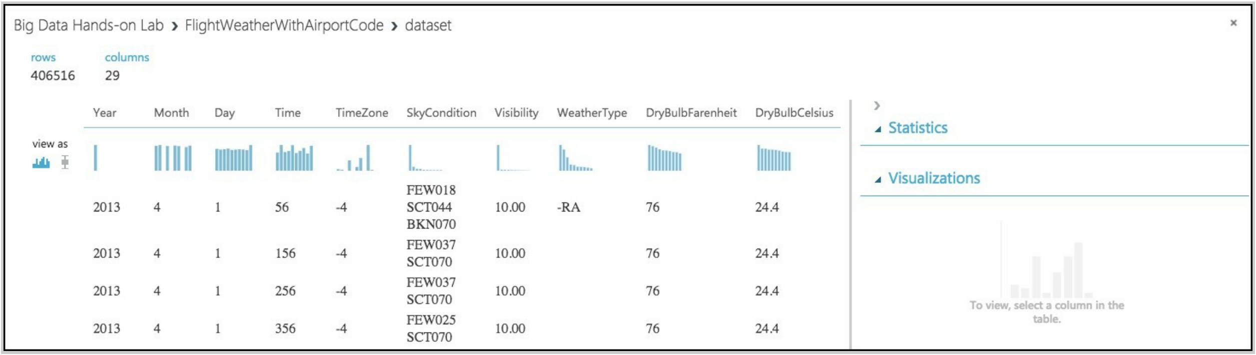
25. At this point the Flight Delay Data is prepared, and we turn to preparing the historical weather data.

Task 5: Prepare weather data

1. To the right of the FlightDelaysWithAirportCodes dataset, add the **FlightWeatherWithAirportCodes** dataset.



2. Right-click the output port of the FlightWeatherWithAirportCodes dataset and select **Visualize**.



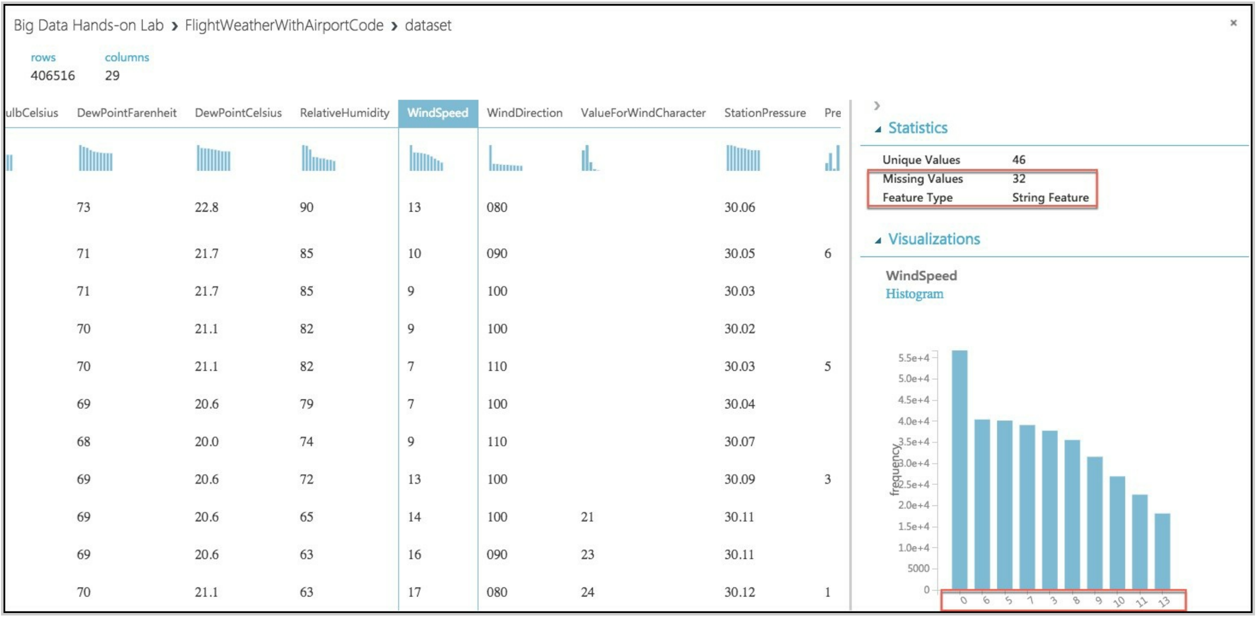
Big Data Hands-on Lab › FlightWeatherWithAirportCode › dataset

rows	columns
406516	29

Year	Month	Day	Time	TimeZone	SkyCondition	Visibility	WeatherType	DryBulbFarenheit	DryBulbCelsius
2013	4	1	56	-4	FEW018 SCT044 BKN070	10.00	-RA	76	24.4
2013	4	1	156	-4	FEW037 SCT070	10.00		76	24.4
2013	4	1	256	-4	FEW037 SCT070	10.00		76	24.4
2013	4	1	356	-4	FEW025 SCT070	10.00		76	24.4

3. Observe that this data set has 406,516 rows and 29 columns. For this model, we are going to focus on predicting delays using WindSpeed (in MPH), SeaLevelPressure (in inches of Hg), and HourlyPrecip (in inches). We will focus on preparing the data for those features.

4. In the dialog, select the **WindSpeed** column, and review the statistics. Observe that the Feature Type was inferred as String and that there are 32 Missing Values. Below that, examine the histogram to see that, even though the type was inferred as string, the values are all actually numbers (e.g. the x-axis values are 0, 6, 5, 7, 3, 8, 9, 10, 11, 13). We will need to ensure that we remove any missing values and convert WindSpeed to its proper type as a numeric feature.



Big Data Hands-on Lab › FlightWeatherWithAirportCode › dataset

rows 406516
columns 29

ulbCelsius	DewPointFarenheit	DewPointCelsius	RelativeHumidity	WindSpeed	WindDirection	ValueForWindCharacter	StationPressure	Pre
	73	22.8	90	13	080		30.06	
	71	21.7	85	10	090		30.05	6
	71	21.7	85	9	100		30.03	
	70	21.1	82	9	100		30.02	
	70	21.1	82	7	110		30.03	5
	69	20.6	79	7	100		30.04	
	68	20.0	74	9	110		30.07	
	69	20.6	72	13	100		30.09	3
	69	20.6	65	14	100	21	30.11	
	69	20.6	63	16	090	23	30.11	
	70	21.1	63	17	080	24	30.12	1

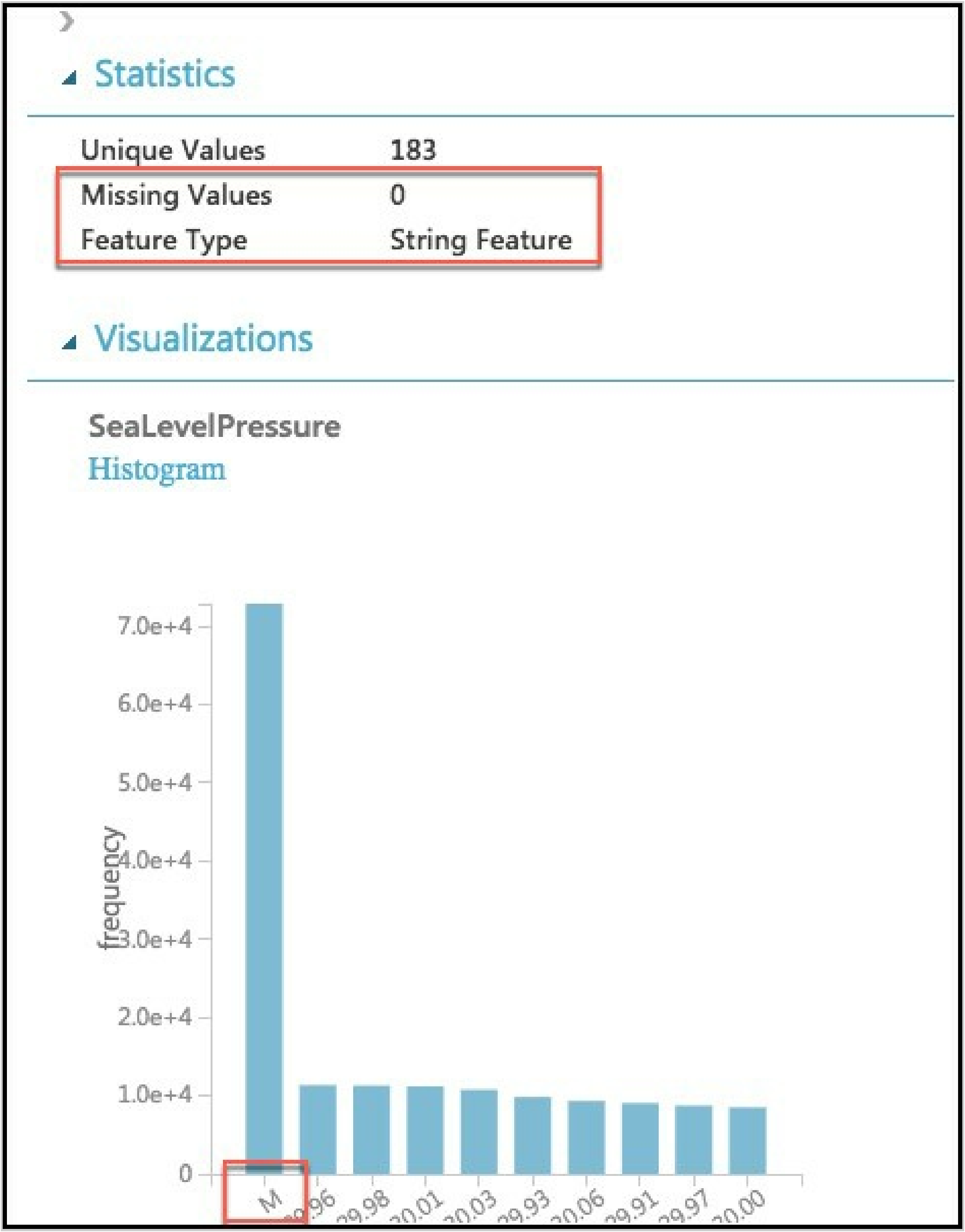
Statistics

Unique Values	46
Missing Values	32
Feature Type	String Feature

Visualizations

WindSpeed
Histogram

5. Next, select the **SeaLevelPressure** column. Observe that the Feature Type was inferred as String and there are 0 Missing Values. Scroll down to the histogram, and observe that many of the features are of a numeric value (e.g., 29.96, 30.01, etc.), but there are many features with the string value of M for Missing. We will need to replace this value of "M" with a suitable numeric value so that we can convert this feature to be a numeric feature.



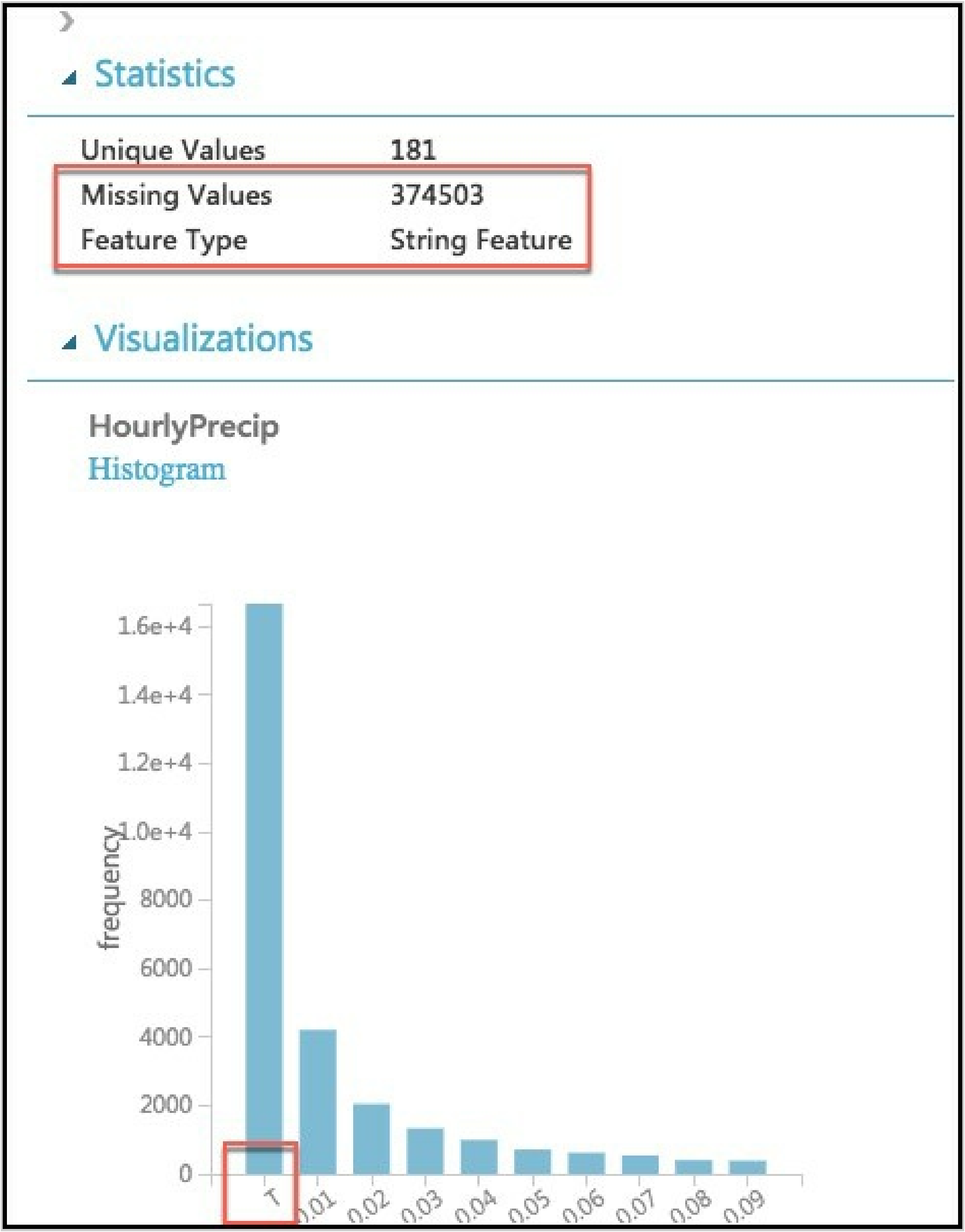
Statistics

Unique Values	183
Missing Values	0
Feature Type	String Feature

Visualizations

SeaLevelPressure
Histogram

6. Finally, examine the **HourlyPrecip** feature. Observe that it too was inferred to have a Feature Type of String and is missing values for 374,503 rows. Looking at the histogram, observe that besides the numeric values, there is a value T (for Trace amount of rain). We need to replace T with a suitable numeric value and covert this to a numeric feature.



Statistics

Unique Values	181
Missing Values	374503
Feature Type	String Feature

Visualizations

HourlyPrecip

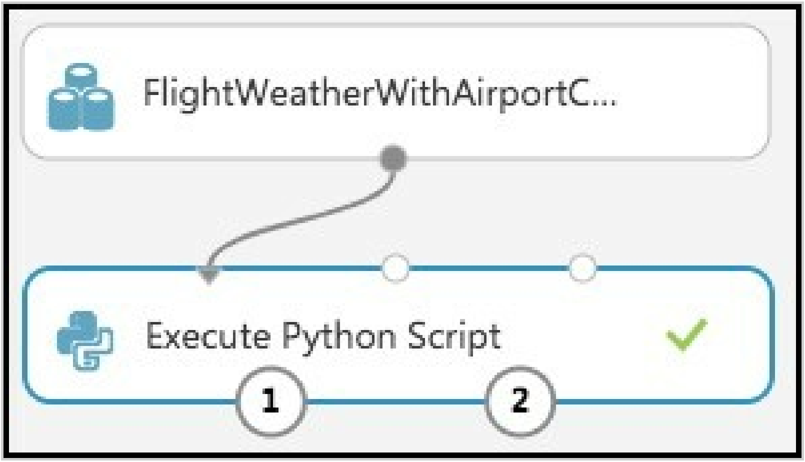
Histogram

7. To preform our data cleanup, we will use a Python script, in which we will perform the following tasks:

- WindSpeed: Replace missing values with 0.0, and “M” values with 0.005
- HourlyPrecip: Replace missing values with 0.0, and “T” values with 0.005
- SeaLevelPressure: Replace “M” values with 29.92 (the average pressure)

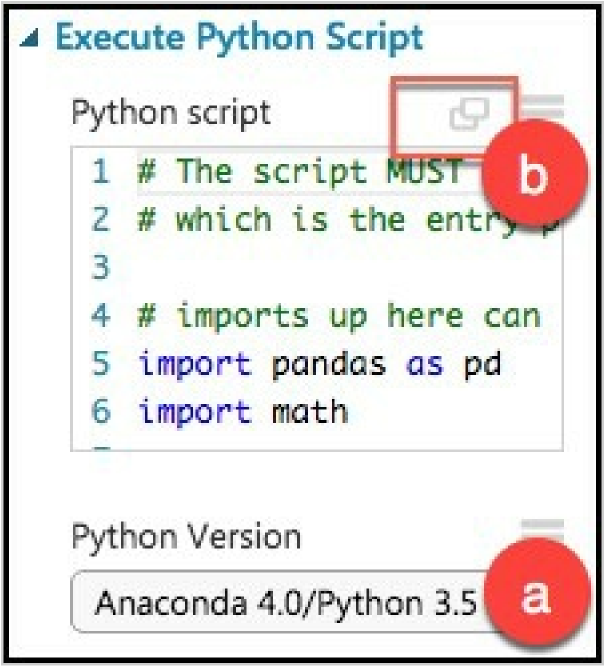
- Convert WindSpeed, HourlyPrecip, and SeaLevelPressure to numeric columns
- Round “Time” column down to the nearest hour, and add value to a new column named “Hour”
- Eliminate unneeded columns from the dataset

8. Add an **Execute Python Script** module below the FlightWeatherWithAirportCode module, and connect the output port of the FlightWeatherWithAirportCode module to the first input port of the Execute Python Script module.



9. In the Properties panel for the Execute Python Script:

- Set the Python Version to Anaconda 4.0/Python 3.5
- Select the Double Windows icon to open the script editor.



Python script

```
# The script MUST
# which is the entry

# imports up here can
import pandas as pd
import math
```

Python Version

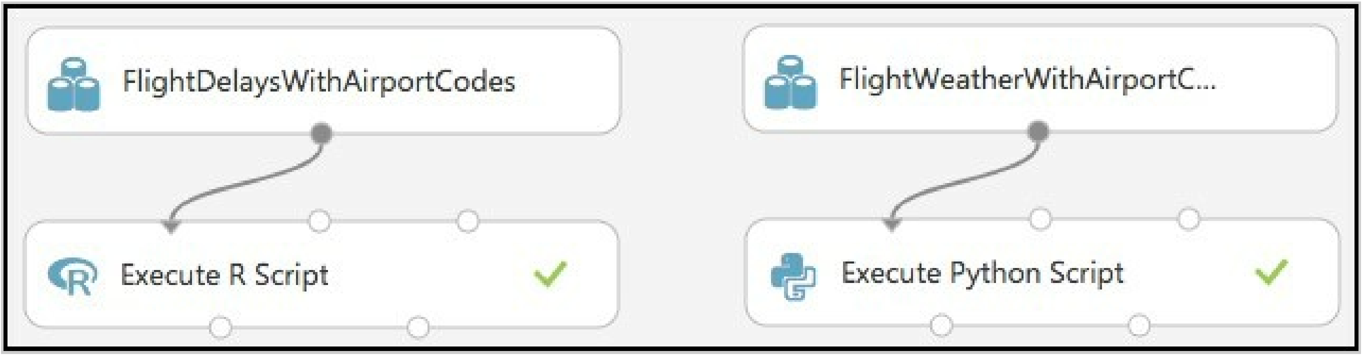
Anaconda 4.0/Python 3.5

10. Paste in the following script into the Python script window, and select the checkmark at the bottom right of the dialog (press CTRL+A to select all then CTRL+V to paste and then

immediately select the checkmark -- don't worry if the formatting is off before hitting the checkmark).

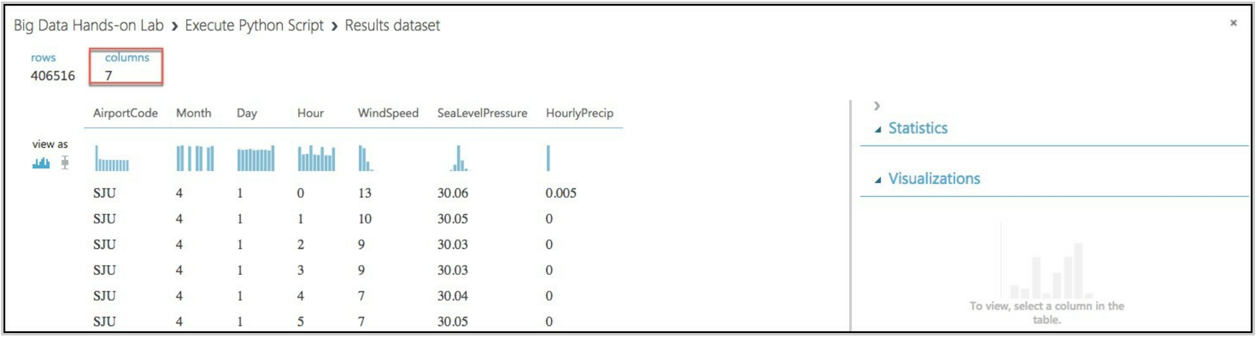
```python
# imports
import pandas as pd
import math

# The entry point function can contain up to two input arguments:
#   Param<dataframe1>: a pandas.DataFrame
#   Param<dataframe2>: a pandas.DataFrame
def azureml_main(dataframe1 = None, dataframe2 = None):

    # Round weather Time down to the next hour, since that is the hour for which we want to use flight dataframe1
    # Add the rounded Time to a new column named "Hour," and append that column to the dataframe1
    dataframe1["Hour"] = dataframe1["Time"].apply(roundDown)

    # Replace any missing HourlyPrecip and WindSpeed values with 0.0
    dataframe1["HourlyPrecip"] = dataframe1["HourlyPrecip"].fillna('0.0')
    dataframe1["WindSpeed"] = dataframe1["WindSpeed"].fillna('0.0')

    # Replace any WindSpeed values of "M" with 0.005
    dataframe1["WindSpeed"] = dataframe1['WindSpeed'].replace(['M'], '0.005')

    # Replace any SeaLevelPressure values of "M" with 29.92 (the average pressure)
    dataframe1["SeaLevelPressure"] = dataframe1['SeaLevelPressure'].replace(['M'], '29.92')

    # Replace any HourlyPrecip values of "T" (trace) with 0.005
    dataframe1["HourlyPrecip"] = dataframe1['HourlyPrecip'].replace(['T'], '0.005')

    # Convert our WindSpeed, SeaLevelPressure, and HourlyPrecip columns to numeric
    dataframe1[['WindSpeed','SeaLevelPressure', 'HourlyPrecip']] = dataframe1[['WindSpeed','SeaLevelPressure', 'HourlyPrecip']].apply(pd.to_numeric)

    # Pare down the variables in the Weather dataset to just the columns being used by the model
    df_result = dataframe1[['AirportCode', 'Month', 'Day', 'Hour', 'WindSpeed', 'SeaLevelPressure', 'HourlyPrecip']]

    # Return value must be of a sequence of pandas.DataFrame
    return df_result

def roundDown(x):
    z = int(math.floor(x/100.0))
    return z
```

11. Run the experiment. Currently it should appear as follows:



12. If you receive an error in the Python script that .to_numeric does not exist, go back and verify that you selected the proper Python version.

13. Right-click the first output port of the Execute Python Script module, and select **Visualize**.

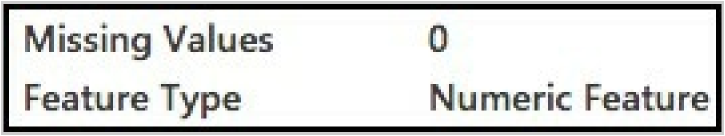


Big Data Hands-on Lab › Execute Python Script › Results dataset

rows 406516
columns 7

AirportCode	Month	Day	Hour	WindSpeed	SeaLevelPressure	HourlyPrecip
SJU	4	1	0	13	30.06	0.005
SJU	4	1	1	10	30.05	0
SJU	4	1	2	9	30.03	0
SJU	4	1	3	9	30.03	0
SJU	4	1	4	7	30.04	0
SJU	4	1	5	7	30.05	0

14. In the statistics, verify that there are now only the 7 columns we are interested in, and that WindSpeed, SeaLevelPressure, and HourlyPrecip are now all Numeric Feature types and that they have no missing values.

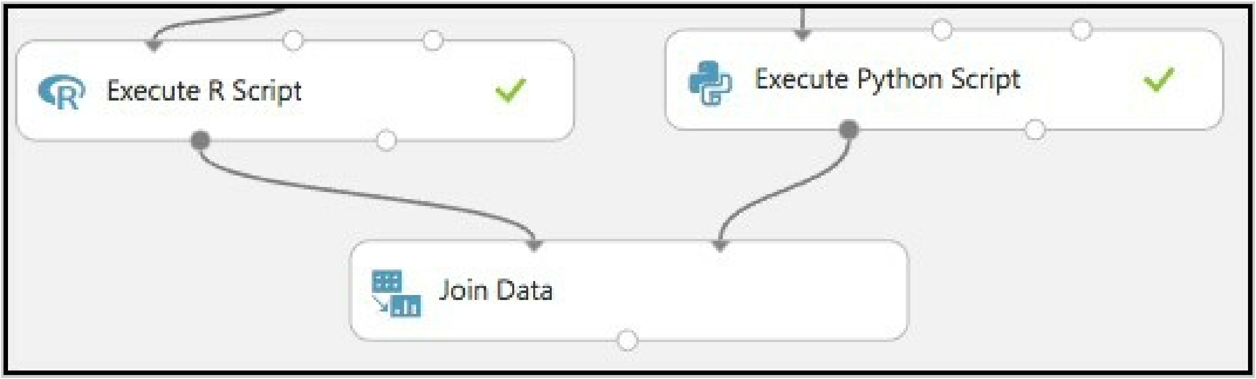


Missing Values 0
Feature Type Numeric Feature

Task 6: Join the Flight and Weather datasets

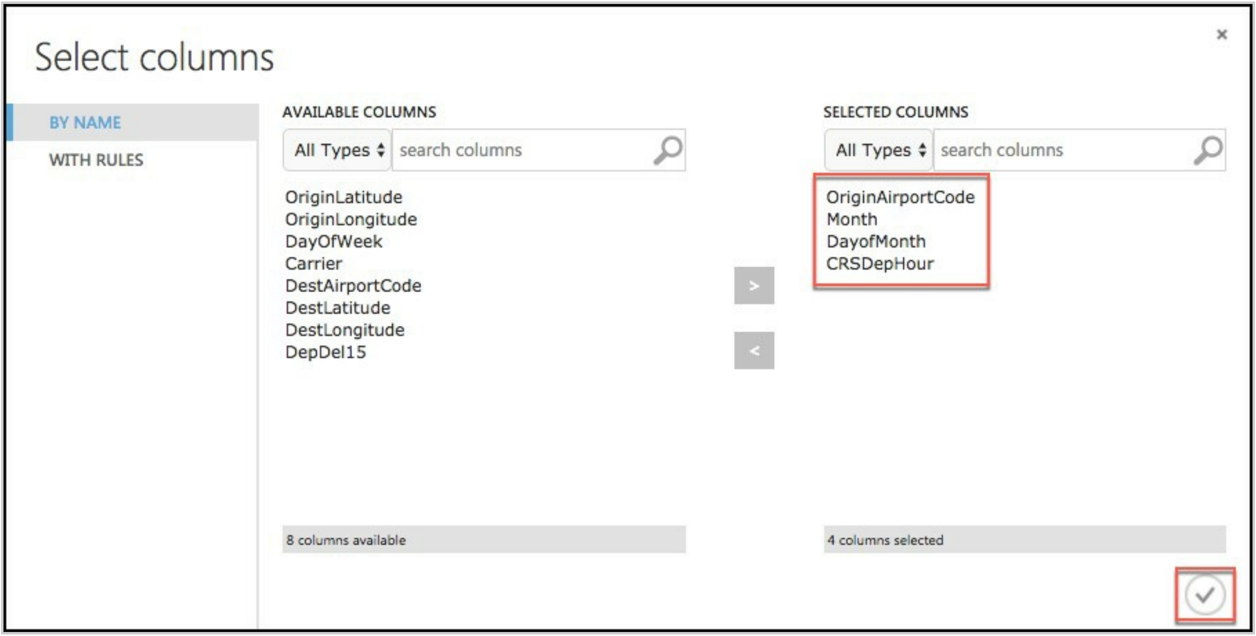
1. With both datasets ready, we want to join them together so that we can associate historical flight delays with the weather data at departure time.

2. Drag a **Join Data** module onto the design surface, beneath and centered between both Execute R and Python Script modules. Connect the output port (1) of the Execute R Script module to input port (1) of the Join Data module, and the output port (1) of the Execute Python Script module to the input port (2) of the Join Data module.



3. In the **Properties** panel for the Join Data module, relate the rows of data between the two sets L (the flight delays) and R (the weather).

4. Select **Launch Column selector** under **Join key columns for L**. Set the Join key columns for L to include OriginAirportCode, Month, DayofMonth, and CRSDepHour, and select the check box in the bottom right.



Select columns

BY NAME
WITH RULES

AVAILABLE COLUMNS

All Types search columns

OriginLatitude
OriginLongitude
DayOfWeek
Carrier
DestAirportCode
DestLatitude
DestLongitude
DepDel15

8 columns available

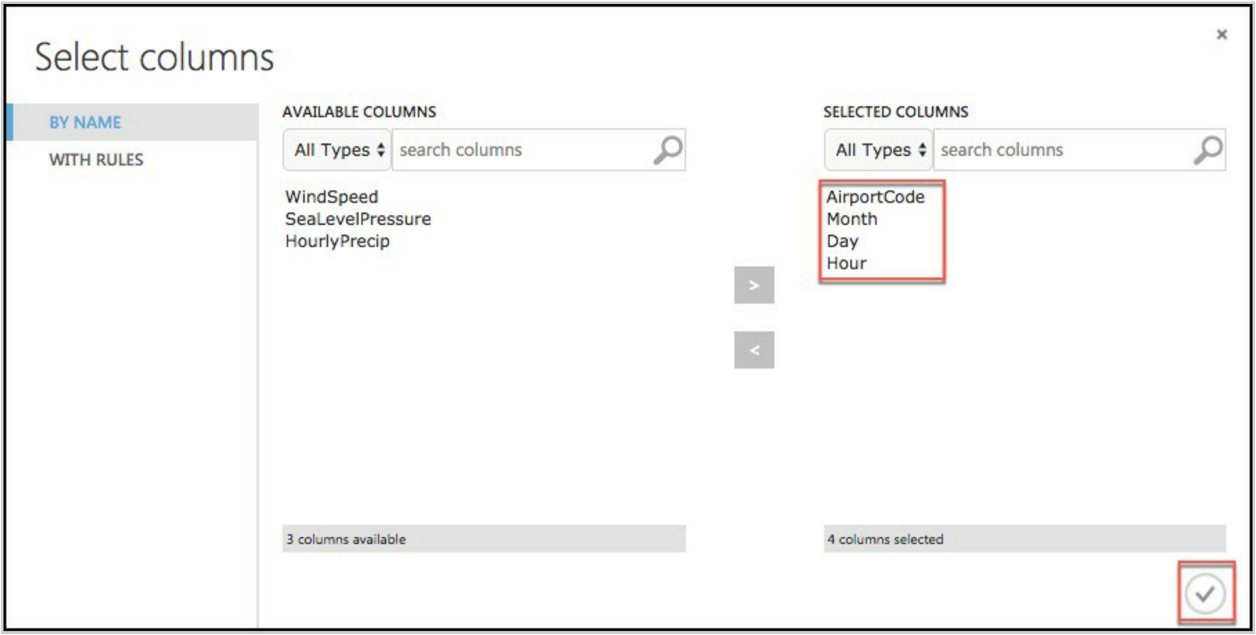
SELECTED COLUMNS

All Types search columns

OriginAirportCode
Month
DayofMonth
CRSDepHour

4 columns selected

5. Select **Launch Column selector** under **Join key columns for R**. Set the join key columns for R to include AirportCode, Month, Day, and Hour, and select the check box in the bottom right.


Select columns

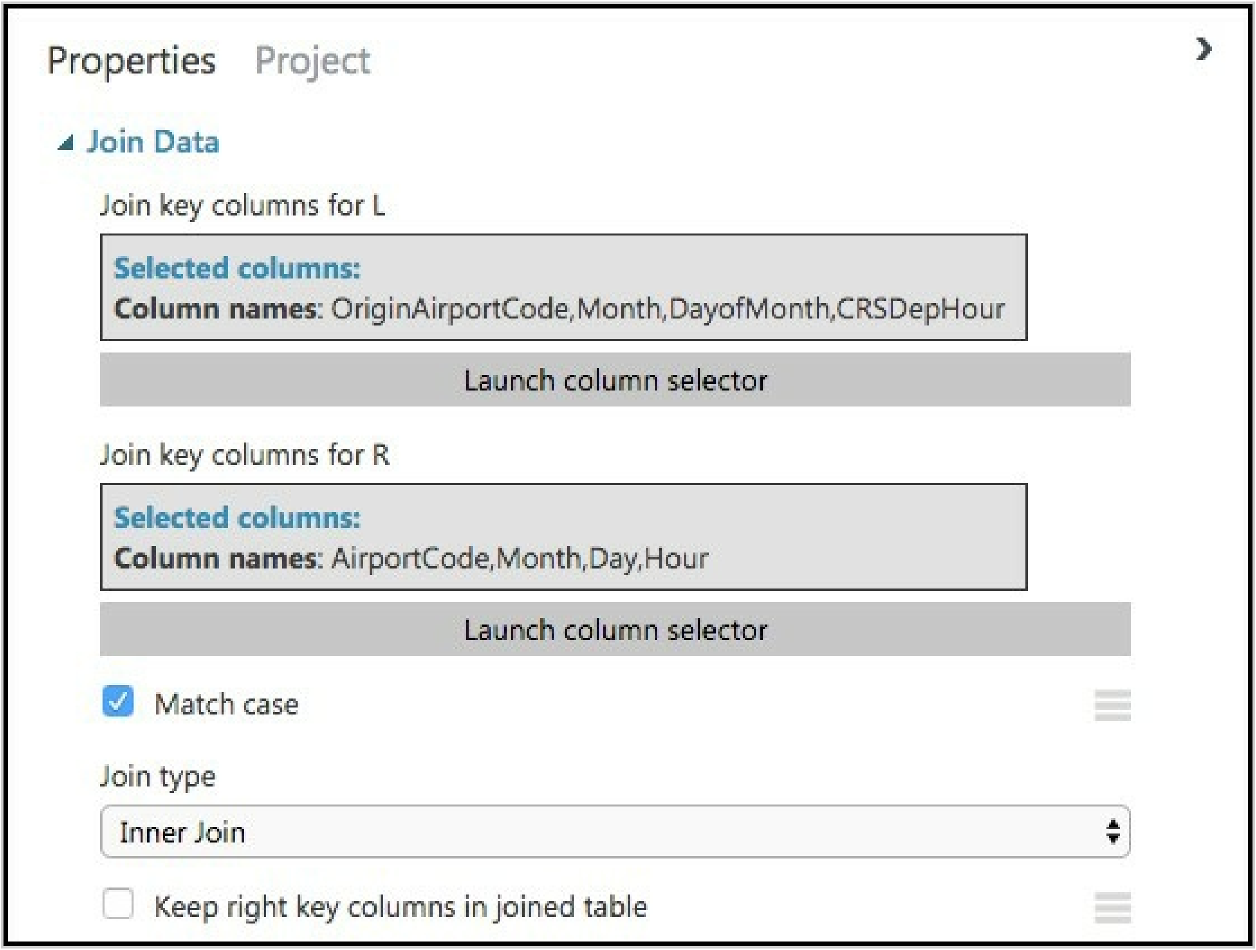
AVAILABLE COLUMNS: WindSpeed, SeaLevelPressure, HourlyPrecip

SELECTED COLUMNS: AirportCode, Month, Day, Hour

3 columns available

4 columns selected

6. Leave the Join Type at Inner Join, and uncheck **Keep right key columns in joined table** (so that we do not include the redundant values of AirportCode, Month, Day, and Hour).



Join Data

Join key columns for L

Selected columns:
Column names: OriginAirportCode,Month,DayofMonth,CRSDepHour

Join key columns for R

Selected columns:
Column names: AirportCode,Month,Day,Hour

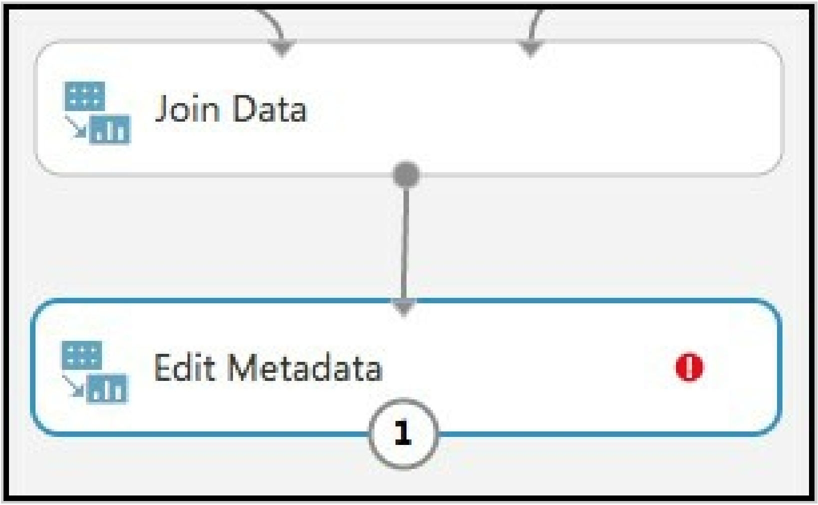
☑ Match case

Join type: Inner Join

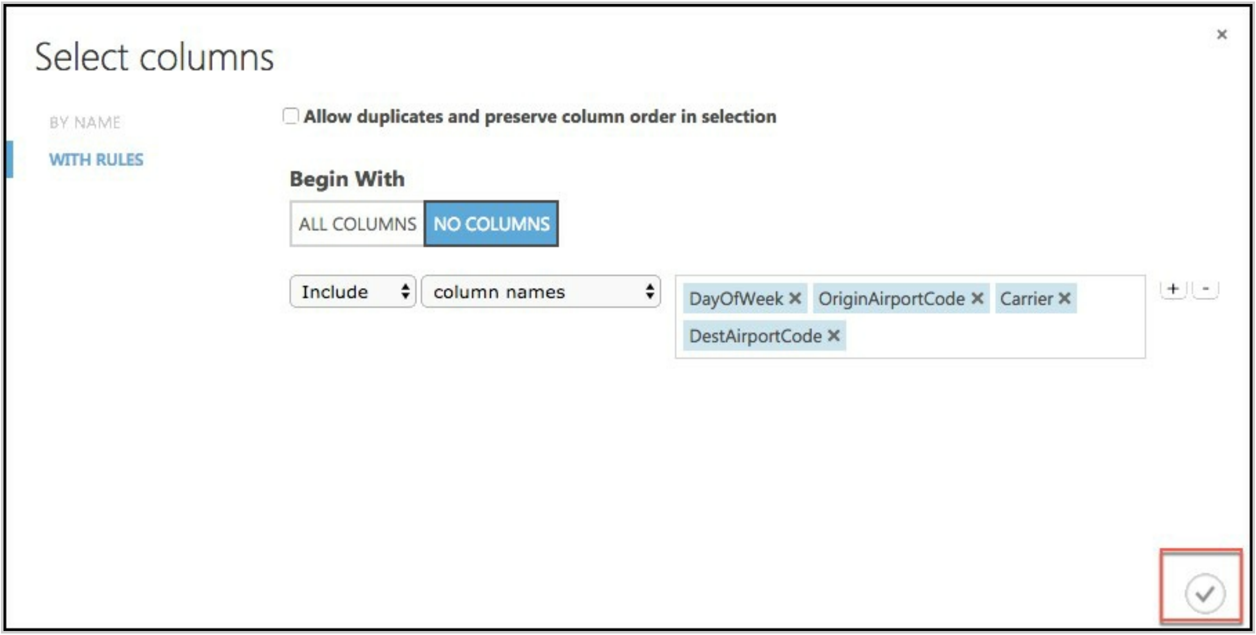
☐ Keep right key columns in joined table

7. Next, drag an **Edit Metadata** module onto the design surface below the Join Data module, and connect its input port to the output port of the Join Data module. We will use this module to convert the fields that were unbounded String feature types, to the enumeration like

Categorical feature.



8. On the **Properties** panel of the Edit Metadata module, select **Launch column selector** and set the Selected columns to DayOfWeek, Carrier, DestAirportCode, and OriginAirportCode, and select the checkbox in the bottom right.



Select columns

BY NAME

WITH RULES

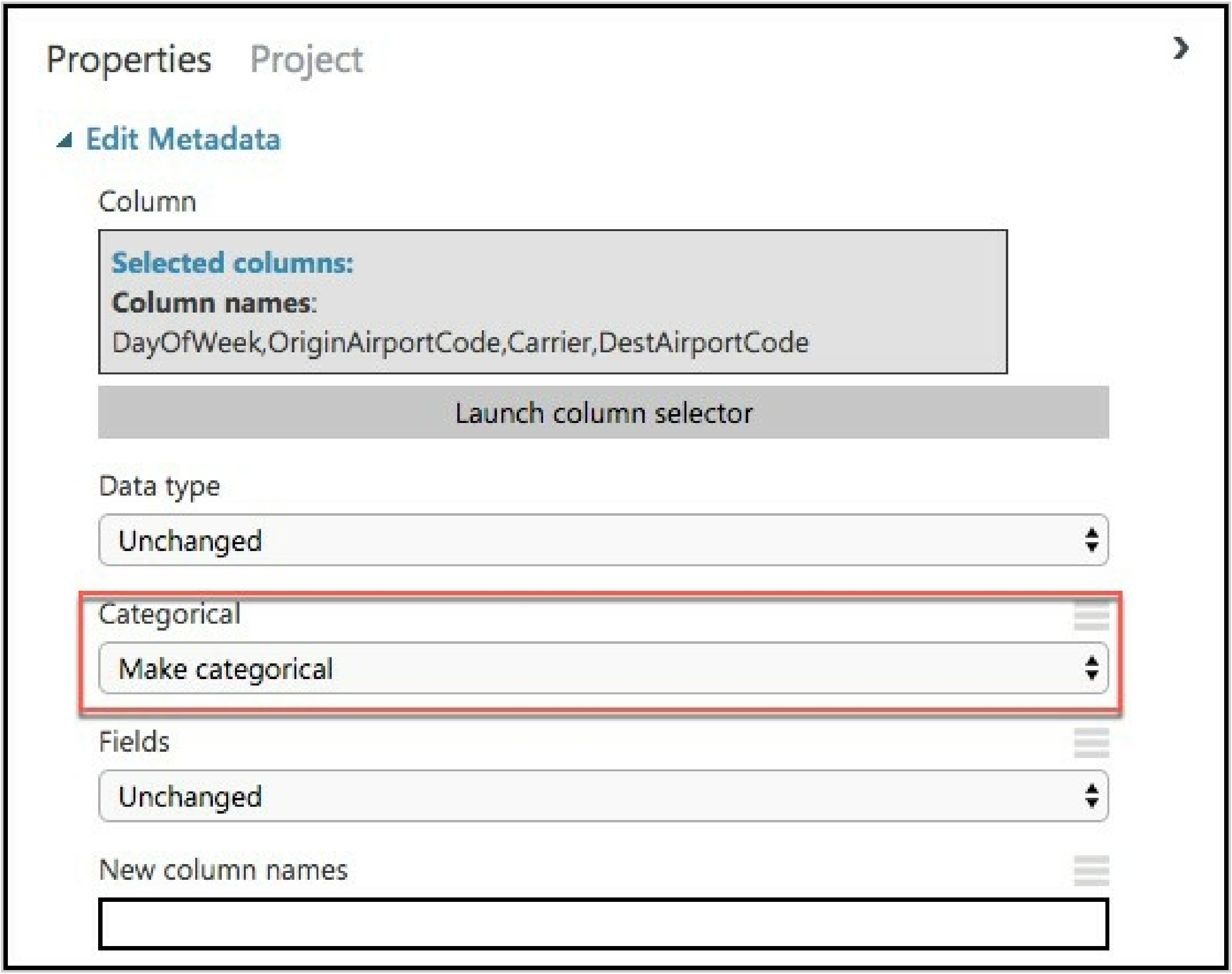
□ Allow duplicates and preserve column order in selection

Begin With

ALL COLUMNS NO COLUMNS

Include column names DayOfWeek ✕ OriginAirportCode ✕ Carrier ✕ DestAirportCode ✕

9. Set the Categorical drop down to **Make categorical**.



Properties Project

Edit Metadata

Column

Selected columns:
Column names: DayOfWeek,OriginAirportCode,Carrier,DestAirportCode

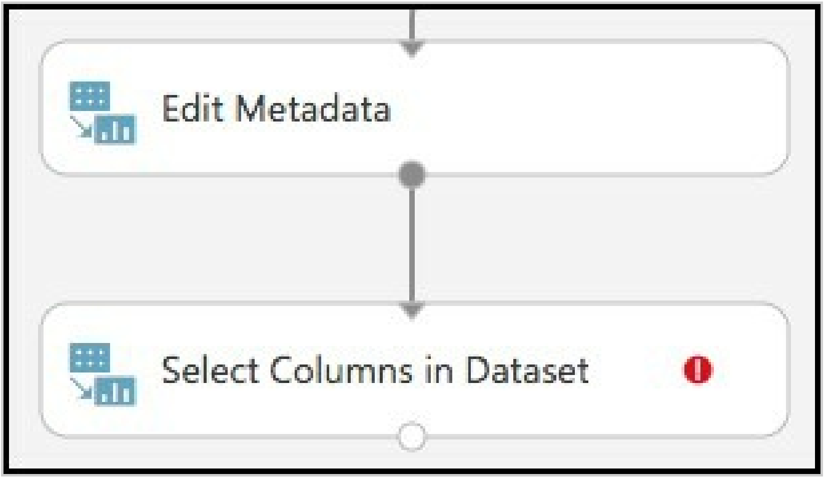
Data type: Unchanged

Categorical: Make categorical

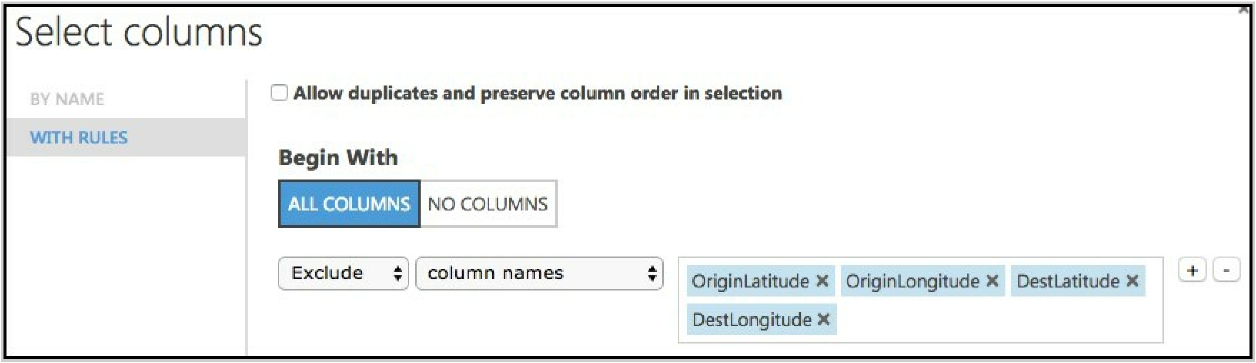
Fields: Unchanged

New column names

10. Drag a **Select Columns in Dataset** module onto the design surface, below the Edit Metadata module. Connect the output of the Edit Metadata module to the input of the Select Columns in Dataset module.



11. Launch the column selector, and choose Begin With **All Columns**, choose **Exclude** and set the selected columns to exclude: OriginLatitude, OriginLongitude, DestLatitude, and DestLongitude.



Select columns

BY NAME

WITH RULES

☐ Allow duplicates and preserve column order in selection

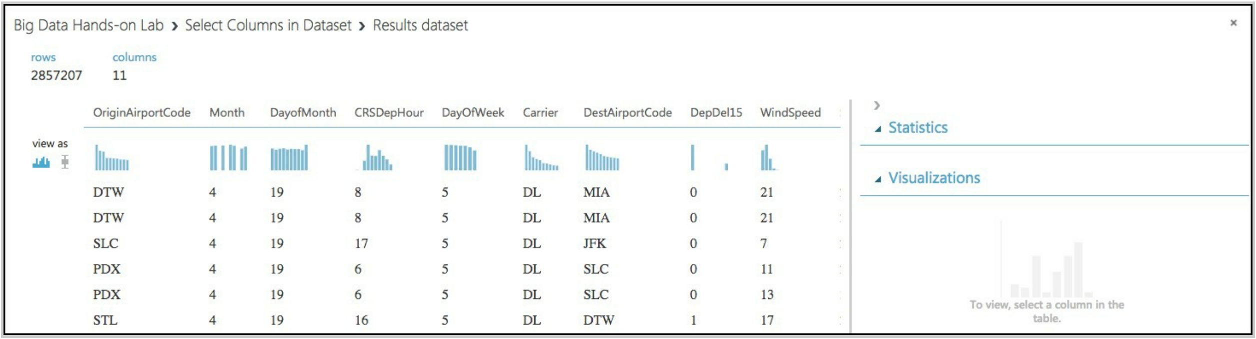
Begin With

ALL COLUMNS | NO COLUMNS

Exclude | column names | OriginLatitude ✕ OriginLongitude ✕ DestLatitude ✕ DestLongitude ✕

12. Save your experiment.

13. Run the experiment to verify everything works as expected and when completed, Visualize by right-clicking on the output of the Select Columns in Dataset module. You will see the joined datasets as output.



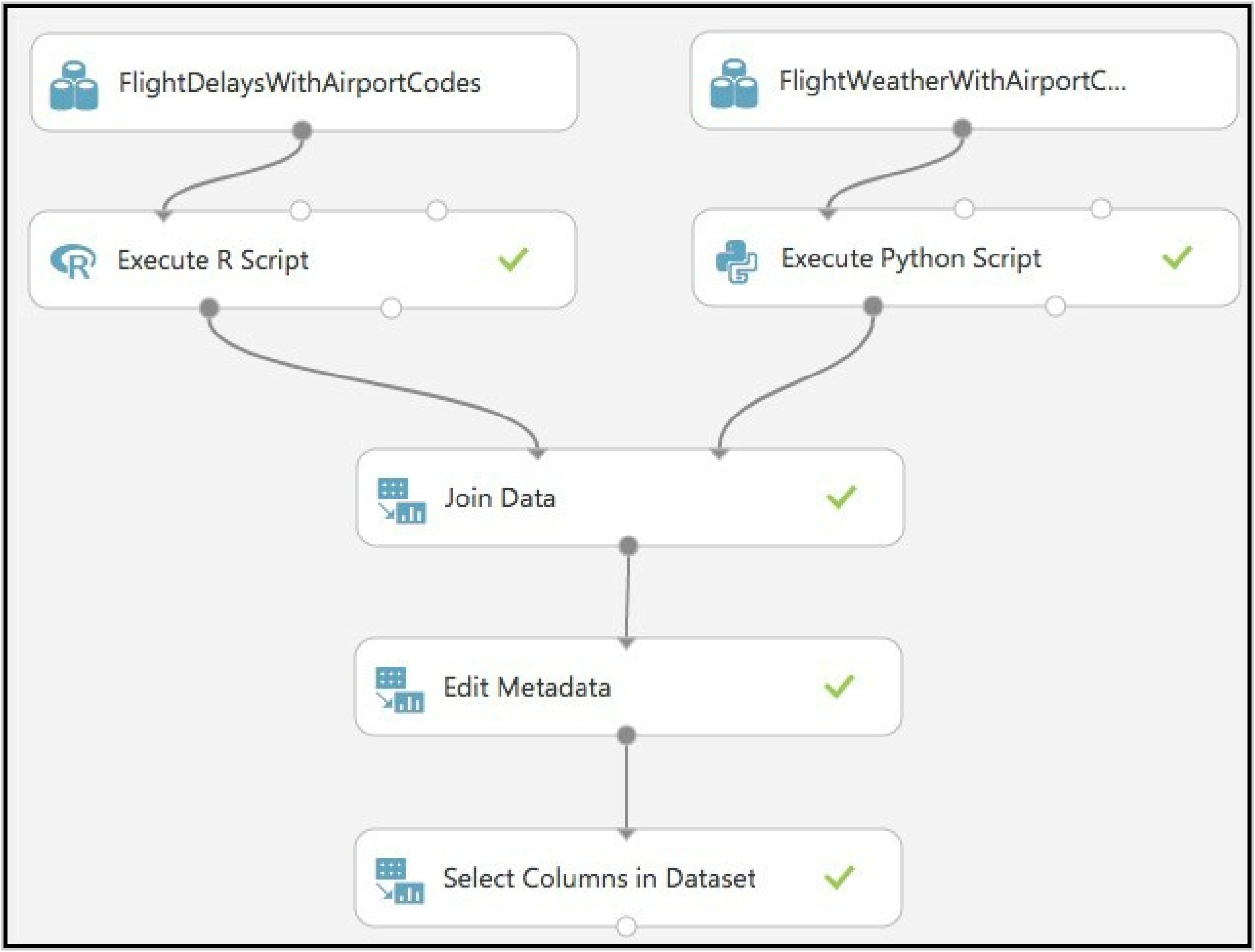
Big Data Hands-on Lab › Select Columns in Dataset › Results dataset

rows 2857207

columns 11

OriginAirportCode	Month	DayofMonth	CRSDepHour	DayOfWeek	Carrier	DestAirportCode	DepDel15	WindSpeed
DTW	4	19	8	5	DL	MIA	0	21
DTW	4	19	8	5	DL	MIA	0	21
SLC	4	19	17	5	DL	JFK	0	7
PDX	4	19	6	5	DL	SLC	0	11
PDX	4	19	6	5	DL	SLC	0	13
STL	4	19	16	5	DL	DTW	1	17

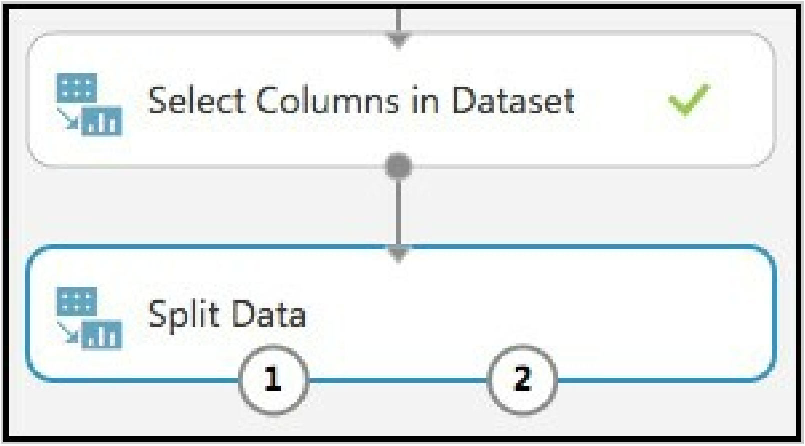
14. The model should now look like the following.


FlightDelaysWithAirportCodes
FlightWeatherWithAirportC...
Execute R Script ✓
Execute Python Script ✓
Join Data ✓
Edit Metadata ✓
Select Columns in Dataset ✓

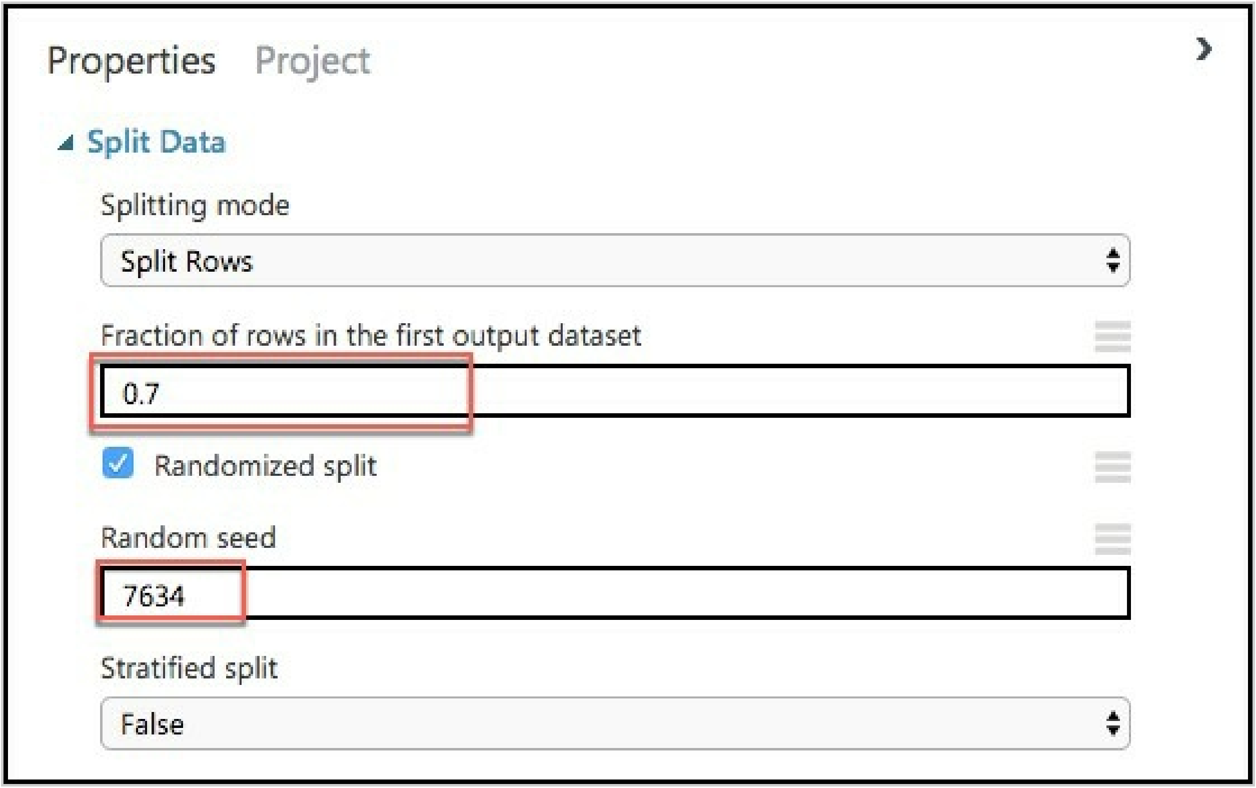
Task 7: Train the model

AdventureWorks Travel wants to build a model to predict if a departing flight will have a 15-minute or greater delay. In the historical data they have provided, the indicator for such a delay is found within the DepDel15 (where a value of 1 means delay, 0 means no delay). To create a model that predicts such a binary outcome, we can choose from the various Two-Class modules that Azure ML offers. For our purposes, we begin with a Two-Class Logistic Regression. This type of classification module needs to be first trained on sample data that includes the features important to making a prediction and must also include the actual historical outcome for those features. The typical pattern is to split the historical data so a portion is shown to the model for training purposes, and another portion is reserved to test just how well the trained model performs against examples it has not seen before.

1. To create our training and validation datasets, drag a **Split Data** module beneath Select Columns in Dataset, and connect the output of the Select Columns in Dataset module to the input of the Split Data module.



2. On the **Properties** panel for the Split Data module, set the Fraction of rows in the first output dataset to **0.7** (so 70% of the historical data will flow to output port 1). Set the Random seed to **7634**.



Split Data

Splitting mode: Split Rows

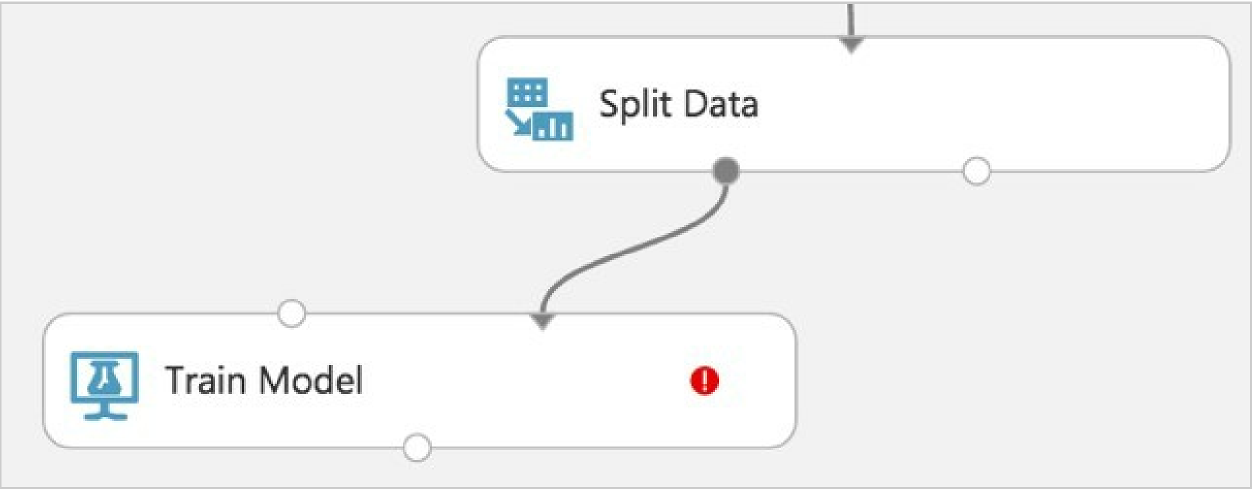
Fraction of rows in the first output dataset: 0.7

☑ Randomized split

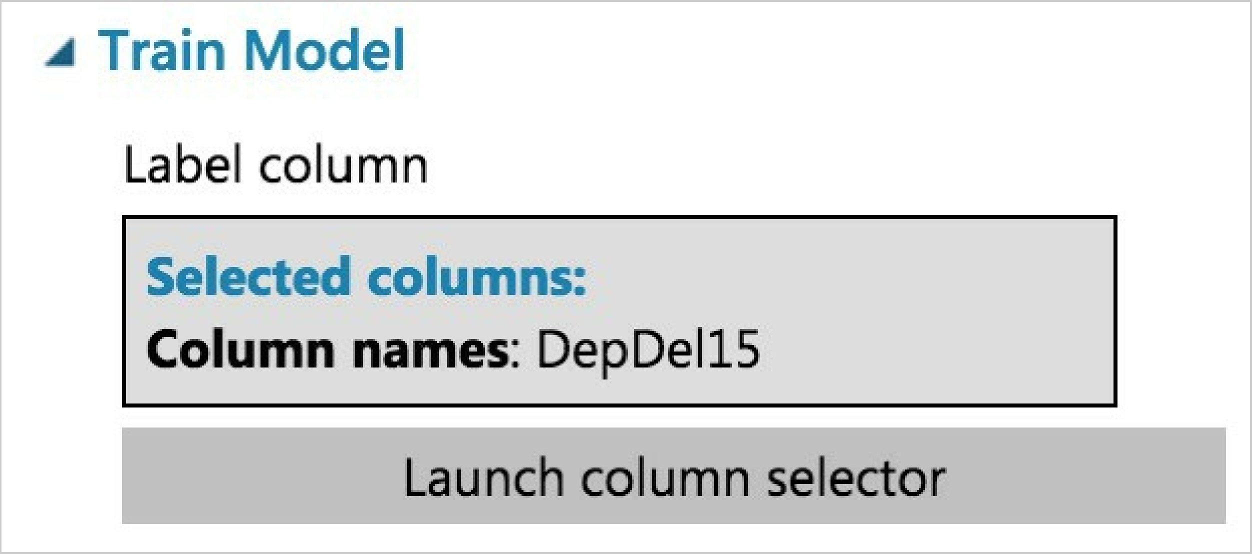
Random seed: 7634

Stratified split: False

3. Next, add a Train Model module and connect it to output 1 of the Split Data module.



4. On the **Properties** panel for the Train Model module, set the Selected columns to **DepDel15**.



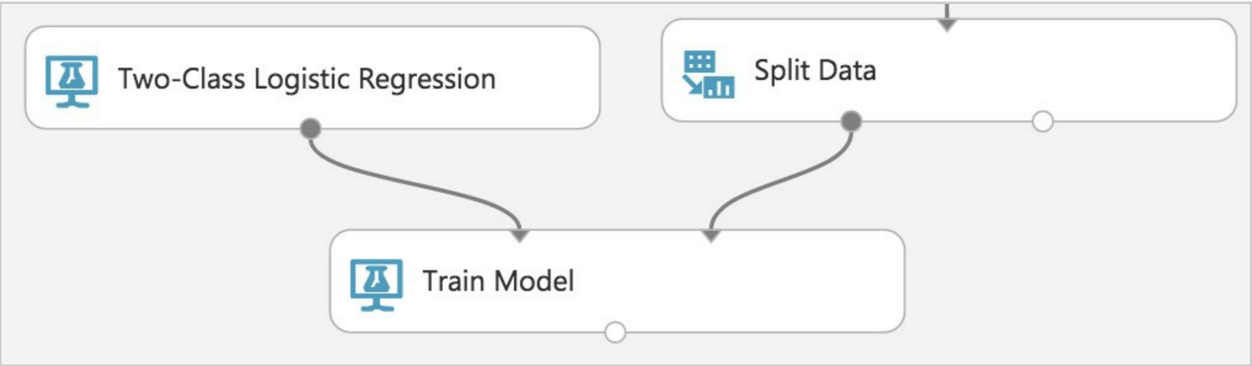
Train Model

Label column

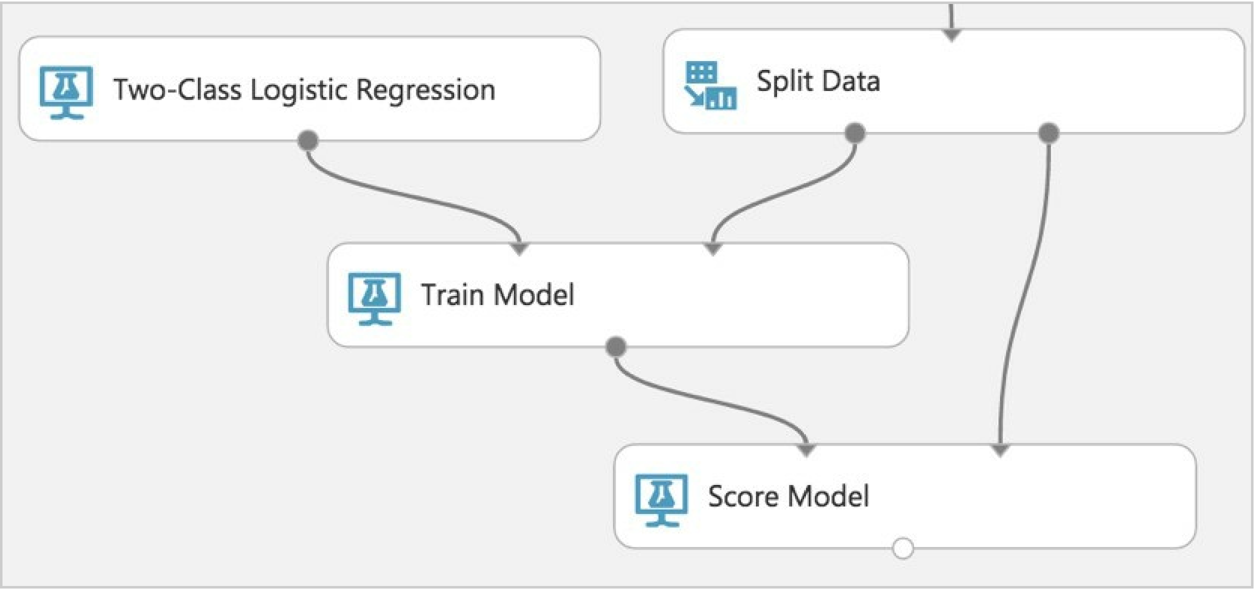
Selected columns:

Column names: DepDel15

5. Drag a **Two-Class Logistic Regression** module above and to the left of the Train Model module and connect the output to the leftmost input of the Train Model module



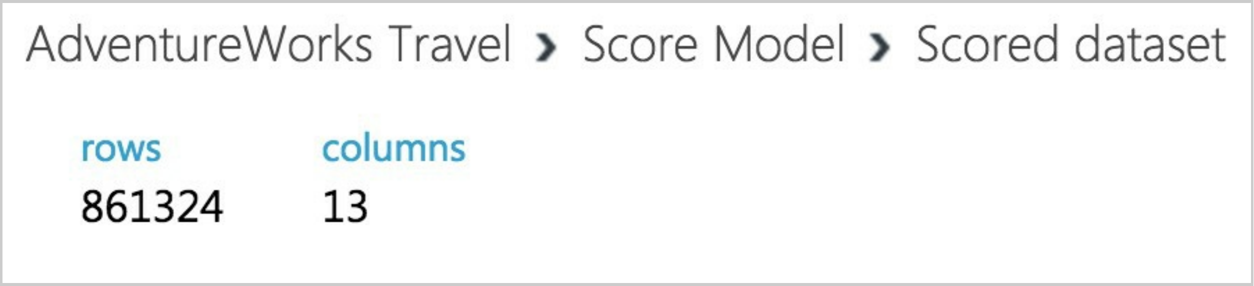
6. Below the Train Model drop a **Score Model** module. Connect the output of the Train Model module to the leftmost input port of the Score Model and connect the rightmost output of the Split Data module to the rightmost input of the Score Model.



7. Save the experiment.

8. Run the experiment.

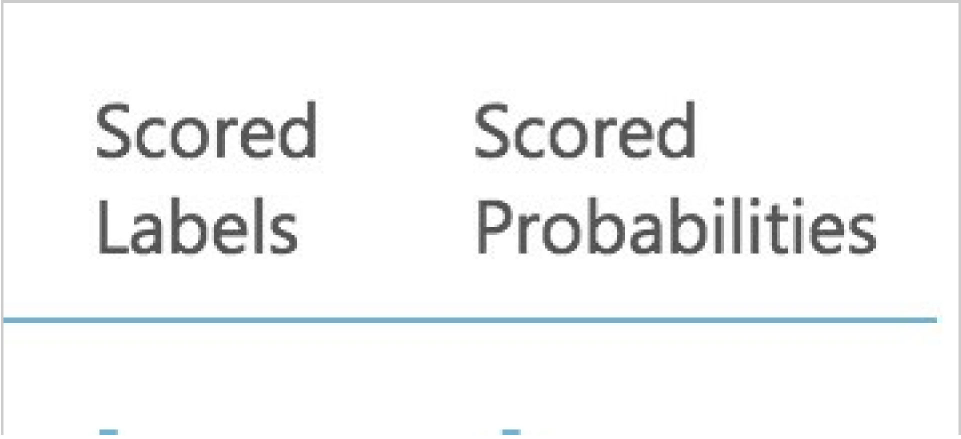
9. When the experiment is finished running (which takes a few minutes), right-click on the output port of the Score Model module and select **Visualize** to see the results of its predictions. **You should have a total of 13 columns**.



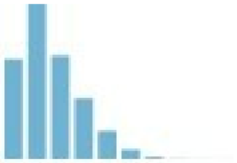
AdventureWorks Travel › Score Model › Scored dataset

rows	columns
861324	13

10. If you scroll to the right so that you can see the last two columns, observe there are **Scored Labels** and **Scored Probabilities** columns. The former is the prediction (1 for predicting delay, 0 for predicting no delay) and the latter is the probability of the prediction. In the following screenshot, for example, the last row shows a delay predication with a 53.1% probability.



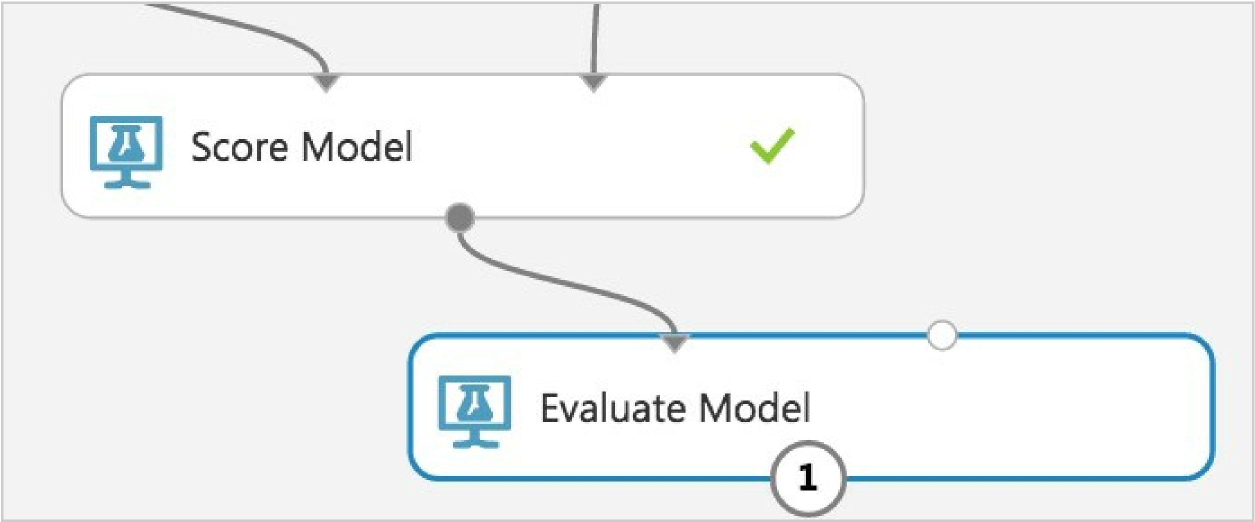
Scored Labels	Scored Probabilities



0	0.224147
0	0.182701
0	0.073537
0	0.270637
0	0.267242
0	0.191029
0	0.095636
0	0.207545
0	0.22024
0	0.082833
1	0.531273

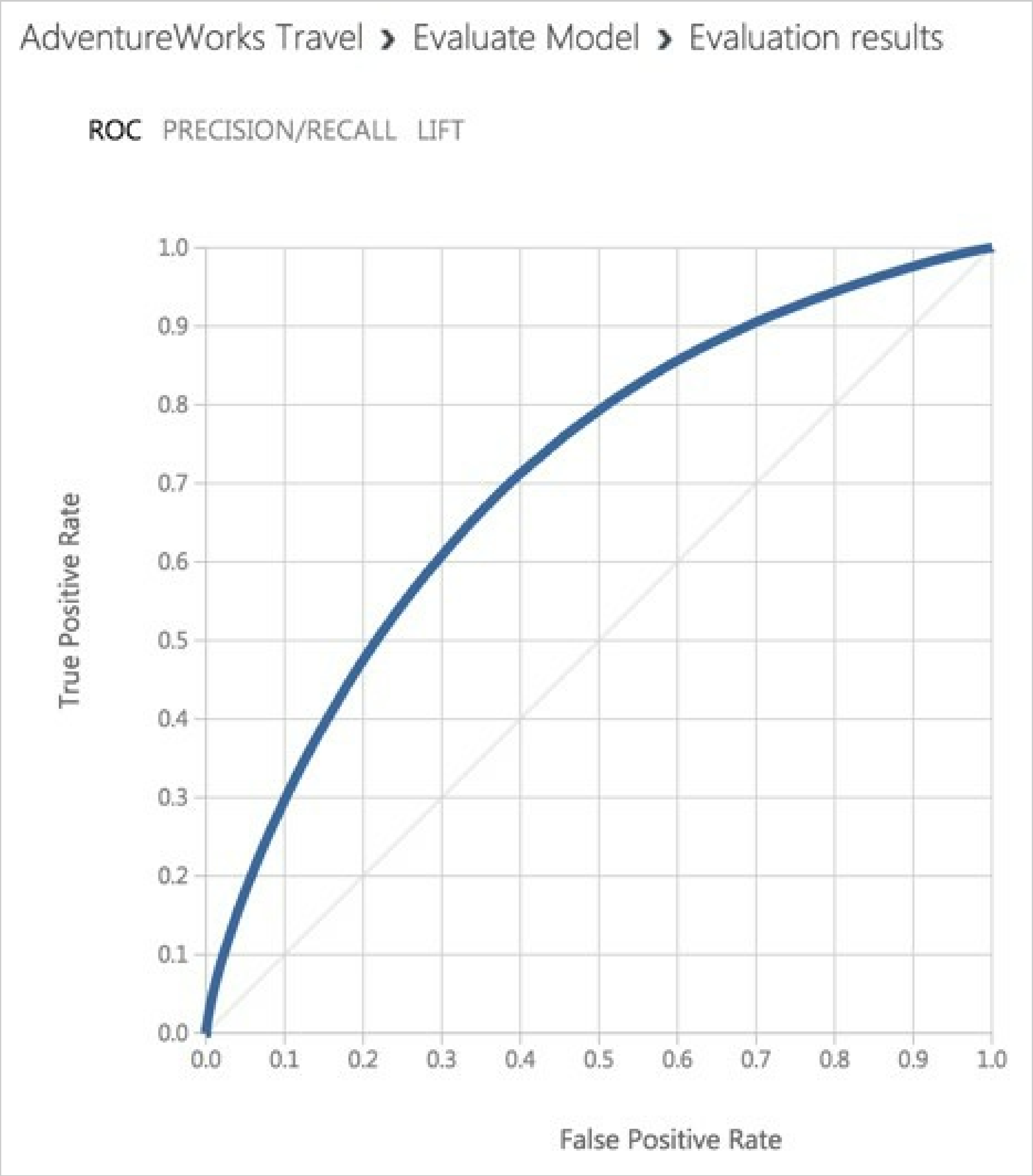
11. While this view enables you to see the prediction results for the first 100 rows, if you want to get more detailed statistics across the prediction results to evaluate your model's performance, you can use the **Evaluate Model** module.

12. Drag an **Evaluate Model** module on to the design surface beneath the Score Model module. Connect the output of the Score Model module to the leftmost input of the Evaluate Model module.

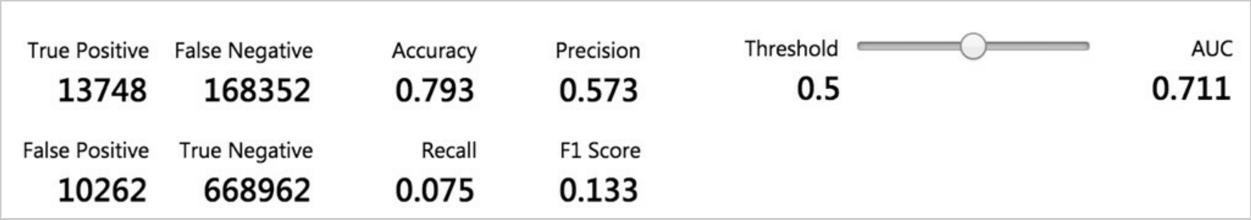


13. Run the experiment.

14. When the experiment is finished running, right-click the output of the Evaluate Model module and select **Visualize**. In this dialog box, you are presented with various ways to understand how your model is performing in the aggregate. While we will not cover how to interpret these results in detail, we can examine the ROC chart that tells us that at least our model (the blue curve) is performing better than random (the light gray straight line going from 0,0 to 1,1)—which is a good start for our first model!



AdventureWorks Travel › Evaluate Model › Evaluation results



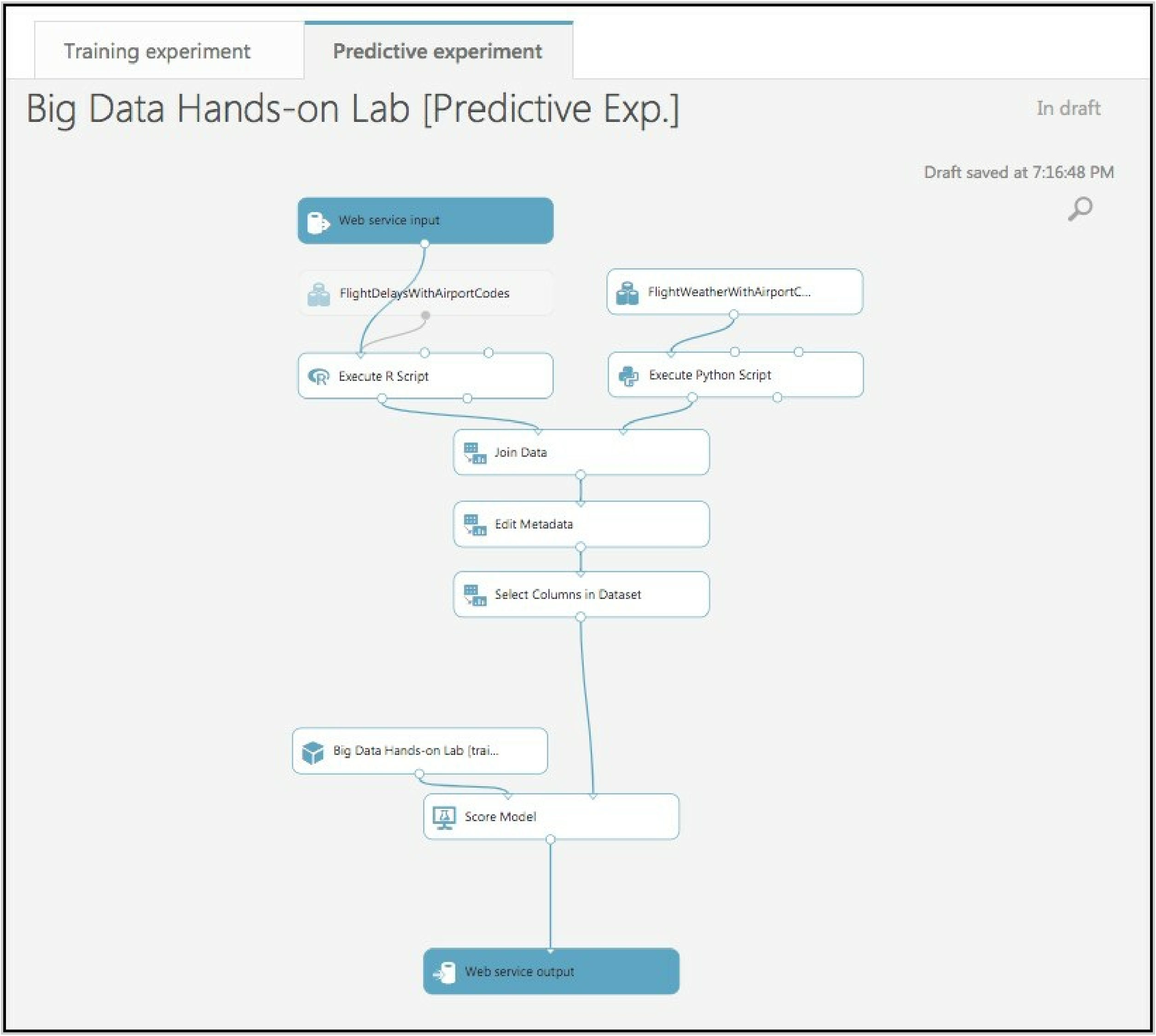
True Positive	False Negative	Accuracy	Precision	Threshold	AUC
13748	168352	0.793	0.573	0.5	0.711
False Positive	True Negative	Recall	F1 Score		
10262	668962	0.075	0.133		

Task 8: Operationalize the experiment

1. Now that we have a functioning model, let us package it up into a predictive experiment that can be called as a web service.

2. In the command bar at the bottom, select **Set Up Web Service** and then select **Predictive Web Service [Recommended]**. (If Predictive Web Service is grayed out, run the experiment again.

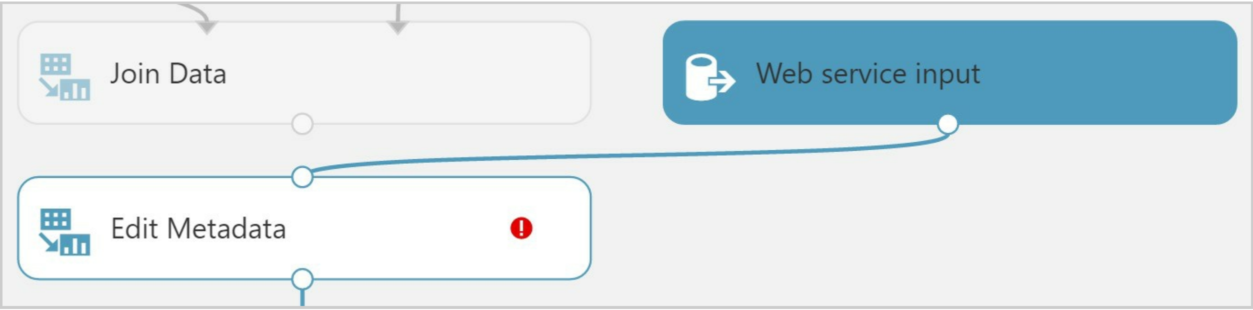
3. A copy of your training experiment is created, and a new tab labeled **Predictive Experiment** is added, which contains the trained model wrapped between web service input (e.g. the web service action you invoke with parameters) and web service output modules (e.g., how the result of scoring the parameters are returned).


Training experiment
Predictive experiment
Big Data Hands-on Lab [Predictive Exp.]
In draft
Draft saved at 7:16:48 PM
Web service input
FlightDelaysWithAirportCodes
FlightWeatherWithAirportC...
Execute R Script
Execute Python Script
Join Data
Edit Metadata
Select Columns in Dataset
Big Data Hands-on Lab [trai...
Score Model
Web service output

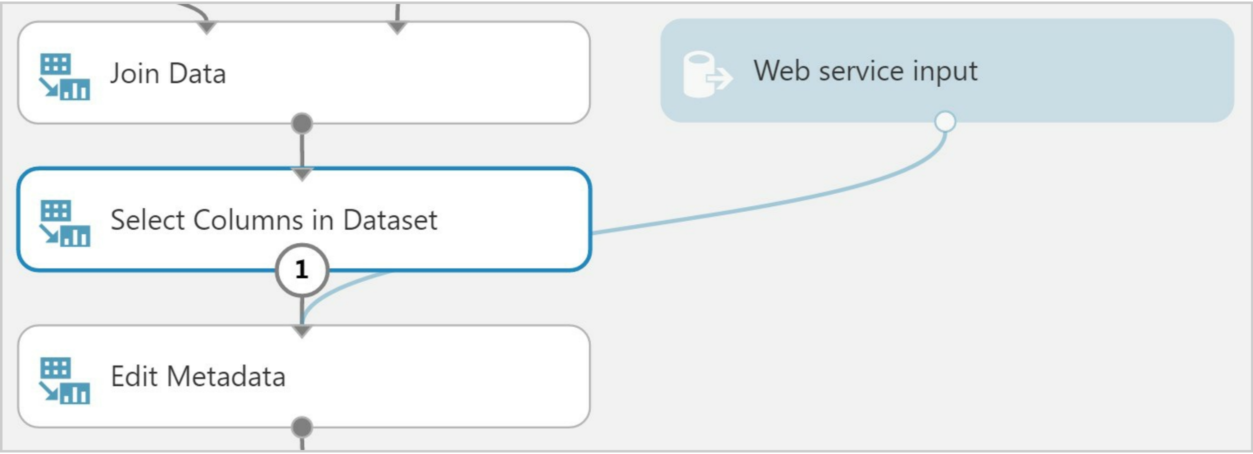
4. We will make some adjustments to the web service input and output modules to control the parameters we require and the results we return.

5. Move the **Web Service Input** module down, so it is to the right of the Join Data module. Connect the output of the Web service input module to input of the Edit Metadata module.

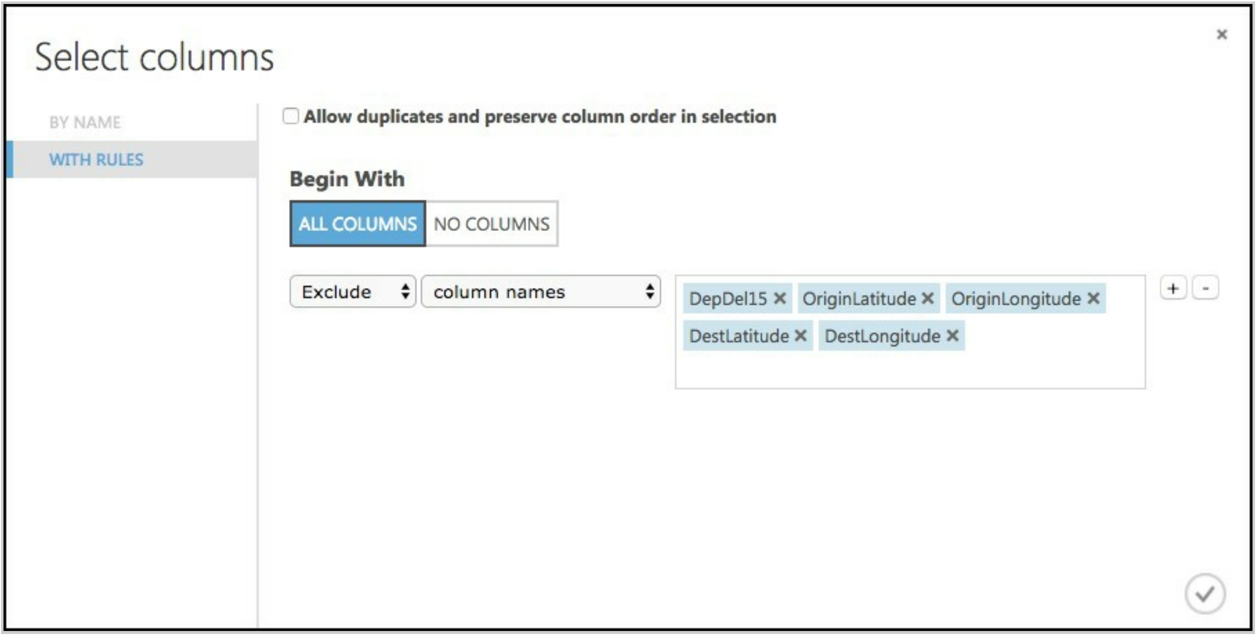
6. Right-click the line connecting the Join Data module and the Edit Metadata module and select **Delete**.



7. In between the Join Data and the Edit Metadata modules, drop a **Select Columns in Dataset** module. Connect the Join Data module's output to the Select Columns module's input, and the Select Columns output to the Edit Metadata module's input.



8. In the Properties panel for the Select Columns in Dataset module, set the Select columns to **All Columns**, and select **Exclude**. Enter columns **DepDel15, OriginLatitude, OriginLongitude, DestLatitude,** and **DestLongitude**.



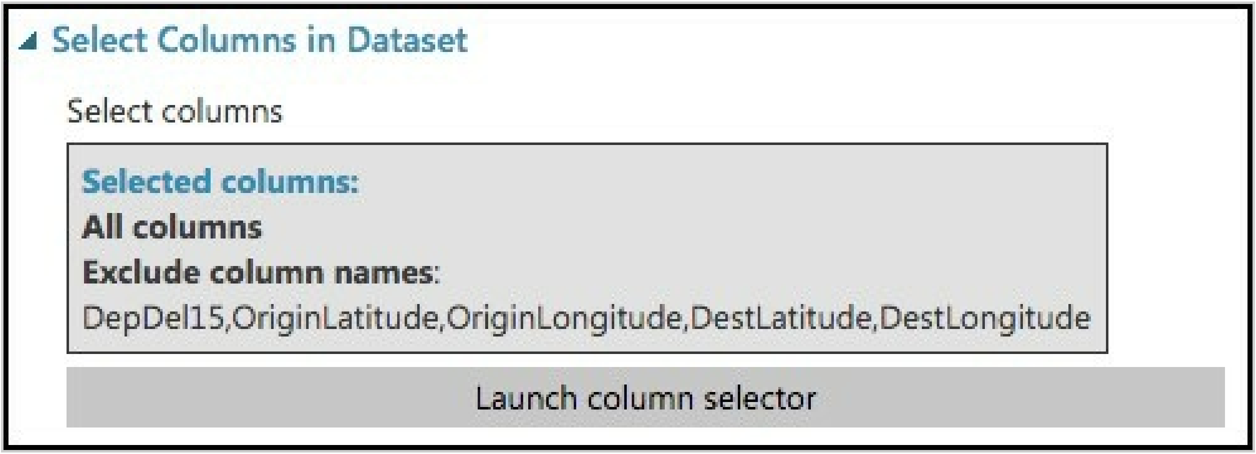
Select columns

□ Allow duplicates and preserve column order in selection

Begin With

Exclude | column names | DepDel15 ✕ OriginLatitude ✕ OriginLongitude ✕ DestLatitude ✕ DestLongitude ✕

9. This configuration will update the web service metadata so that these columns do not appear as required input parameters for the web service.



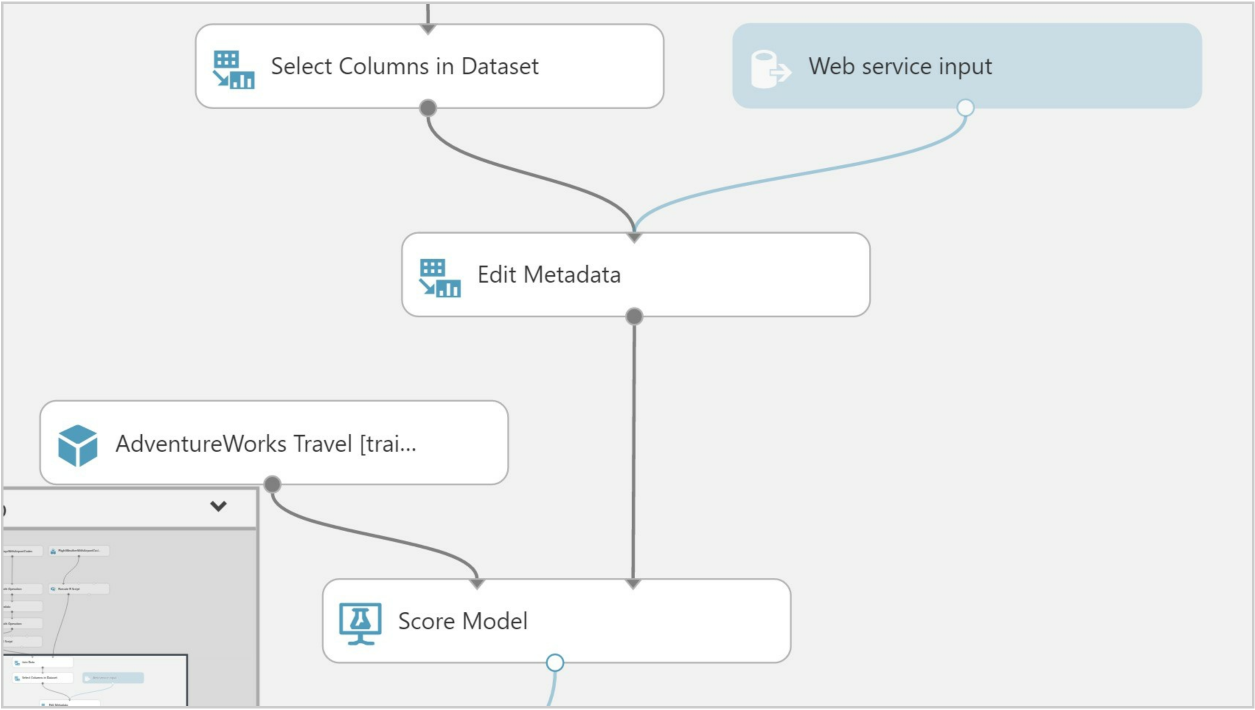
Select Columns in Dataset

Select columns

Selected columns:
All columns
Exclude column names:
DepDel15,OriginLatitude,OriginLongitude,DestLatitude,DestLongitude

10. Select the Select Columns in Dataset module that comes **after the Metadata Editor module**, and delete it.

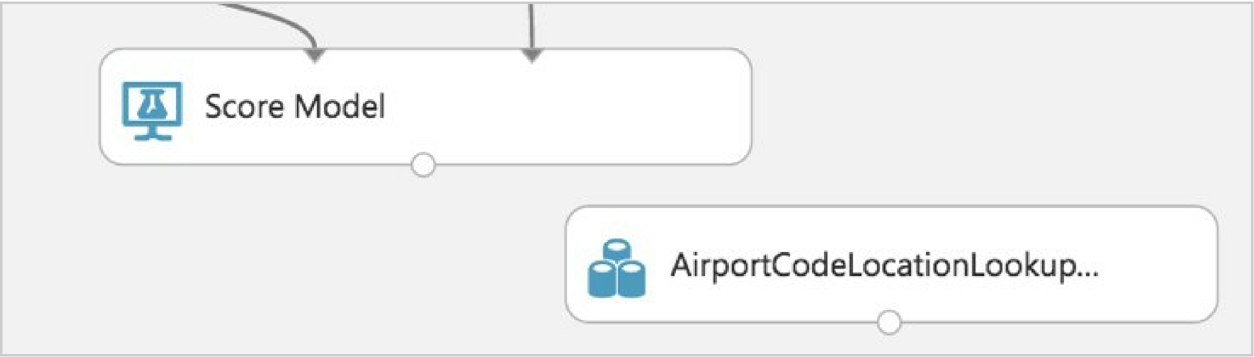
11. Connect the output of the Edit Metadata module directly to the right input of the Score Model module.



12. As we removed the latitude and longitude columns from the dataset to remove them as input to the web service, we have to add them back in before we return the result so that the results can be easily visualized on a map.

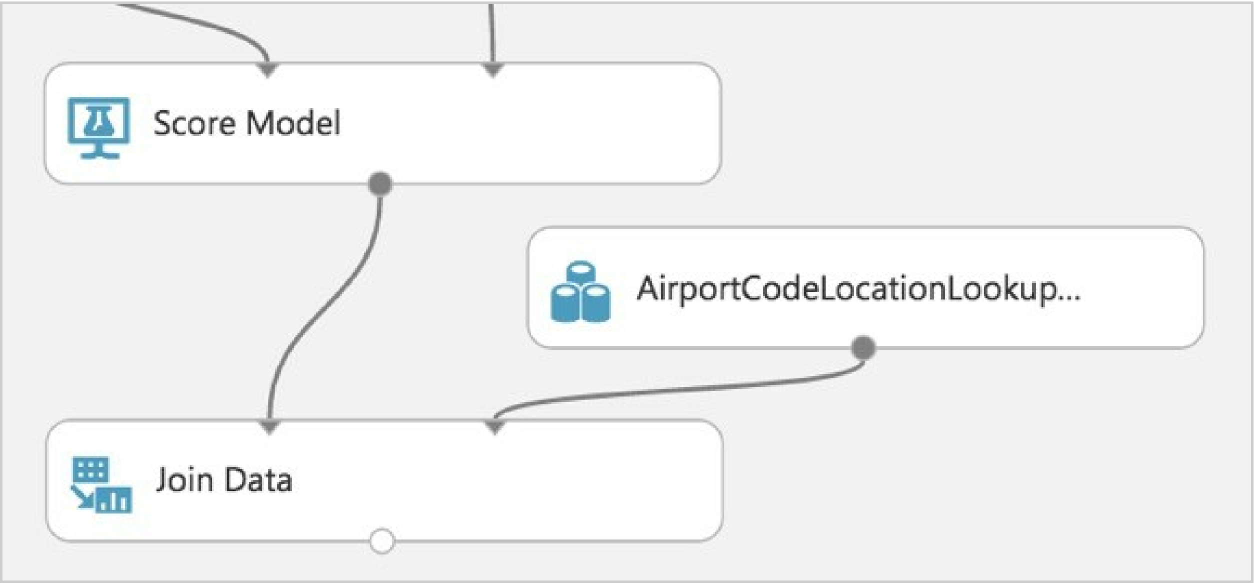
13. To add these fields back, begin by **deleting the line between the Score Model and Web service output**.

14. Drag the **AirportCodeLocationLookupClean** dataset on to the design surface, positioning it below and to the right of the Score Model module.

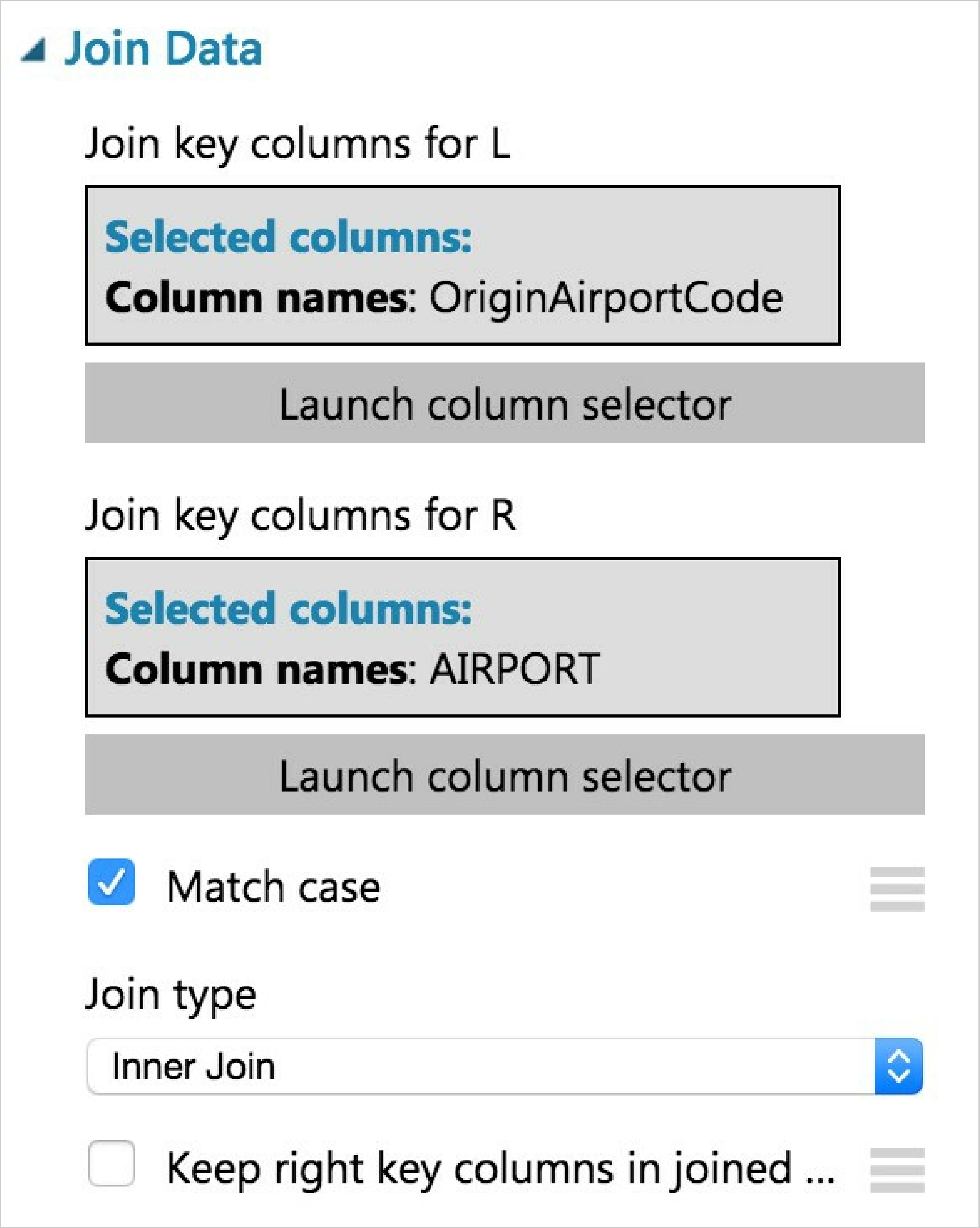


15. Add a **Join Data** module, and position it below and to the left of the AirportCodeLocationLookupClean module.

16. Connect the output of the **Score Model** module to the leftmost input of the **Join Data** module and the output of the **AirportCodeLocationLookupClean** dataset to the rightmost input of the **Join Data** module.



17. In the **Properties** panel for the Join Data module, for the Join key columns for L set the selected columns to **OriginAirportCode**. For the Join key columns for R, set the Selected columns to **AIRPORT**. Uncheck Keep right key columns in joined table.



◢ Join Data

Join key columns for L

Selected columns:
Column names: OriginAirportCode

Join key columns for R

Selected columns:
Column names: AIRPORT

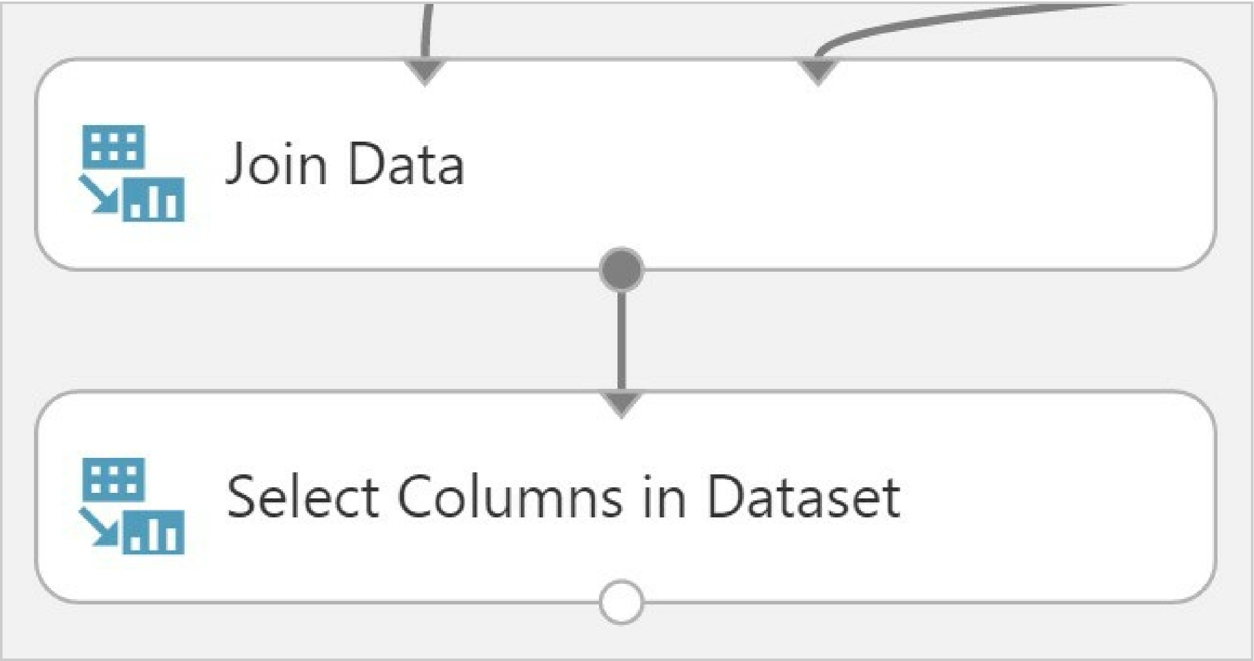
☑ Match case

Join type

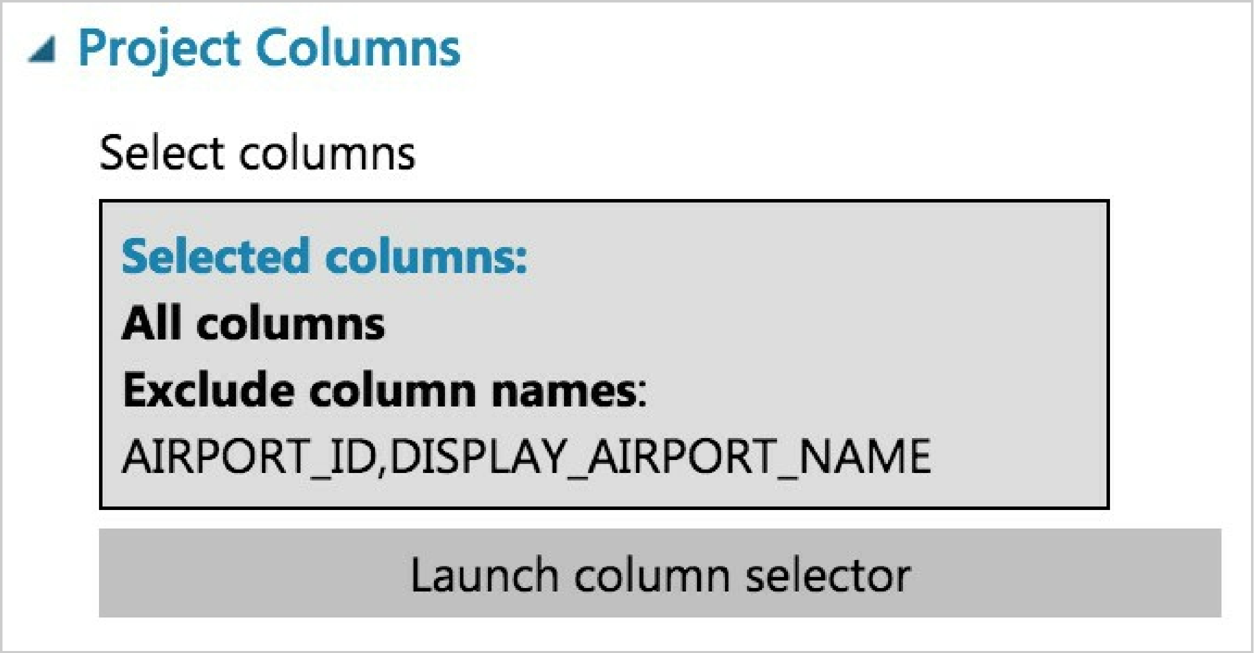
Inner Join

☐ Keep right key columns in joined ...

18. Add a **Select Columns in Dataset** module beneath the Join Data module. Connect the Join Data output to the input of the Select Columns in Dataset module.



19. In the **Property** panel, begin with **All Columns**, and set the Selected columns to **Exclude** the columns: **AIRPORT_ID** and **DISPLAY_AIRPORT_NAME**.

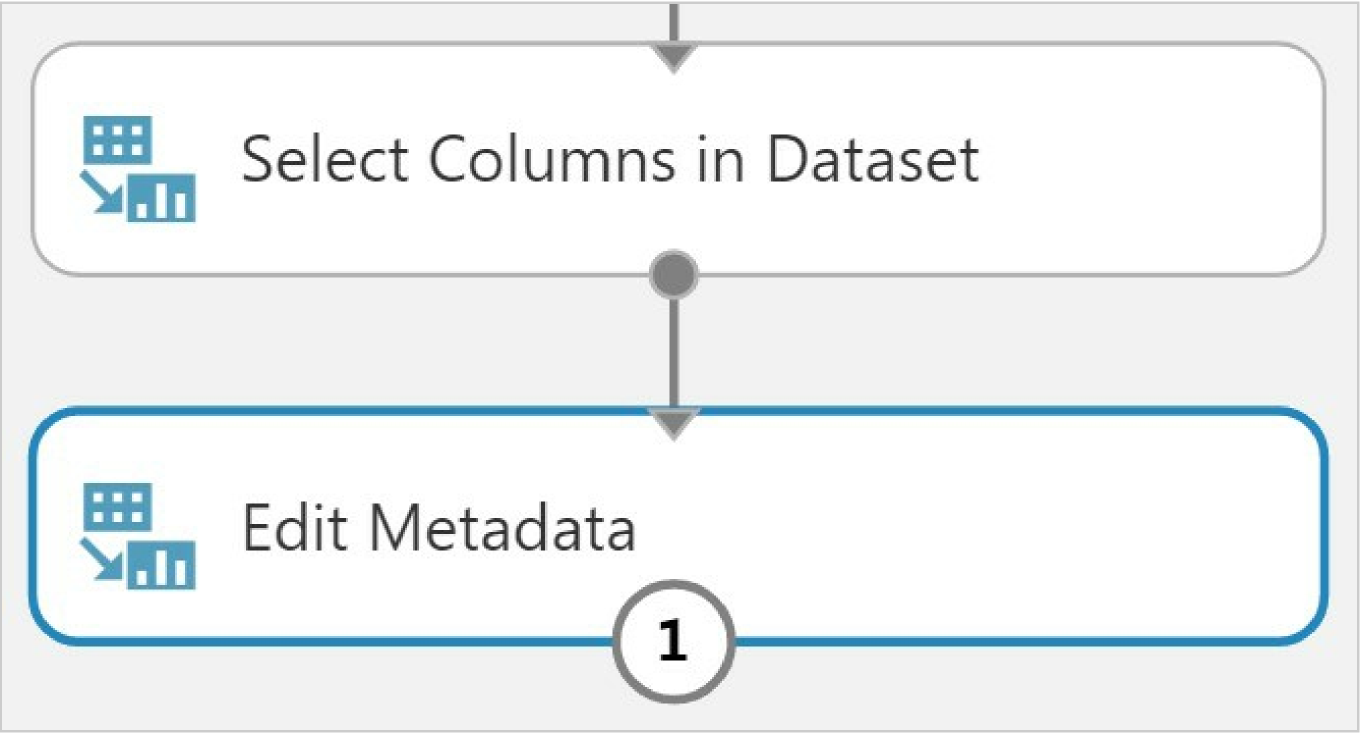


Project Columns

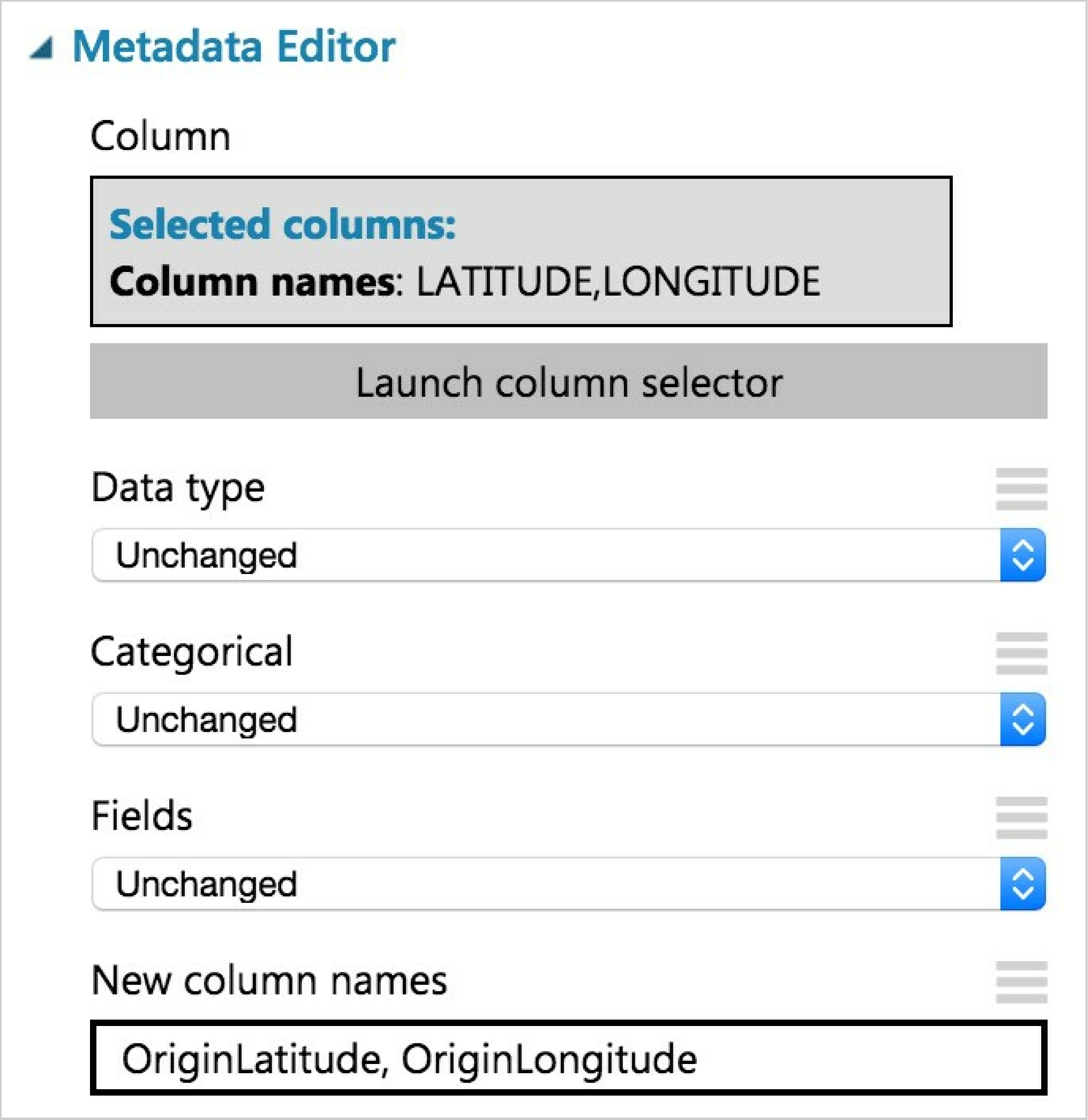
Select columns

Selected columns:
All columns
Exclude column names:
AIRPORT_ID,DISPLAY_AIRPORT_NAME

20. Add an **Edit Metadata** module. Connect the output of the Select Columns in Dataset module to the input of the Edit Metadata module.



21. In the Properties panel for the Metadata Editor, use the column selector to set the Selected columns to **LATITUDE** and **LONGITUDE**. In the New column names enter: **OriginLatitude**, **OriginLongitude**.



Column

Selected columns:

Column names: LATITUDE,LONGITUDE

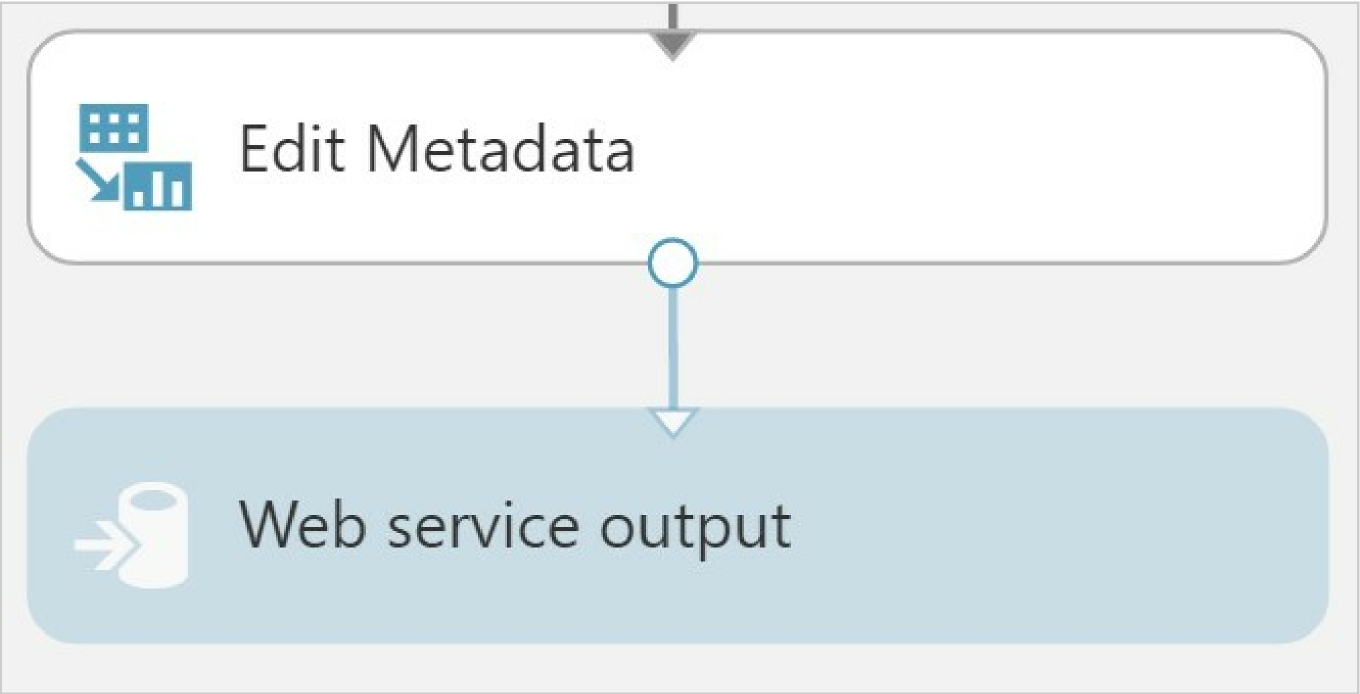
Data type: Unchanged

Categorical: Unchanged

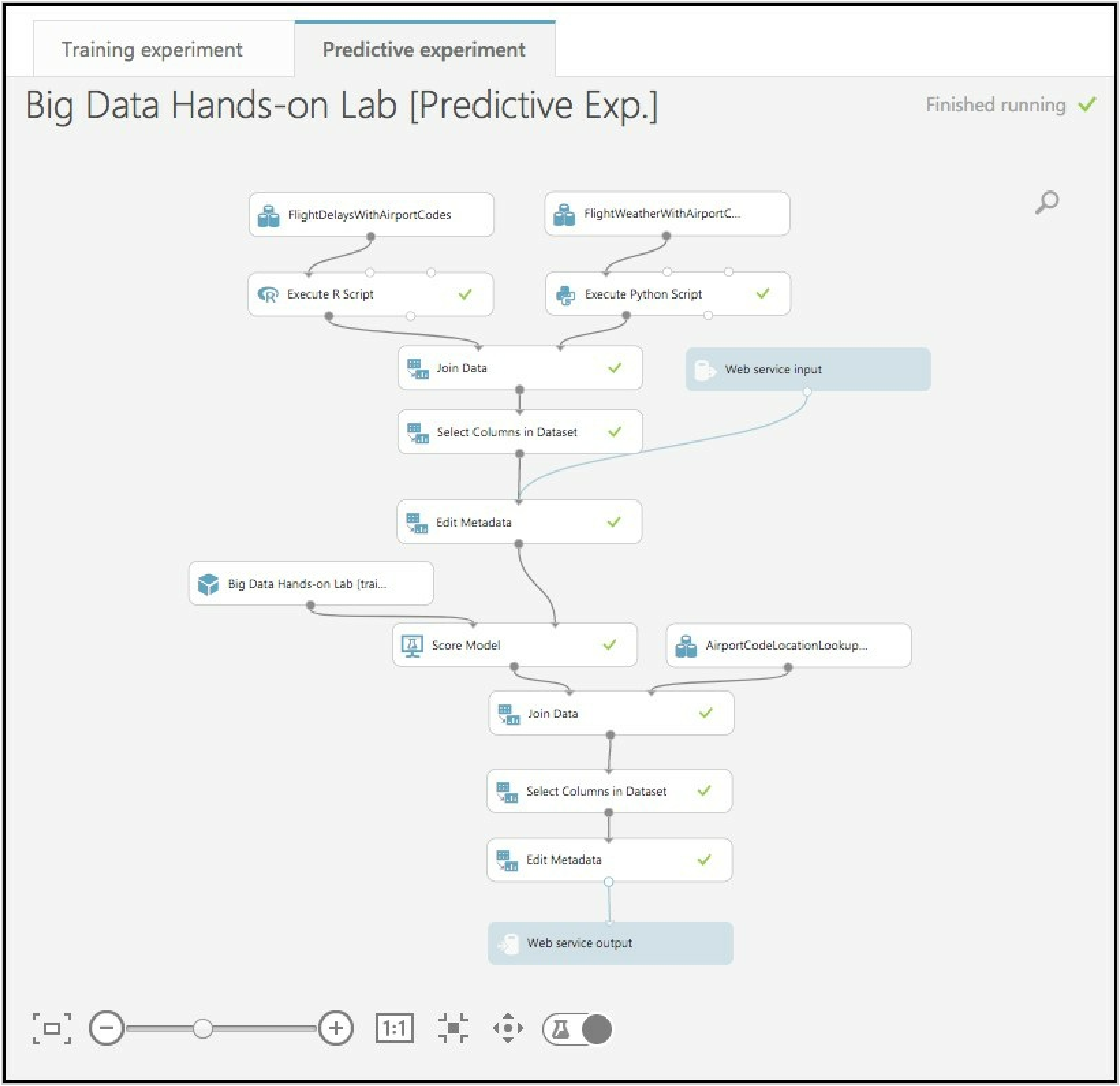
Fields: Unchanged

New column names: OriginLatitude, OriginLongitude

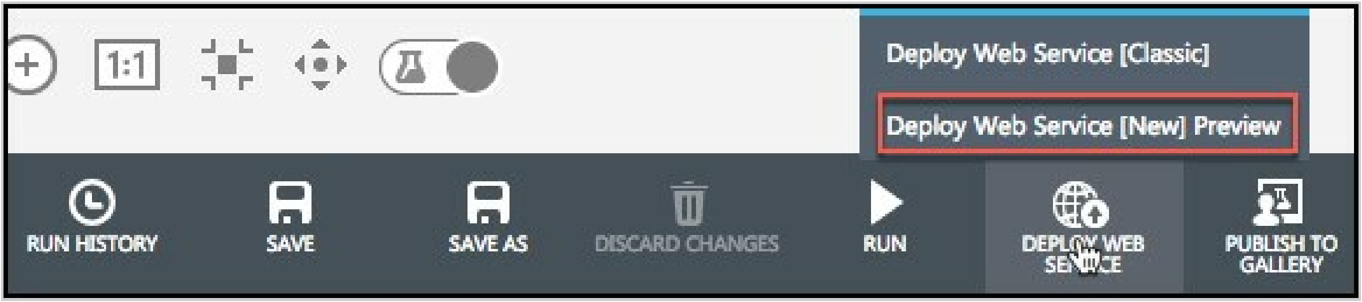
22. Connect the output of the Edit Metadata to the input of the web service output module.



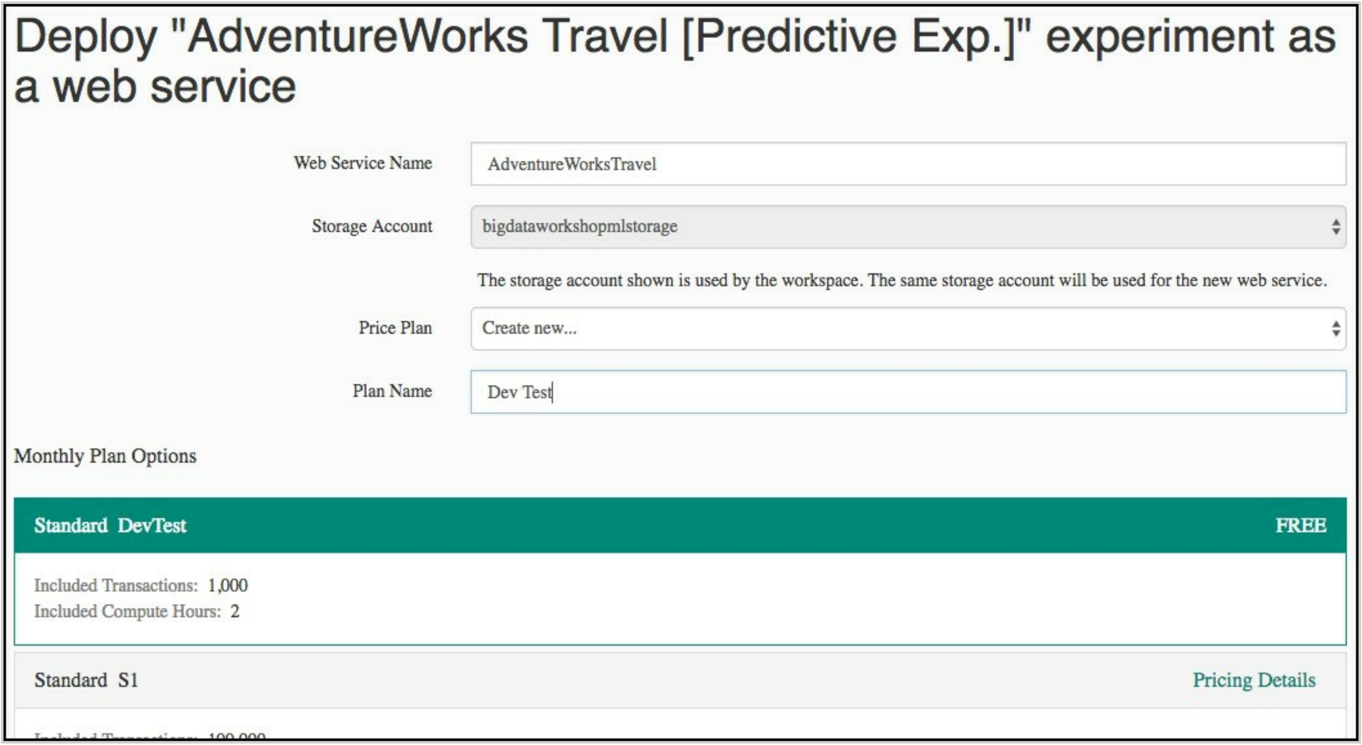
23. Run the experiment.


Training experiment
Predictive experiment
Big Data Hands-on Lab [Predictive Exp.]
Finished running ✓
FlightDelaysWithAirportCodes
FlightWeatherWithAirportC...
Execute R Script
Execute Python Script
Join Data
Web service input
Select Columns in Dataset
Edit Metadata
Big Data Hands-on Lab [trai...
Score Model
AirportCodeLocationLookup...
Join Data
Select Columns in Dataset
Edit Metadata
Web service output
1:1

24. When the experiment is finished running, select **Deploy Web Service, Deploy Web Service [NEW] Preview**.



25. On the Deploy experiment page, select **Create New…** in the Price Plan drop down, and enter **Dev Test** as the Plan Name. Select **Standard DevTest (FREE)** under Monthly Plan Options.



Deploy "AdventureWorks Travel [Predictive Exp.]" experiment as a web service

Web Service Name: AdventureWorksTravel

Storage Account: bigdataworkshopmlstorage

The storage account shown is used by the workspace. The same storage account will be used for the new web service.

Price Plan: Create new...

Plan Name: Dev Test

Monthly Plan Options

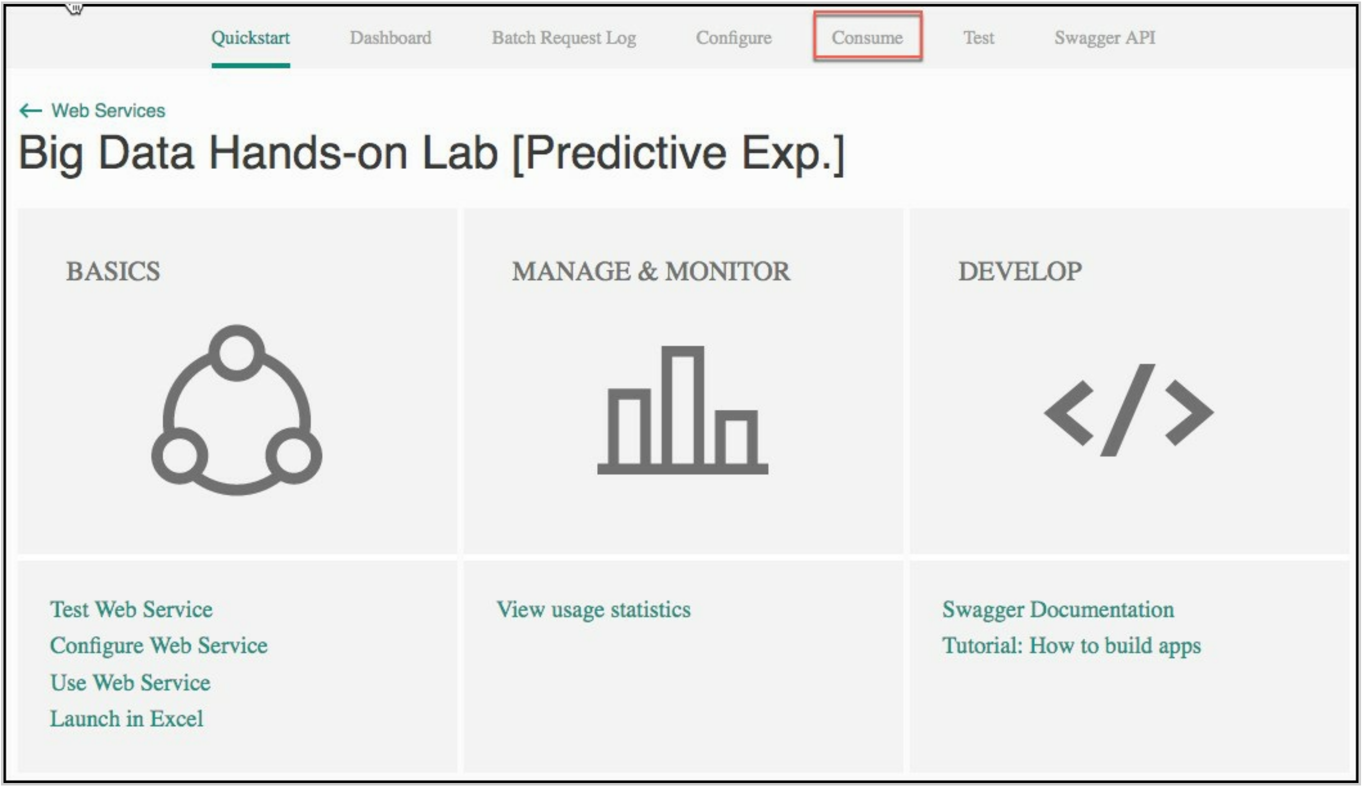
Standard DevTest — FREE

Included Transactions: 1,000
Included Compute Hours: 2

Standard S1 — Pricing Details

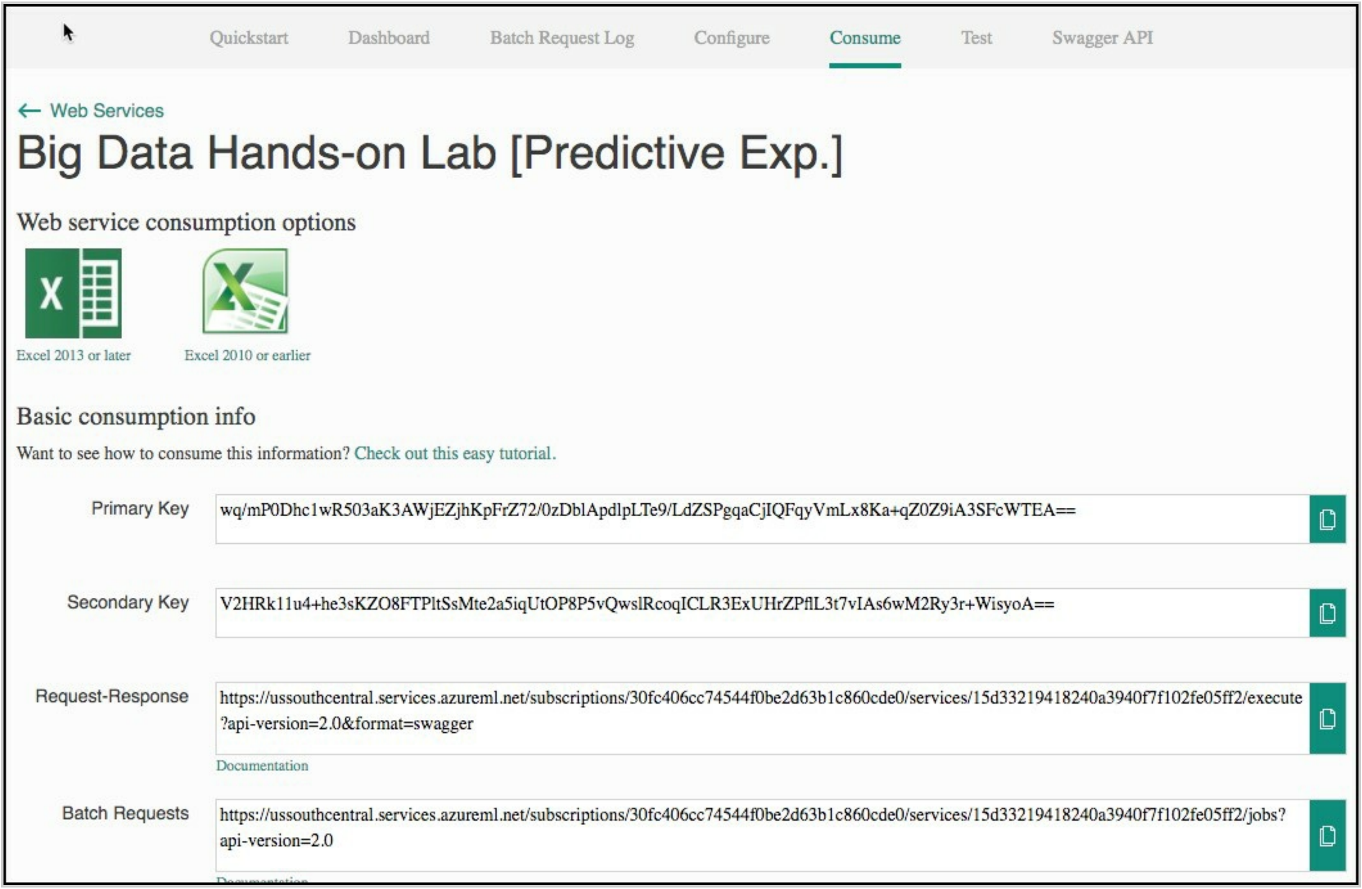
26. Select **Deploy**.

27. When the deployment is complete, you will be taken to the Web Service Quickstart page. Select the **Consume** tab.



Big Data Hands-on Lab [Predictive Exp.]

28. Leave the Consume page open for reference during **Exercise 4, Task 1**. At that point, you need to copy the Primary Key and Batch Requests Uri (omitting the querystring – "?api-version=2.0



Big Data Hands-on Lab [Predictive Exp.]

Web service consumption options

Basic consumption info

Want to see how to consume this information? Check out this easy tutorial.

Primary Key: wq/mP0Dhc1wR503aK3AWjEZjhKpFrZ72/0zDblApdlpLTe9/LdZSPgqaCjIQFqyVmLx8Ka+qZ0Z9iA3SFcWTEA==

Secondary Key: V2HRk11u4+he3sKZO8FTPltSsMte2a5iqUtOP8P5vQwslRcoqICLR3ExUHrZPflL3t7vIAs6wM2Ry3r+WisyoA==

Request-Response: https://ussouthcentral.services.azureml.net/subscriptions/30fc406cc74544f0be2d63b1c860cde0/services/15d33219418240a3940f7f102fe05ff2/execute?api-version=2.0&format=swagger

Documentation

Batch Requests: https://ussouthcentral.services.azureml.net/subscriptions/30fc406cc74544f0be2d63b1c860cde0/services/15d33219418240a3940f7f102fe05ff2/jobs?api-version=2.0

Setup Azure Data Factory

Exercise 2: Setup Azure Data Factory

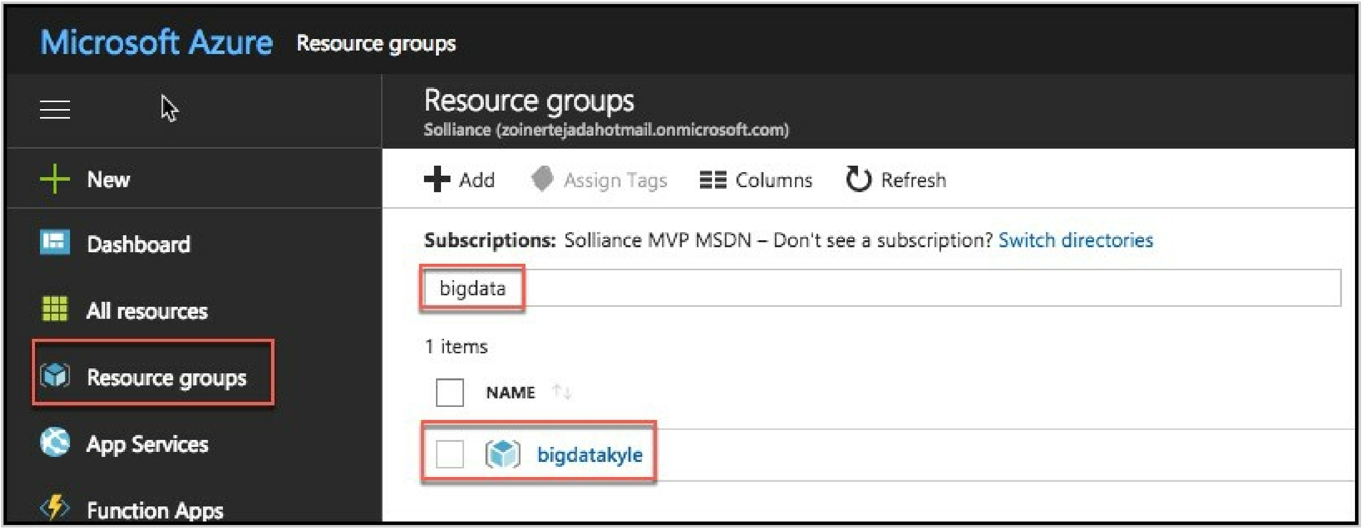
Duration: 20 mins

Synopsis: In this exercise, attendees 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 Integration Runtime which is the agent that facilitates data movement from on-premises to Microsoft Azure.

Task 1: Connect to the Lab VM

1. NOTE: If you are already, connected to your Lab VM, skip to Task 2.

2. From the left side menu in the Azure portal, click on **Resource groups**, then enter your resource group name into the filter box, and select it from the list.



Solliance (zoinertejadahotmail.onmicrosoft.com)

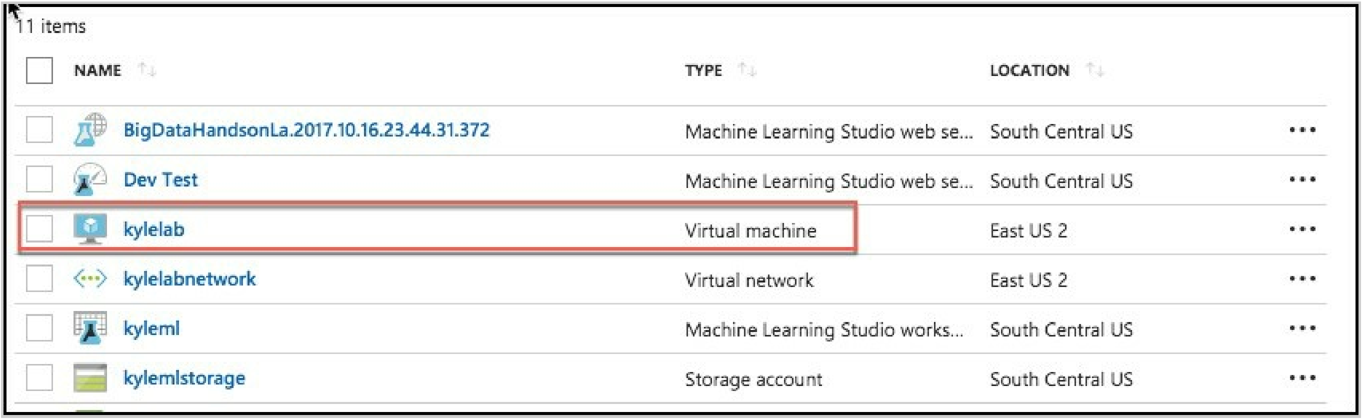
Subscriptions: Solliance MVP MSDN – Don't see a subscription? Switch directories

bigdata

1 items

□ bigdatakyle

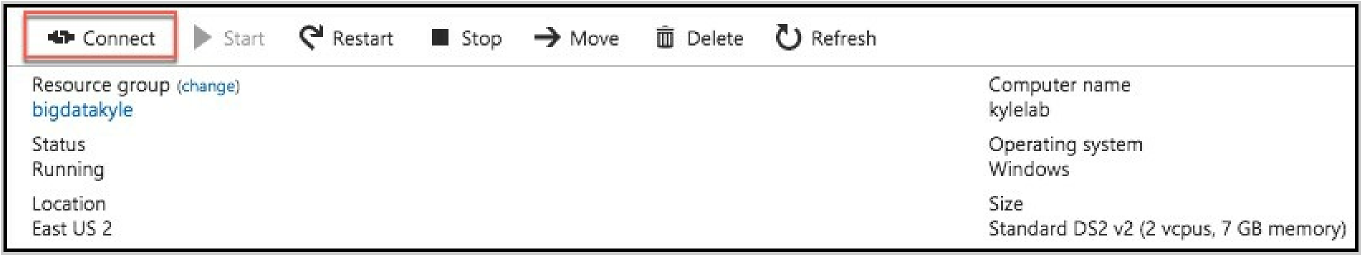
3. Next, select your lab virtual machine from the list.



11 items

NAME	TYPE	LOCATION
BigDataHandsonLa.2017.10.16.23.44.31.372	Machine Learning Studio web se...	South Central US
Dev Test	Machine Learning Studio web se...	South Central US
kylelab	Virtual machine	East US 2
kylelabnetwork	Virtual network	East US 2
kyleml	Machine Learning Studio works...	South Central US
kylemlstorage	Storage account	South Central US

4. On your Lab VM blade, select **Connect** from the top menu.



Resource group (change)
bigdatakyle

Status
Running

Location
East US 2

Computer name
kylelab

Operating system
Windows

Size
Standard DS2 v2 (2 vcpus, 7 GB memory)

5. Download and open the RDP file.

6. Select **Connect**, and enter the following credentials:

   - User name: demouser
   - Password: Password.1!!

Task 2: Download and stage data to be processed

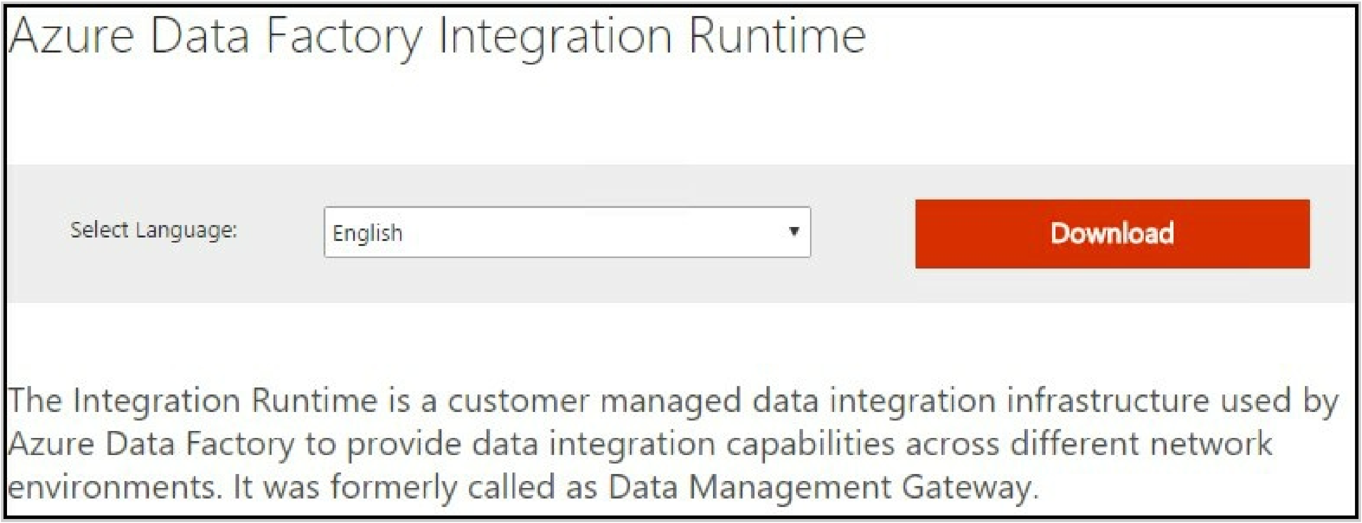
1. Once you have logged into the Lab VM, open a web browser. A shortcut for Chrome is on the desktop.

2. Download the AdventureWorks sample data from http://bit.ly/2zi4Sqa.

3. Extract it to a new folder called **C:\Data**.

Task 3: Install and configure Azure Data Factory Integration Runtime on the Lab VM

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

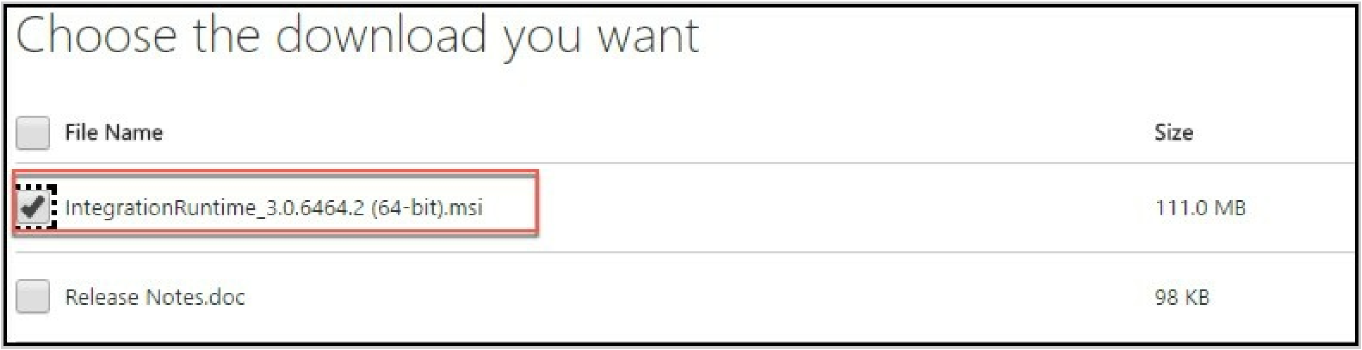


Azure Data Factory Integration Runtime

Select Language: English

The Integration Runtime is a customer managed data integration infrastructure used by Azure Data Factory to provide data integration capabilities across different network environments. It was formerly called as Data Management Gateway.

2. Select **Download**, then choose the download you want from the next screen.

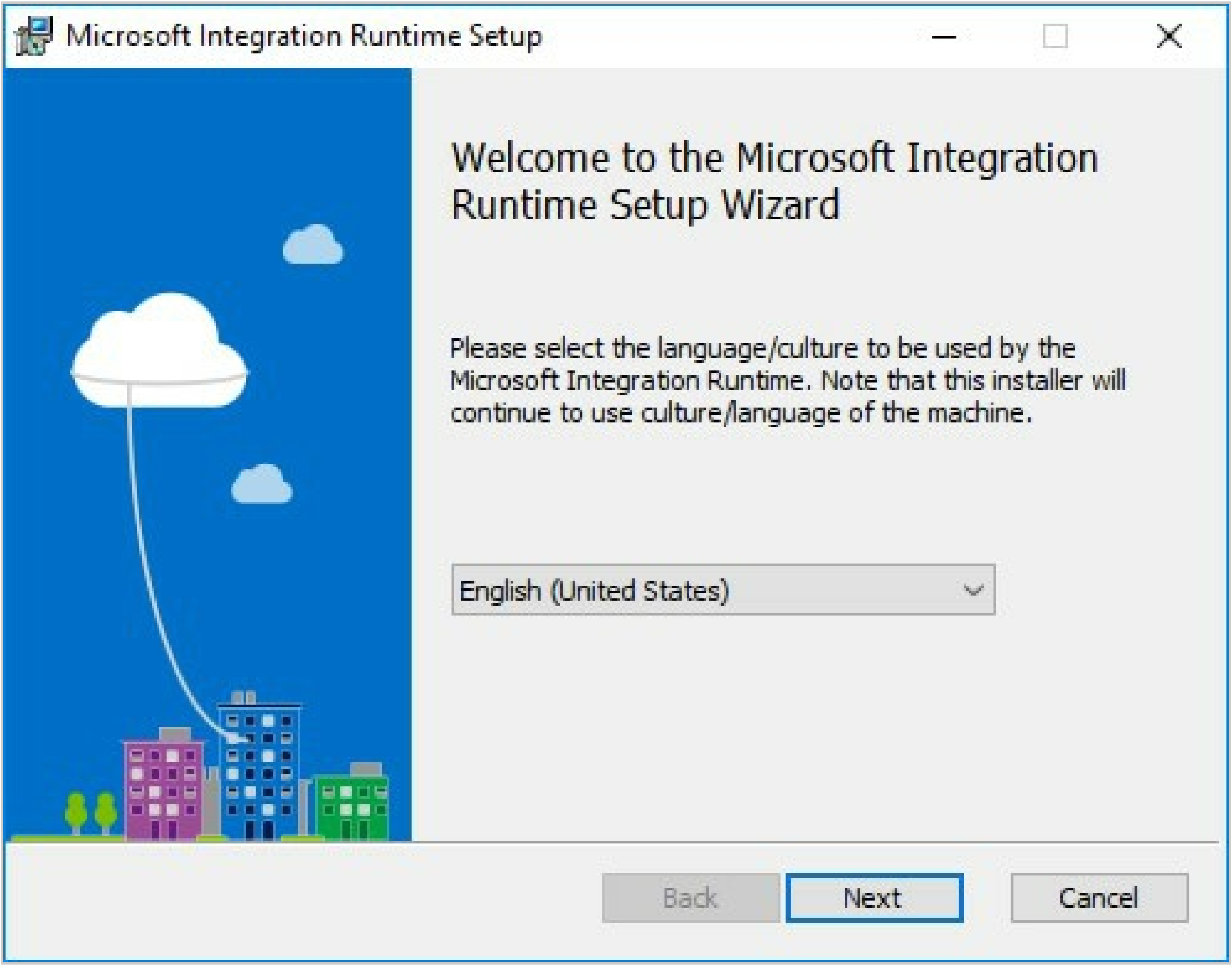


Choose the download you want

□ File Name	Size
☑ IntegrationRuntime_3.0.6464.2 (64-bit).msi	111.0 MB
□ Release Notes.doc	98 KB

3. Run the installer, once downloaded.

4. When you see the following screen, select **Next**.



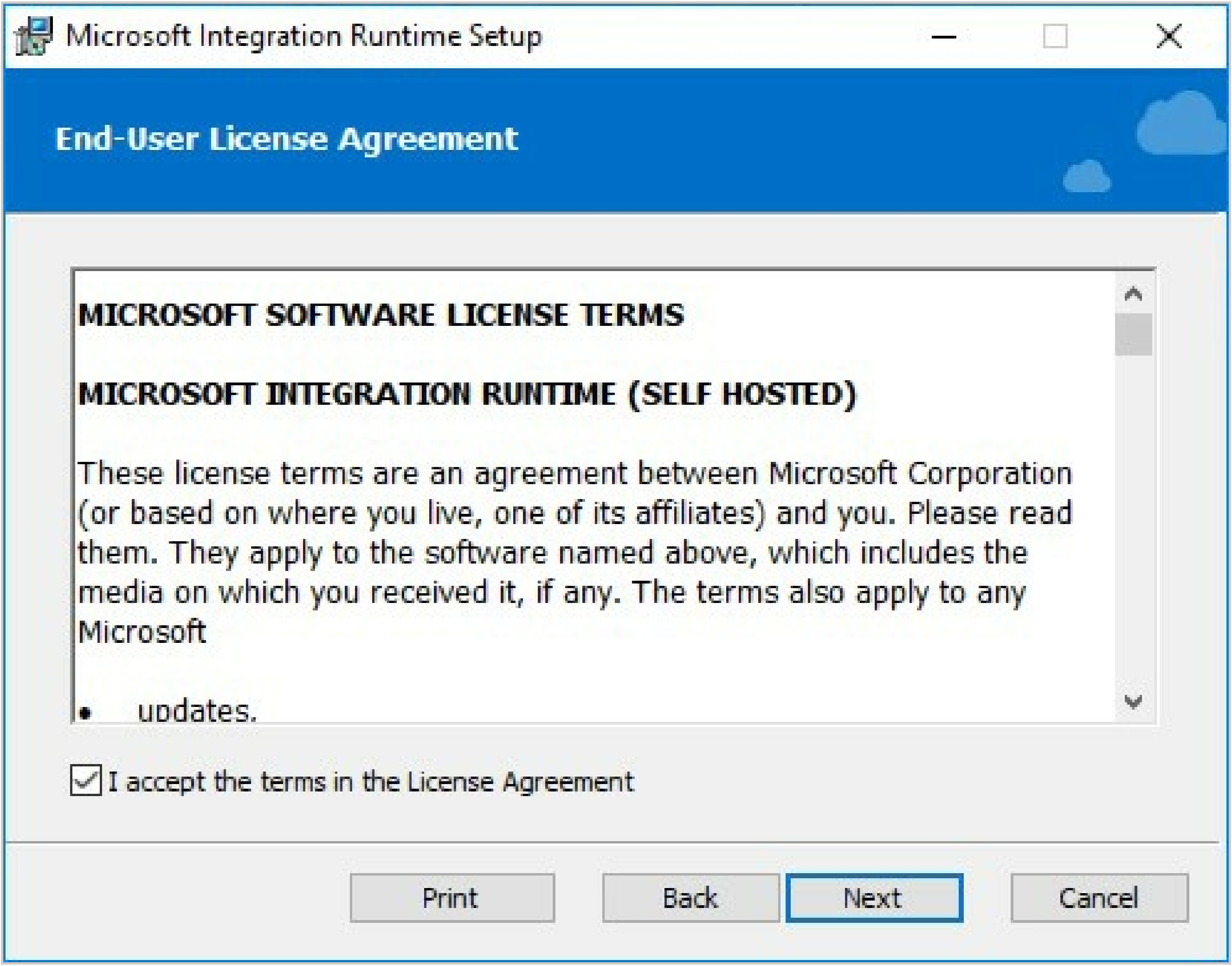
Microsoft Integration Runtime Setup

Welcome to the Microsoft Integration Runtime Setup Wizard

Please select the language/culture to be used by the Microsoft Integration Runtime. Note that this installer will continue to use culture/language of the machine.

English (United States)

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



Microsoft Integration Runtime Setup
End-User License Agreement

MICROSOFT SOFTWARE LICENSE TERMS

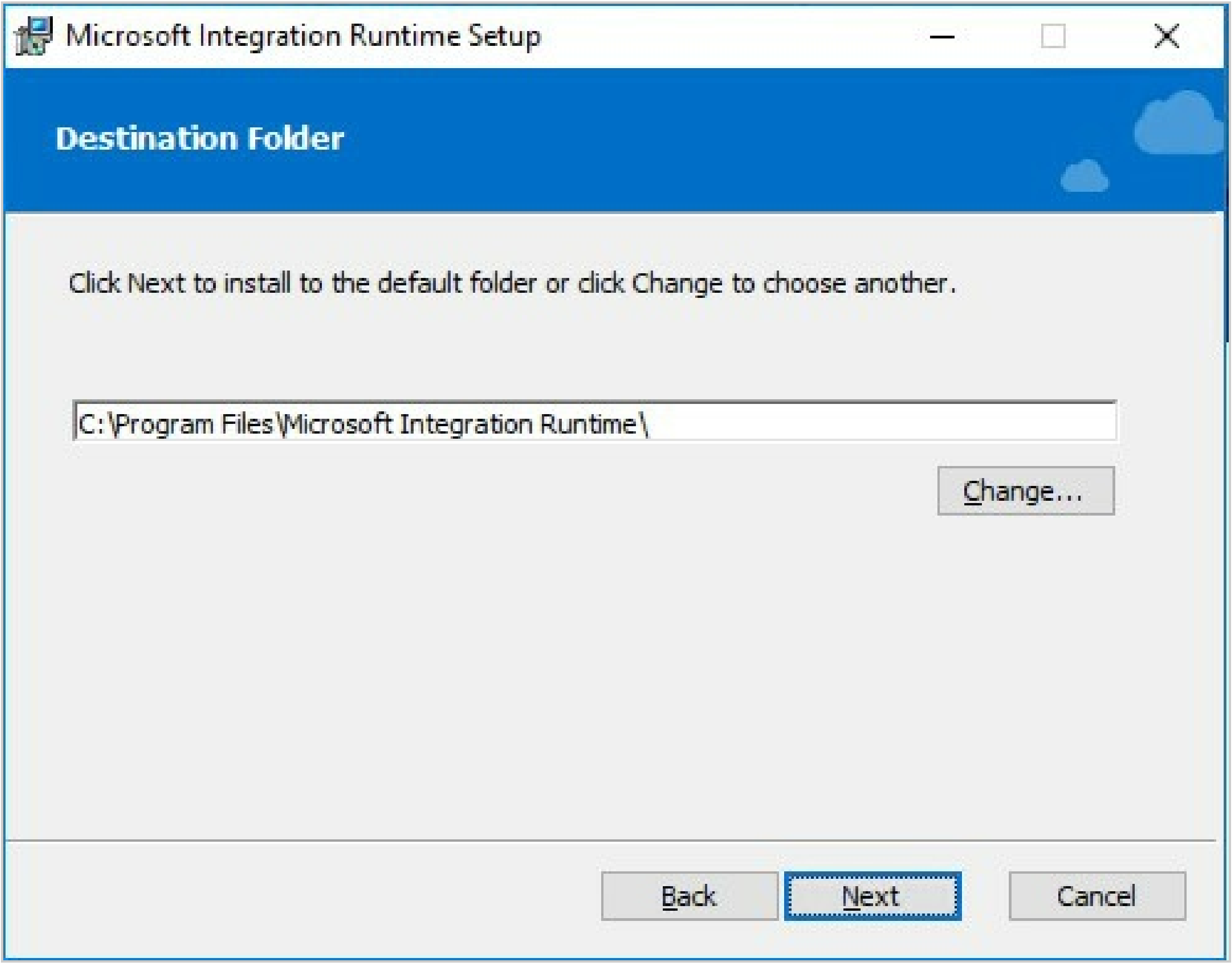
MICROSOFT INTEGRATION RUNTIME (SELF HOSTED)

These license terms are an agreement between Microsoft Corporation (or based on where you live, one of its affiliates) and you. Please read them. They apply to the software named above, which includes the media on which you received it, if any. The terms also apply to any Microsoft

- updates,

☑ I accept the terms in the License Agreement

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



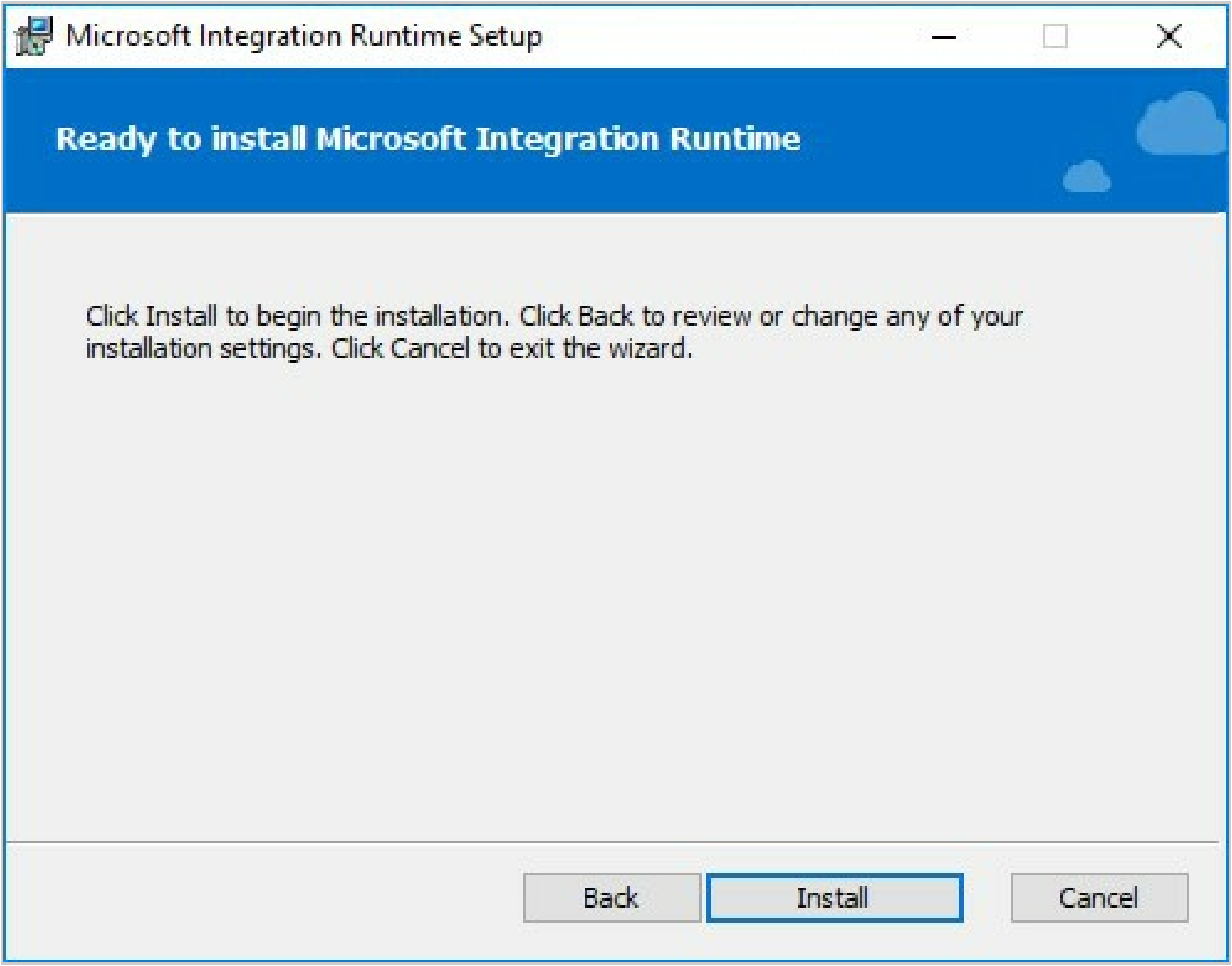
Microsoft Integration Runtime Setup

Destination Folder

Click Next to install to the default folder or click Change to choose another.

C:\Program Files\Microsoft Integration Runtime\

7. Select **Install** to complete the installation.

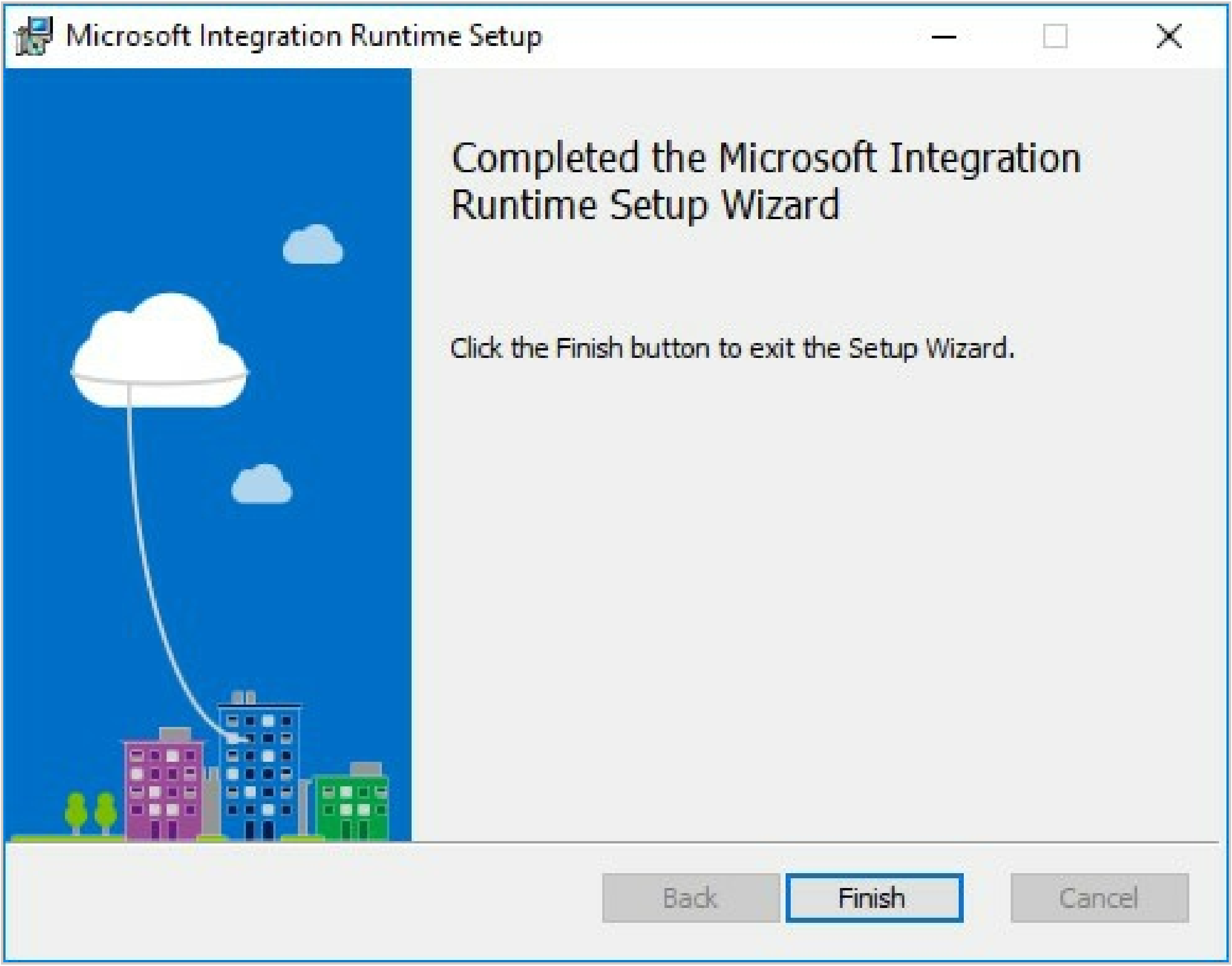


Microsoft Integration Runtime Setup

Ready to install Microsoft Integration Runtime

Click Install to begin the installation. Click Back to review or change any of your installation settings. Click Cancel to exit the wizard.

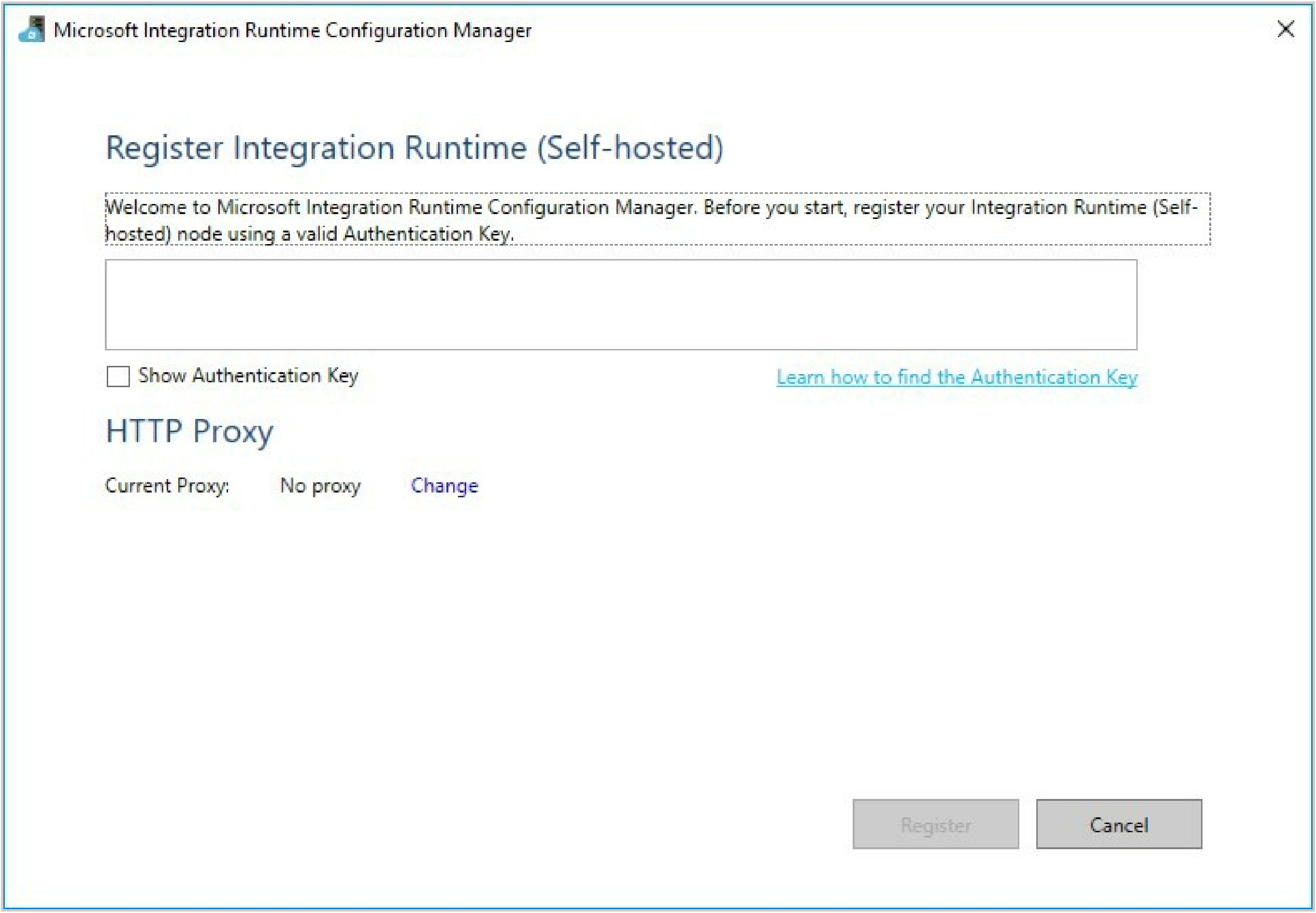
8. Select **Finish** once the installation has completed.


Microsoft Integration Runtime Setup

Completed the Microsoft Integration Runtime Setup Wizard

Click the Finish button to exit the Setup Wizard.

9. After clicking Finish, the following screen will appear. Keep it open for now. We will come back to this screen once we have provisioned the Data Factory in Azure, and obtain the gateway key so we can connect Data Factory to this “on-premises” server


Microsoft Integration Runtime Configuration Manager

Register Integration Runtime (Self-hosted)

Welcome to Microsoft Integration Runtime Configuration Manager. Before you start, register your Integration Runtime (Self-hosted) node using a valid Authentication Key.

□ Show Authentication Key

Learn how to find the Authentication Key

HTTP Proxy

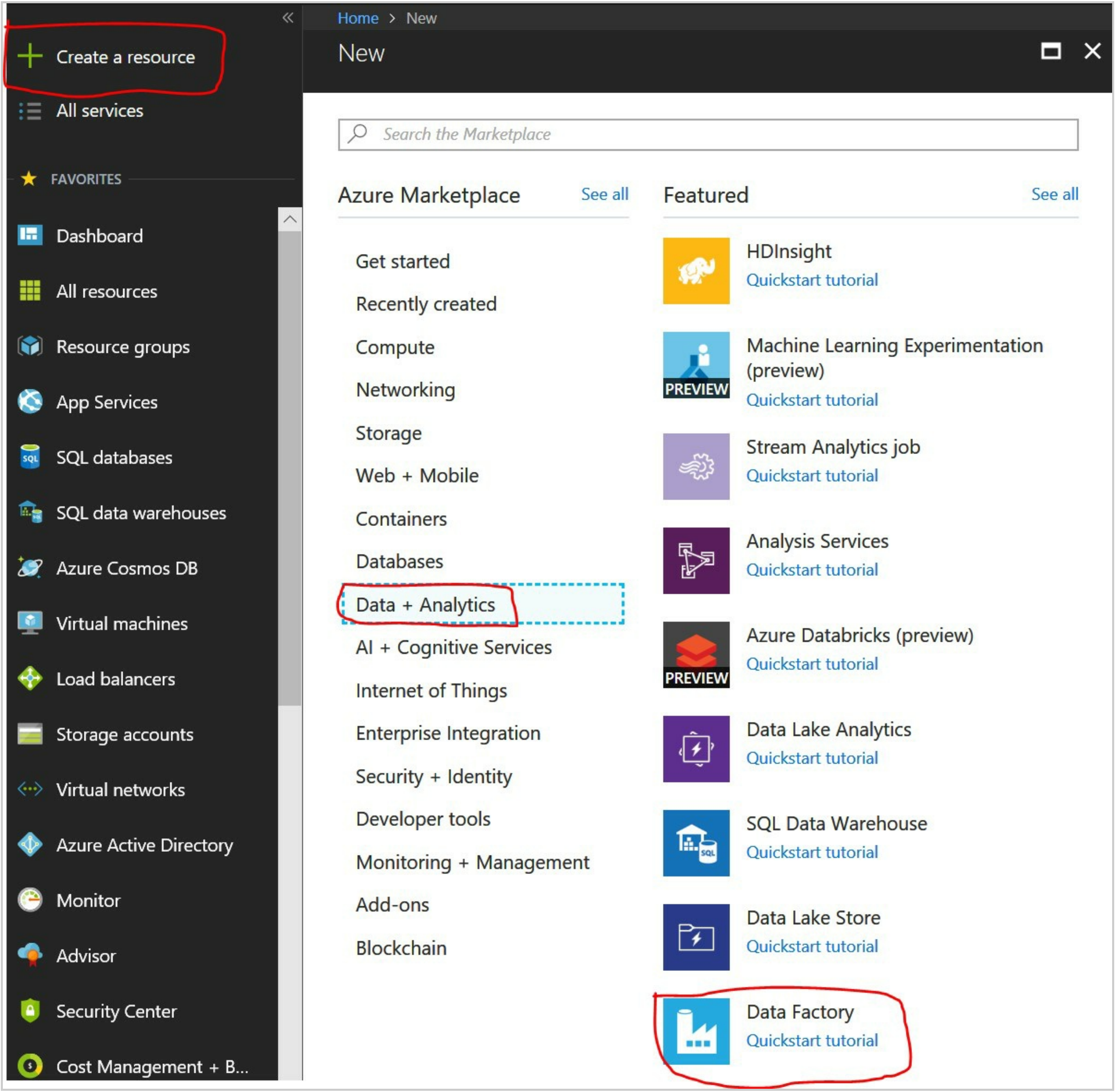
Current Proxy: No proxy

Task 4: Create an Azure Data Factory

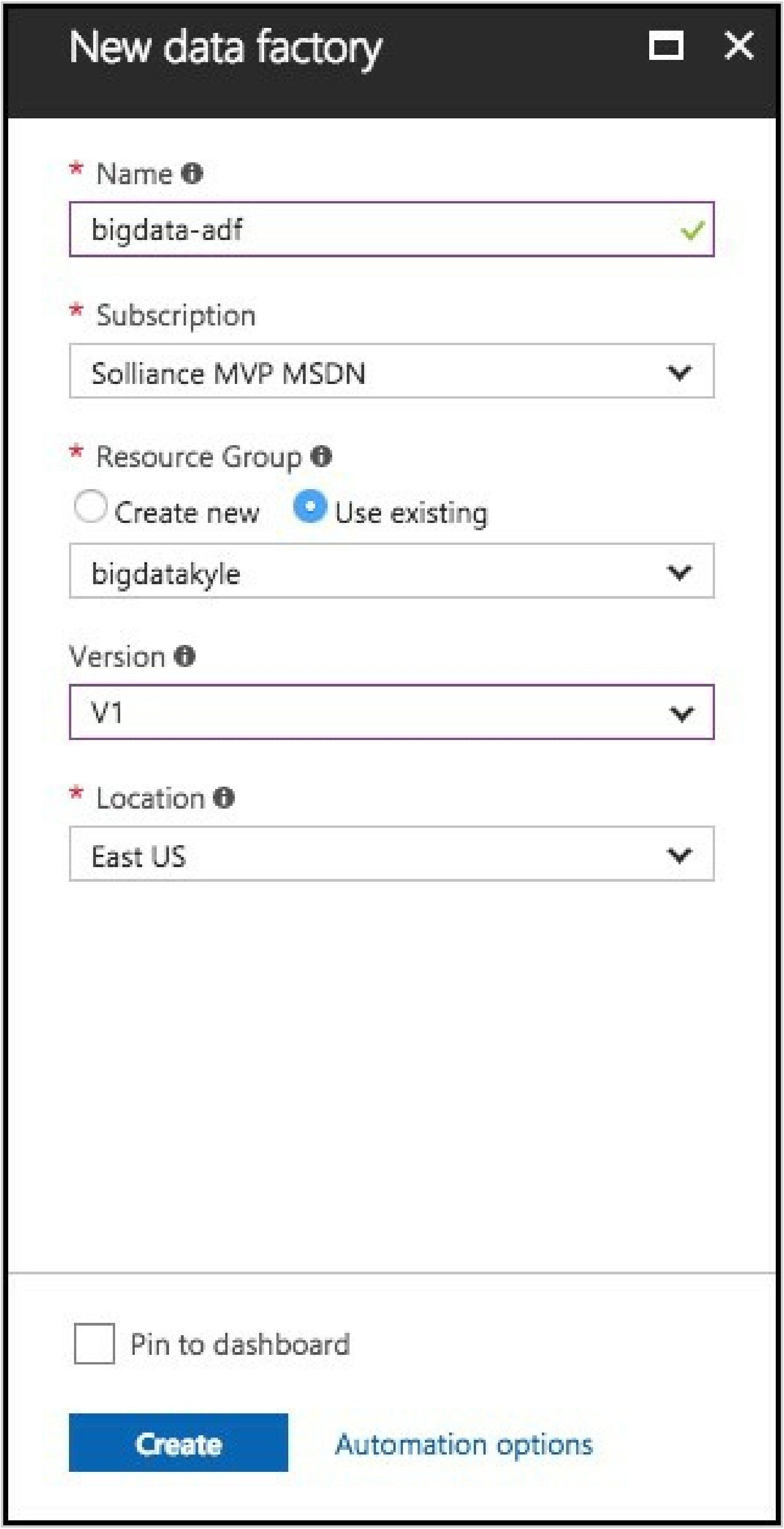
1. Launch a new browser window, and navigate to the Azure portal (https://portal.azure.com). Once prompted, log in with your Microsoft Azure credentials. If prompted, choose whether your account is an organization account or a Microsoft account. This will be based on which account was used to provision your Azure subscription that is being used for this lab.

   - **Note:** You may need to launch an InPrivate/Incognito session in your browser if you have multiple Microsoft accounts.

2. From the top left corner of the Azure portal, select **+ Create a resource**, and select **Data + Analytics**, then select **Data Factory**.



3. On the New data factory blade, enter the following:
   - Name: Provide a name, such as bigdata-adf
   - Subscription: Select your subscription
   - Resource Group: Choose Use existing, and select the Resource Group you created when deploying the lab prerequisites
   - Version: Select V1
   - Location: Select one of the available locations from the list nearest the one used by your Resource Group
   - Select **Create**



New data factory

* Name

bigdata-adf ✓

* Subscription

Solliance MVP MSDN

* Resource Group

○ Create new ● Use existing

bigdatakyle

Version

V1

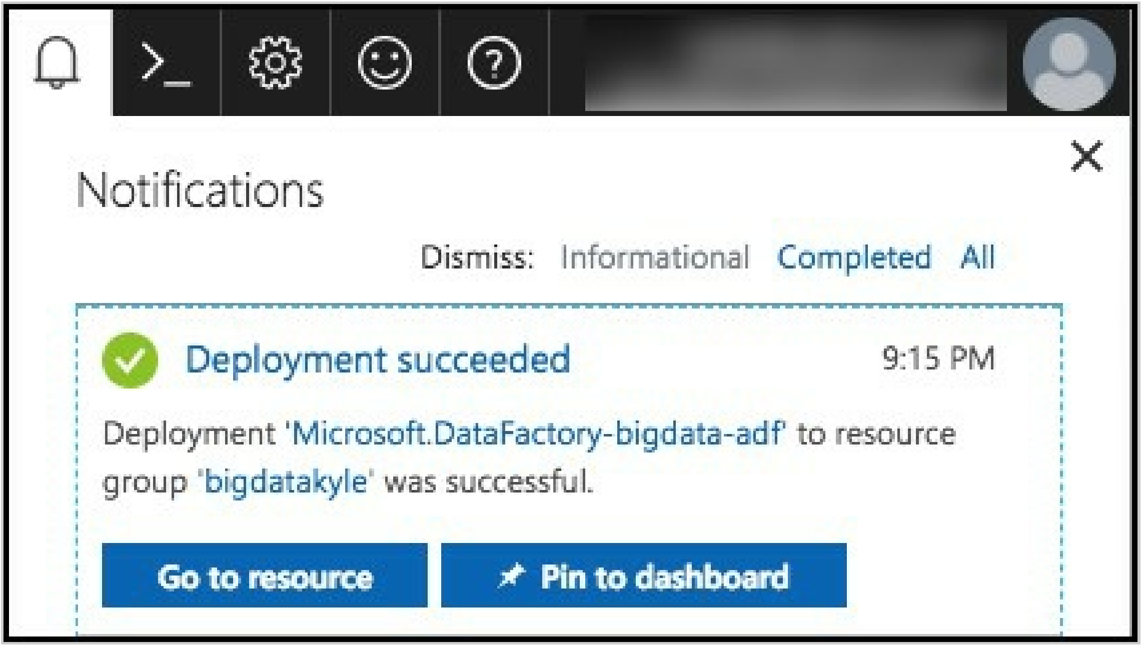
* Location

East US

□ Pin to dashboard

4. The ADF deployment will take several minutes.

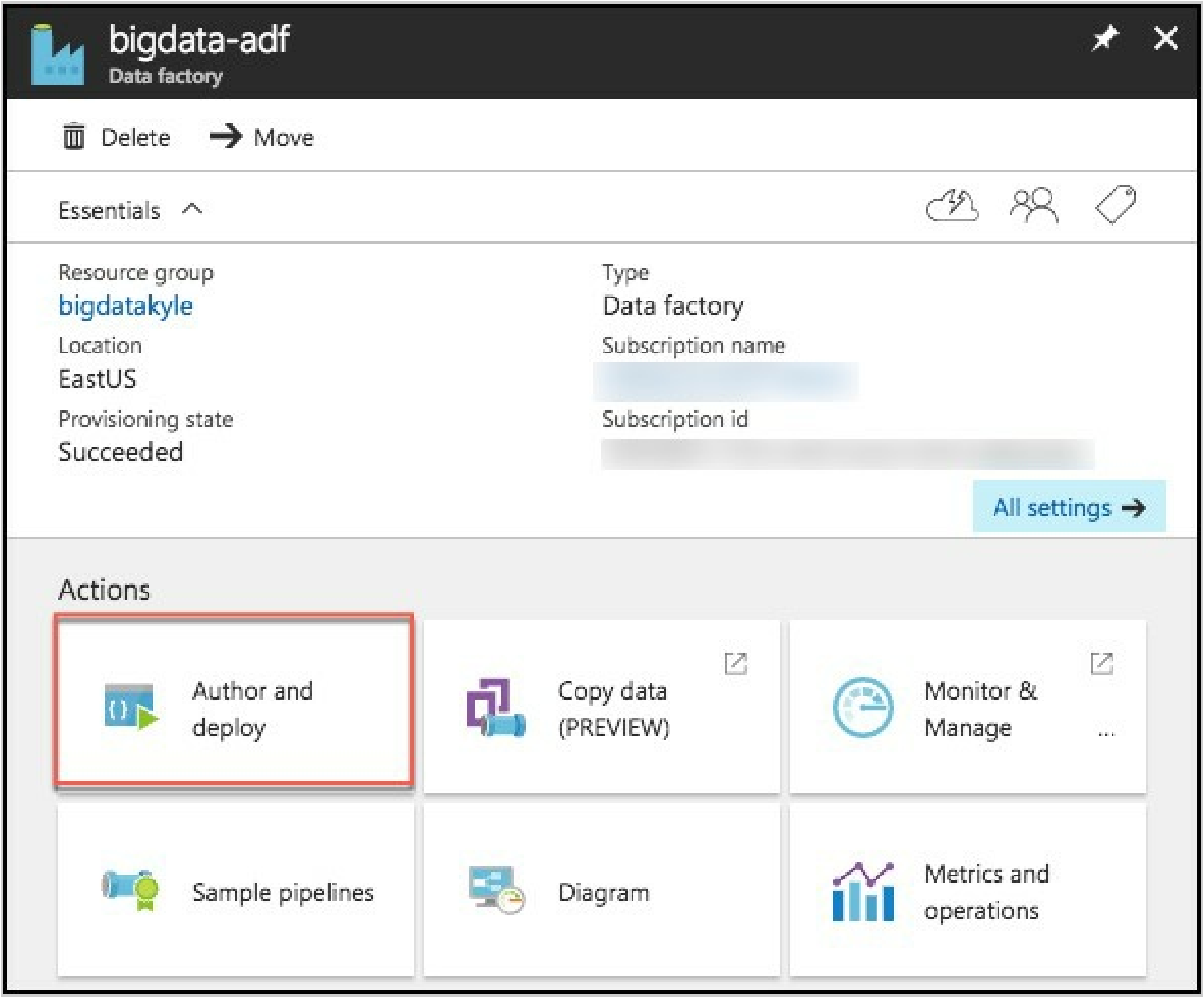
5. Once the deployment is completed, you will receive a notification that it succeeded.



Deployment succeeded 9:15 PM

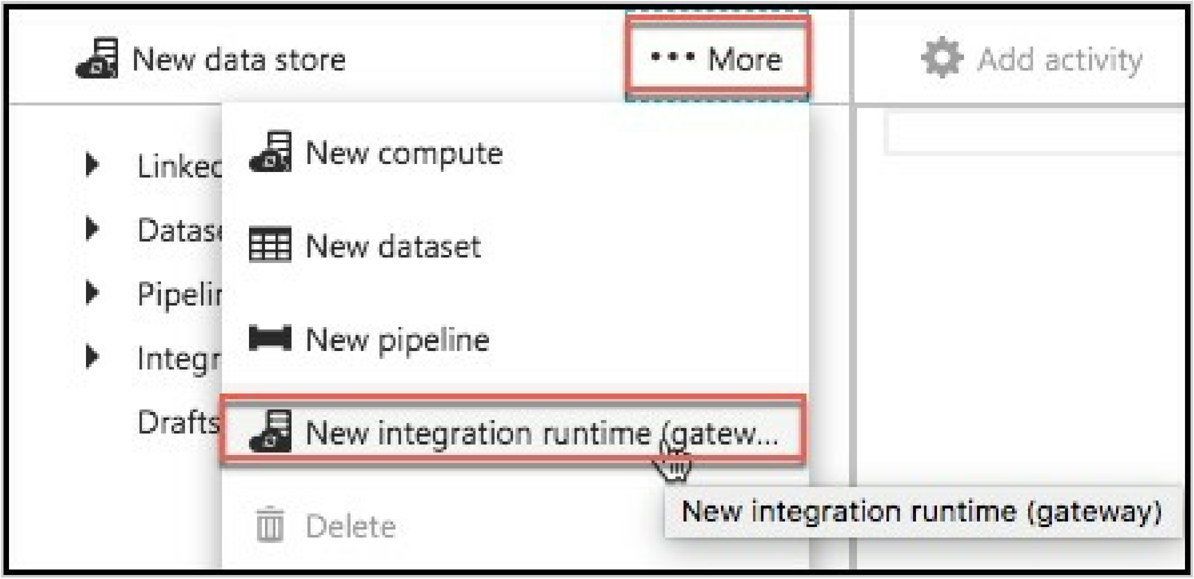
Deployment 'Microsoft.DataFactory-bigdata-adf' to resource group 'bigdatakyle' was successful.

6. Select the **Go to resource button**, to navigate to the newly created Data Factory.
7. On the Data Factory blade, select **Author and Deploy** under Actions.

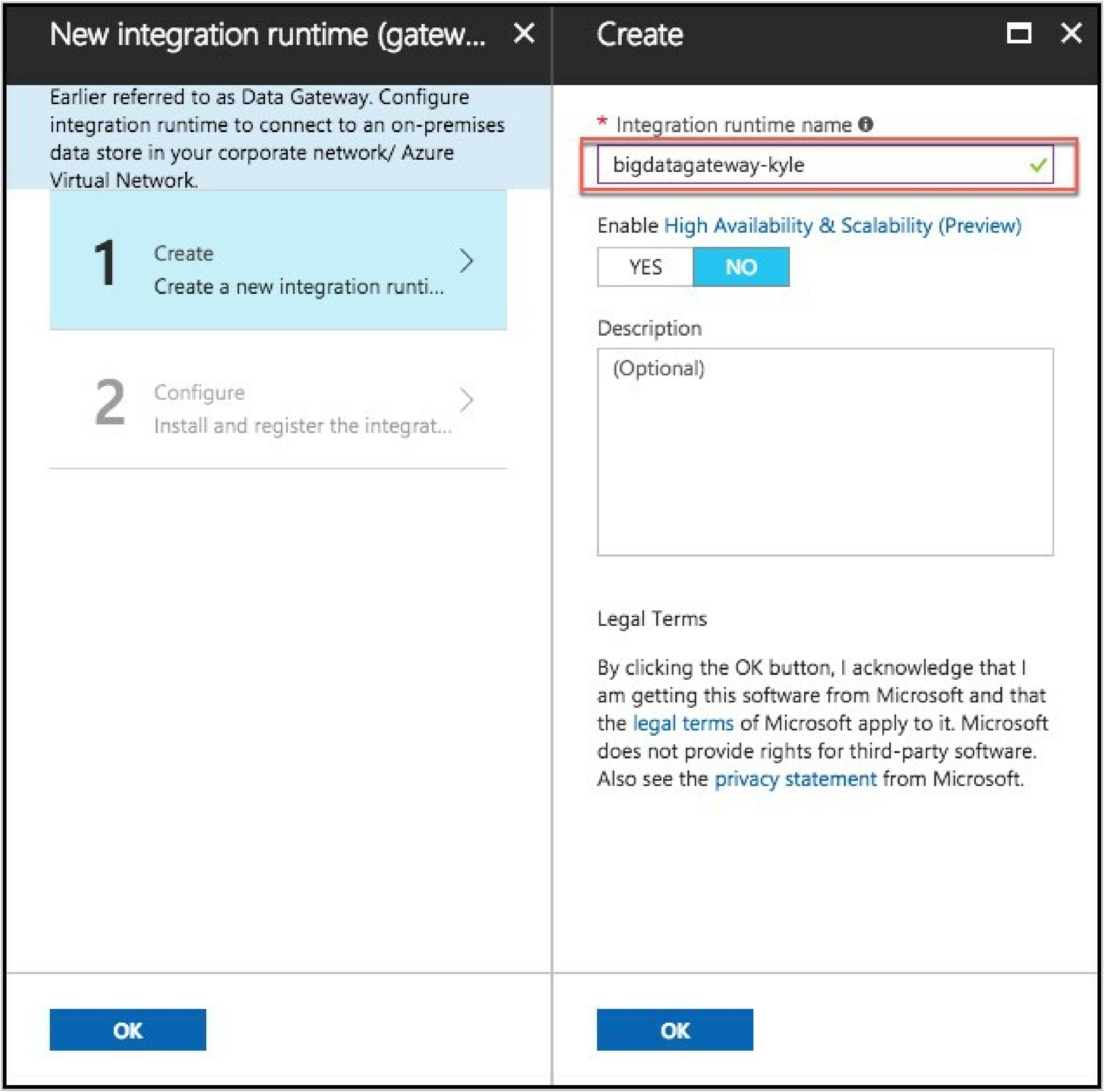


Resource group
bigdatakyle
Location
EastUS
Provisioning state
Succeeded
Type
Data factory
Subscription name
[redacted]
Subscription id
[redacted]

8. Next, select …More, then New integration runtime (gateway).



9. Enter an Integration runtime name, such as bigdatagateway-[initials], and select **OK**.



New integration runtime (gatew... ✕

Earlier referred to as Data Gateway. Configure integration runtime to connect to an on-premises data store in your corporate network/ Azure Virtual Network.

1 Create
Create a new integration runti...

2 Configure
Install and register the integrat...

Create

* Integration runtime name
bigdatagateway-kyle ✓

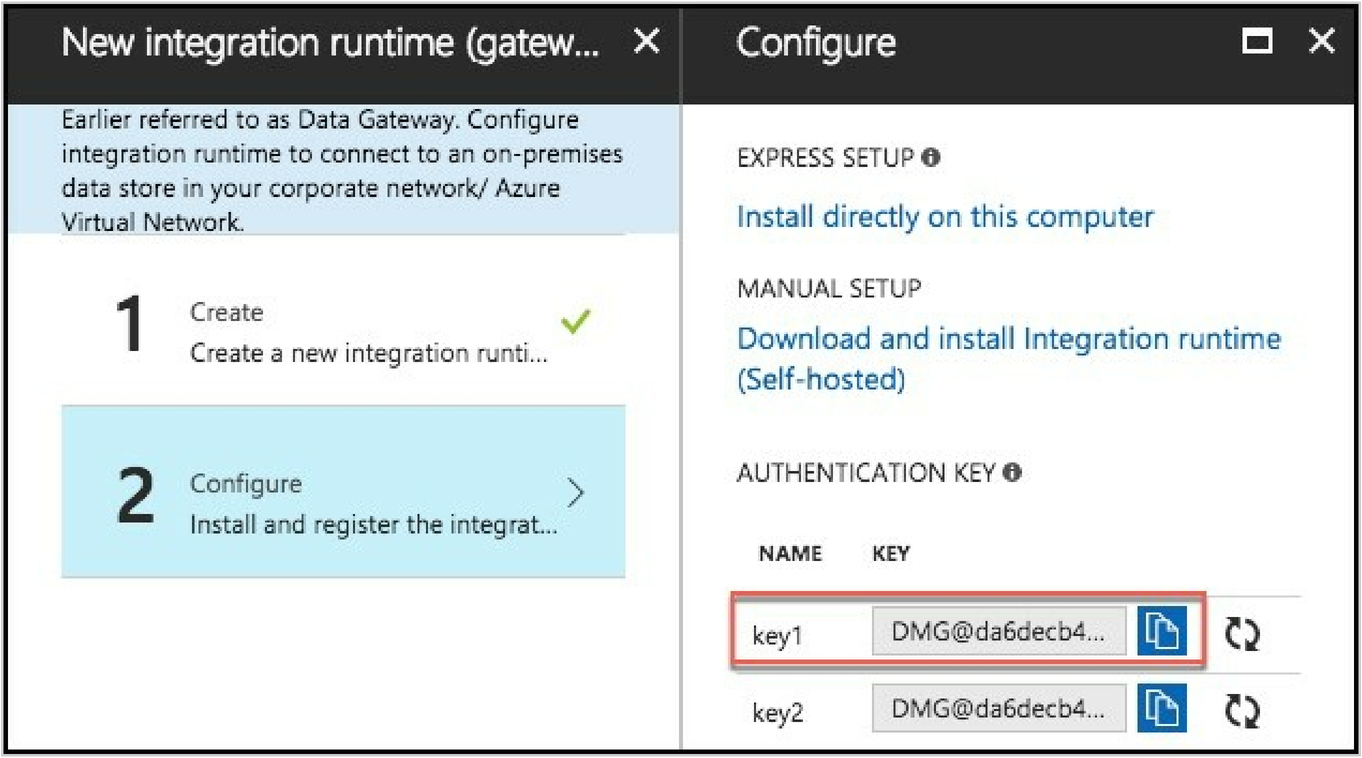
Enable High Availability & Scalability (Preview)
YES [✓ NO]

Description
(Optional)

Legal Terms

By clicking the OK button, I acknowledge that I am getting this software from Microsoft and that the legal terms of Microsoft apply to it. Microsoft does not provide rights for third-party software. Also see the privacy statement from Microsoft.

10. On the Configure screen, copy the key1 value by selecting the Copy button, then select OK.



New integration runtime (gatew...

Earlier referred to as Data Gateway. Configure integration runtime to connect to an on-premises data store in your corporate network/ Azure Virtual Network.

1 Create
Create a new integration runti...

2 Configure
Install and register the integrat...

Configure

EXPRESS SETUP

Install directly on this computer

MANUAL SETUP

Download and install Integration runtime (Self-hosted)

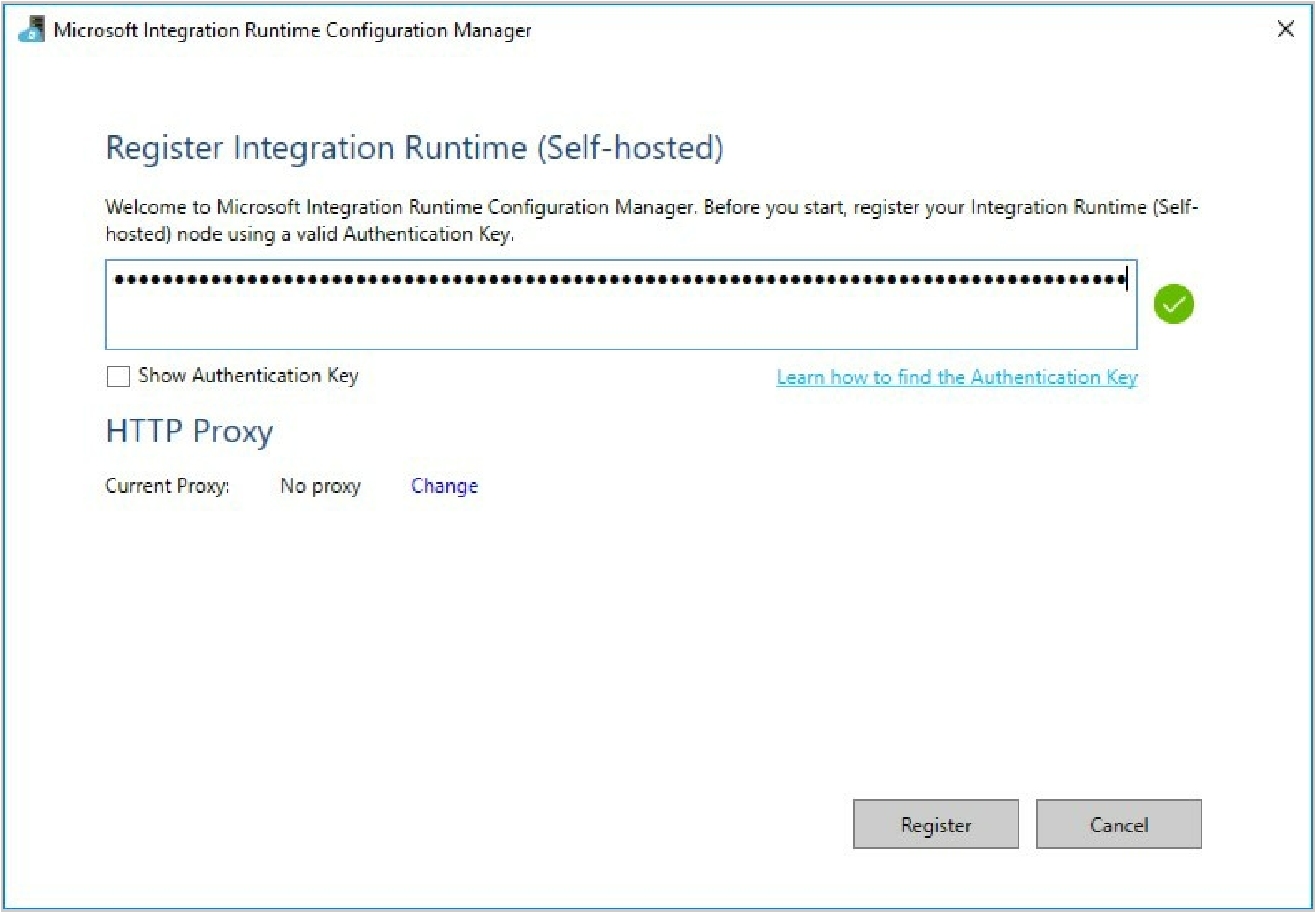
AUTHENTICATION KEY

NAME	KEY
key1	DMG@da6decb4...
key2	DMG@da6decb4...

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

12. Go back to the Remote Deskop session of the Lab VM.

13. Paste the **key1** value into the box in the middle of the Microsoft Integration Runtime Configuration Manager screen.



Microsoft Integration Runtime Configuration Manager

Register Integration Runtime (Self-hosted)

Welcome to Microsoft Integration Runtime Configuration Manager. Before you start, register your Integration Runtime (Self-hosted) node using a valid Authentication Key.

□ Show Authentication Key

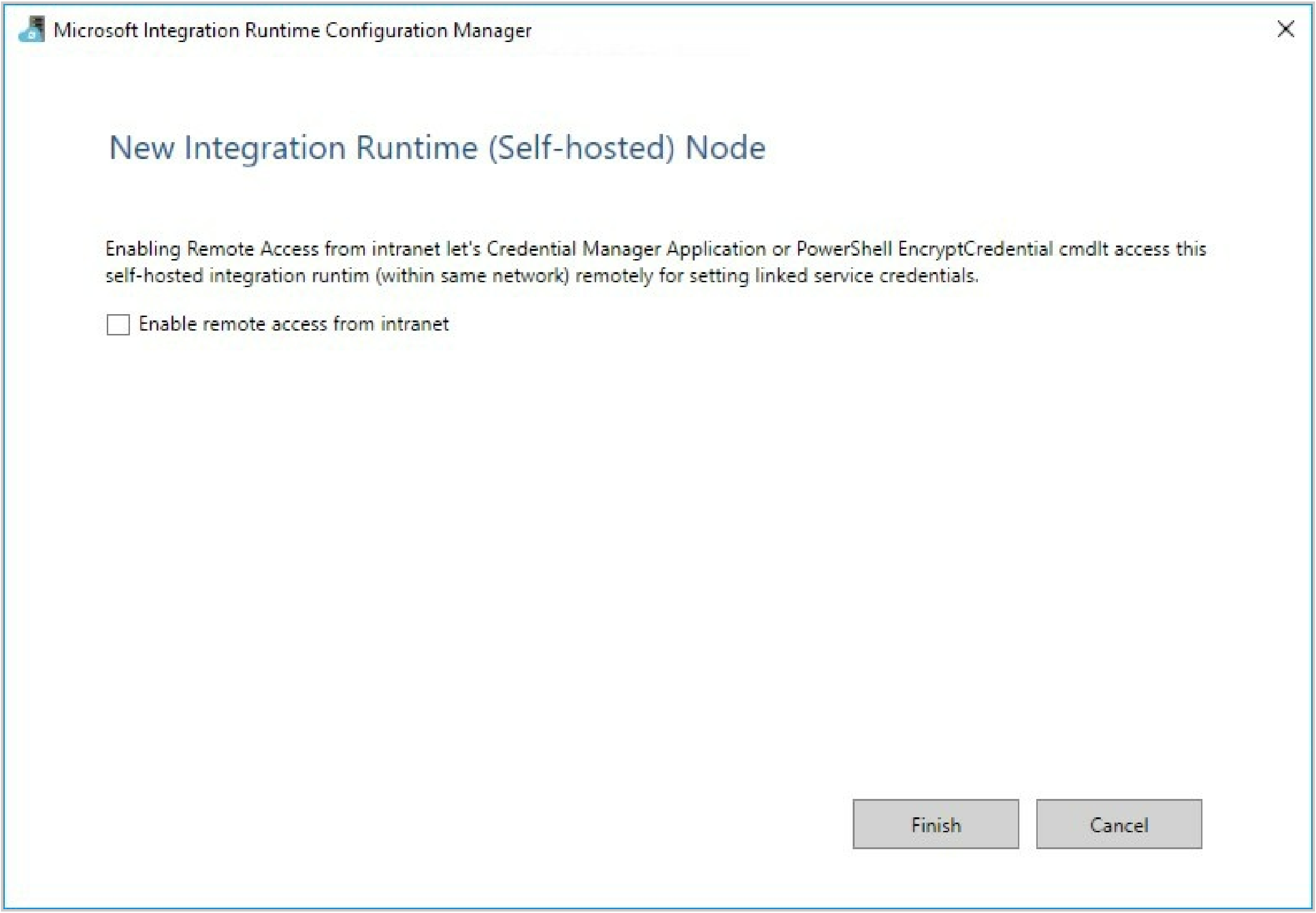
HTTP Proxy

Current Proxy: No proxy

14. Select **Register**.

15. It will take 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 clicking the **Cancel** button.

16. The next screen will be New Integration Runtime (Self-hosted) Node. Select **Finish**.



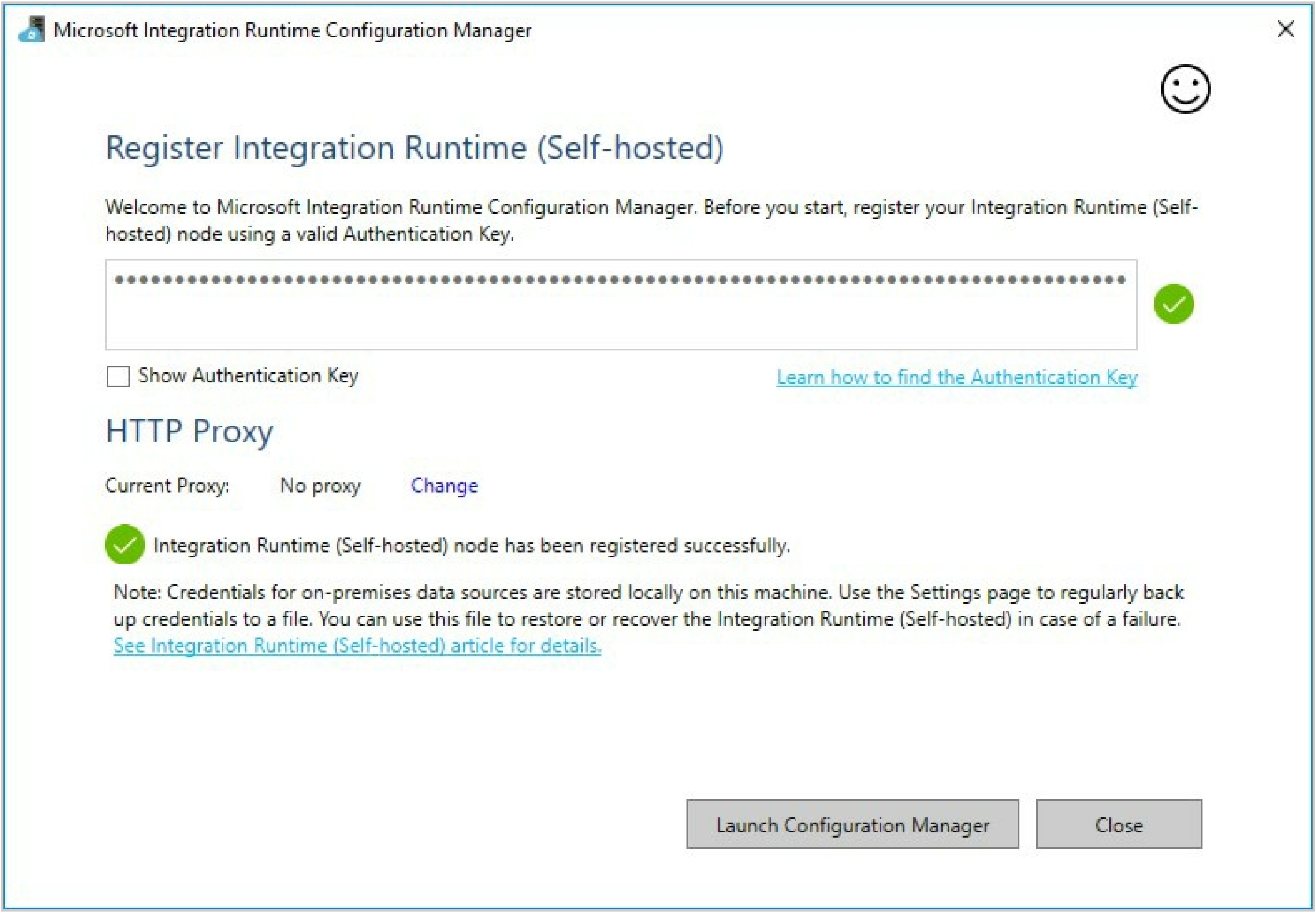
Microsoft Integration Runtime Configuration Manager

New Integration Runtime (Self-hosted) Node

Enabling Remote Access from intranet let's Credential Manager Application or PowerShell EncryptCredential cmdlt access this self-hosted integration runtim (within same network) remotely for setting linked service credentials.

□ Enable remote access from intranet

17. You will then get a screen with a confirmation message.



Register Integration Runtime (Self-hosted)

Welcome to Microsoft Integration Runtime Configuration Manager. Before you start, register your Integration Runtime (Self-hosted) node using a valid Authentication Key.

••• ✓

□ Show Authentication Key

Learn how to find the Authentication Key

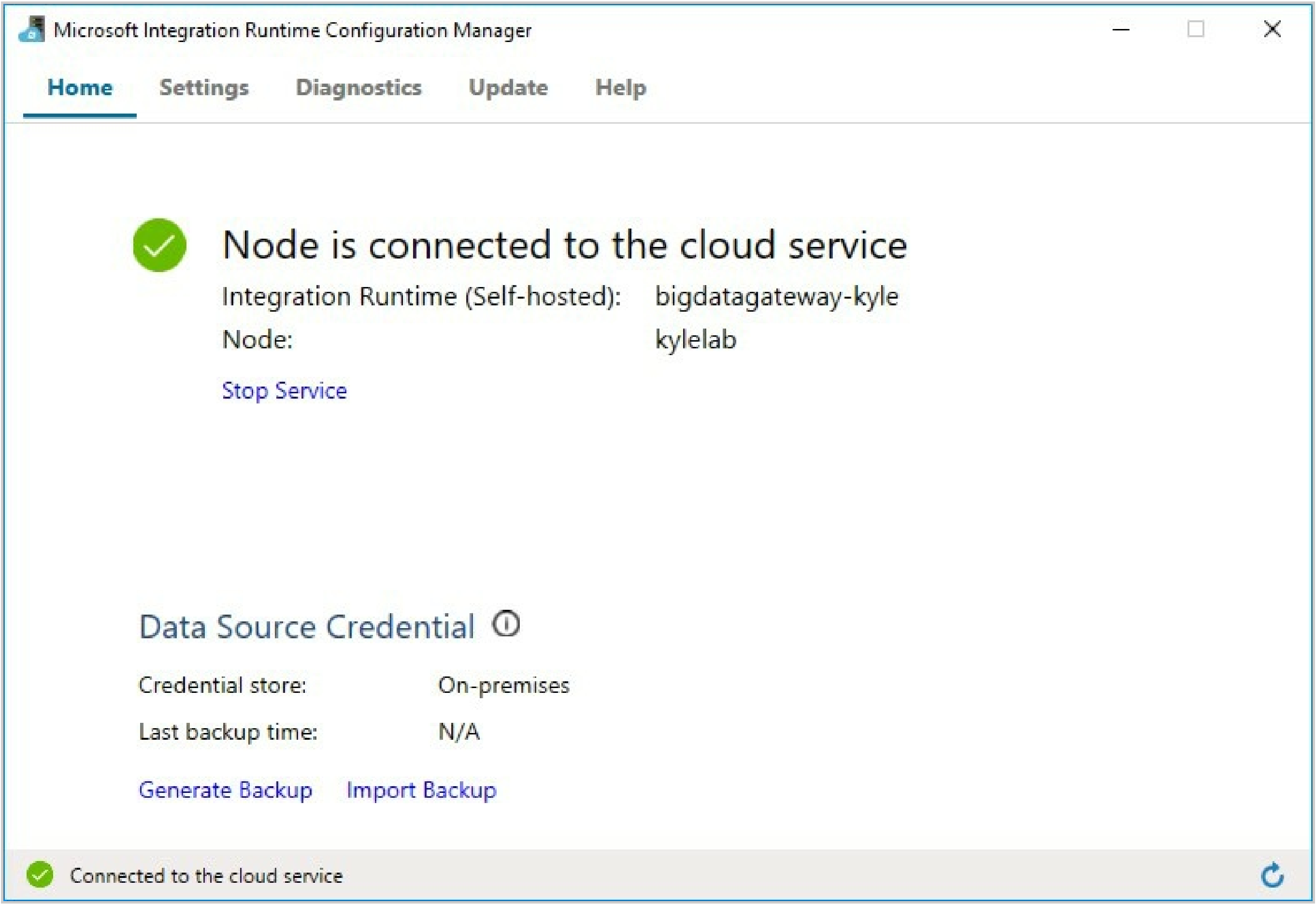
HTTP Proxy

Current Proxy: No proxy Change

✓ Integration Runtime (Self-hosted) node has been registered successfully.

Note: Credentials for on-premises data sources are stored locally on this machine. Use the Settings page to regularly back up credentials to a file. You can use this file to restore or recover the Integration Runtime (Self-hosted) in case of a failure. See Integration Runtime (Self-hosted) article for details.

18. Select the **Launch Configuration Manager** button to view the connection details.



Node is connected to the cloud service

Integration Runtime (Self-hosted): bigdatagateway-kyle

Node: kylelab

Data Source Credential

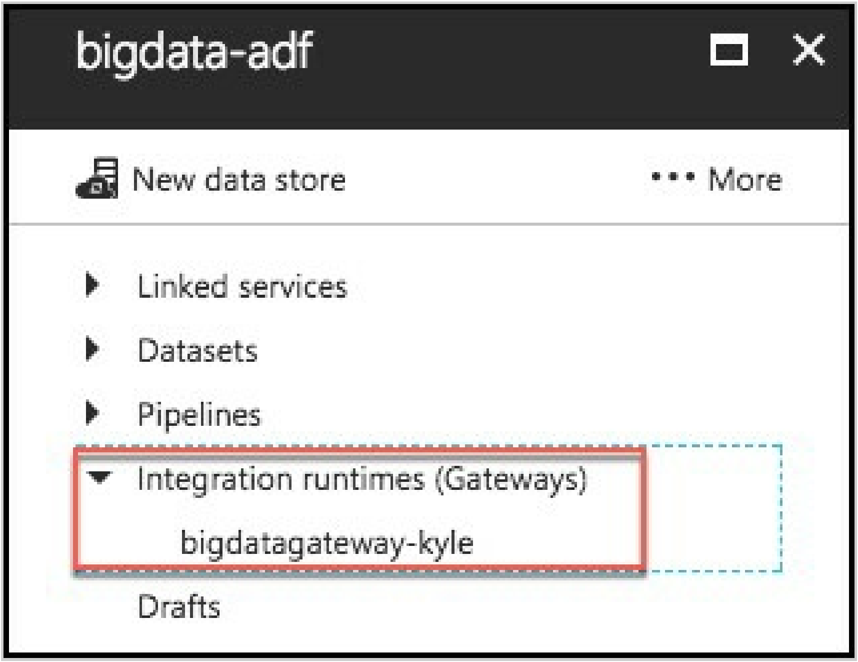
Credential store: On-premises

Last backup time: N/A

Connected to the cloud service

19. You can now return to the Azure portal, and click **OK** twice to complete the Integration Runtime setup.

20. You can view the Integration Runtime by expanding Integration runtimes on the Author and Deploy blade.



21. Close the Author and Deploy blade, to return to the the Azure Data Factory blade. Leave this open for the next exercise.

Develop a data factory pipeline for data movement

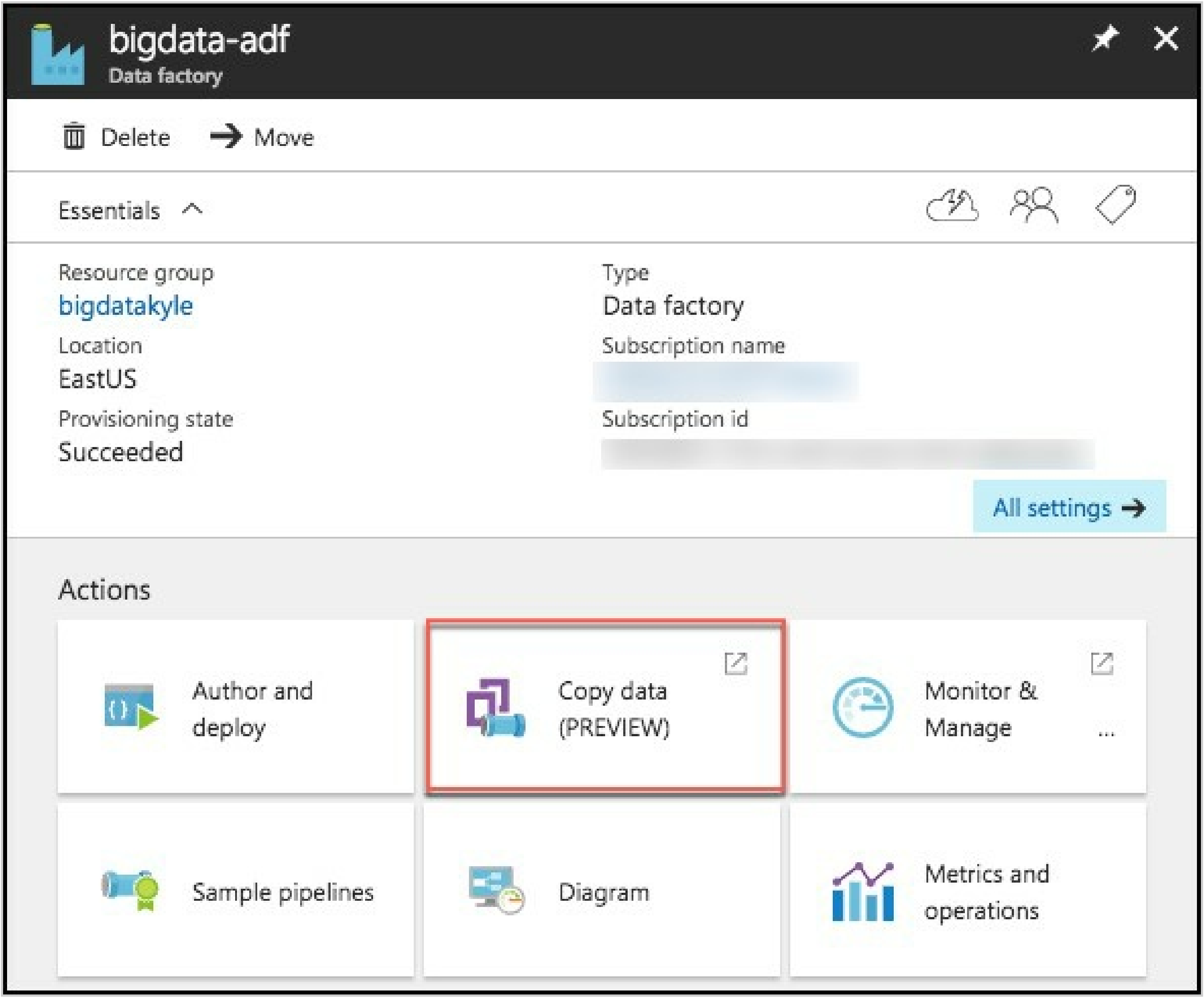
Exercise 3: Develop a data factory pipeline for data movement

Duration: 20 mins

Synopsis: In this exercise, you will create an Azure Data Factory pipeline to copy data (.CSV file) from an on-premises server (Lab VM) 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). You will see how assets are created, deployed, executed, and monitored.

Task 1: Create copy pipeline using the Copy Data Wizard

1. On your Azure Data Factory blade in the Azure portal, select **Copy Data (PREVIEW)**, under Actions.



Resource group
bigdatakyle

Location
EastUS

Provisioning state
Succeeded

Type
Data factory

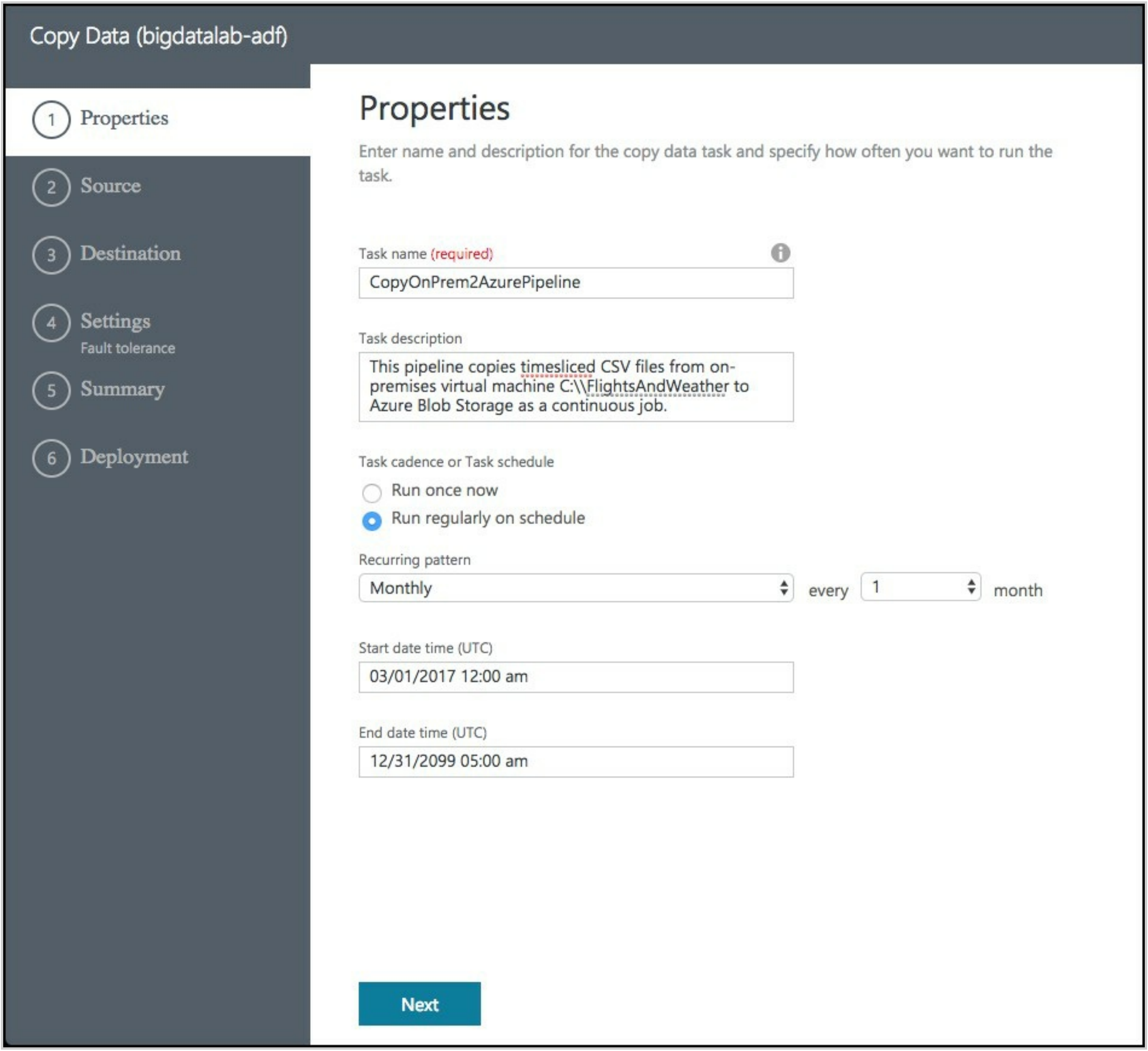
Subscription name
[redacted]

Subscription id
[redacted]

2. This will launch a new browser window. Log in with the same credentials you used to create the Data Factory.

3. In the new browser window, enter the following:

   - Task name: Enter “CopyOnPrem2AzurePipeline”
   - Task description: (Optional) Enter a description, such as “This pipeline copies timesliced CSV files from on-premises virtual machine C:\FlightsAndWeather to Azure Blob Storage as a continuous job.”
   - Task cadence (or) Task schedule: Select Run regularly on schedule.
   - Recurring pattern: Select Monthly, and every 1 month.
   - Start date time (UTC): Set to 03/01/2017 12:00 am
   - End date time (UTC): Set to 12/31/2099 11:59 pm
   - Select **Next**



Properties

Enter name and description for the copy data task and specify how often you want to run the task.

Task name (required): CopyOnPrem2AzurePipeline

Task description: This pipeline copies timesliced CSV files from on-premises virtual machine C:\\FlightsAndWeather to Azure Blob Storage as a continuous job.

Task cadence or Task schedule

○ Run once now

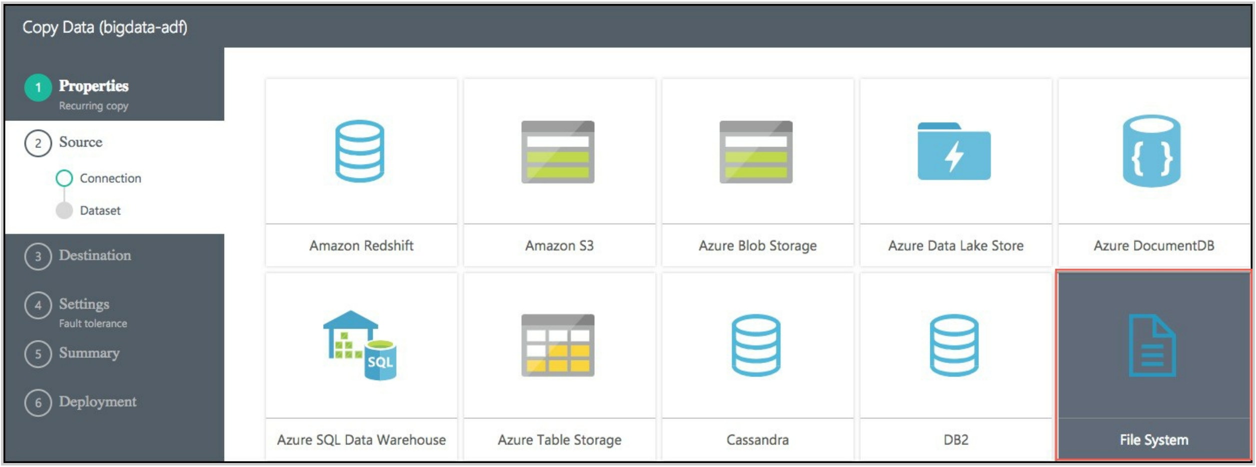
● Run regularly on schedule

Recurring pattern: Monthly every 1 month

Start date time (UTC): 03/01/2017 12:00 am

End date time (UTC): 12/31/2099 05:00 am

4. On the Source screen, select **File System**, then select **Next**.

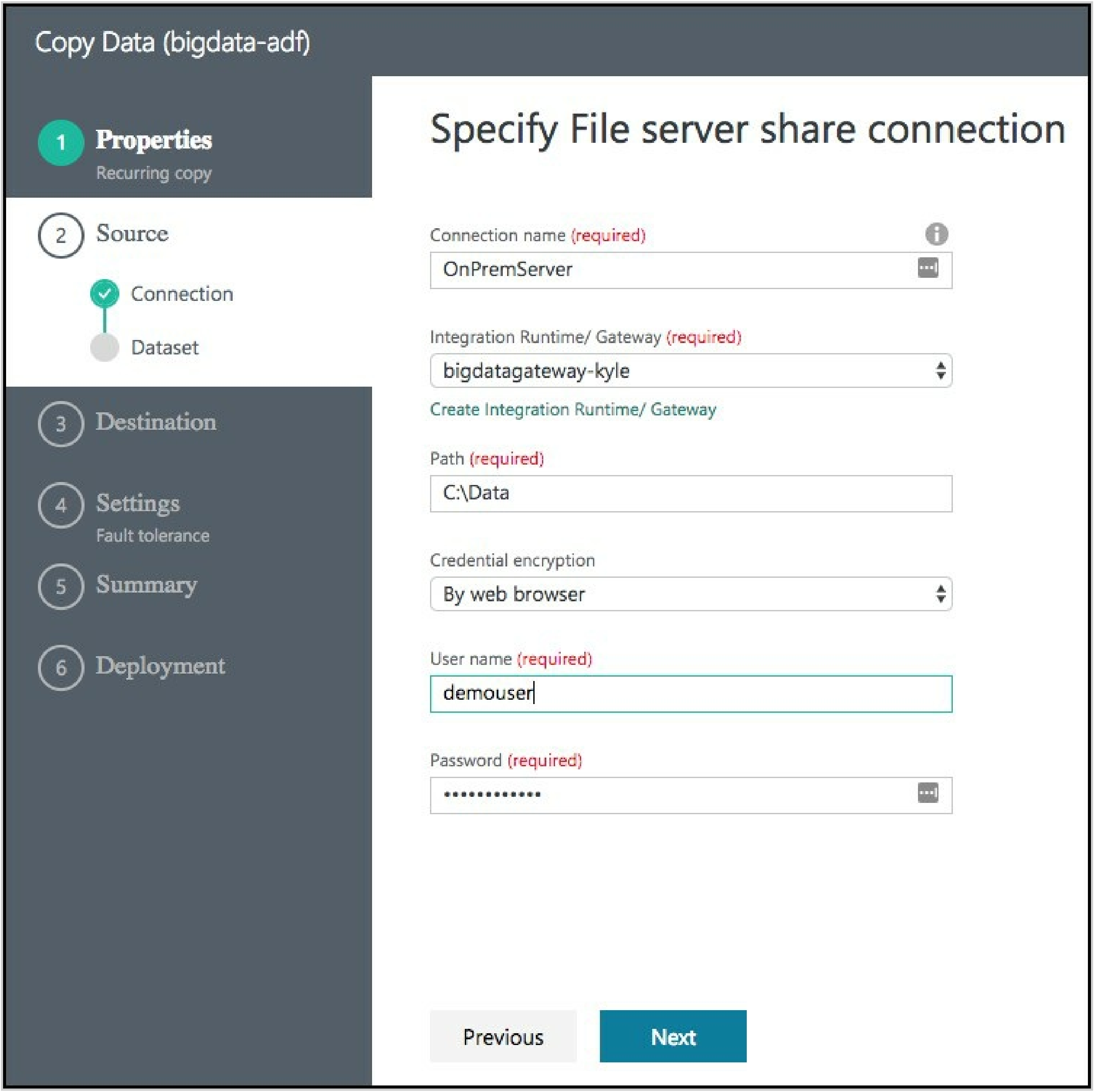


5. From the Specify File server share connection screen, enter the following:

- Connection name: **OnPremServer**
- Integration Runtime/Gateway: Select the Integration runtime created previously in

this exercise (this value should already be populated)

- Path: Enter **C:\Data**
- Credential encryption: Select **By web browser**
- User name: Enter **demouser**
- Password: Enter **Password.1!!**
- Select **Next**



Specify File server share connection

Connection name (required): OnPremServer

Integration Runtime/ Gateway (required): bigdatagateway-kyle

Create Integration Runtime/ Gateway

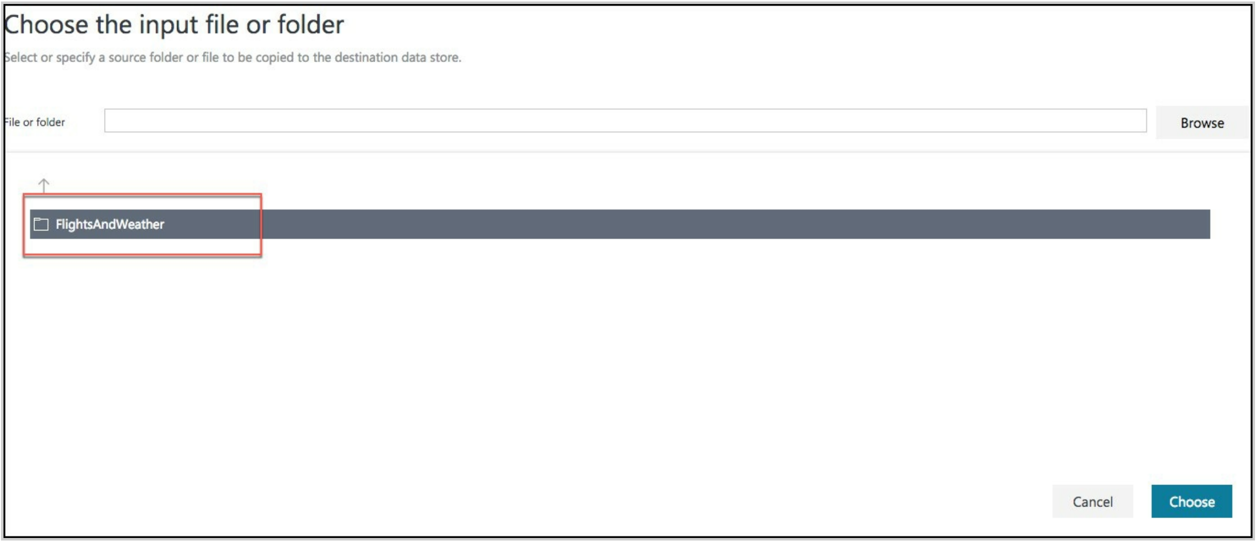
Path (required): C:\Data

Credential encryption: By web browser

User name (required): demouser

Password (required): •••••••••••••

6. On the Choose the input file or folder screen, select the folder **FlightsAndWeather**, and select Choose.

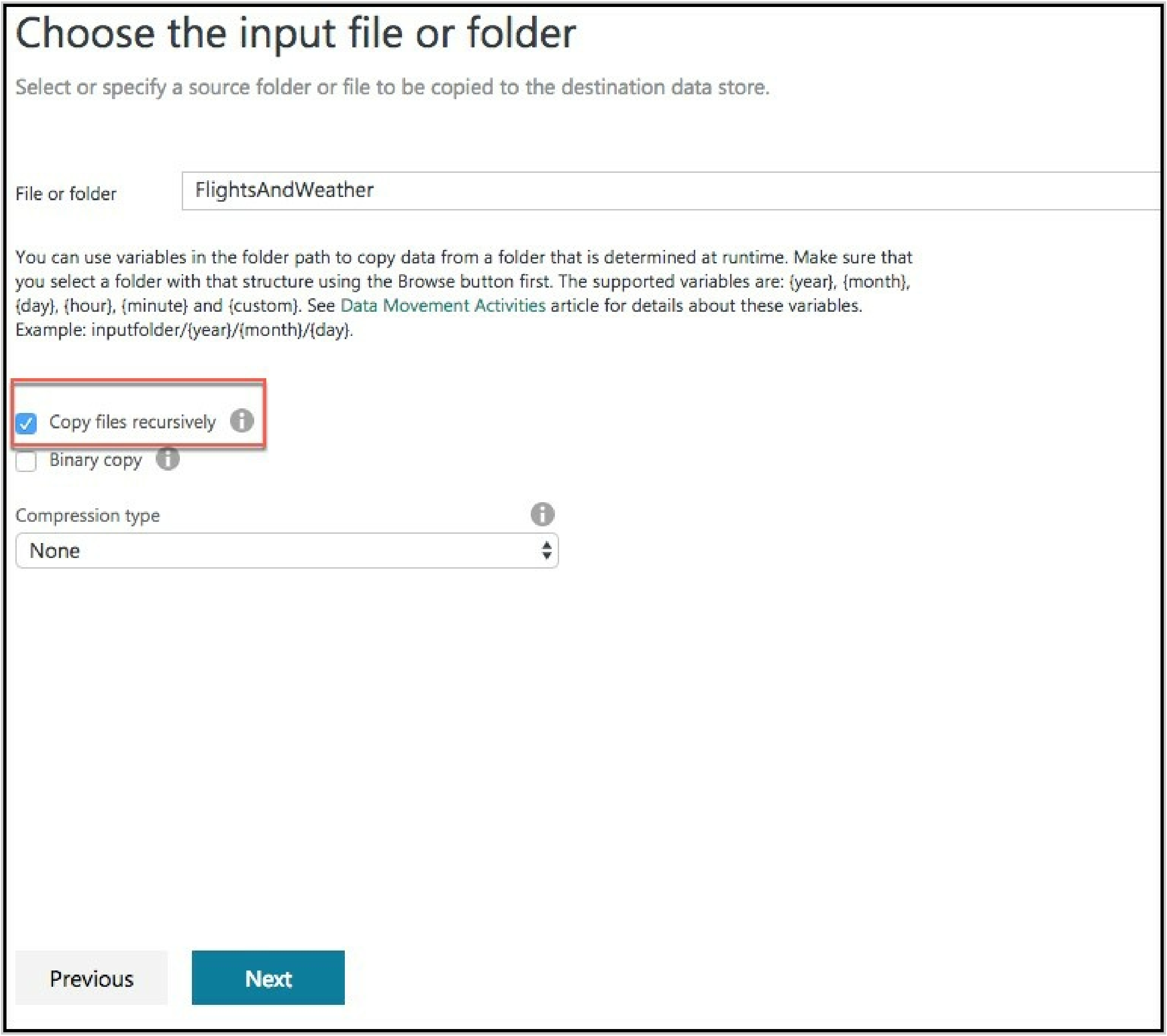


Choose the input file or folder

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

File or folder

7. On the next screen, check the **Copy files recursively** check box, and select **Next**.



Choose the input file or folder

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

File or folder: FlightsAndWeather

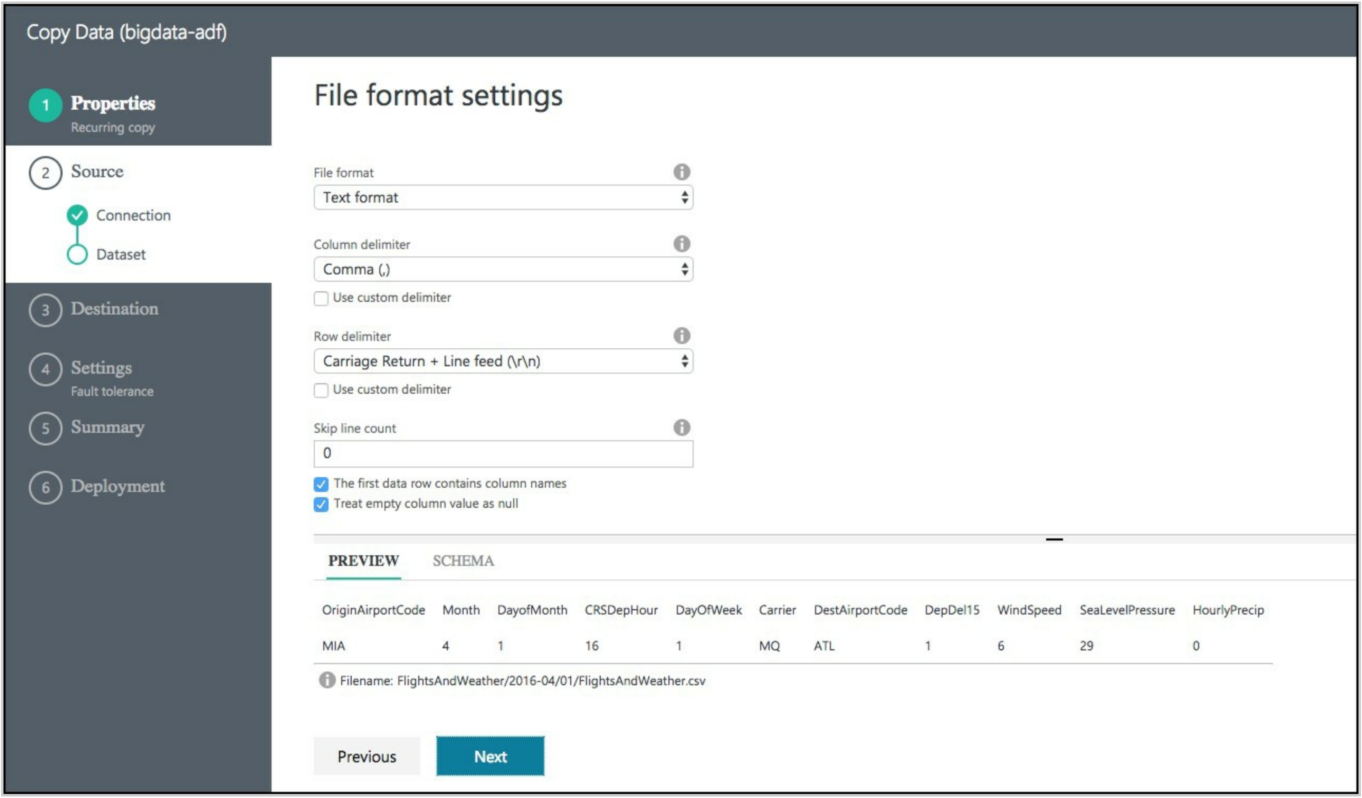
You can use variables in the folder path to copy data from a folder that is determined at runtime. Make sure that you select a folder with that structure using the Browse button first. The supported variables are: {year}, {month}, {day}, {hour}, {minute} and {custom}. See Data Movement Activities article for details about these variables. Example: inputfolder/{year}/{month}/{day}.

☑ Copy files recursively

□ Binary copy

Compression type: None

8. On the File format settings page, leave the default settings, and select **Next**.



Copy Data (bigdata-adf)

1 Properties
Recurring copy

2 Source
Connection
Dataset

3 Destination

4 Settings
Fault tolerance

5 Summary

6 Deployment

File format settings

File format: Text format

Column delimiter: Comma (,)

□ Use custom delimiter

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

□ Use custom delimiter

Skip line count: 0

☑ The first data row contains column names

☑ Treat empty column value as null

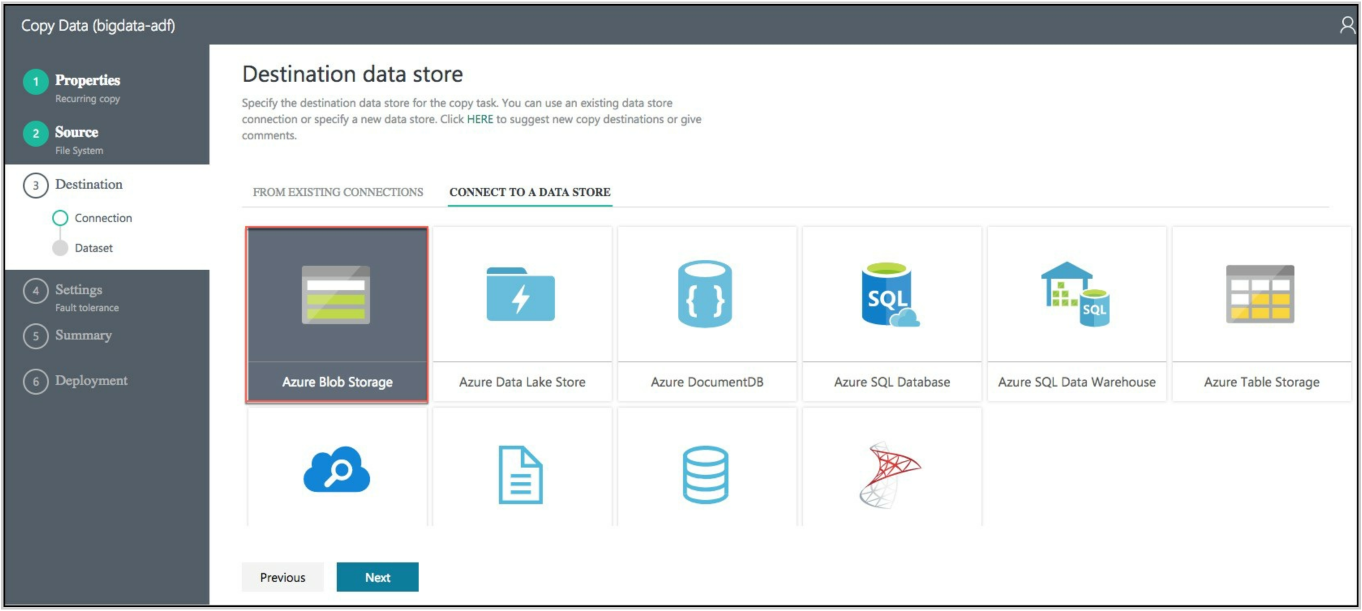
PREVIEW SCHEMA

OriginAirportCode	Month	DayofMonth	CRSDepHour	DayOfWeek	Carrier	DestAirportCode	DepDel15	WindSpeed	SeaLevelPressure	HourlyPrecip
MIA	4	1	16	1	MQ	ATL	1	6	29	0

Filename: FlightsAndWeather/2016-04/01/FlightsAndWeather.csv

Previous Next

9. On the Destination screen, select **Azure Blob Storage**, and select **Next**.



Copy Data (bigdata-adf)

1 Properties
Recurring copy

2 Source
File System

3 Destination
Connection
Dataset

4 Settings
Fault tolerance

5 Summary

6 Deployment

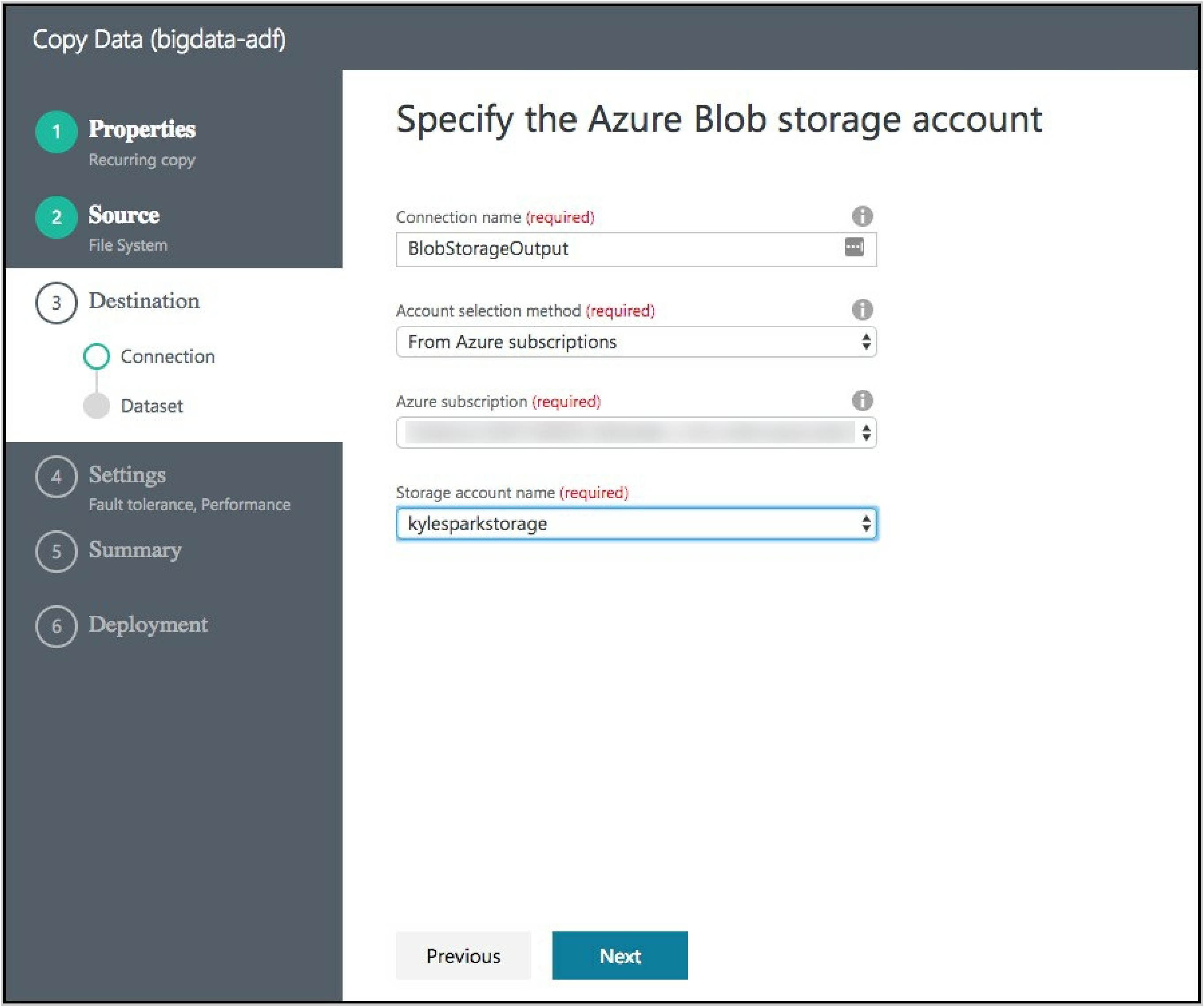
Destination data store

Specify the destination data store for the copy task. You can use an existing data store connection or specify a new data store. Click HERE to suggest new copy destinations or give comments.

FROM EXISTING CONNECTIONS CONNECT TO A DATA STORE

Previous Next

10. On the Specify the Azure Blob storage account screen, enter the following:
    - Connection name: BlobStorageOutput
    - Account selection method: Leave as From Azure subscriptions
    - Azure Subscription: Select your subscription
    - Storage account name: Select <YOUR_APP_NAME>sparkstorage. Make sure you select the storage account with the **sparkstorage** suffix, or you will have issues with subsequent exercises. This ensures data will be copied to the storage account that the Spark cluster users for its data files.



Specify the Azure Blob storage account

Connection name (required): BlobStorageOutput

Account selection method (required): From Azure subscriptions

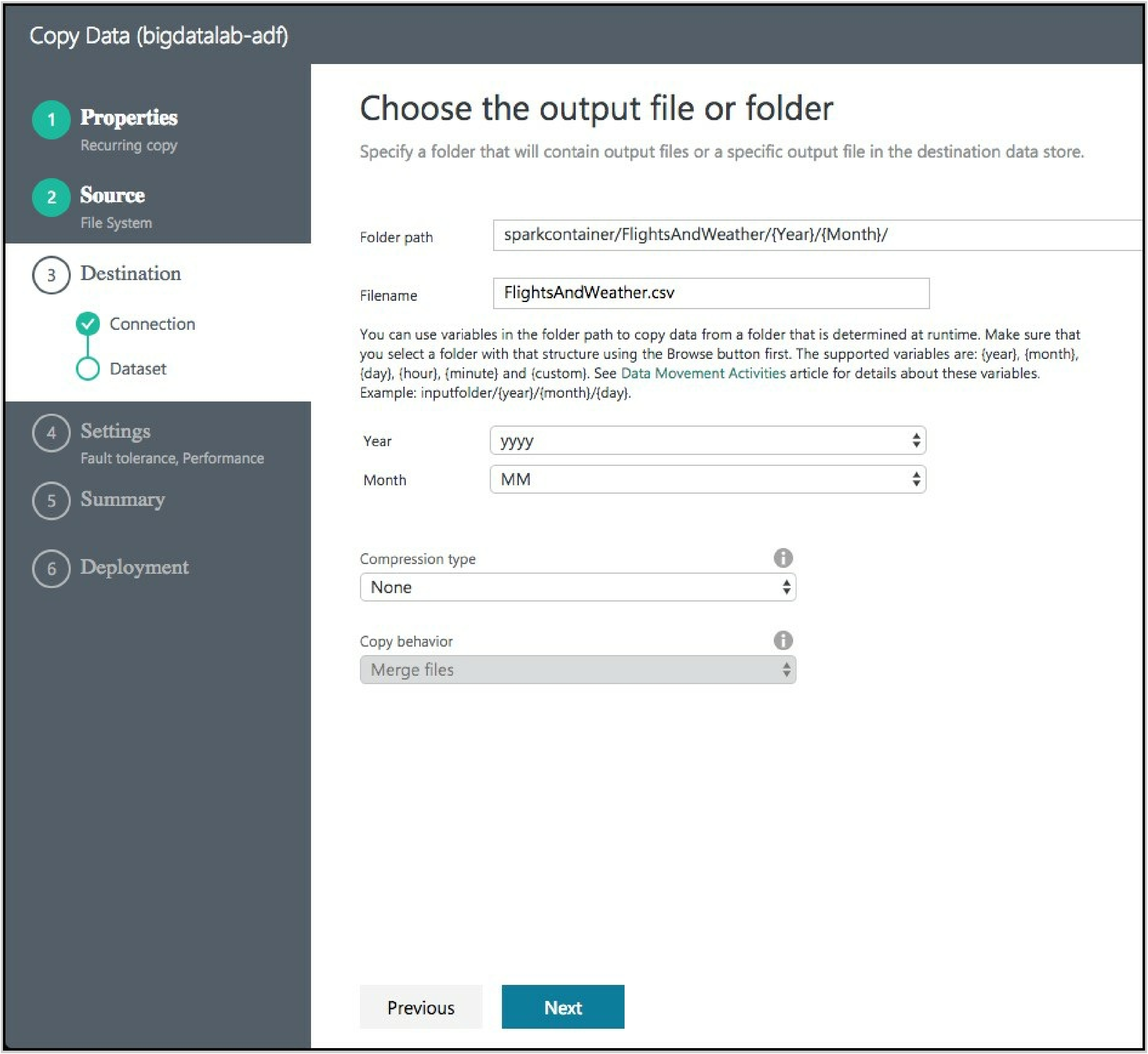
Azure subscription (required): [redacted]

Storage account name (required): kylesparkstorage

11. Before selecting Next, please ensure you have selected the proper **sparkstorage** account. Finally, select **Next**.

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

    - Folder path: Enter **sparkcontainer/FlightsAndWeather/{Year}/{Month}/**
    - Filename: Enter **FlightsAndWeather.csv**
    - Year: Select **yyyy** from the drop down
    - Month: Leave as **MM**
    - Select **Next**.



Copy Data (bigdatalab-adf)

Choose the output file or folder

Specify a folder that will contain output files or a specific output file in the destination data store.

Folder path: sparkcontainer/FlightsAndWeather/{Year}/{Month}/

Filename: FlightsAndWeather.csv

You can use variables in the folder path to copy data from a folder that is determined at runtime. Make sure that you select a folder with that structure using the Browse button first. The supported variables are: {year}, {month}, {day}, {hour}, {minute} and {custom}. See Data Movement Activities article for details about these variables. Example: inputfolder/{year}/{month}/{day}.

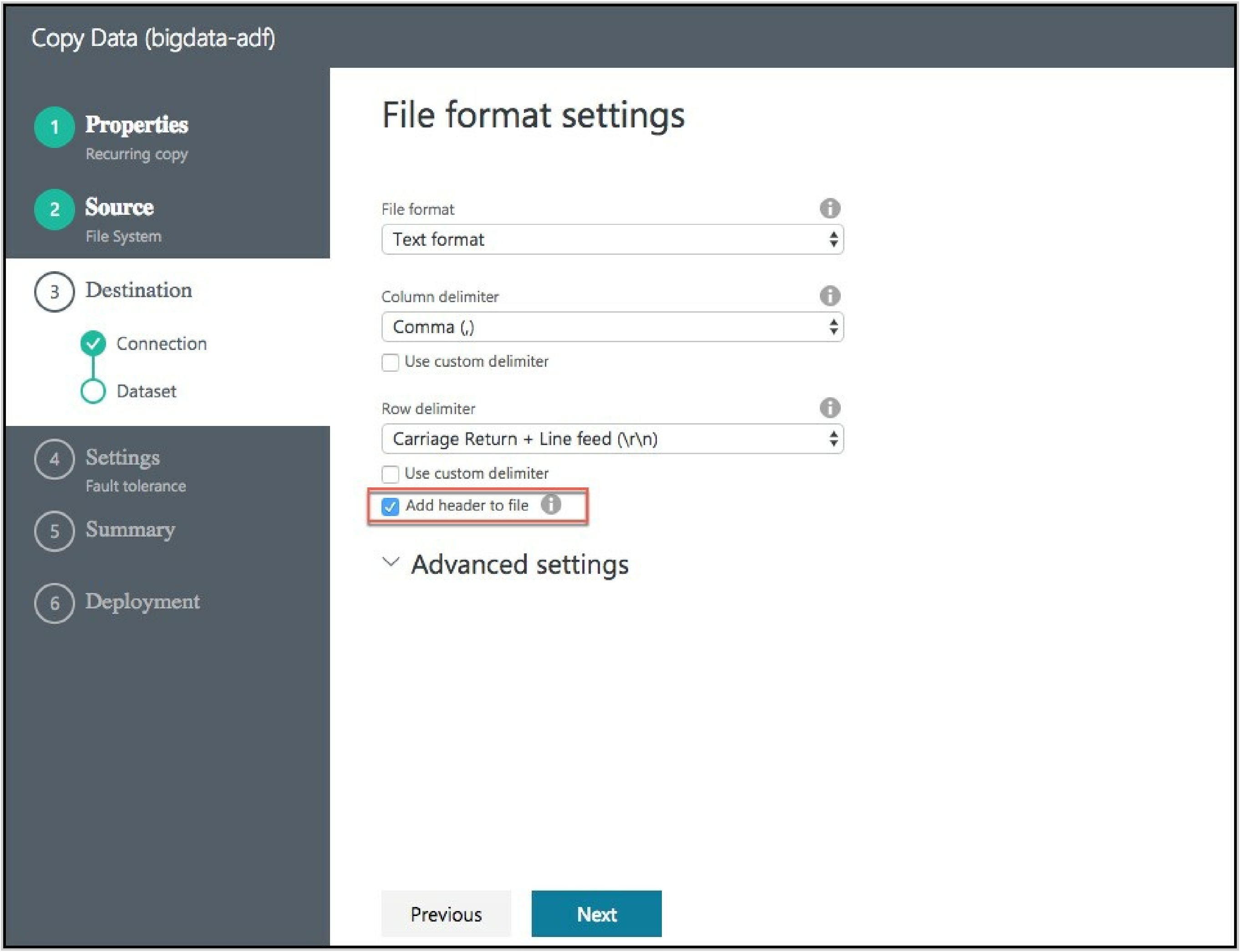
Year: yyyy

Month: MM

Compression type: None

Copy behavior: Merge files

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



File format settings

File format: Text format

Column delimiter: Comma (,)

□ Use custom delimiter

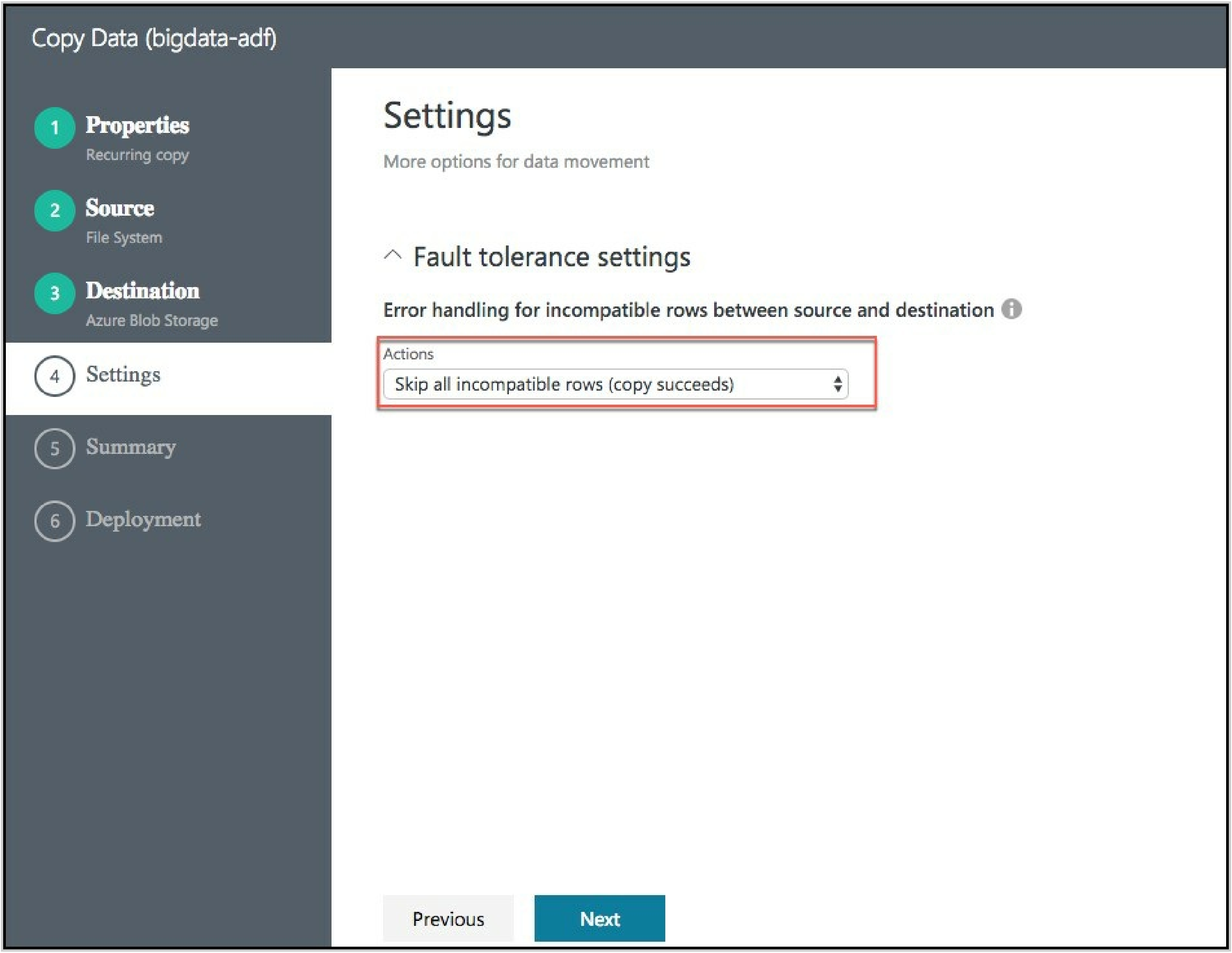
Row delimiter: Carriage Return + Line feed (\r\n)

□ Use custom delimiter

☑ Add header to file

Advanced settings

13. On the Settings screen, select **Skip all incompatible rows** under Actions, then select **Next**.



Settings

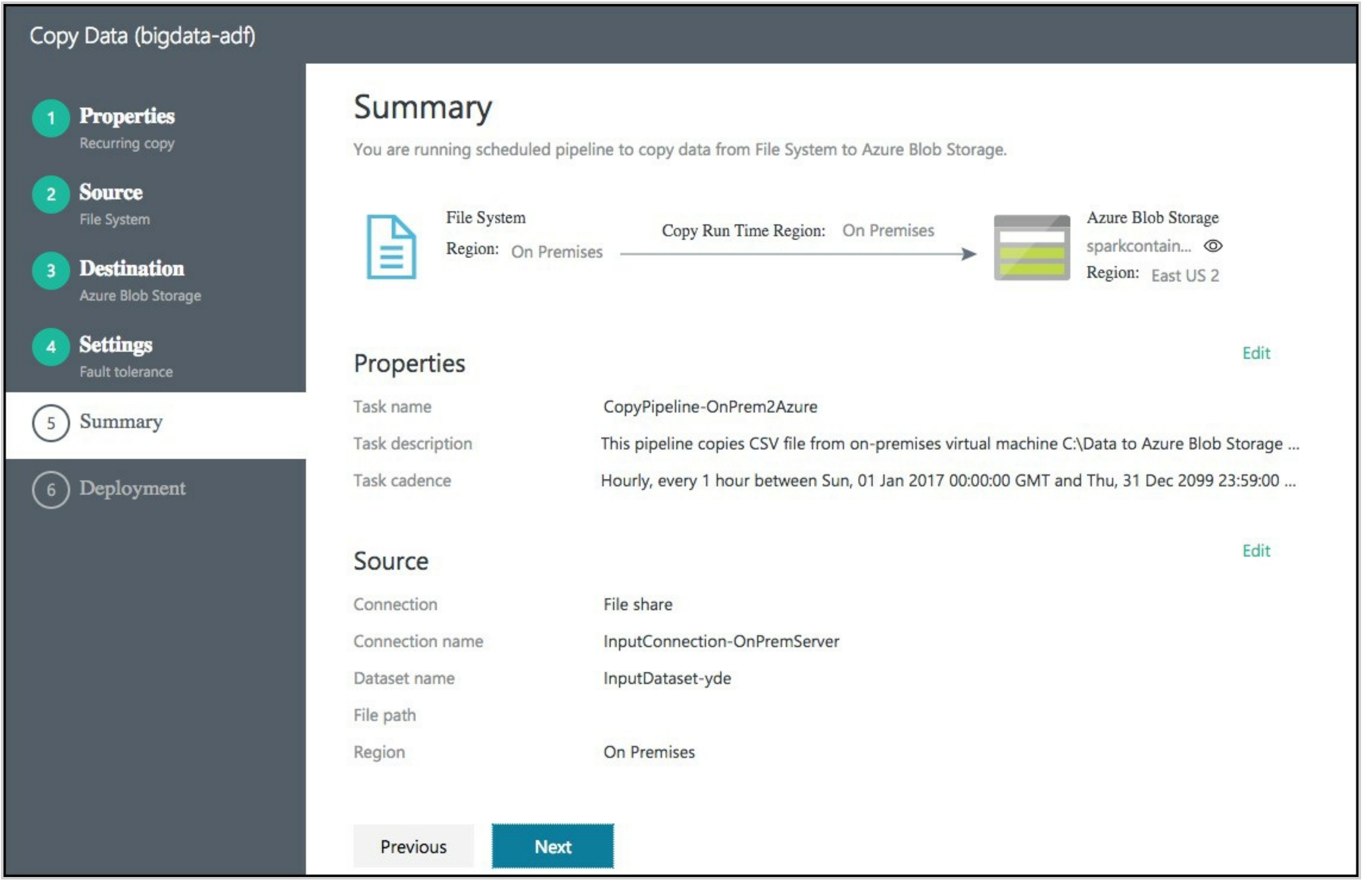
More options for data movement

Fault tolerance settings

Error handling for incompatible rows between source and destination

Actions: Skip all incompatible rows (copy succeeds)

14. Review settings on the Summary tab.



Copy Data (bigdata-adf)

1 **Properties**
Recurring copy

2 **Source**
File System

3 **Destination**
Azure Blob Storage

4 **Settings**
Fault tolerance

5 Summary

6 Deployment

Summary

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

File System
Region: On Premises

Copy Run Time Region: On Premises

Azure Blob Storage
sparkcontain...
Region: East US 2

Properties

Edit

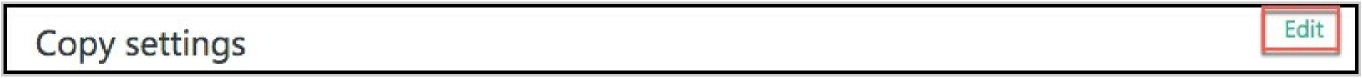
Task name	CopyPipeline-OnPrem2Azure
Task description	This pipeline copies CSV file from on-premises virtual machine C:\Data to Azure Blob Storage ...
Task cadence	Hourly, every 1 hour between Sun, 01 Jan 2017 00:00:00 GMT and Thu, 31 Dec 2099 23:59:00 ...

Source

Edit

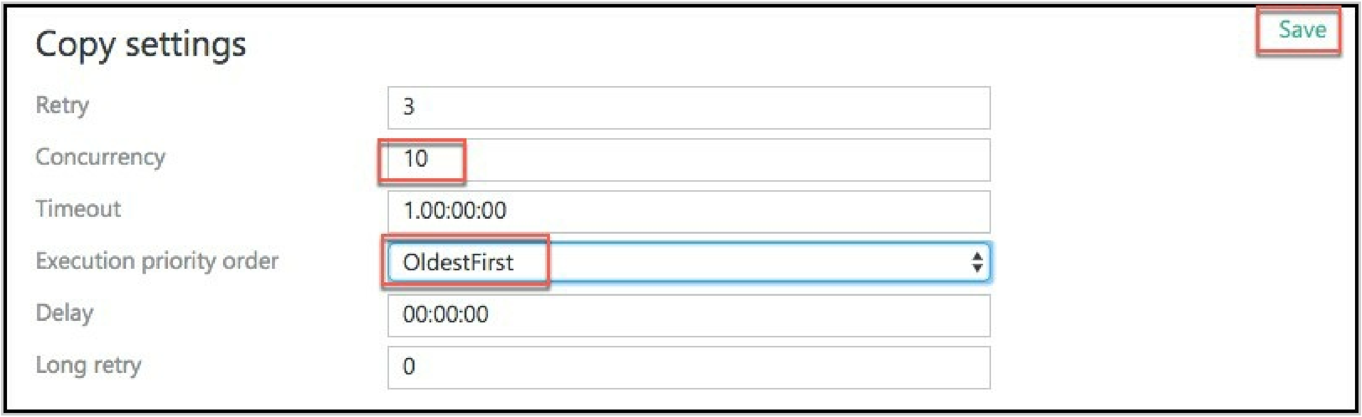
Connection	File share
Connection name	InputConnection-OnPremServer
Dataset name	InputDataset-yde
File path	
Region	On Premises

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



Copy settings

16. Change the following Copy settings
    - Concurrency: Set to **10**
    - Execution priority order: Change to **OldestFirst**
    - Select **Save**

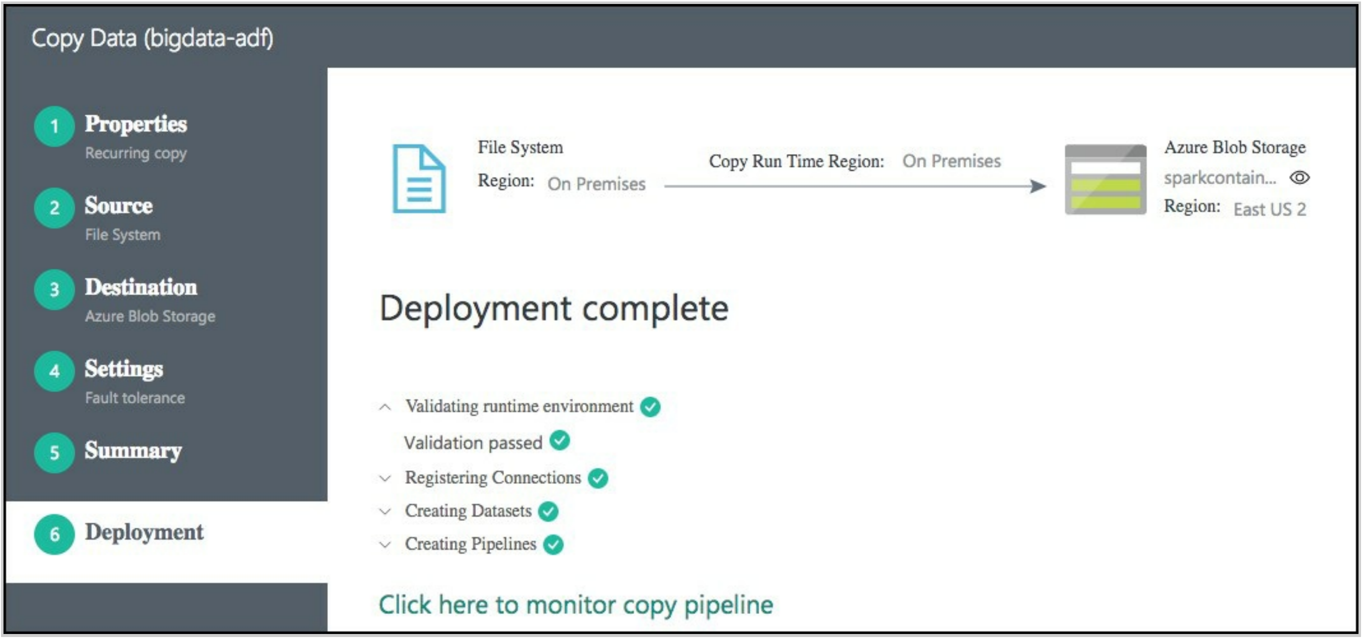


Copy settings

Retry	3
Concurrency	10
Timeout	1.00:00:00
Execution priority order	OldestFirst
Delay	00:00:00
Long retry	0

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

18. On the Deployment screen you will see a message that the deployment in is progress, and after a minute or two that the deployment completed.



Copy Data (bigdata-adf)

1 **Properties** Recurring copy
2 **Source** File System
3 **Destination** Azure Blob Storage
4 **Settings** Fault tolerance
5 **Summary**
6 **Deployment**

File System
Region: On Premises

Copy Run Time Region: On Premises

Azure Blob Storage
sparkcontain...
Region: East US 2

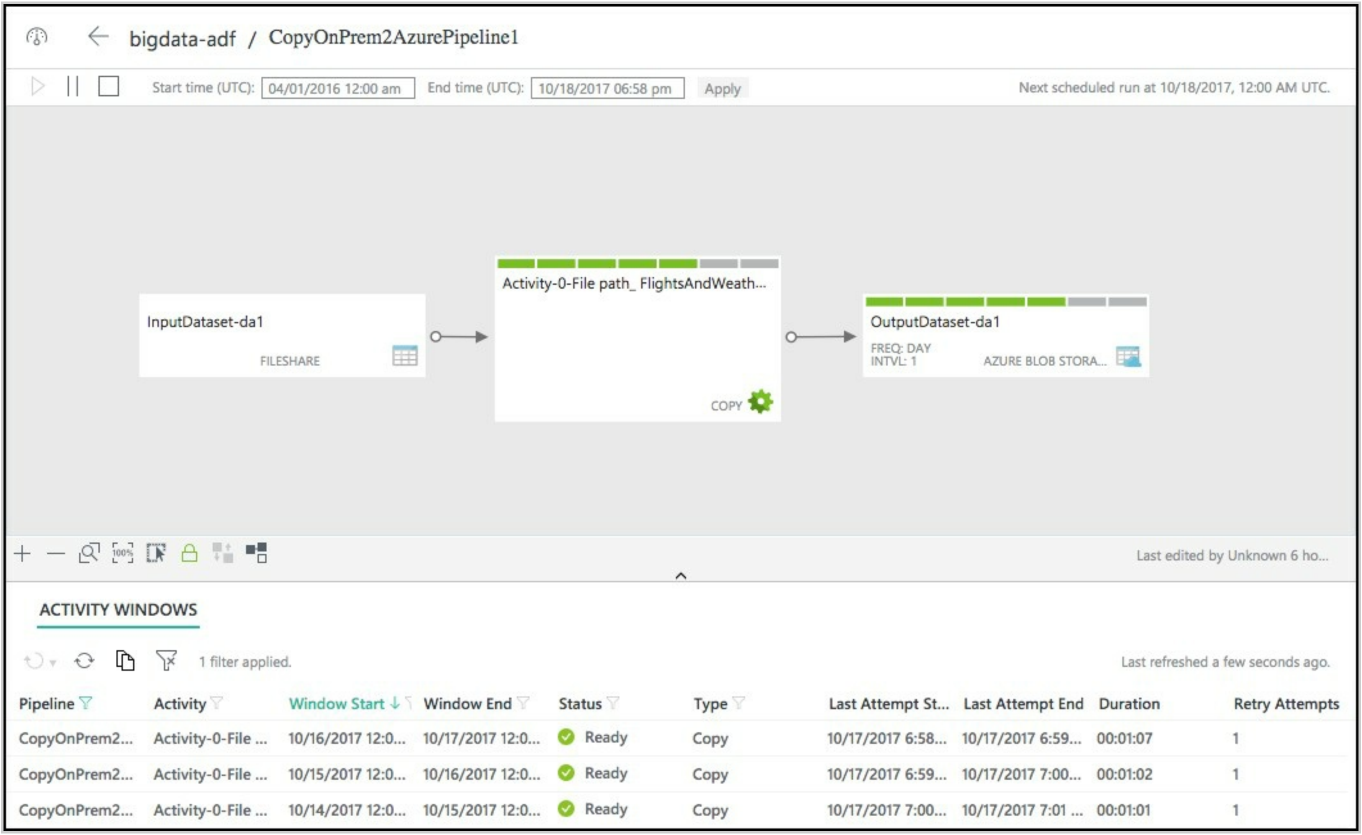
Deployment complete

- Validating runtime environment ✓
  - Validation passed ✓
- Registering Connections ✓
- Creating Datasets ✓
- Creating Pipelines ✓

Click here to monitor copy pipeline

19. Select the **Click here to monitor copy pipeline** link at the bottom of the **Deployment** screen.

20. From the Data Factory Resource Explorer, you should see the pipeline activity status **Ready**. This indicates the CSV files are successfully copied from your VM to your Azure Blob Storage location.



bigdata-adf / CopyOnPrem2AzurePipeline1

Start time (UTC): 04/01/2016 12:00 am End time (UTC): 10/18/2017 06:58 pm Apply

Next scheduled run at 10/18/2017, 12:00 AM UTC.

Last edited by Unknown 6 ho...

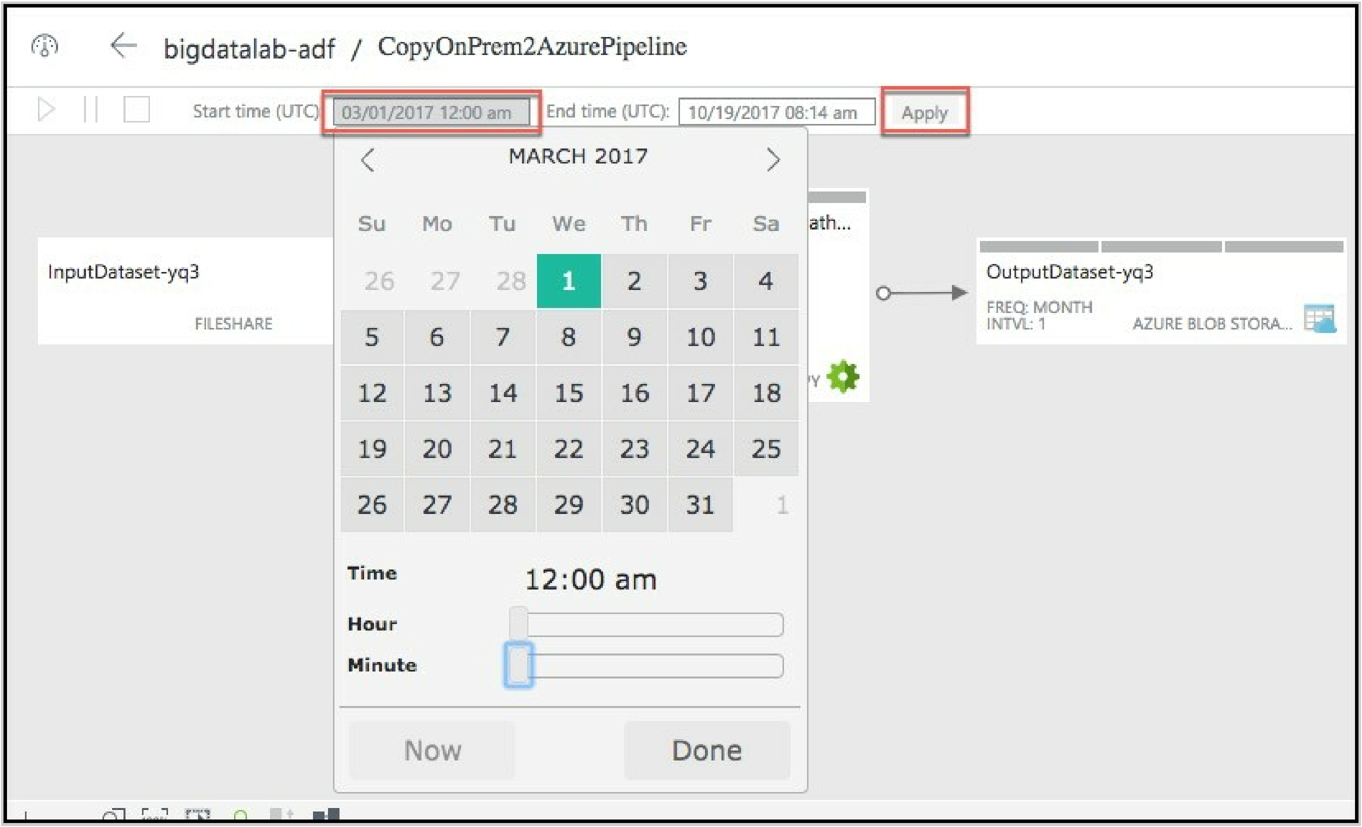
ACTIVITY WINDOWS

1 filter applied.

Last refreshed a few seconds ago.

Pipeline	Activity	Window Start ↓	Window End	Status	Type	Last Attempt St...	Last Attempt End	Duration	Retry Attempts
CopyOnPrem2...	Activity-0-File ...	10/16/2017 12:0...	10/17/2017 12:0...	✓ Ready	Copy	10/17/2017 6:58...	10/17/2017 6:59...	00:01:07	1
CopyOnPrem2...	Activity-0-File ...	10/15/2017 12:0...	10/16/2017 12:0...	✓ Ready	Copy	10/17/2017 6:59...	10/17/2017 7:00...	00:01:02	1
CopyOnPrem2...	Activity-0-File ...	10/14/2017 12:0...	10/15/2017 12:0...	✓ Ready	Copy	10/17/2017 7:00...	10/17/2017 7:01 ...	00:01:01	1

4. You may need to adjust the Start time in the window, as follows, and then select **Apply**.



bigdatalab-adf / CopyOnPrem2AzurePipeline

Start time (UTC) 03/01/2017 12:00 am End time (UTC): 10/19/2017 08:14 am Apply

MARCH 2017

Su	Mo	Tu	We	Th	Fr	Sa
26	27	28	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	1

Time 12:00 am

Hour

Minute

Operationalize ML Scoring with Azure ML and Data Factory

Exercise 4: Operationalize ML scoring with Azure ML and Data Factory

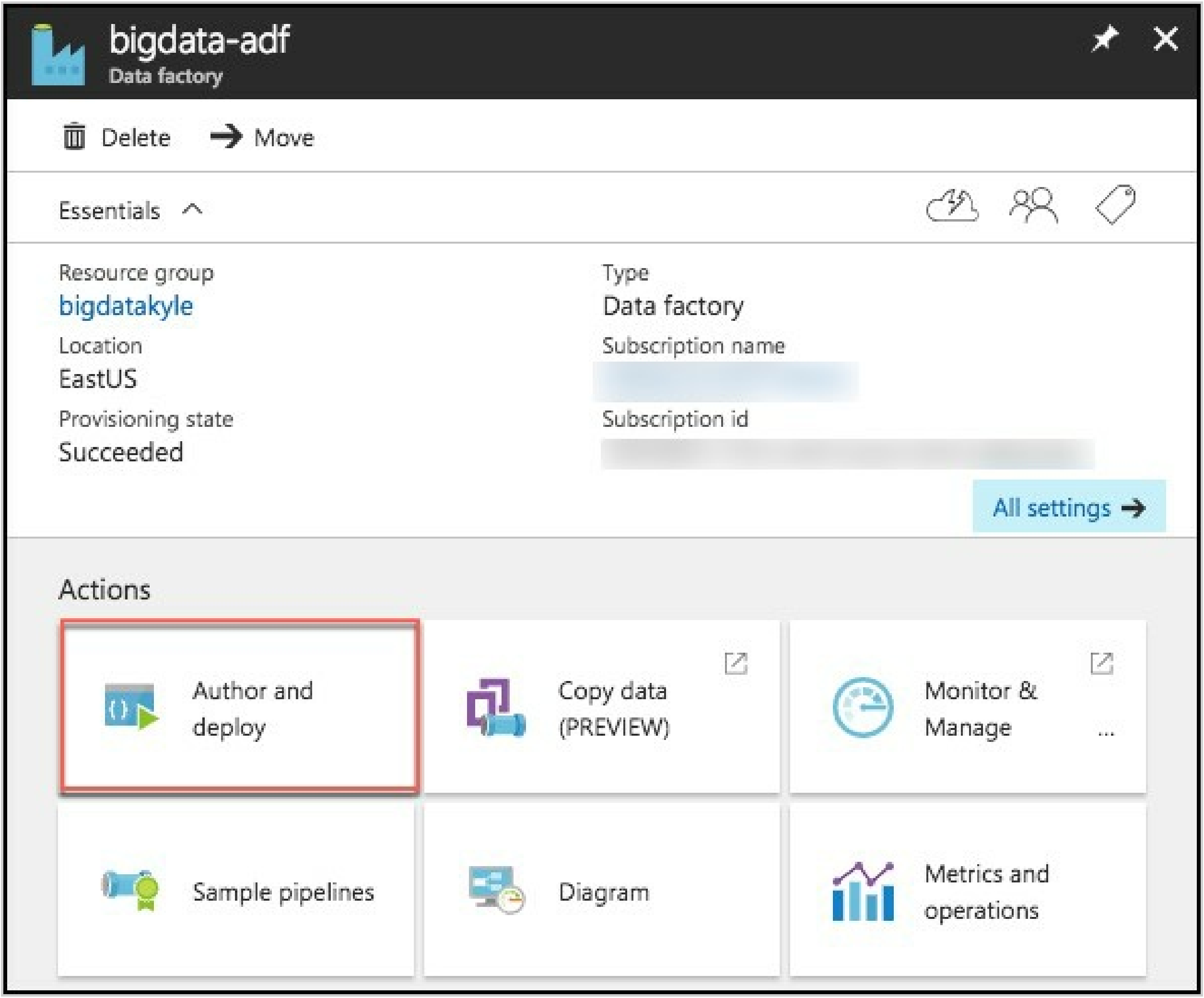
Duration: 20 mins

Synopsis: In this exercise, you will extend the Data Factory to operationalize the scoring of data using the previously created Azure Machine Learning (ML) model.

Task 1: Create Azure ML Linked Service

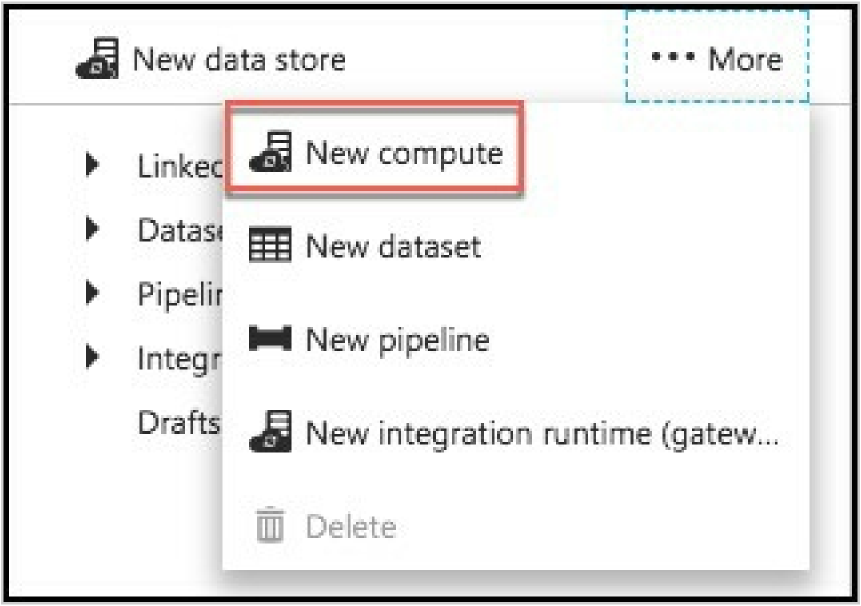
1. Return to the Azure Data Factory blade in the Azure portal

2. Select **Author and Deploy** below Actions.

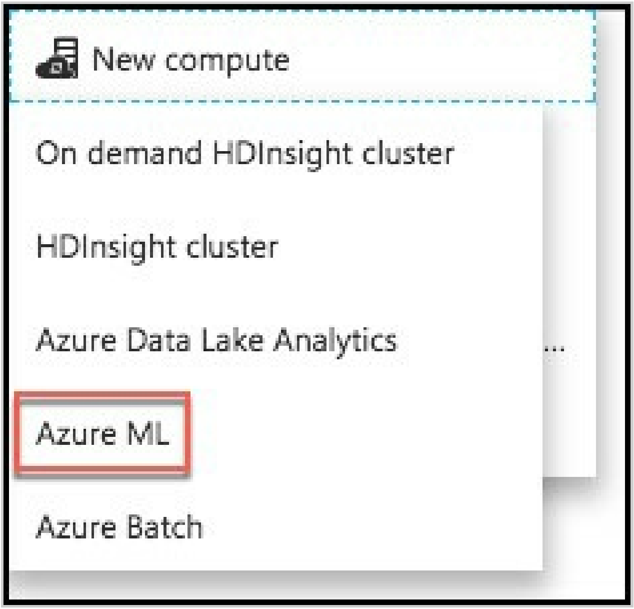


Resource group
bigdatakyle
Location
EastUS
Provisioning state
Succeeded
Type
Data factory
Subscription name
[redacted]
Subscription id
[redacted]

3. On the Author and Deploy blade, select **…More**, the select **New Compute**.



4. Select Azure ML from the New Compute list.

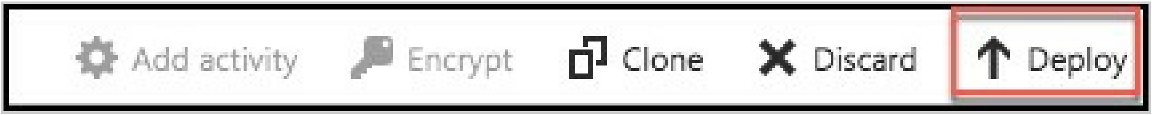


5. In the new window, replace the contents of the file with the following JSON.

   - Back in Exercise 1, Task 9, you left your ML Web Service’s Consume page open. Return to that page, and copy and paste the following values into the JSON below.
   - The value of **mlEndpoint** below is your web service’s **Batch Request URL**, remember to **remove the query string (e.g., “?api_version=2.0”).
   - **apiKey** is the Primary Key of your web service.
   - Your tenant string should be populated automatically.
   - Delete the other optional settings (updateResourceEndpoint, servicePrincipalId, servicePrincipalKey).

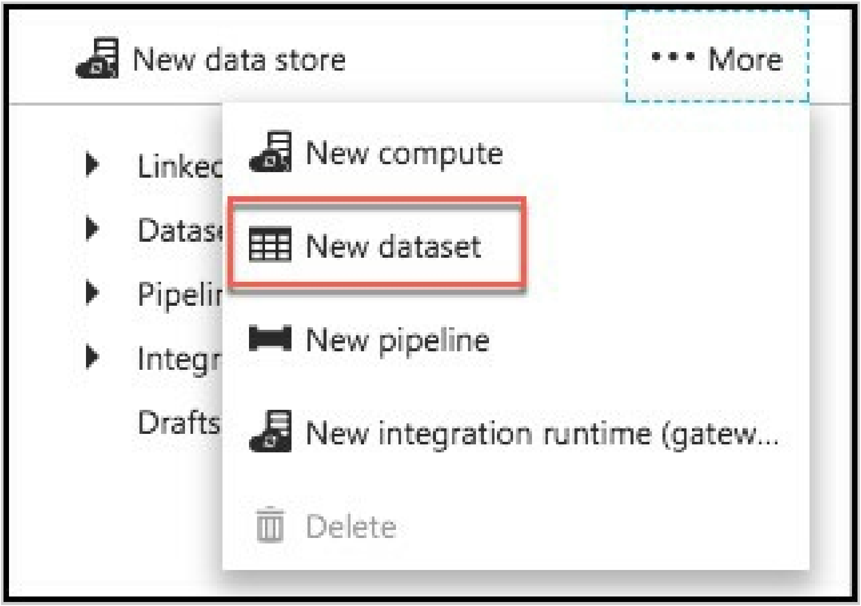
```
{
"name": "AzureMLLinkedService",
"properties": {
    "type": "AzureML",
    "description": "",
    "typeProperties": {
        "mlEndpoint": "<Specify the batch scoring URL>",
        "apiKey": "<Specify the published workspace model’s API key>",
        "tenant": "<Specify your tenant string>"
                        }
                }
}
```

6. Select **Deploy**.

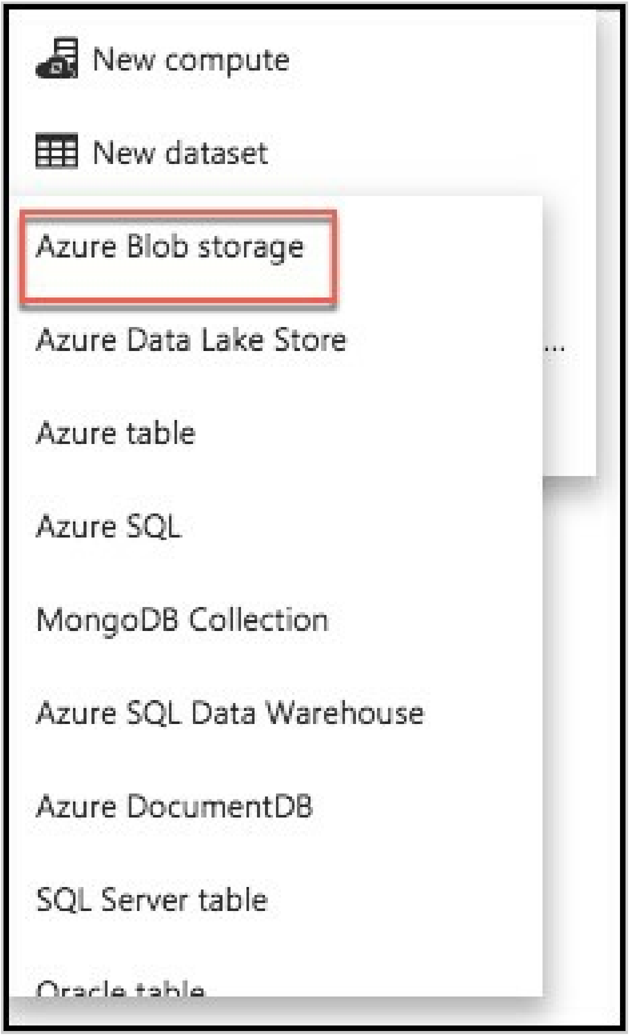


Task 2: Create Azure ML input dataset

1. Still on the Author and Deploy blade, select **…More** again.
2. To create a new dataset that will be copied into Azure Blob storage, select **New dataset** from the top.



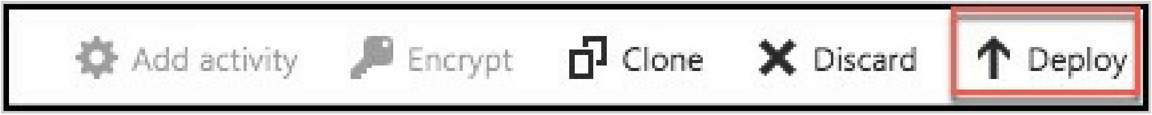
3. Select **Azure Blob storage** from the list of available datasets.



4. Replace the JSON text in the draft window with following JSON.

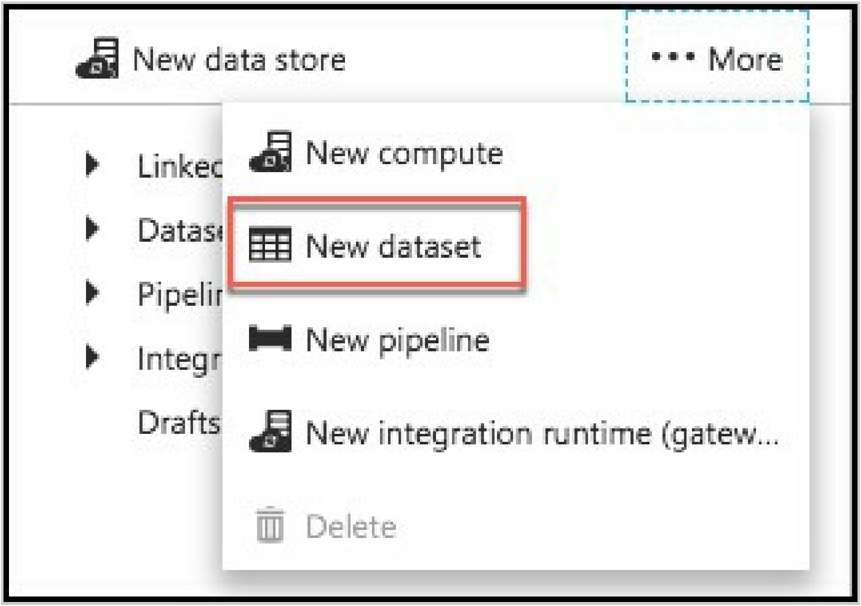
```
{
    "name": "PartitionedBlobInput",
    "properties": {
        "published": false,
        "type": "AzureBlob",
        "linkedServiceName": "BlobStorageOutput",
        "typeProperties": {
            "fileName": "FlightsAndWeather.csv",
            "folderPath": "sparkcontainer/FlightsAndWeather/{Year}/{Month}/",
            "format": {
                "type": "TextFormat"
            },
            "partitionedBy": [
                {
                    "name": "Year",
                    "value": {
                        "type": "DateTime",
                        "date": "SliceStart",
                        "format": "yyyy"
                    }
                },
                {
                    "name": "Month",
                    "value": {
                        "type": "DateTime",
                        "date": "SliceStart",
                        "format": "MM"
                    }
                }
            ]
        },
        "availability": {
            "frequency": "Month",
            "interval": 1
        },
        "external": true,
        "policy": {}
    }
}
```

5. Select **Deploy**.

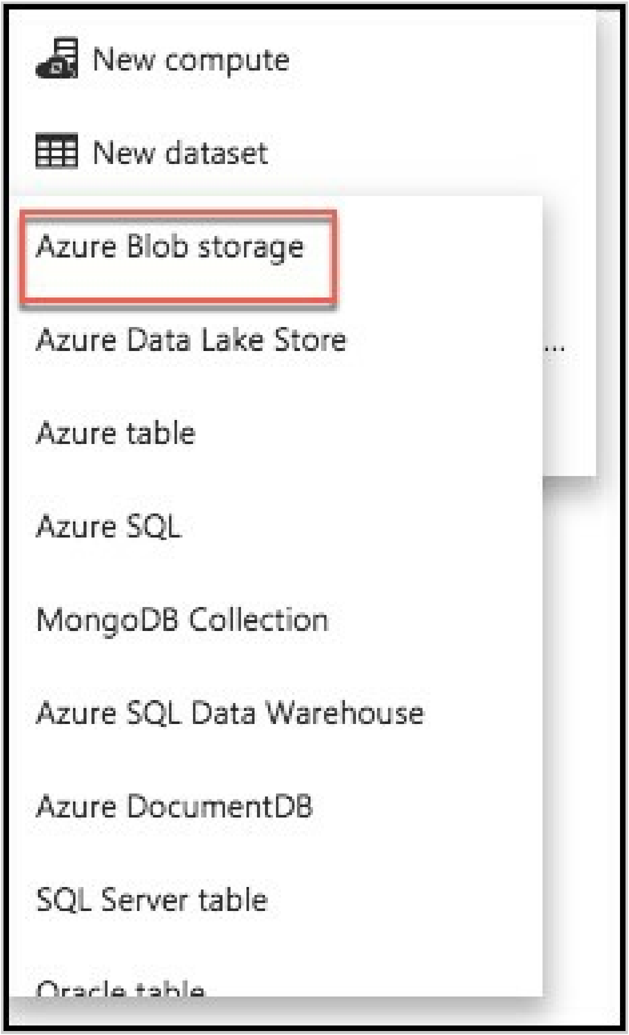


Task 3: Create Azure ML scored dataset

1. Select **…More** again, and select **New dataset**.



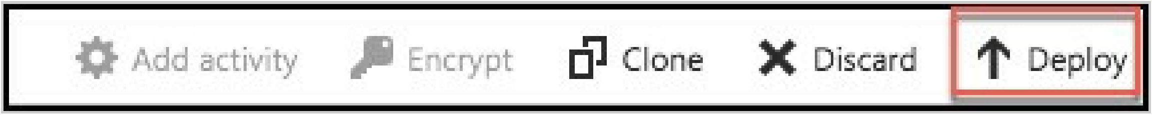
2. Select Azure Blob storage from the list of available datasets.



3. Replace the JSON text in the draft window with following JSON.

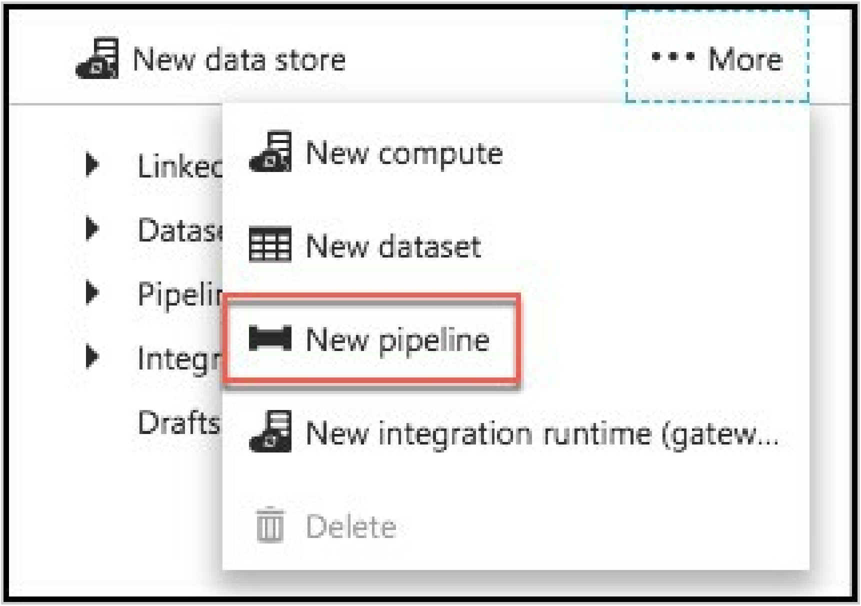
```
{
    "name": "ScoredBlobOutput",
    "properties": {
        "published": false,
        "type": "AzureBlob",
        "linkedServiceName": "BlobStorageOutput",
        "typeProperties": {
            "fileName": "Scored_FlightsAndWeather{Year}{Month}.csv",
            "folderPath": "sparkcontainer/ScoredFlightsAndWeather",
            "format": {
                "type": "TextFormat"
            },
            "partitionedBy": [
                {
                    "name": "Year",
                    "value": {
                        "type": "DateTime",
                        "date": "SliceStart",
                        "format": "yyyy"
                    }
                },
                {
                    "name": "Month",
                    "value": {
                        "type": "DateTime",
                        "date": "SliceStart",
                        "format": "MM"
                    }
                }
            ]
        },
        "availability": {
            "frequency": "Month",
            "interval": 1
        }
    }
}
```

4. Select **Deploy**.



Task 4: Create Azure ML predictive pipeline

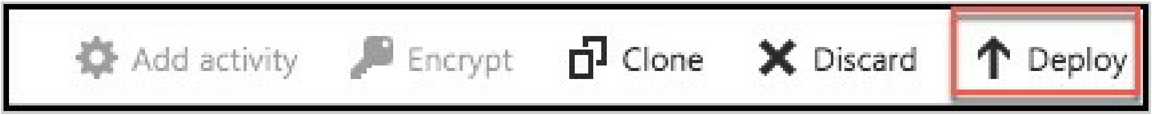
1. Select **…More** again, and select **New pipeline**.



2. Replace the JSON text in the draft window with following JSON.

```json
{
    "name": "MLPredictivePipeline",
    "properties": {
        "description": "Use AzureML model",
        "activities": [
            {
                "type": "AzureMLBatchExecution",
                "typeProperties": {
                    "webServiceInput": "PartitionedBlobInput",
                    "webServiceOutputs": {
                        "output1": "ScoredBlobOutput"
                    },
                    "webServiceInputs": {},
                    "globalParameters": {}
                },
                "inputs": [
                    {
                        "name": "PartitionedBlobInput"
                    }
                ],
                "outputs": [
                    {
                        "name": "ScoredBlobOutPut"
                    }
                ],
                "policy": {
                    "timeout": "02:00:00",
                    "concurrency": 10,
                    "retry": 1
                },
                "scheduler": {
                    "frequency": "Month",
                    "interval": 1
                },
                "name": "MLActivity",
                "description": "prediction analysis on batch input",
                "linkedServiceName": "AzureMLLinkedService"
            }
        ],
        "start": "2017-03-01T00:00:00Z",
        "end": "2099-12-31T11:59:59Z",
        "isPaused": false,
        "pipelineMode": "Scheduled"
    }
}
```

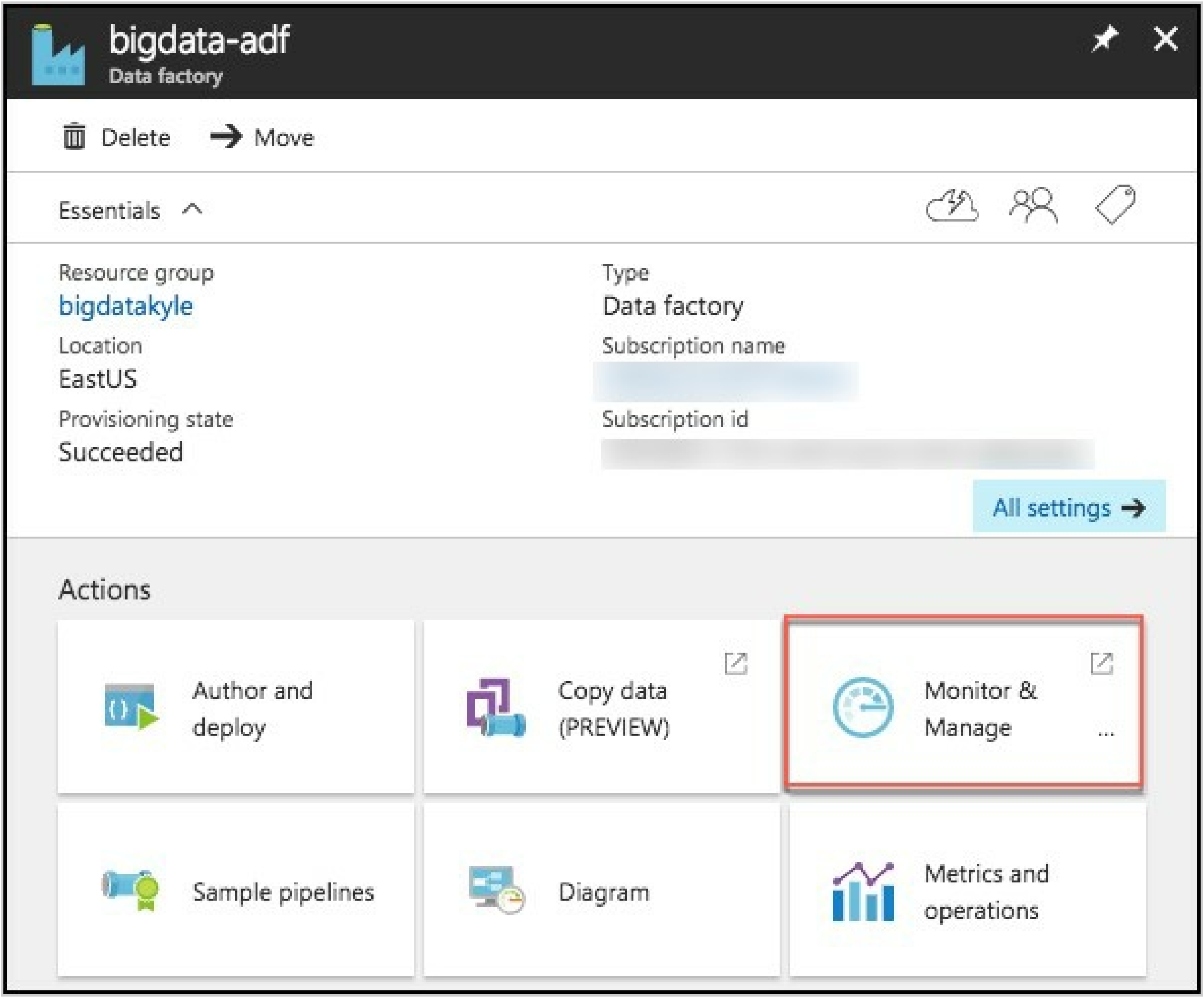
3. Select **Deploy**.



Task 5: Monitor pipeline activities

1. **Close the Author and Deploy blade**, and return to the Data Factory overview.

2. Select **Monitor & Manage** under **Actions**.



Resource group: bigdatakyle

Location: EastUS

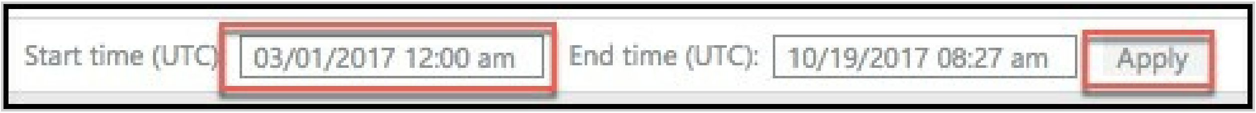
Provisioning state: Succeeded

Type: Data factory

Subscription name: [redacted]

Subscription id: [redacted]

3. Once again, you may need to shift the start time in order to see the items in progress and ready states.



Start time (UTC): 03/01/2017 12:00 am

End time (UTC): 10/19/2017 08:27 am

4. Close the Monitor & Manage browser tab.

Summarize data using HDInsight Spark

Exercise 5: Summarize data using HDInsight Spark

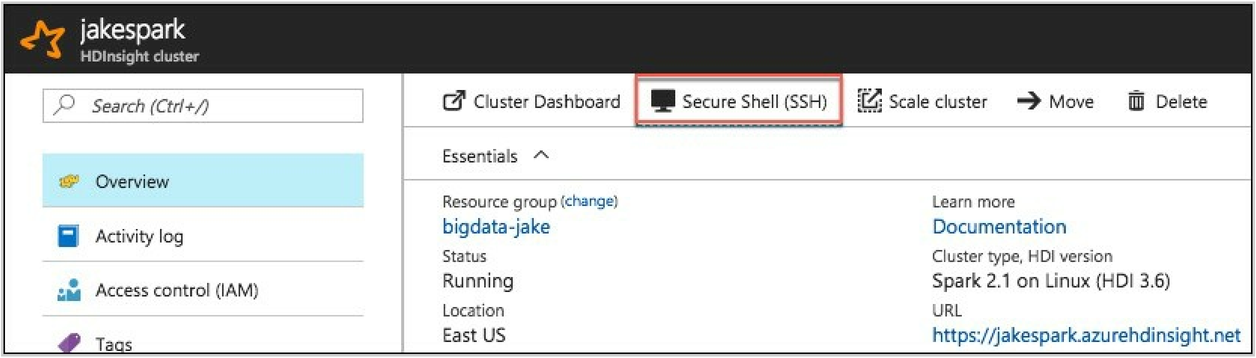
Duration: 20 mins

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

Task 1: Install pandas on the HDInsight cluster

In this task, you will upgrade the version of panda on the HDInsight cluster, to ensure the Jupyter notebook's autovixwidget has the necessary 'api' module installed.

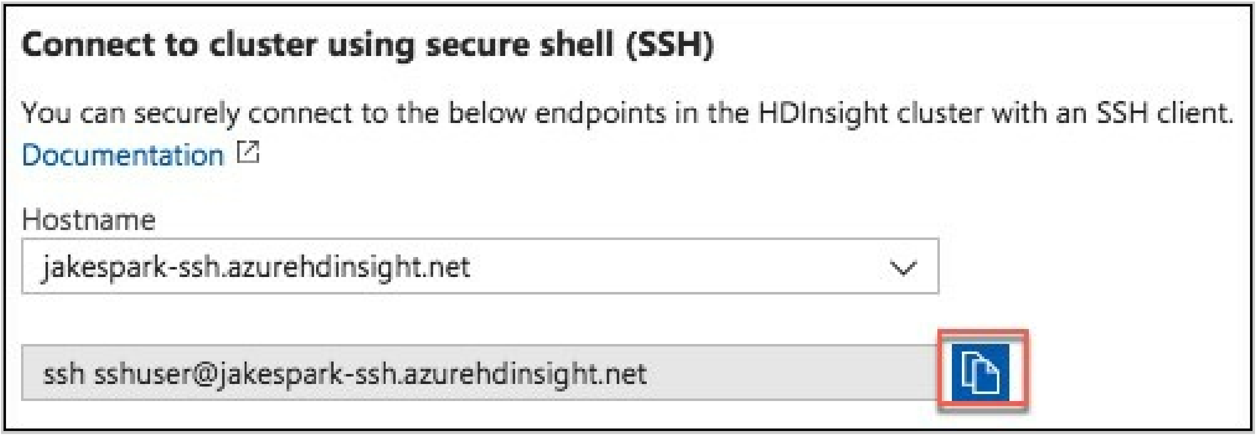
1. In the Azure portal, navigate to your HDInsight cluster, and from the Overview blade select Secure Shell (SSH).



Resource group (change)
bigdata-jake
Status
Running
Location
East US

Learn more
Documentation
Cluster type, HDI version
Spark 2.1 on Linux (HDI 3.6)
URL
https://jakespark.azurehdinsight.net

2. On the SSH + Cluster login blade, select your cluster from the Hostname drop down, then select the copy button next to the SSH command.



Connect to cluster using secure shell (SSH)

You can securely connect to the below endpoints in the HDInsight cluster with an SSH client.
Documentation

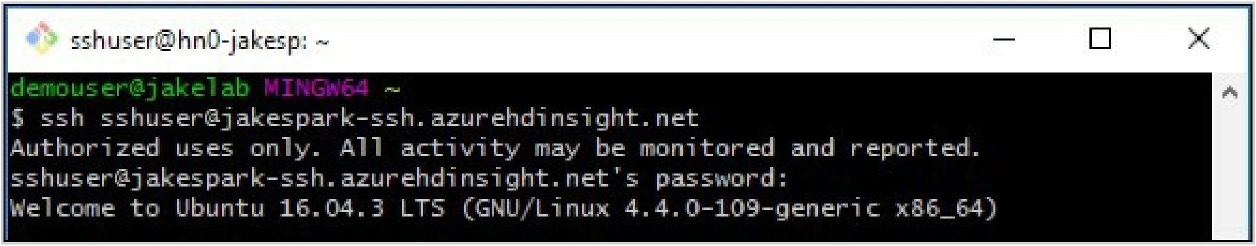
Hostname

jakespark-ssh.azurehdinsight.net

```
ssh sshuser@jakespark-ssh.azurehdinsight.net
```

3. On your Lab VM, open a **new Git Bash terminal** window.

4. At the prompt, paste the SSH command you copied from your HDInsight SSH + Cluster login blade.



```
demouser@jakelab MINGW64 ~
$ ssh sshuser@jakespark-ssh.azurehdinsight.net
Authorized uses only. All activity may be monitored and reported.
sshuser@jakespark-ssh.azurehdinsight.net's password:
Welcome to Ubuntu 16.04.3 LTS (GNU/Linux 4.4.0-109-generic x86_64)
```

5. Enter **yes**, if prompted about continuing, and **enter the following password** for the sshuser:

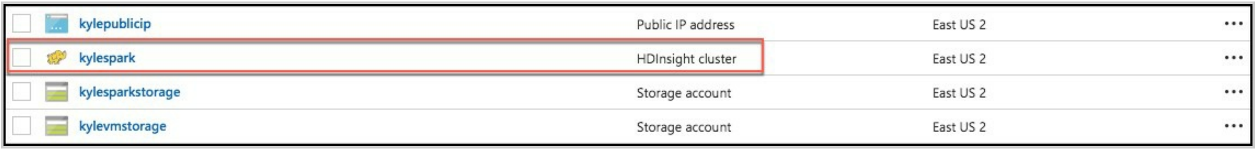
   - Abc!1234567890

6. At the sshuser prompt within the bash terminal, **enter the following command** to install pandas on the cluster:

   - sudo -HE /usr/bin/anaconda/bin/conda install pandas

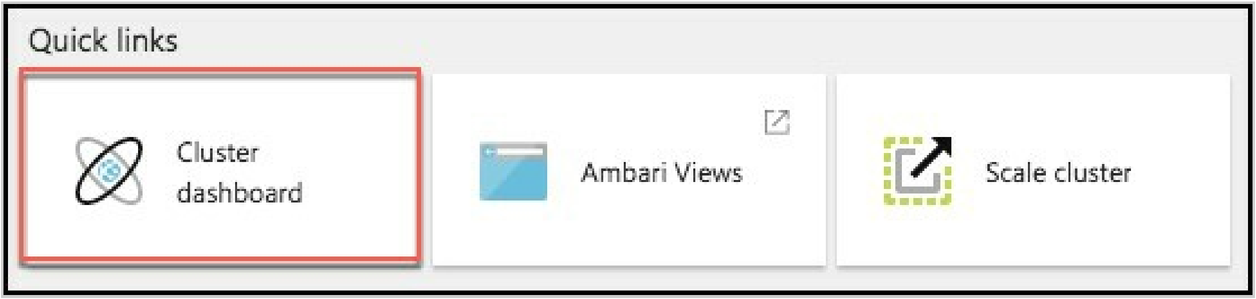
Task 2: Summarize delays by airport

1. In the Azure portal (https://portal.azure.com), navigate to the blade for your Spark cluster. Do this by going to the resource group you created during the lab setup, using the Resource Group link in the left-hand menu. Once you select your resource group, you will see a list of the resources within that group, including your Spark cluster. Select your Spark cluster.

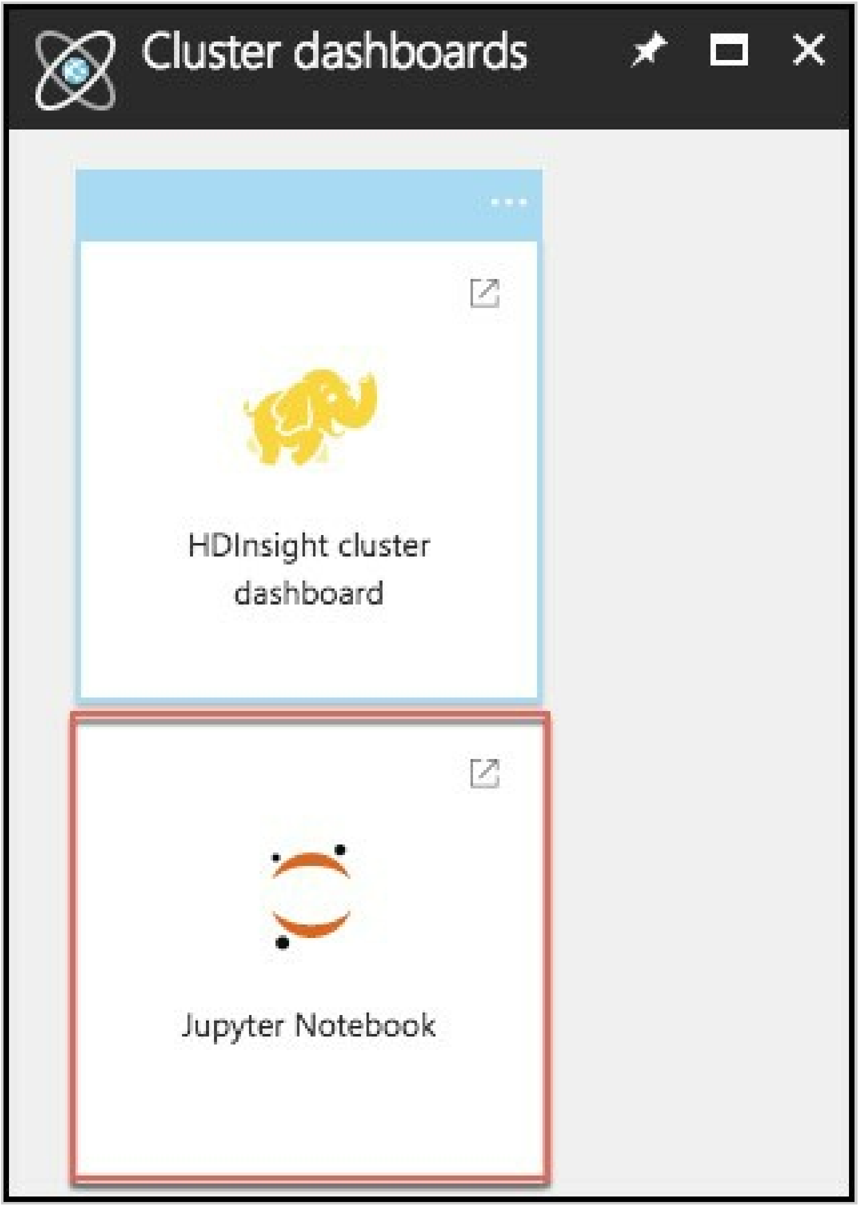


□ kylepublicip	Public IP address	East US 2	···
□ kylespark	HDInsight cluster	East US 2	···
□ kylesparkstorage	Storage account	East US 2	···
□ kylevmstorage	Storage account	East US 2	···

2. In the **Quick links** section, select **Cluster dashboard**.



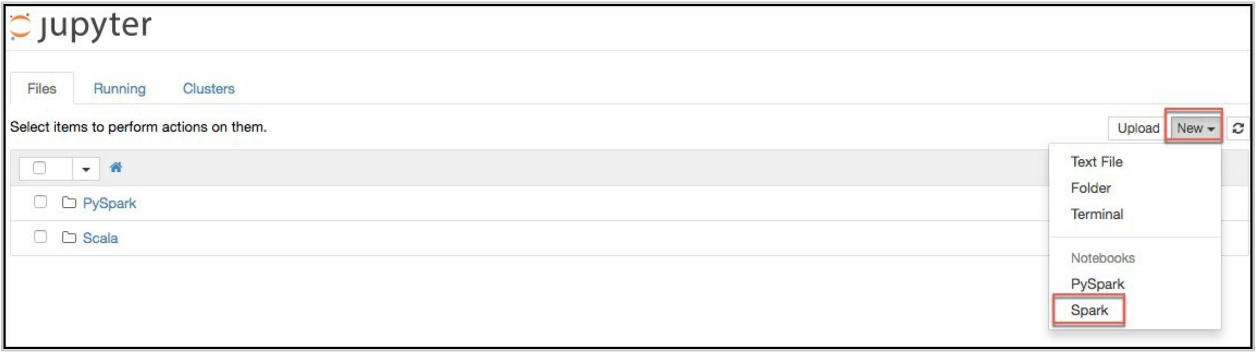
3. From the **Cluster dashboards** blade, select **Jupyter Notebook**.



4. Juptyer Notebook will open in a new browser window. **Log in** with the following credentials:

   - User name: demouser
   - Password: Password.1!!
   - Note: If you get a 403 – Forbidden: Access is denied error, try to open the jupyter URL in a private or incognito browser window. You can also clear the browser cache.

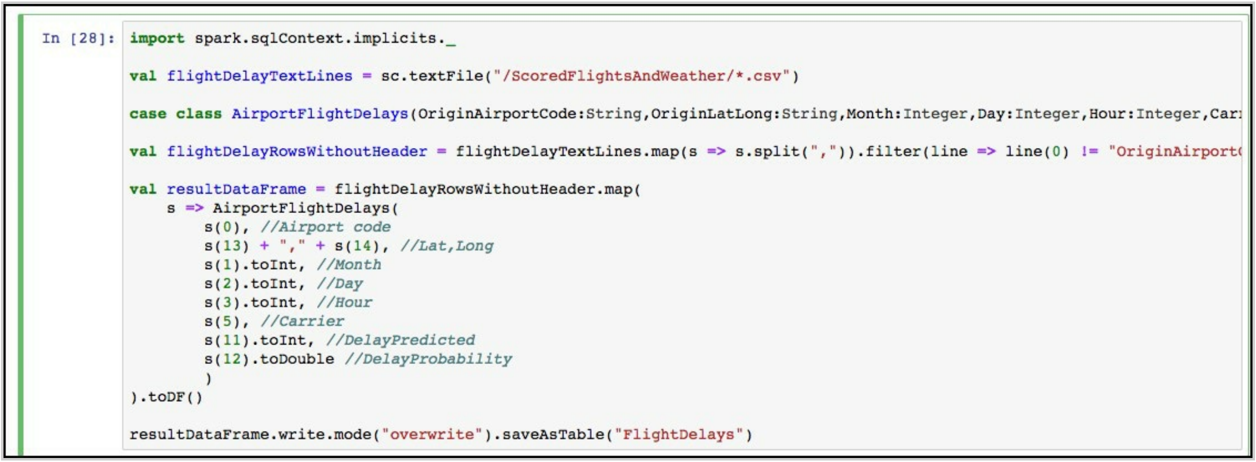
5. On the Jupyter Notebook screen, select **New**, and **Spark**. This will open a Jupyter notebook in a new browser tab.



6. **Copy the text below**, and **paste** it into the **first cell** in the Jupyter notebook. This will read the data from our Scored_FlightsAndWeather.csv file, and output it into a Hive table named “FlightDelays.”

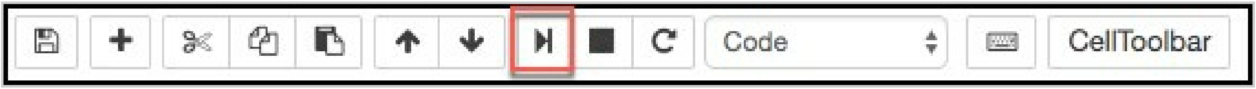
```
import spark.sqlContext.implicits._

val flightDelayTextLines = sc.textFile("/ScoredFlightsAndWeather/*.csv")

case class AirportFlightDelays(OriginAirportCode:String,OriginLatLong:String,Mo
nth:Integer,Day:Integer,Hour:Integer,Carrier:String,DelayPredicted:Integer,Dela
yProbability:Double)

val flightDelayRowsWithoutHeader = flightDelayTextLines.map(s => s.split(",")).
filter(line => line(0) != "OriginAirportCode")

val resultDataFrame = flightDelayRowsWithoutHeader.map(
    s => AirportFlightDelays(
        s(0), //Airport code
        s(13) + "," + s(14), //Lat,Long
        s(1).toInt, //Month
        s(2).toInt, //Day
        s(3).toInt, //Hour
        s(5), //Carrier
        s(11).toInt, //DelayPredicted
        s(12).toDouble //DelayProbability
        )
).toDF()

resultDataFrame.write.mode("overwrite").saveAsTable("FlightDelays")
```

7. The notebook should now look like the image below.

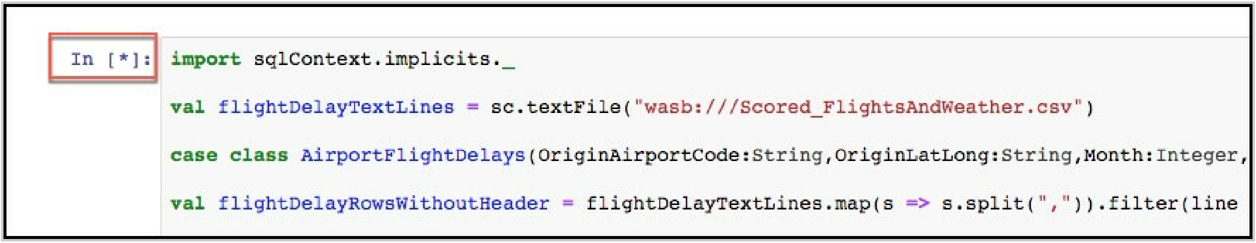


```
In [28]: import spark.sqlContext.implicits._

         val flightDelayTextLines = sc.textFile("/ScoredFlightsAndWeather/*.csv")

         case class AirportFlightDelays(OriginAirportCode:String,OriginLatLong:String,Month:Integer,Day:Integer,Hour:Integer,Car

         val flightDelayRowsWithoutHeader = flightDelayTextLines.map(s => s.split(",")).filter(line => line(0) != "OriginAirport

         val resultDataFrame = flightDelayRowsWithoutHeader.map(
             s => AirportFlightDelays(
                 s(0), //Airport code
                 s(13) + "," + s(14), //Lat,Long
                 s(1).toInt, //Month
                 s(2).toInt, //Day
                 s(3).toInt, //Hour
                 s(5), //Carrier
                 s(11).toInt, //DelayPredicted
                 s(12).toDouble //DelayProbability
                 )
         ).toDF()

         resultDataFrame.write.mode("overwrite").saveAsTable("FlightDelays")
```

8. Select the Run cell button on the toolbar.

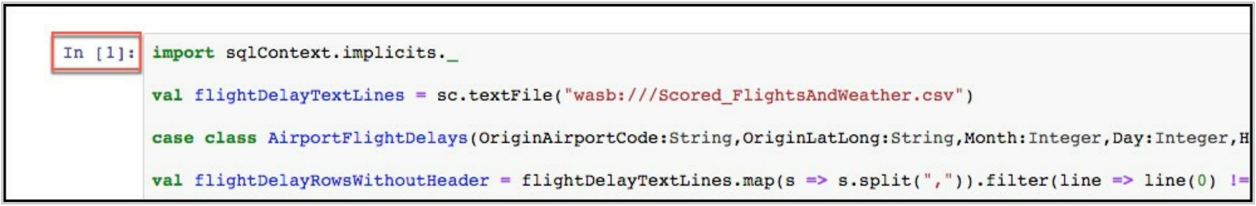


9. You will see in asterisk appear between the brackets in front of the cell.



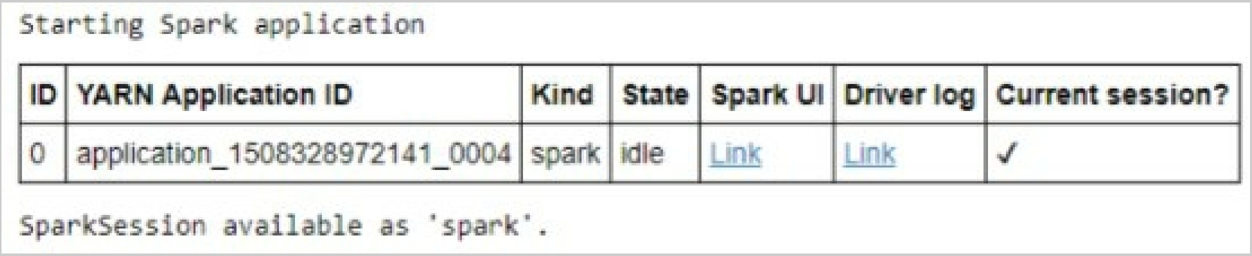
```
In [*]: import sqlContext.implicits._

        val flightDelayTextLines = sc.textFile("wasb:///Scored_FlightsAndWeather.csv")

        case class AirportFlightDelays(OriginAirportCode:String,OriginLatLong:String,Month:Integer,

        val flightDelayRowsWithoutHeader = flightDelayTextLines.map(s => s.split(",")).filter(line
```

10. This will change to a number once the command is complete.



```
In [1]: import sqlContext.implicits._

        val flightDelayTextLines = sc.textFile("wasb:///Scored_FlightsAndWeather.csv")

        case class AirportFlightDelays(OriginAirportCode:String,OriginLatLong:String,Month:Integer,Day:Integer,H

        val flightDelayRowsWithoutHeader = flightDelayTextLines.map(s => s.split(",")).filter(line => line(0) !=
```

11. Below the cell, you will see the output from executing the command.



Starting Spark application

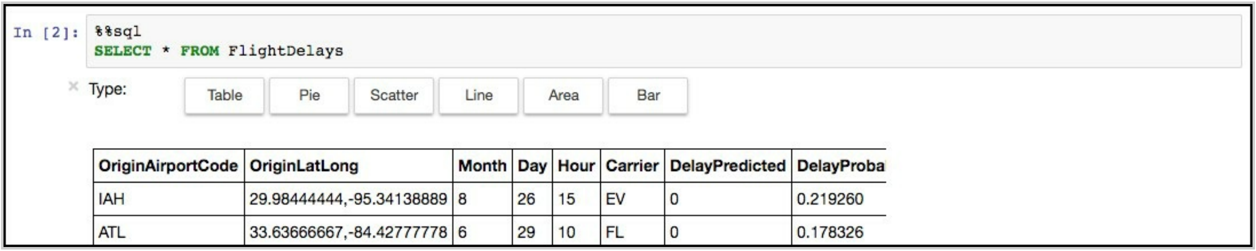
ID	YARN Application ID	Kind	State	Spark UI	Driver log	Current session?
0	application_1508328972141_0004	spark	idle	Link	Link	✓

SparkSession available as 'spark'.

12. Now, we can query the hive table which was created by the previous command. **Paste** the text below into the **empty cell** at the bottom on the notebook, and **select** the **Run cell** button for that cell.

```
%%sql

SELECT * FROM FlightDelays
```

13. Once completed you will see the results displayed as a table.



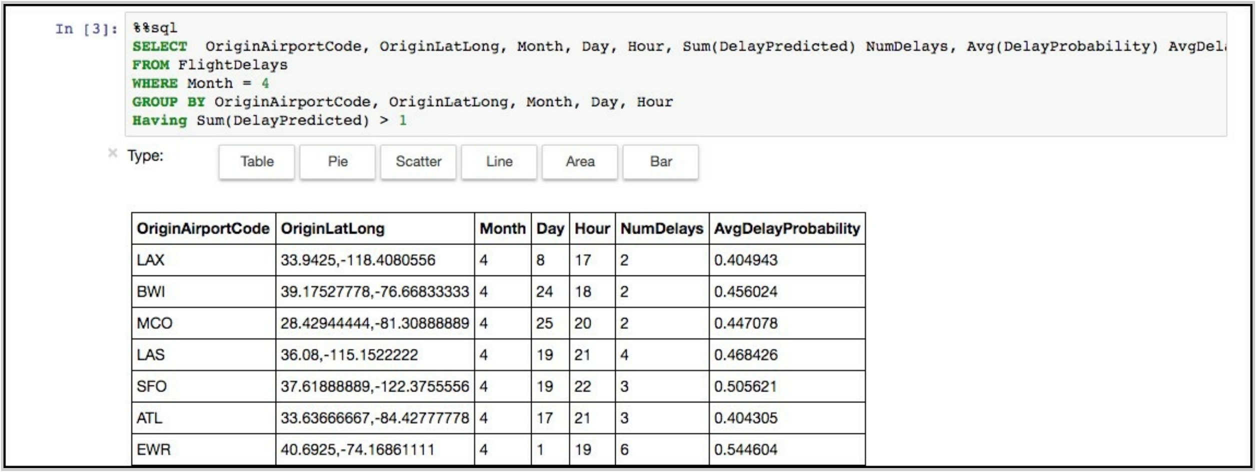
```
In [2]: %%sql
SELECT * FROM FlightDelays
```

OriginAirportCode	OriginLatLong	Month	Day	Hour	Carrier	DelayPredicted	DelayProba
IAH	29.98444444,-95.34138889	8	26	15	EV	0	0.219260
ATL	33.63666667,-84.42777778	6	29	10	FL	0	0.178326

14. Next, you will create a table that summarizes the flight delays data. Instead of containing one row per flight, this new summary table will contain one row per origin airport at a given hour, along with a count of the quantity of anticipated delays. In a **new cell** below the results of our previous cell, **paste** the following text, and **select** the **Run cell** button from the toolbar.

```
%%sql

SELECT  OriginAirportCode, OriginLatLong, Month, Day, Hour, Sum(DelayPredicted)
 NumDelays, Avg(DelayProbability) AvgDelayProbability
FROM FlightDelays
WHERE Month = 4
GROUP BY OriginAirportCode, OriginLatLong, Month, Day, Hour
Having Sum(DelayPredicted) > 1
```

15. Execution of this cell should return a results table like the following.



```
In [3]: %%sql
SELECT  OriginAirportCode, OriginLatLong, Month, Day, Hour, Sum(DelayPredicted) NumDelays, Avg(DelayProbability) AvgDel
FROM FlightDelays
WHERE Month = 4
GROUP BY OriginAirportCode, OriginLatLong, Month, Day, Hour
Having Sum(DelayPredicted) > 1
```

OriginAirportCode	OriginLatLong	Month	Day	Hour	NumDelays	AvgDelayProbability
LAX	33.9425,-118.4080556	4	8	17	2	0.404943
BWI	39.17527778,-76.66833333	4	24	18	2	0.456024
MCO	28.42944444,-81.30888889	4	25	20	2	0.447078
LAS	36.08,-115.1522222	4	19	21	4	0.468426
SFO	37.61888889,-122.3755556	4	19	22	3	0.505621
ATL	33.63666667,-84.42777778	4	17	21	3	0.404305
EWR	40.6925,-74.16861111	4	1	19	6	0.544604

16. Since the summary data looks good, the final step is to save this summary calculation as a table, which we can later query using Power BI (in the next exercise).

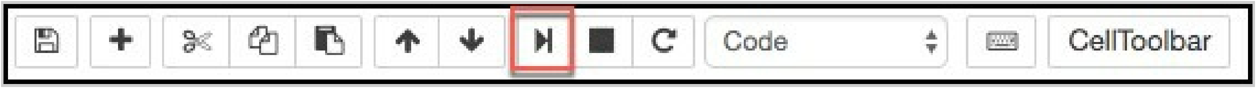
17. To accomplish this, **paste** the text below into a **new cell**, and **select** the **Run cell** button from the toolbar.

```
val summary = spark.sqlContext.sql("SELECT  OriginAirportCode, OriginLatLong, M
onth, Day, Hour, Sum(DelayPredicted) NumDelays, Avg(DelayProbability) AvgDelayP
robability FROM FlightDelays WHERE Month = 4 GROUP BY OriginAirportCode, Origin
LatLong, Month, Day, Hour Having Sum(DelayPredicted) > 1")
summary.write.mode("overwrite").saveAsTable("FlightDelaysSummary")
```

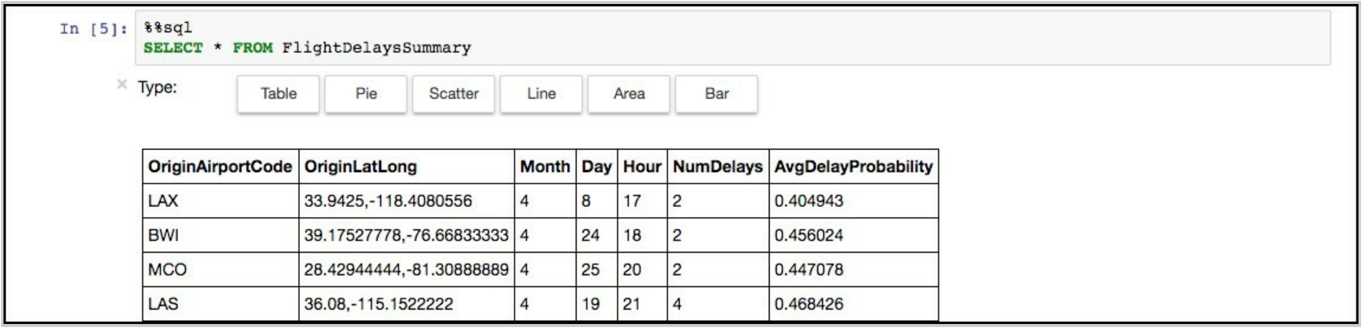
18. To verify the table was successfully created, go to another **new cell**, and **enter the following query**.

```
%%sql

SELECT * FROM FlightDelaysSummary
```

19. Select the Run cell button on the toolbar.



20. You should see a results table similar to the following.



```
In [5]: %%sql
SELECT * FROM FlightDelaysSummary
```

OriginAirportCode	OriginLatLong	Month	Day	Hour	NumDelays	AvgDelayProbability
LAX	33.9425,-118.4080556	4	8	17	2	0.404943
BWI	39.17527778,-76.66833333	4	24	18	2	0.456024
MCO	28.42944444,-81.30888889	4	25	20	2	0.447078
LAS	36.08,-115.1522222	4	19	21	4	0.468426

21. You can also select Pie, Scatter, Line, Area, and Bar chart visualizations of the dataset.

Visualizing in PowerBI Desktop

Exercise 6: Visualizing in Power BI Desktop

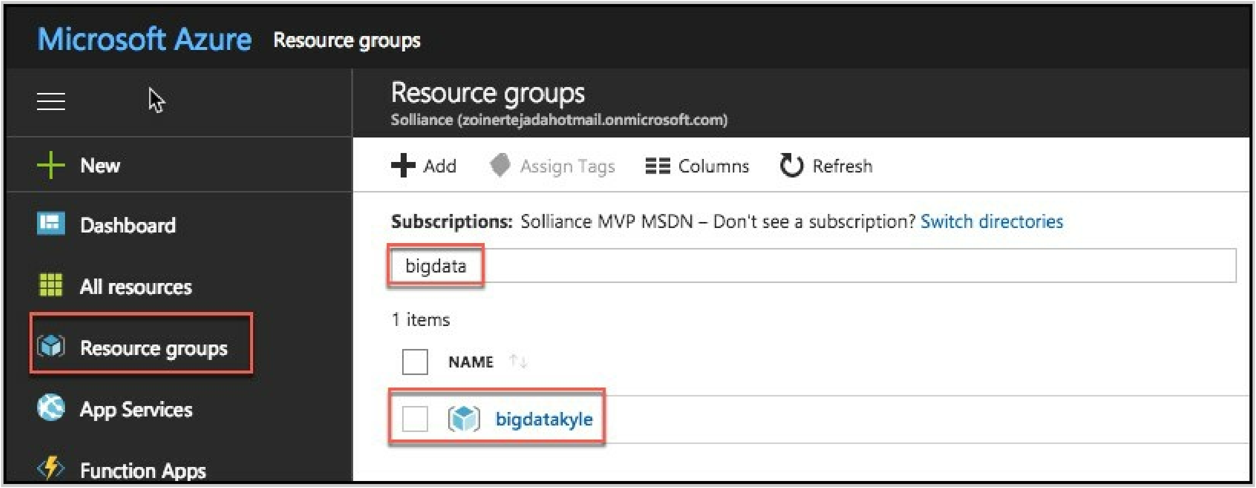
Duration: 20 mins

Synopsis: In this exercise, you will create a Power BI Report to visualize the data in HDInsight Spark.

Task 1: Connect to the Lab VM

1. NOTE: If you are already, connected to your Lab VM, skip to Task 2.

2. From the left side menu in the Azure portal, click on **Resource groups**, then enter your resource group name into the filter box, and select it from the list.

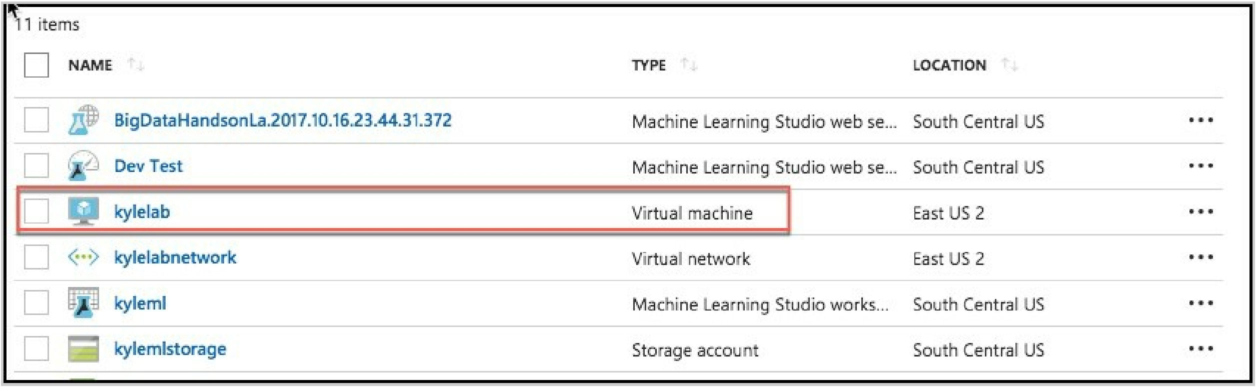


Subscriptions: Solliance MVP MSDN – Don't see a subscription? Switch directories

bigdata

1 items

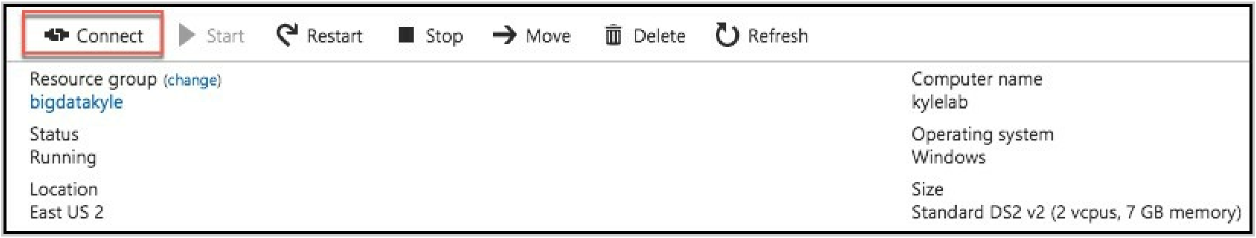
3. Next, select your lab virtual machine from the list.



11 items

NAME	TYPE	LOCATION
BigDataHandsonLa.2017.10.16.23.44.31.372	Machine Learning Studio web se...	South Central US
Dev Test	Machine Learning Studio web se...	South Central US
kylelab	Virtual machine	East US 2
kylelabnetwork	Virtual network	East US 2
kyleml	Machine Learning Studio works...	South Central US
kylemlstorage	Storage account	South Central US

4. On your Lab VM blade, select **Connect** from the top menu.



Resource group (change)
bigdatakyle

Status
Running

Location
East US 2

Computer name
kylelab

Operating system
Windows

Size
Standard DS2 v2 (2 vcpus, 7 GB memory)

5. **Download** and **open** the RDP file.

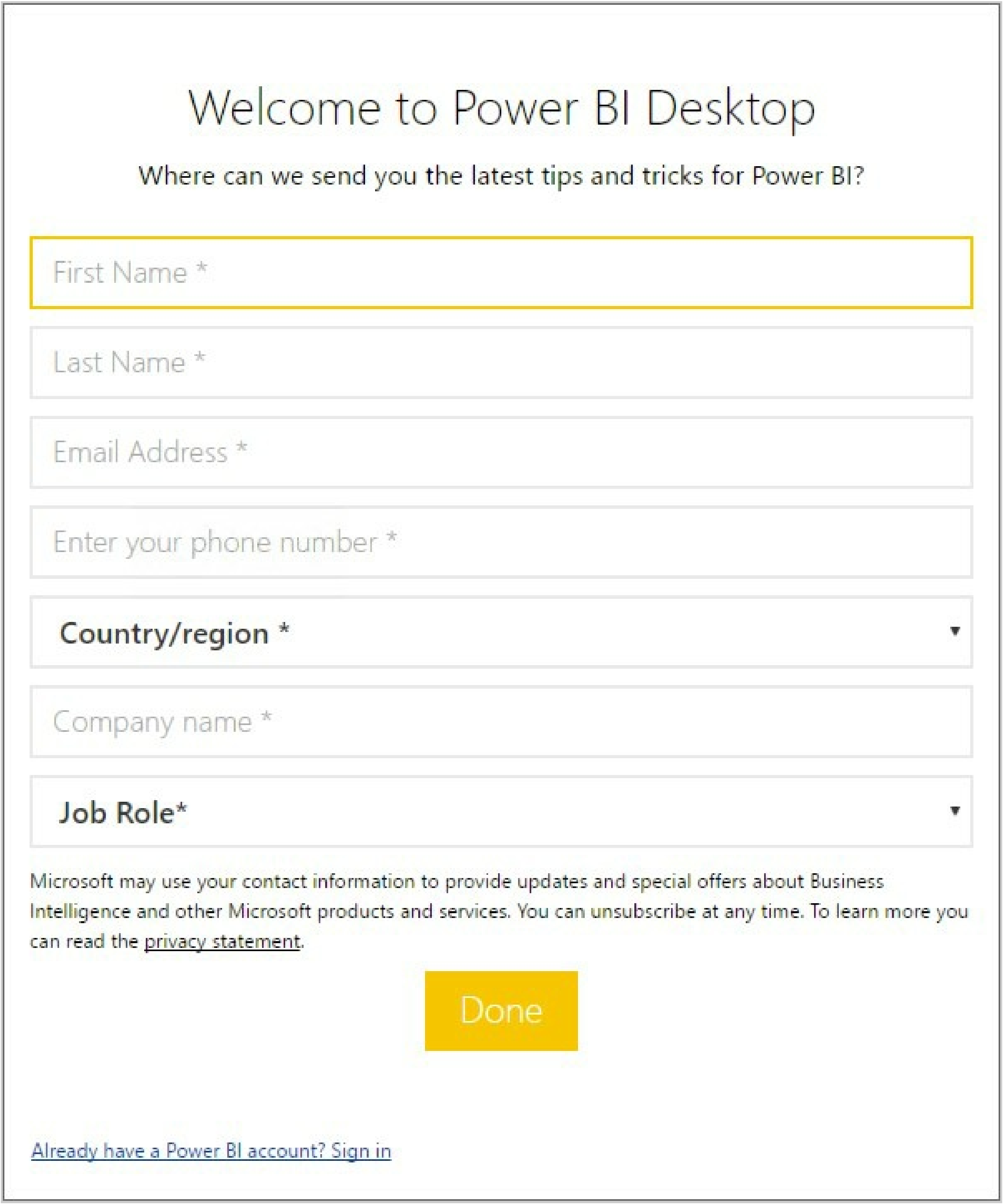
6. Select **Connect**, and enter the following credentials:

   - User name: demouser
   - Password: Password.1!!

Task 2: Connect to HDInsight Spark using Power BI Desktop

1. On your Lab VM, launch Power BI Desktop by **double-clicking on the desktop shortcut** you created in the pre-lab setup.

2. When Power BI Desktop opens, 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 *

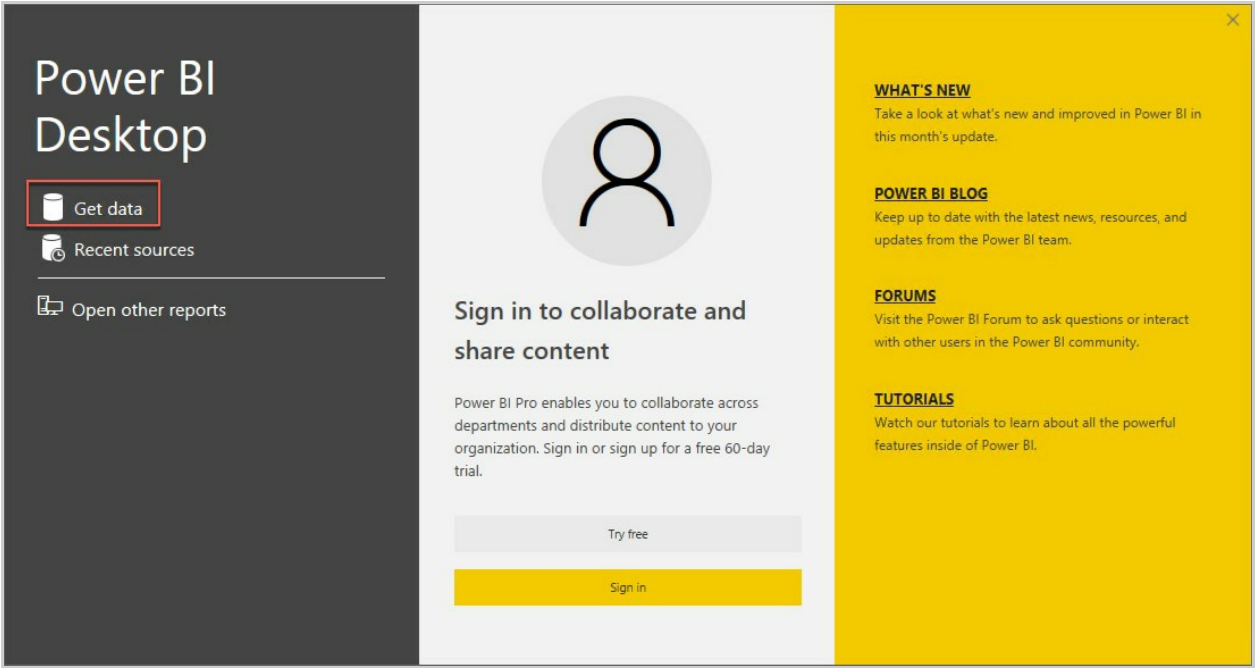
Country/region * ▾

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.

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



Sign in to collaborate and share content

Power BI Pro enables you to collaborate across departments and distribute content to your organization. Sign in or sign up for a free 60-day trial.

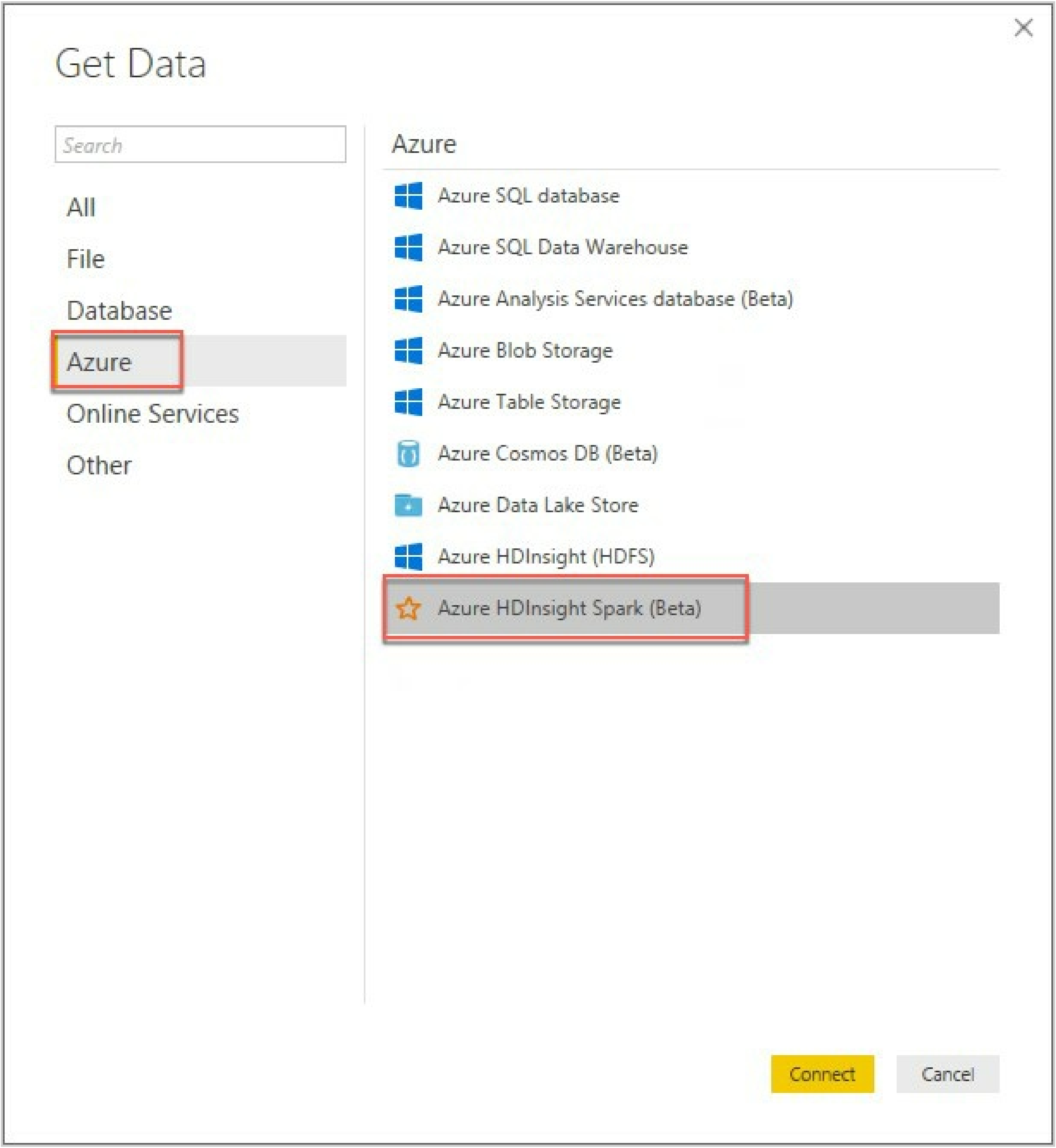
WHAT'S NEW
Take a look at what's new and improved in Power BI in this month's update.

POWER BI BLOG
Keep up to date with the latest news, resources, and updates from the Power BI team.

FORUMS
Visit the Power BI Forum to ask questions or interact with other users in the Power BI community.

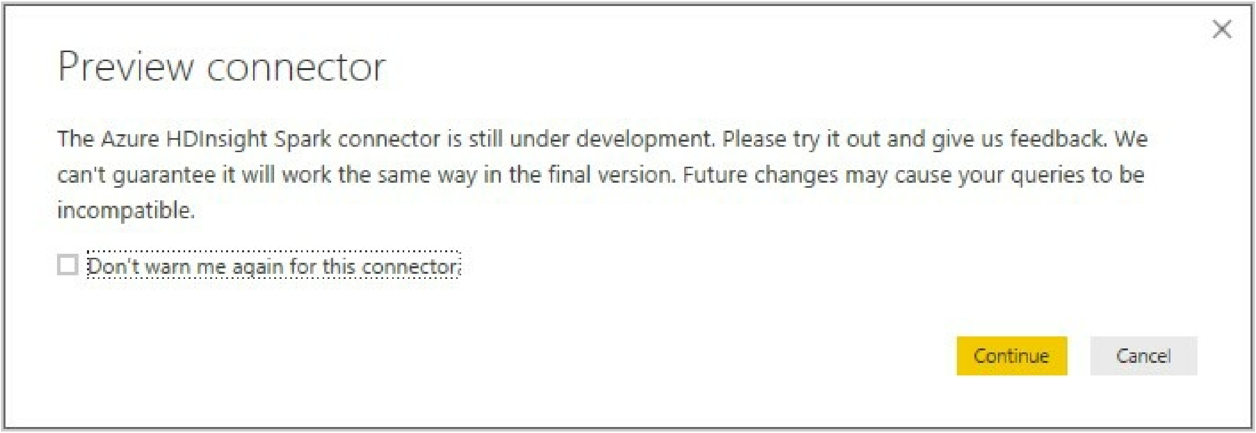
TUTORIALS
Watch our tutorials to learn about all the powerful features inside of Power BI.

4. Select **Azure** from the left, and select **Azure HDInsight Spark (Beta)** from the list of available data sources.



5. Select **Connect**.

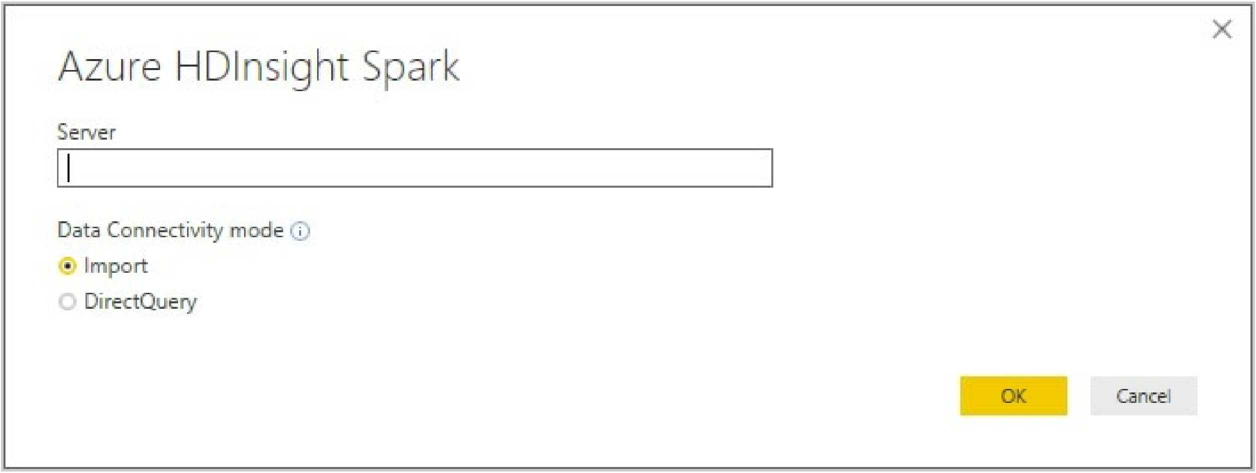
6. You will receive a prompt warning you that the Spark connector is still in preview. Select **Continue**.


Preview connector

The Azure HDInsight Spark connector is still under development. Please try it out and give us feedback. We can't guarantee it will work the same way in the final version. Future changes may cause your queries to be incompatible.

☐ Don't warn me again for this connector

7. On the next screen, you will be prompted for your HDInsight Spark cluster URL.


Azure HDInsight Spark

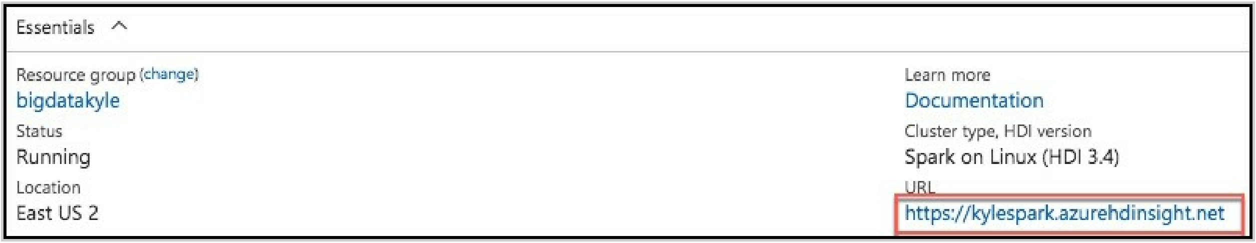
Server

Data Connectivity mode

● Import

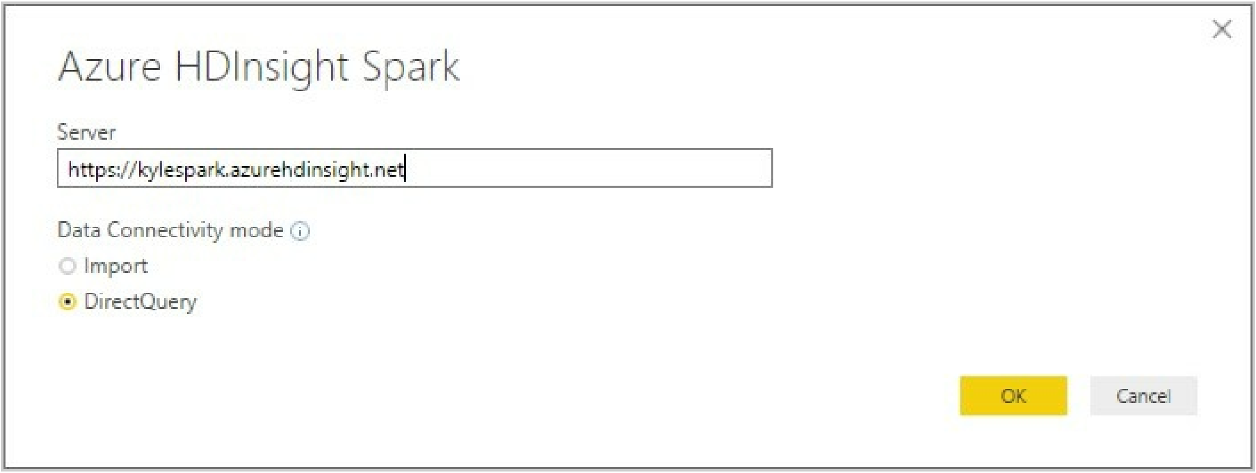
○ DirectQuery

8. To find your Spark cluster URL, go into the **Azure portal**, and navigate to your **Spark cluster**, as you did in **Exercise 5, Task 1**. Once on the cluster blade, look for the URL under the **Essentials** section


Essentials

Resource group (change) bigdatakyle	Learn more Documentation
Status Running	Cluster type, HDI version Spark on Linux (HDI 3.4)
Location East US 2	URL https://kylespark.azurehdinsight.net

9. Copy the URL, and **paste it into the Server box** on the Power BI Azure HDInsight Spark dialog.



Azure HDInsight Spark

Server

https://kylespark.azurehdinsight.net

Data Connectivity mode

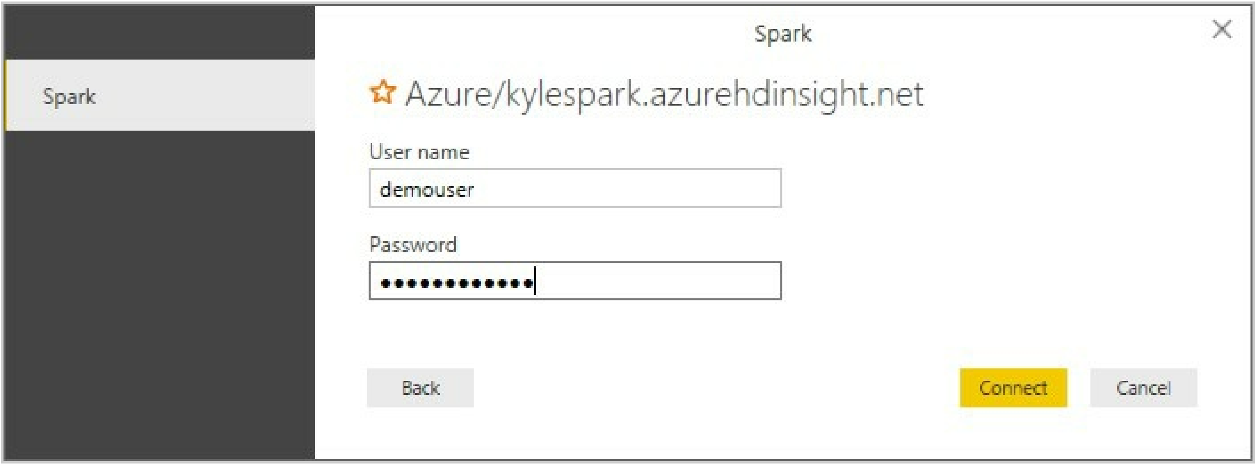
○ Import

● DirectQuery

10. Select **DirectQuery** for the Data Connectivity mode, and select **OK**.

11. Enter your credentials on the next screen as follows.

    - User name: demouser
    - Password: Password.1!!



Spark

Spark

☆ Azure/kylespark.azurehdinsight.net

User name

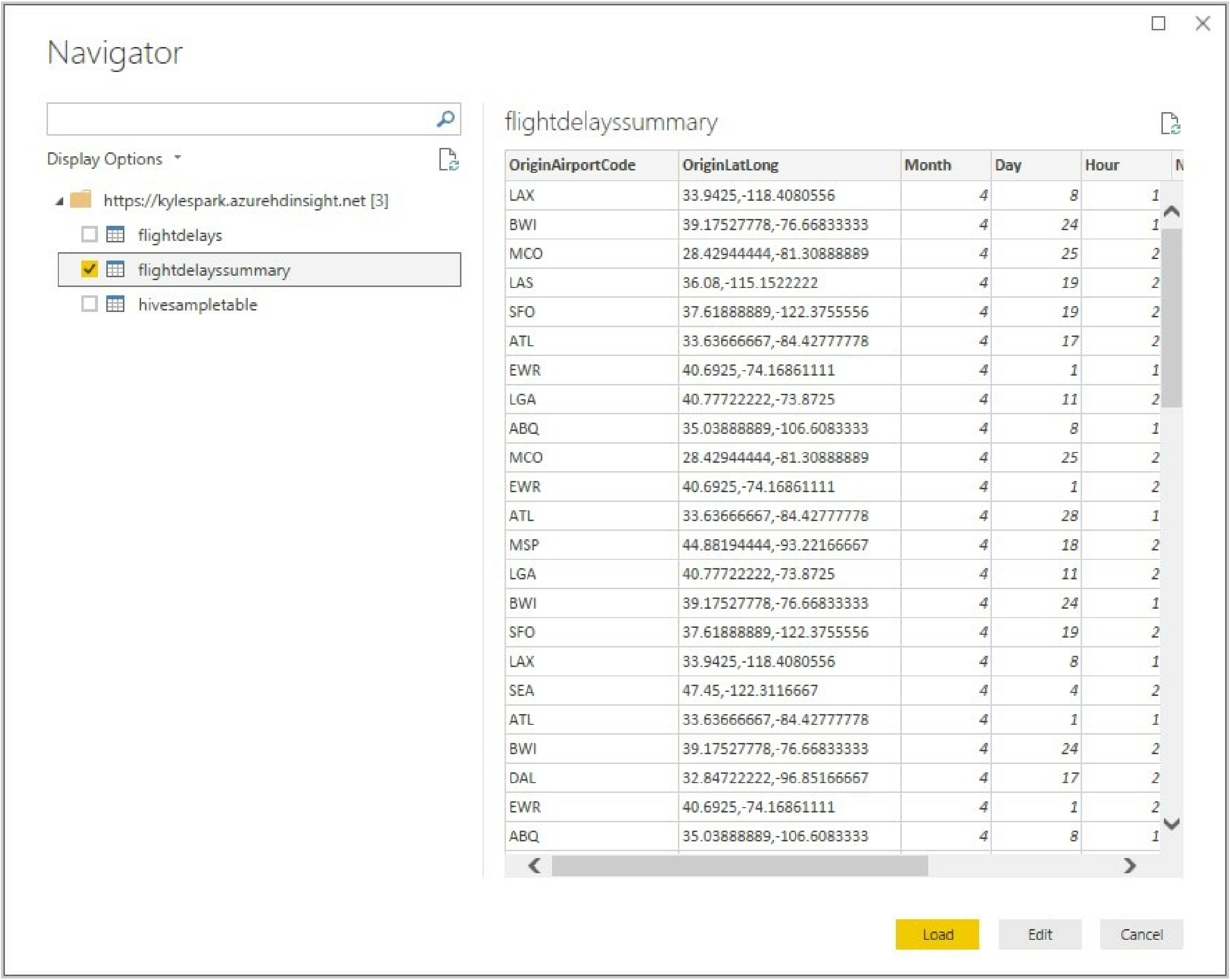
demouser

Password

•••••••••••••

12. Select **Connect**.

13. In the Navigator dialog, **check the box** next to **flightdelayssummary**, and select **Load**.



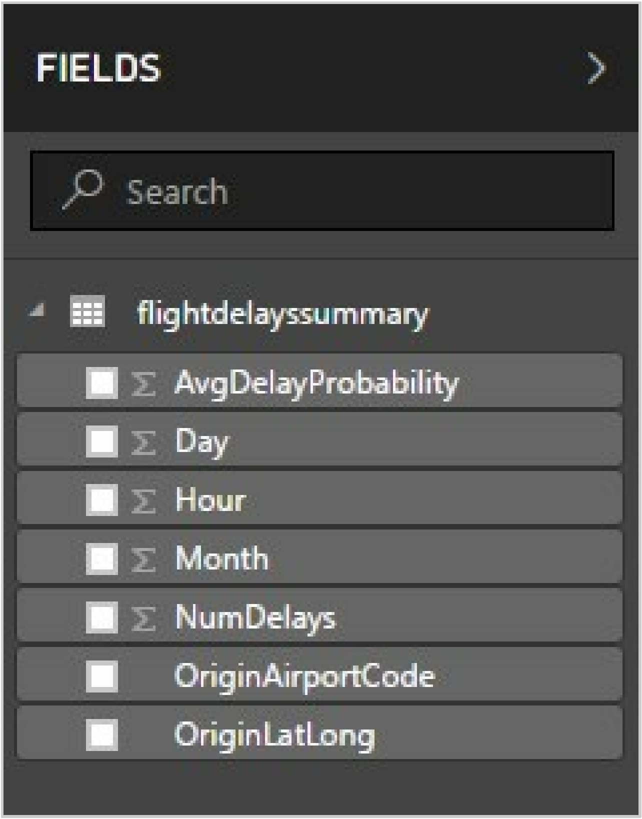
flightdelayssummary

OriginAirportCode	OriginLatLong	Month	Day	Hour
LAX	33.9425,-118.4080556	4	8	1
BWI	39.17527778,-76.66833333	4	24	1
MCO	28.42944444,-81.30888889	4	25	2
LAS	36.08,-115.1522222	4	19	2
SFO	37.61888889,-122.3755556	4	19	2
ATL	33.63666667,-84.42777778	4	17	2
EWR	40.6925,-74.16861111	4	1	1
LGA	40.77722222,-73.8725	4	11	2
ABQ	35.03888889,-106.6083333	4	8	1
MCO	28.42944444,-81.30888889	4	25	2
EWR	40.6925,-74.16861111	4	1	2
ATL	33.63666667,-84.42777778	4	28	1
MSP	44.88194444,-93.22166667	4	18	2
LGA	40.77722222,-73.8725	4	11	2
BWI	39.17527778,-76.66833333	4	24	1
SFO	37.61888889,-122.3755556	4	19	2
LAX	33.9425,-118.4080556	4	8	1
SEA	47.45,-122.3116667	4	4	2
ATL	33.63666667,-84.42777778	4	1	1
BWI	39.17527778,-76.66833333	4	24	2
DAL	32.84722222,-96.85166667	4	17	2
EWR	40.6925,-74.16861111	4	1	2
ABQ	35.03888889,-106.6083333	4	8	1

14. It will take several minutes for the data to load into the Power BI Desktop client.

Task 3: Create Power BI report

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



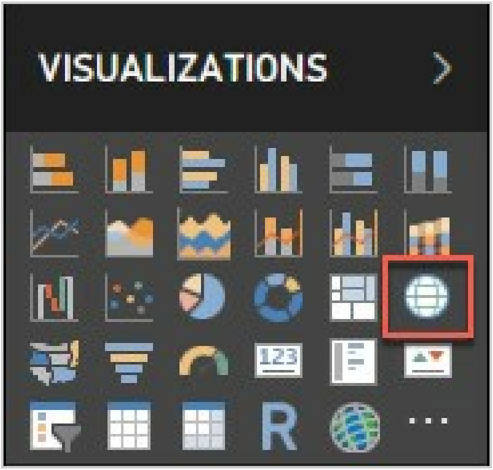
FIELDS

Search

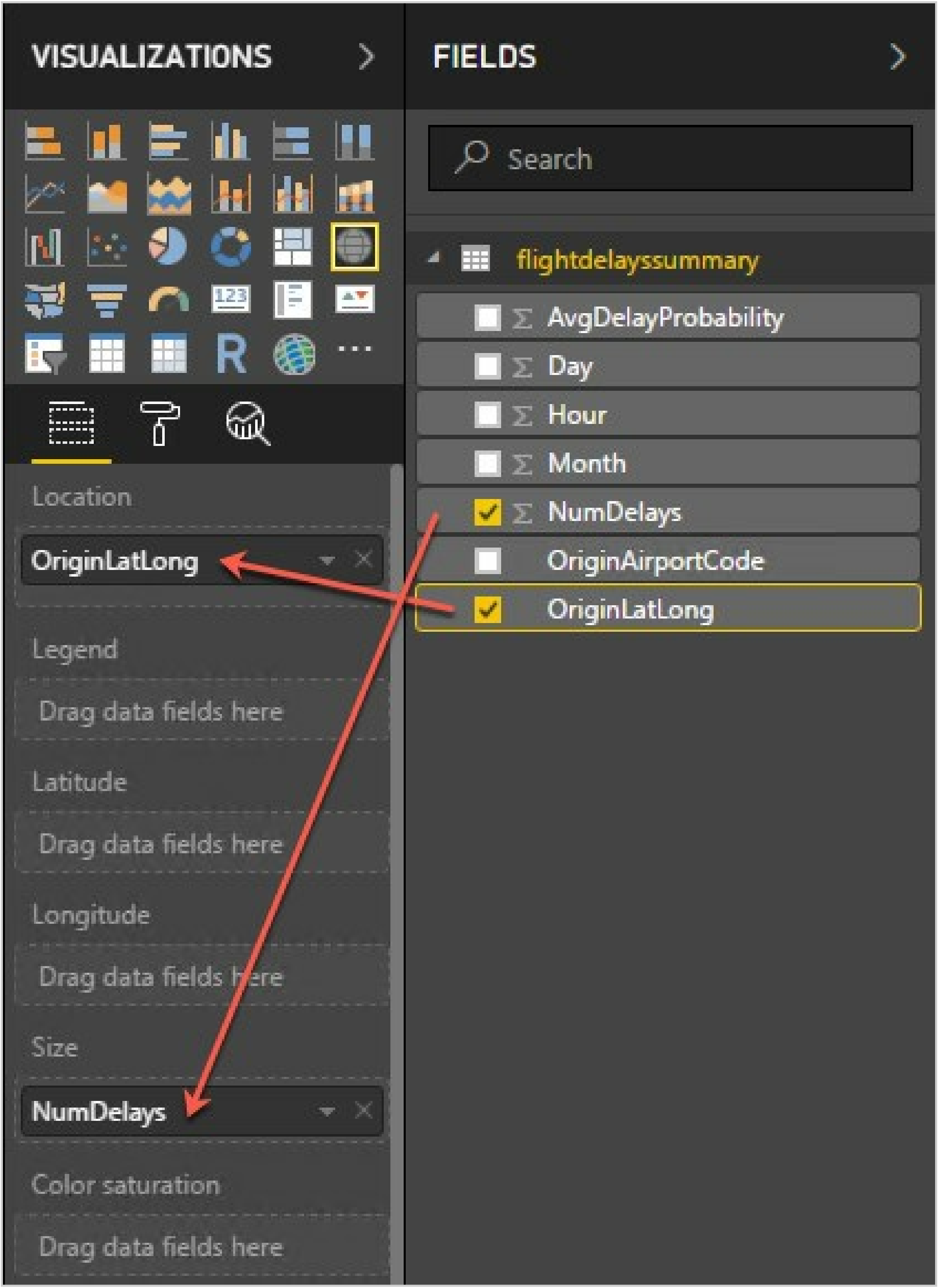
flightdelayssummary

- □ Σ AvgDelayProbability
- □ Σ Day
- □ Σ Hour
- □ Σ Month
- □ Σ NumDelays
- □ OriginAirportCode
- □ OriginLatLong

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



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

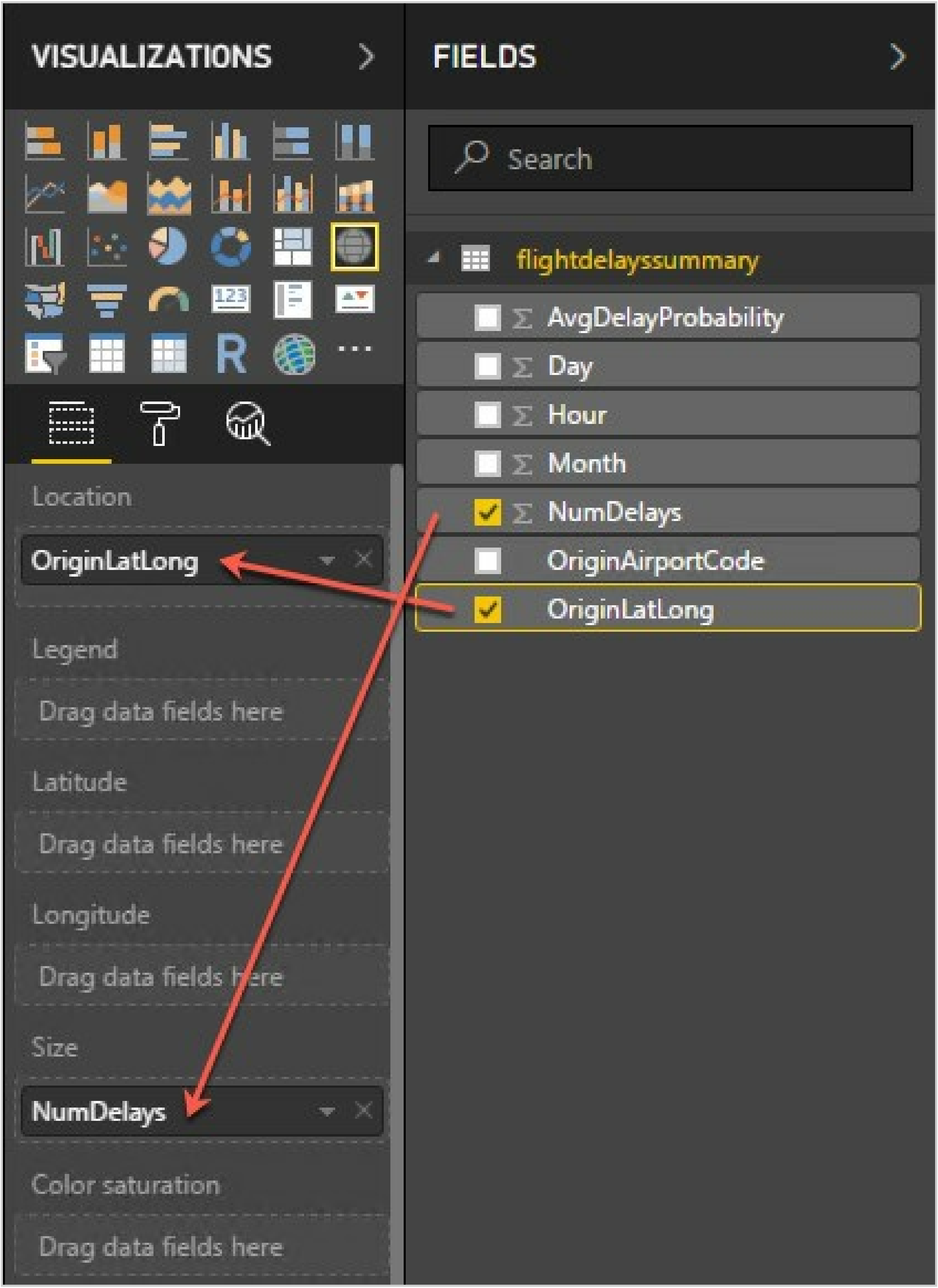


Location
OriginLatLong

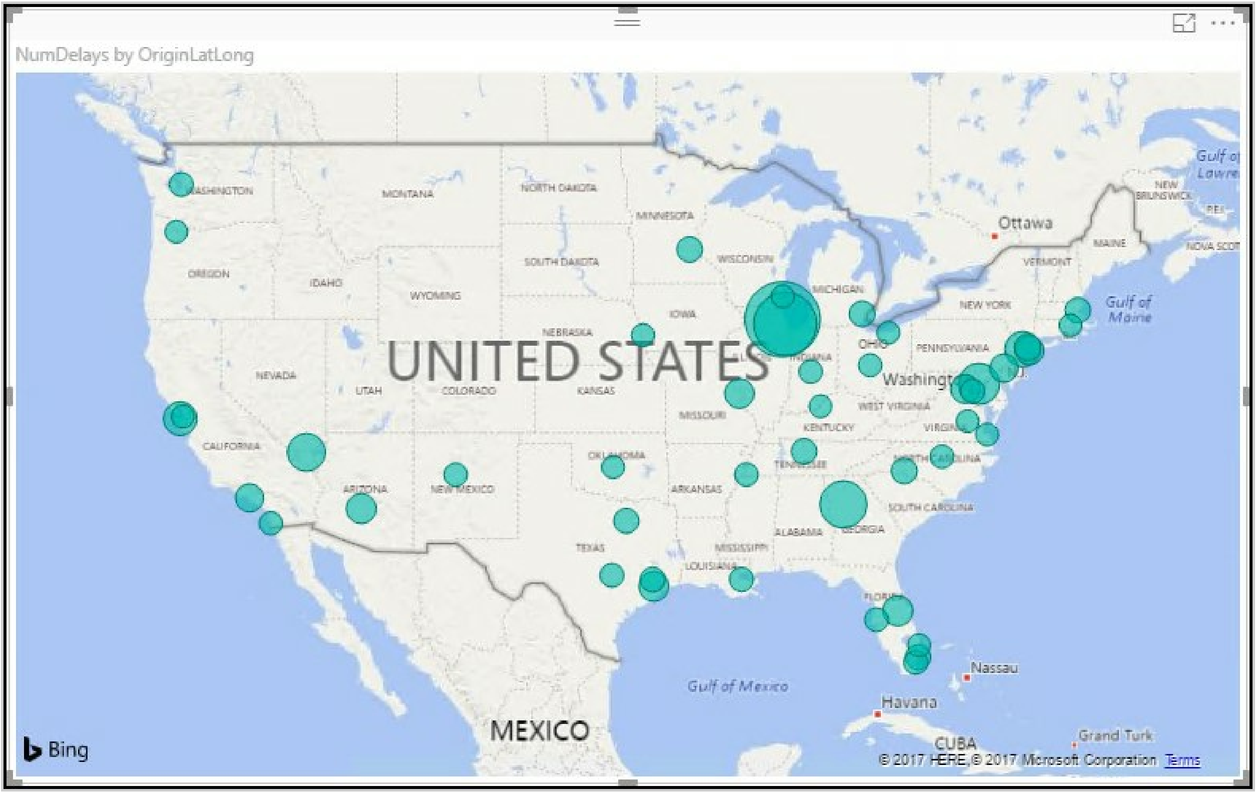
Size
NumDelays

□ Σ AvgDelayProbability
□ Σ Day
□ Σ Hour
□ Σ Month
☑ Σ NumDelays
□ OriginAirportCode
☑ OriginLatLong

4. Next, drag the **NumDelays** field to the **Size** field under Visualizations.

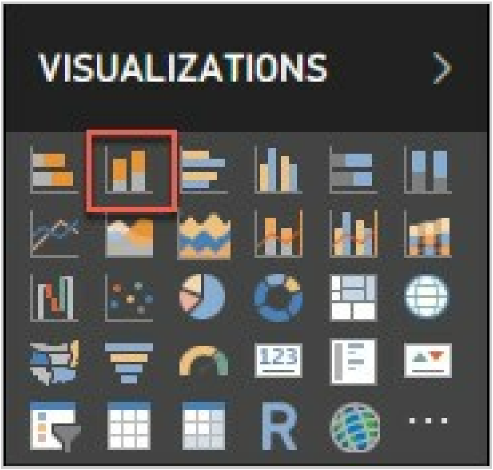


5. You should now see a map that looks similar to the following (resize and zoom on your map if necessary):



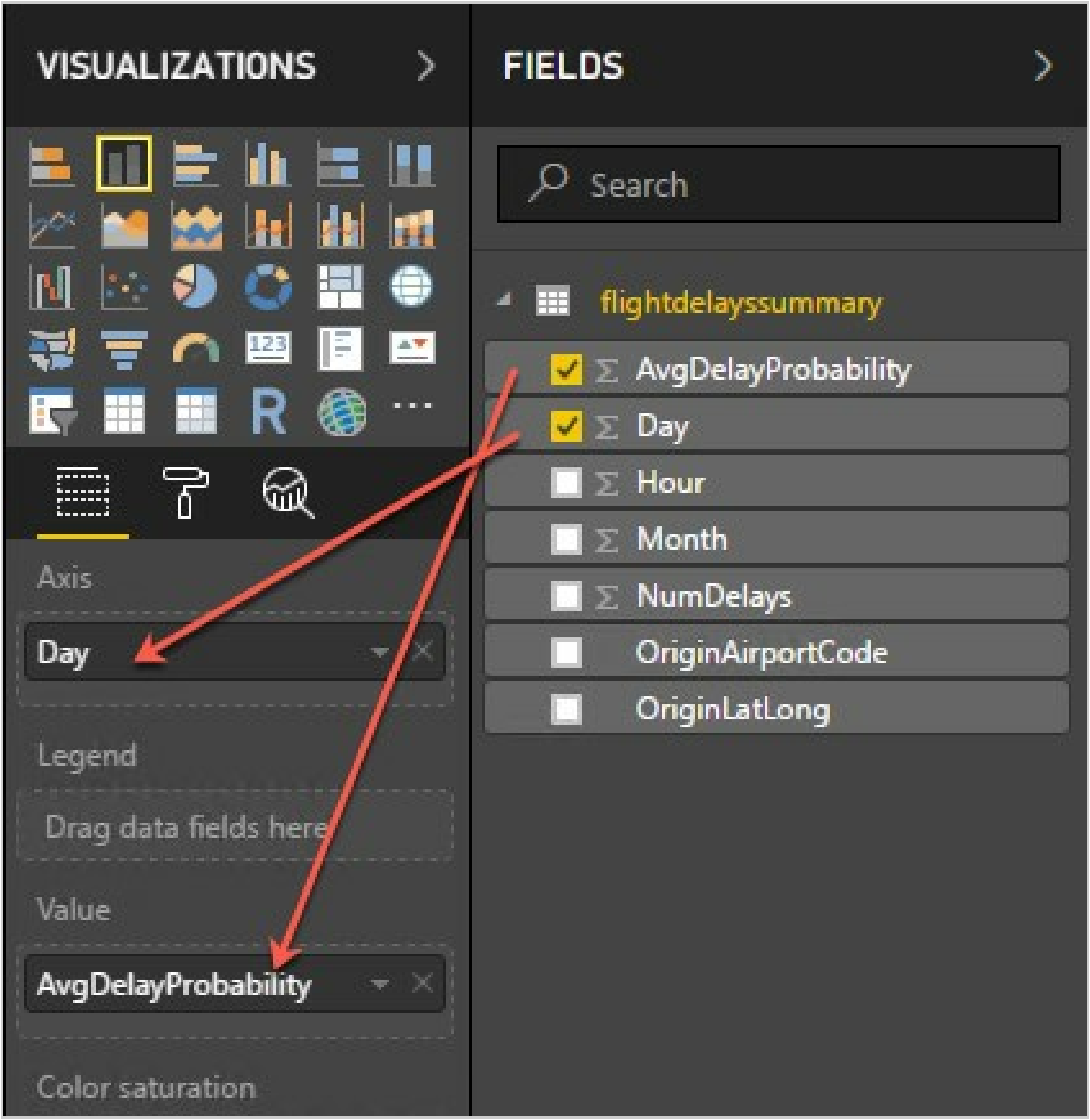
6. **Unselect** the Map visualization by clicking on the white space next to the map in the report area.

7. From the Visualizations area, select the **Stacked Column Chart** icon to add a bar chart visual to the report's design surface.



8. With the Stacked Column Chart still selected, drag the **Day** field and drop it into the **Axis** field located under Visualizations.

9. Next, drag the **AvgDelayProbability** field over, and drop it into the **Value** field.



- ☑ Σ AvgDelayProbability
- ☑ Σ Day
- □ Σ Hour
- □ Σ Month
- □ Σ NumDelays
- □ OriginAirportCode
- □ OriginLatLong

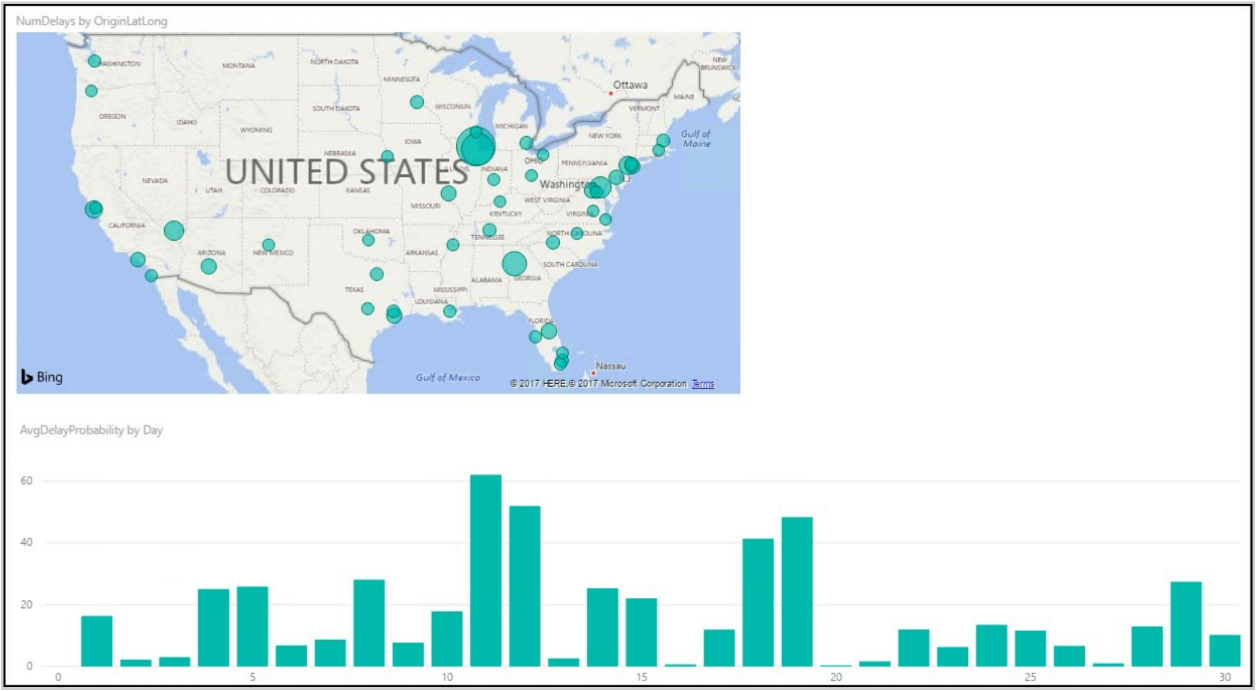
Axis: Day

Legend: Drag data fields here

Value: AvgDelayProbability

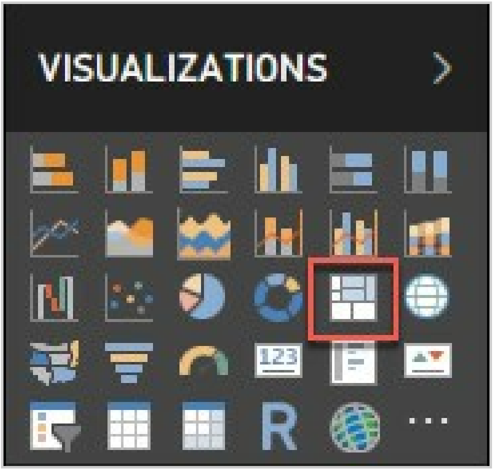
Color saturation

10. 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.


NumDelays by OriginLatLong
UNITED STATES
Washington
Ottawa
Gulf of Maine
Gulf of Mexico
Nassau
Bing
© 2017 HERE, © 2017 Microsoft Corporation Terms
AvgDelayProbability by Day
60
40
20
0
0
5
10
15
20
25
30

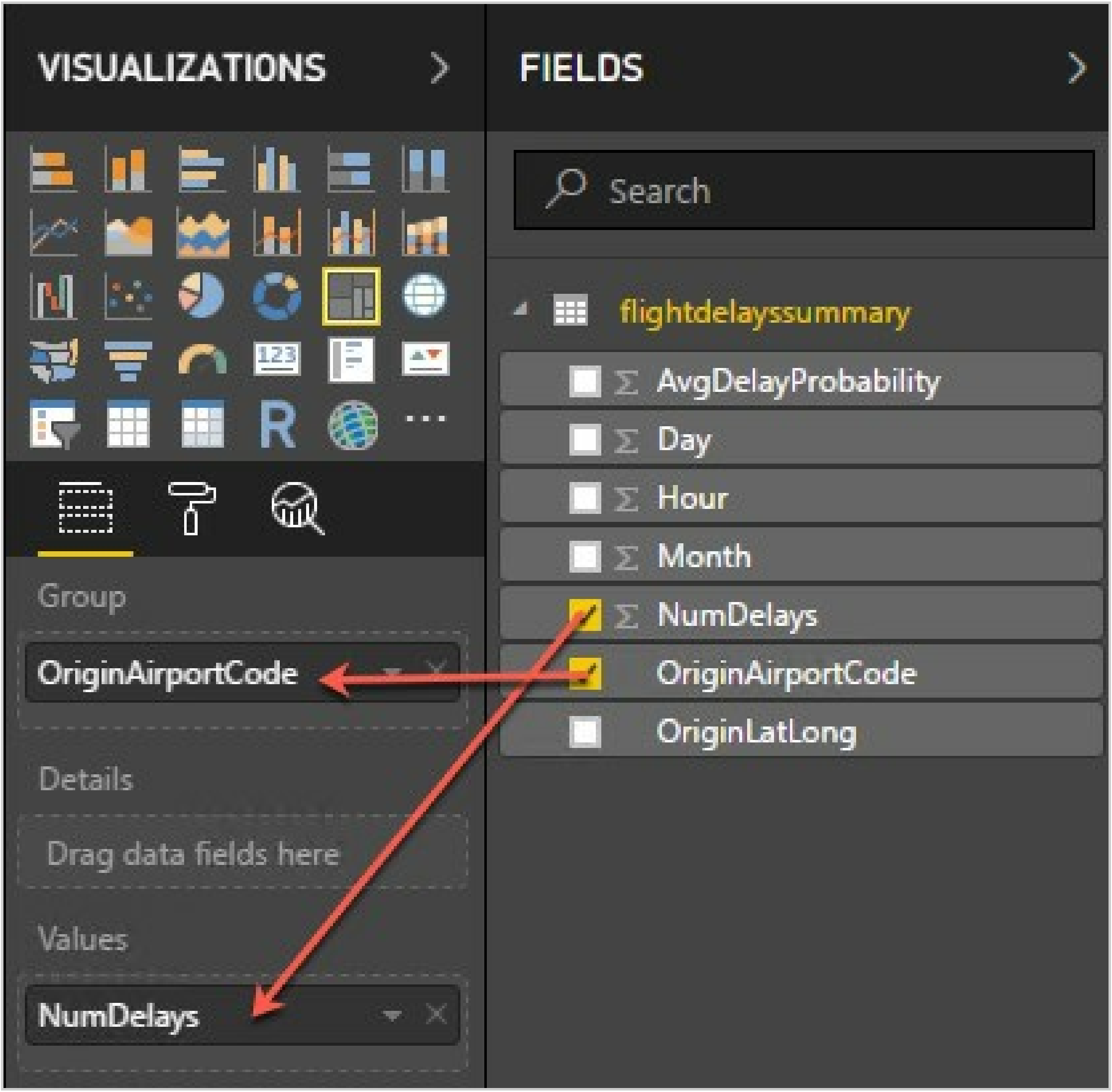
11. **Unselect** the **Stacked Column Chart** visual by clicking on the white space next to the map on the design surface.

12. From the Visualizations area, select the **Treemap** icon to add this visualization to the report.



13. With the Treemap visualization selected, drag the **OriginAirportCode** field into the **Group** field under Visualizations.

14. Next, drag the **NumDelays** field over, and drop it into the **Values** field.



VISUALIZATIONS

FIELDS

Search

flightdelayssummary

- □ Σ AvgDelayProbability
- □ Σ Day
- □ Σ Hour
- □ Σ Month
- ☑ Σ NumDelays
- ☑ OriginAirportCode
- □ OriginLatLong

Group

OriginAirportCode

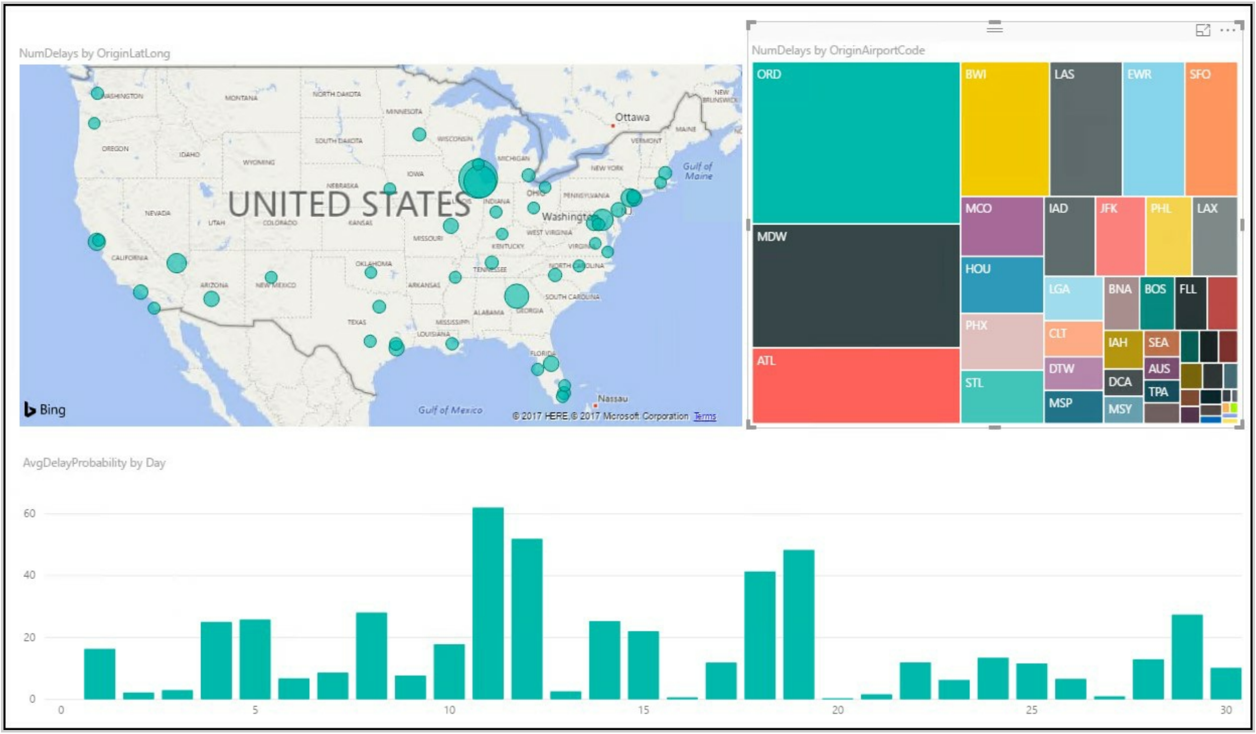
Details

Drag data fields here

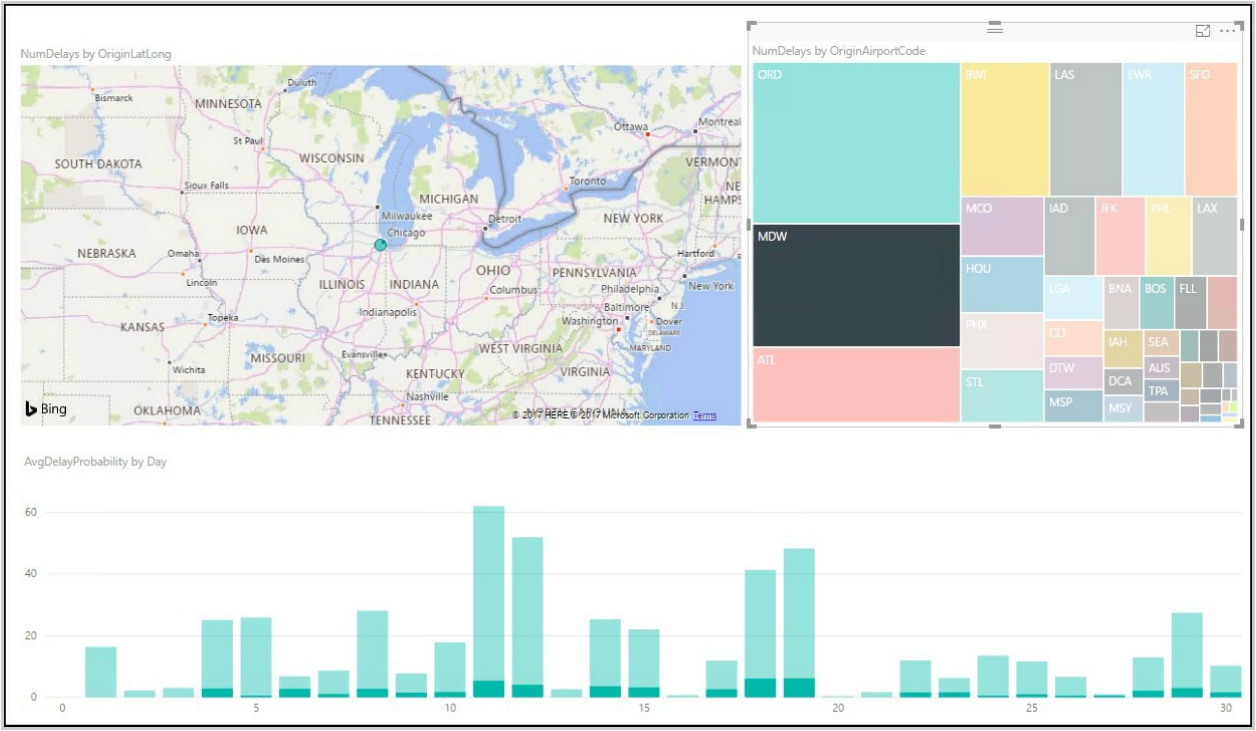
Values

NumDelays

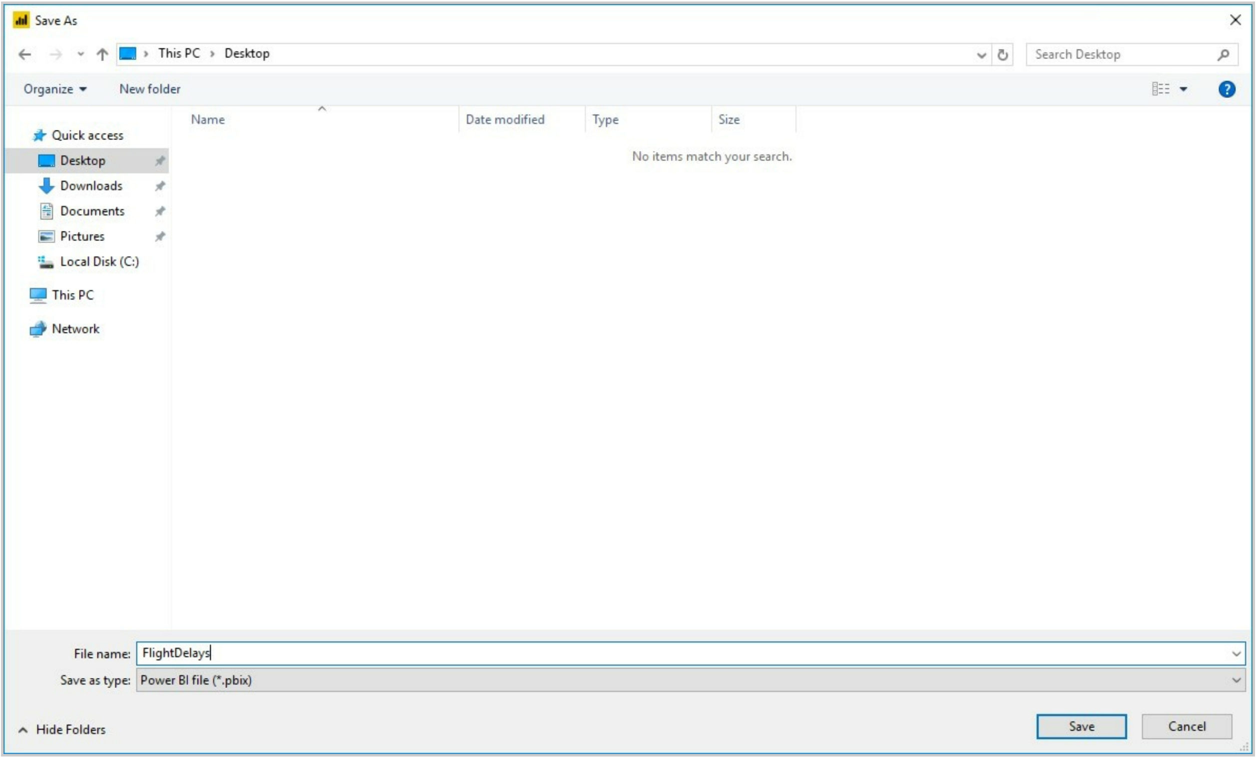
15. Grab the corner of the Treemap visual on the report design surface, and expand it to fill the area between the map and the right edge of the design surface. The report should now look similar to the following.



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



17. You can save the report, by clicking **Save** from the **File** menu, and entering a name and location for the file.



No items match your search.

File name: FlightDelays

Save as type: Power BI file (*.pbix)

Deploy Intelligent Web App

Exercise 7: Deploy Intelligent Web App

Duration: 20 mins

Synopsis: In this exercise, you will deploy an intelligent web application to Azure from GitHub. This application leverages the operationalized machine learning model that was deployed in Exercise 1 to bring action-oriented insight to an already existing business process.

Task 1: Deploy web app from GitHub

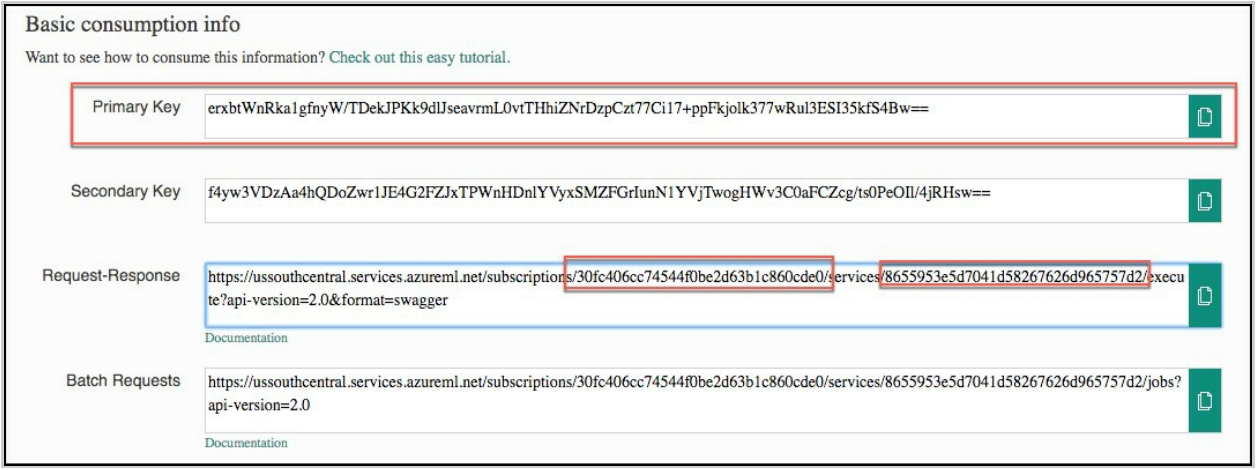
1. Navigate to https://github.com/ZoinerTejada/mcw-big-data-and-visualization/blob/master/AdventureWorksTravel/README.md in your browser of choice, but where you are already authenticated to the Azure portal.

2. Read through the README information on the GitHub page and capture the required parameters.

3. Click the **Deploy to Azure** button.

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.
   - Finally, enter the ML API and Weather API information.
     - Recall that you recorded the ML API information back in Exercise 1, Task 9.
       1. This information can be obtained on your Machine Learning web service page (https://services.azureml.net, then go to the **Consume** tab.
       2. The Primary Key listed is your ML API key
       3. In the Request-Response URL, the **GUID after subscriptions/** is your **ML Workspace Id**
       4. In the Request-Response URL, the **GUID after services/** is your **ML Service Id**

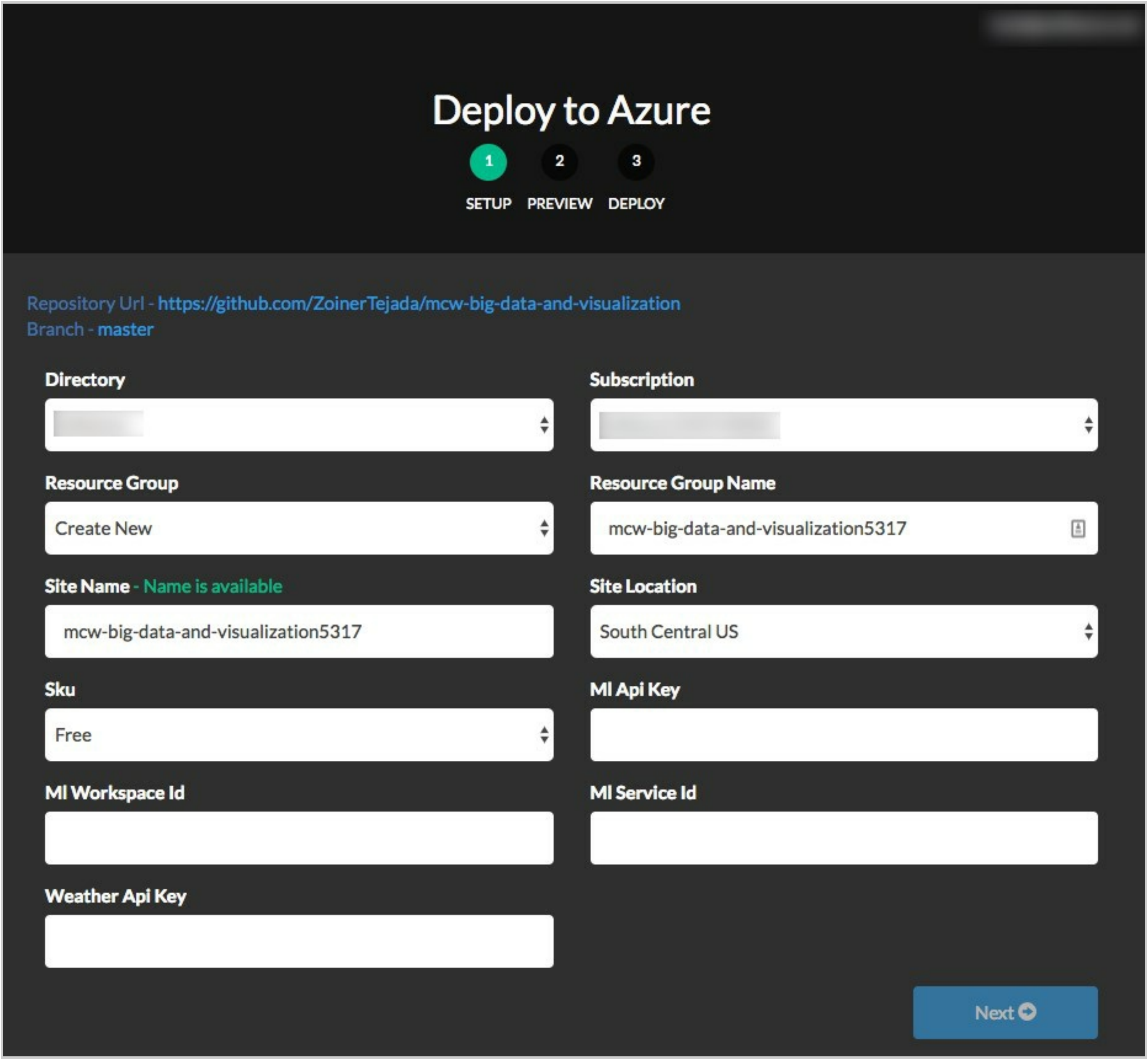


Basic consumption info

Want to see how to consume this information? Check out this easy tutorial.

Primary Key	erxbtWnRka1gfnyW/TDekJPKk9dlJseavrmL0vtTHhiZNrDzpCzt77Ci17+ppFkjolk377wRul3ESI35kfS4Bw==
Secondary Key	f4yw3VDzAa4hQDoZwr1JE4G2FZJxTPWnHDnlYVyxSMZFGrIunN1YVjTwogHWv3C0aFCZcg/ts0PeOIl/4jRHsw==
Request-Response	https://ussouthcentral.services.azureml.net/subscriptions/30fc406cc74544f0be2d63b1c860cde0/services/8655953e5d7041d58267626d965757d2/execute?api-version=2.0&format=swagger Documentation
Batch Requests	https://ussouthcentral.services.azureml.net/subscriptions/30fc406cc74544f0be2d63b1c860cde0/services/8655953e5d7041d58267626d965757d2/jobs?api-version=2.0 Documentation

- Also, recall that you obtained the **Weather API key** back in the **Task 3** of the **prerequisite steps** for the lab. Insert that key into the **Weather Api Key field**.



[redacted]

Deploy to Azure

Repository Url - https://github.com/ZoinerTejada/mcw-big-data-and-visualization
Branch - master

Directory	[redacted]
Subscription	[redacted]
Resource Group	Create New
Resource Group Name	mcw-big-data-and-visualization5317
Site Name - Name is available	mcw-big-data-and-visualization5317
Site Location	South Central US
Sku	Free
Ml Api Key	
Ml Workspace Id	
Ml Service Id	
Weather Api Key	

5. Select **Next**, and on the following screen, select **Deploy**.

6. 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 an account that is either a Service Administrator or a Co-Administrator. You won't have permissions to deploy the website otherwise.

7. After a short time, the deployment will complete, and you will be presented with a link to your newly deployed web application. **CTRL+Click** to open it in a new tab.

8. 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 Weather Underground API.

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

Cleanup

Exercise 8: Cleanup After the hands-on workshop

Duration: 10 mins

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

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

Task 1: Delete resource group

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

Task 2: Delete the Machine Learning Workspace

1. From the Azure Portal, select **Machine Learning Studio workspaces**.
2. In the list of Workspaces, select the workspace you created.
3. Click **Delete** in the command bar at the bottom.
4. When prompted to confirm the deletion, click **Yes**.