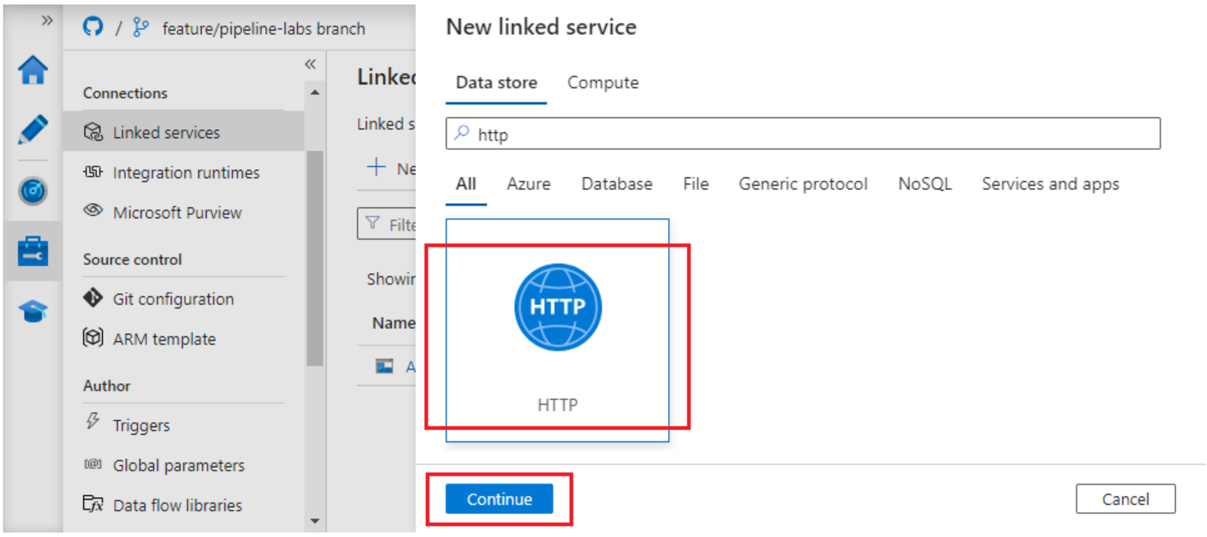
Lab 2 – Build a copy pipeline

In this lab you will create a pipeline to ingest data from a website and into your data lake. You'll be copying data from Microsoft's "AdventureWorks" sample OLTP database, which is available from the company's "sql-server-samples" repository on GitHub. The pipeline you build will copy a file directly from the website and into your data lake.

Lab 2.1 – Create source linked service

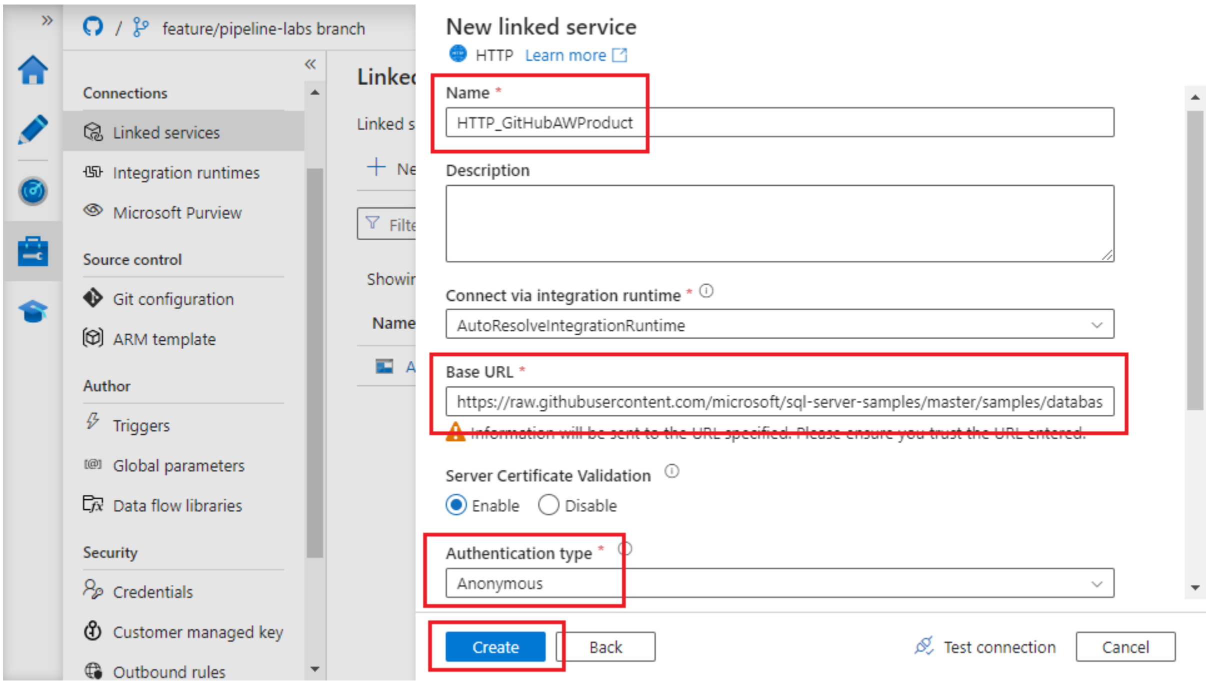
The linked service you created in Lab 1 defines a connection to your data lake. To copy data from an external web resource, a similar connection is required – in this section you'll create an HTTP linked service that enables access to a file on GitHub.

1. As in Lab 1.4, navigate to the management hub, open the "Linked services" page, then click "+ New". This time, choose a linked service of type "HTTP" and click "Continue".



2. Configure the linked service like this:

- Give it a **Name**
- Set its **Base URL** to "https://raw.githubusercontent.com/microsoft/sql-server-samples/master/samples/databases/adventure-works/oltp-install-script/Product.csv". This is the URL of a raw text file containing Adventure Works product data.
- Set **Authentication type** to "Anonymous"
- Click "Create"



New linked service

HTTP

Name *: HTTP_GitHubAWProduct

Description:

Connect via integration runtime *: AutoResolveIntegrationRuntime

Base URL *: https://raw.githubusercontent.com/microsoft/sql-server-samples/master/samples/databas

Information will be sent to the URL specified. Please ensure you trust the URL entered.

Server Certificate Validation: ● Enable ○ Disable

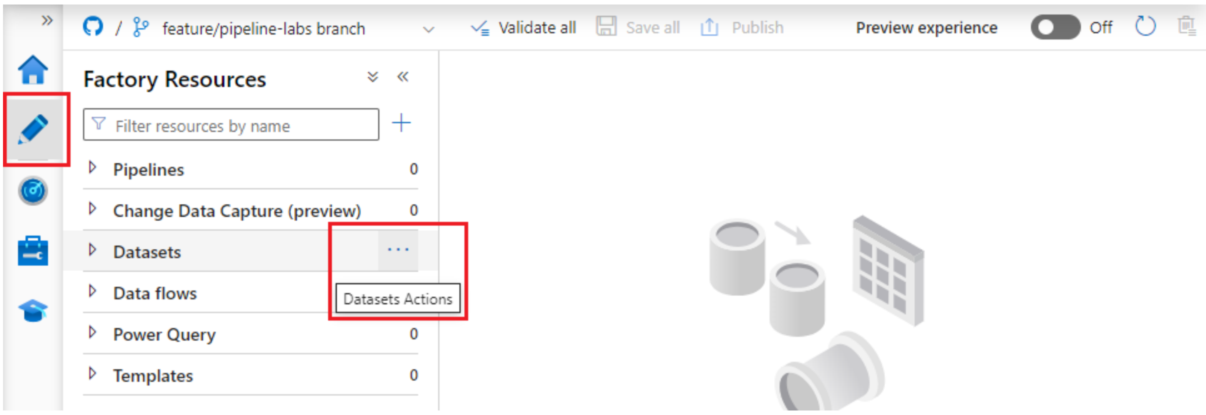
Authentication type *: Anonymous

Lab 2.2 – Create datasets

In ADF, the linked services you created in Lab 1.4 and Lab 2.1 represent connections to external systems, but not to the data objects inside those systems. Data stored by those systems must be represented using datasets.

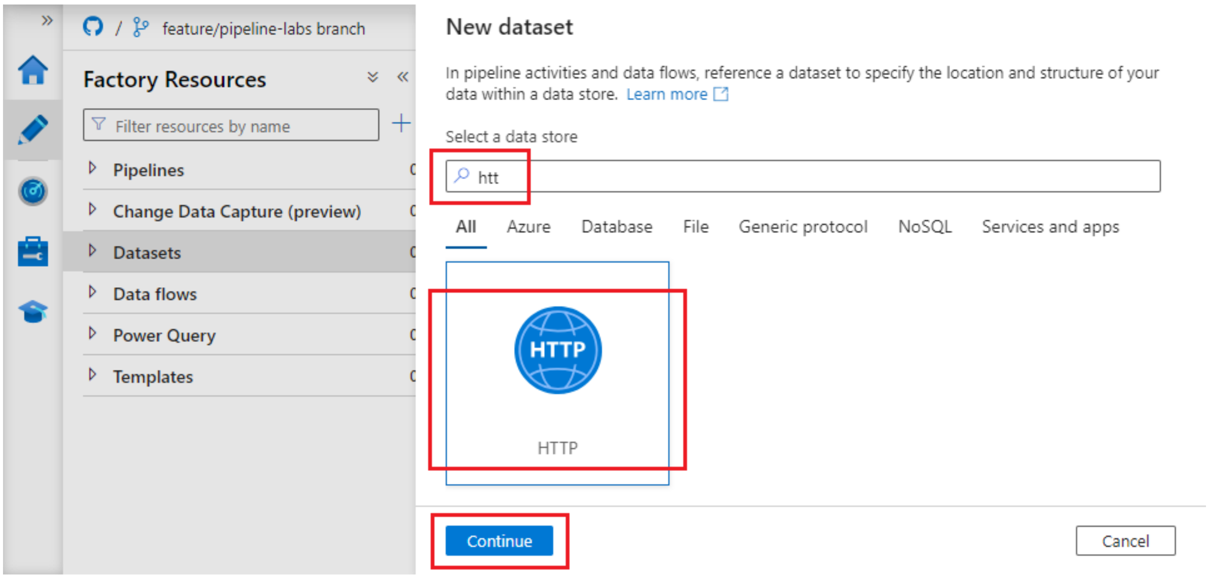
In this section you will create a dataset representing the source (GitHub) and sink (data lake) files between which your pipeline will copy data.

1. Navigate to the authoring canvas in ADF Studio using the “Author” button (pencil icon) in the leftmost toolbar. Hover over the number to the right of “Datasets” in the “Factory Resources” menu to reveal an ellipsis button. Click the button to expand the “Dataset Actions” menu.



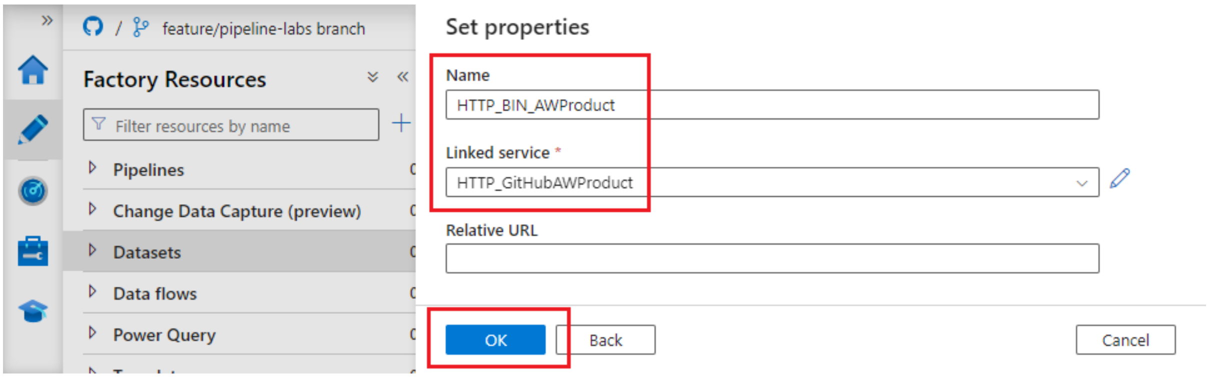
Factory Resources	
Pipelines	0
Change Data Capture (preview)	0
Datasets	···
Data flows	
Power Query	0
Templates	0

2. From the “Dataset Actions” menu, choose “New dataset”, then search for and select the “HTTP” data store in the “New dataset” flyout. Click “Continue”.



3. Choose the “Binary” format from the “Select format” page. The format does not suggest that a file has no internal structure – it means simply that at this point we are not specifying one.

4. Name the dataset “HTTP_BIN_AWProduct”, then select the linked service you created in the previous lab. Leave “Relative URL” blank, then click “OK”.



Set properties

Name: HTTP_BIN_AWProduct

Linked service *: HTTP_GitHubAWProduct

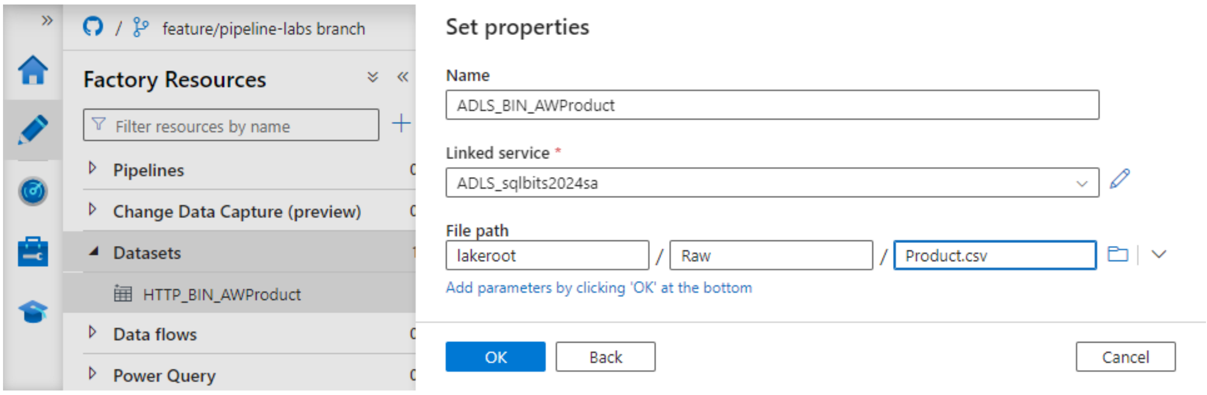
Relative URL:

5. Dataset changes are **not** automatically committed to the factory’s GitHub repository. Click “Save” below the dataset’s tab (or “Save all” in the ADF Studio header bar) to commit your changes. Unsaved changes are discarded when you quit ADF Studio.

6. Repeat steps 2-5 to create a second integration dataset, this time to represent the sink file to be written into the data lake.

   - Choose the “Azure Data Lake Storage Gen2” data store.
   - Choose the “Binary” file format.
   - Name the dataset “ADLS_BIN_AWProduct” and select the data lake linked service you created in Lab 1.
   - Specify the location (file path) into which you want the file to be copied. Use the “lakeroot” file system (container) created in Lab 1, the “raw” directory, and set “File name” to “Product.csv” (consistent with the source URL used in Lab 2.1).

   Click OK, then remember to commit (save) your changes to GitHub.



Set properties

Name: ADLS_BIN_AWProduct

Linked service *: ADLS_sqlbits2024sa

File path: lakeroot / Raw / Product.csv

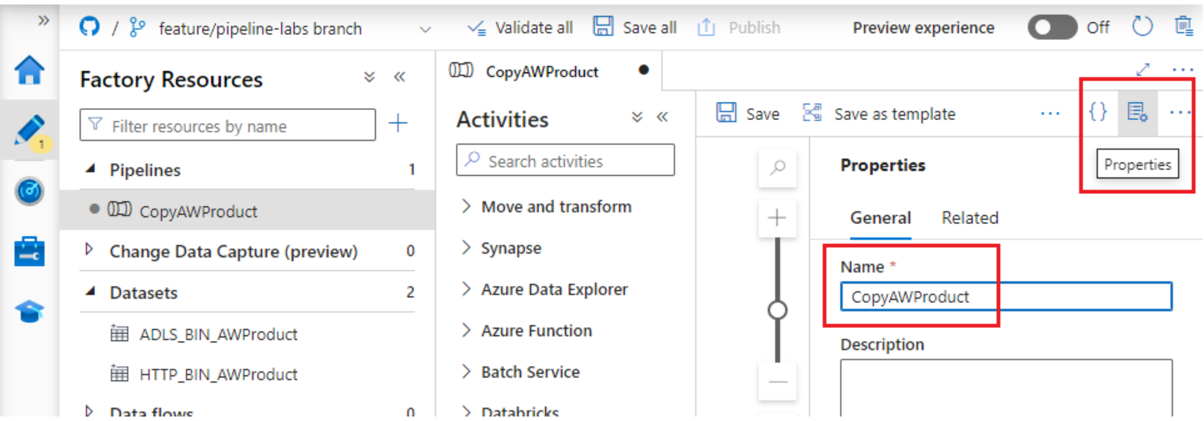
Add parameters by clicking 'OK' at the bottom

Lab 2.3 – Create a copy pipeline

In this section you will create a pipeline that copies data from the source dataset and into the sink dataset, both of which you created in Lab 2.2.

1. Open the “Pipelines Actions” menu in the same way you accessed the “Dataset Actions” menu, by clicking the ellipsis to the right of “Pipelines” in the “Factory Resources” menu.

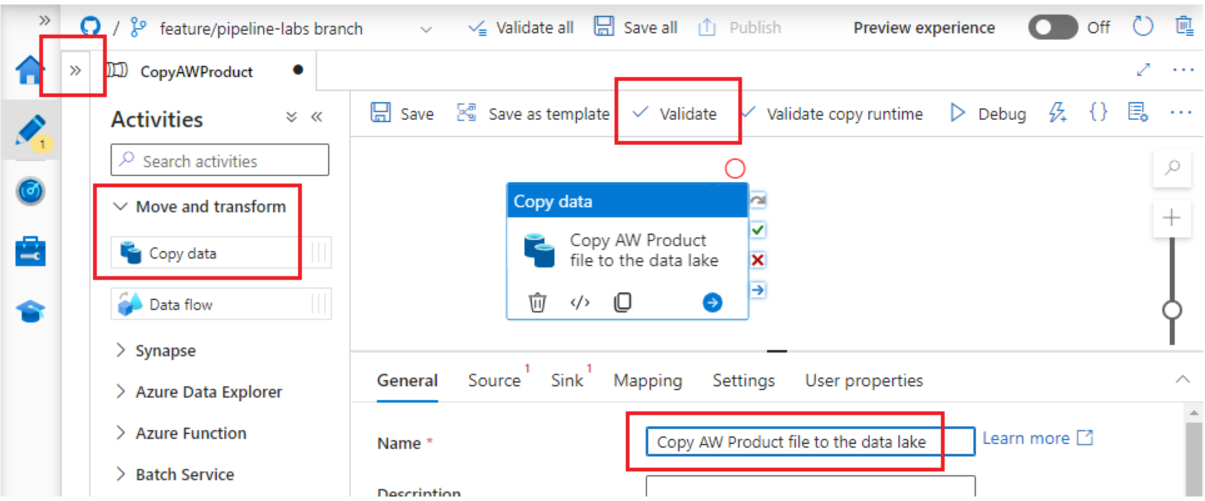
2. Select “New pipeline”. A new pipeline opens in the tabbed pane to the right, and the pipeline’s “Properties” flyout appears. Name the pipeline appropriately using the “Properties” flyout, then click the “Properties” toggle button to dismiss the flyout.



Properties — General

Name *: CopyAWProduct

3. If you need more space, collapse the “Factory Resources” sidebar using the left chevron button. Drag a “Copy” activity from the “Move & transform” section of the activity toolbox and drop it onto the pipeline canvas. Give the activity a meaningful name.



Name * Copy AW Product file to the data lake

4. Select the “Source” tab below the canvas and select the “HTTP_BIN_AWProduct” dataset from the “Source dataset” dropdown.

5. Select the “Sink” tab and select the “ADLS_BIN_AWProduct” dataset from the “Sink dataset” dropdown.

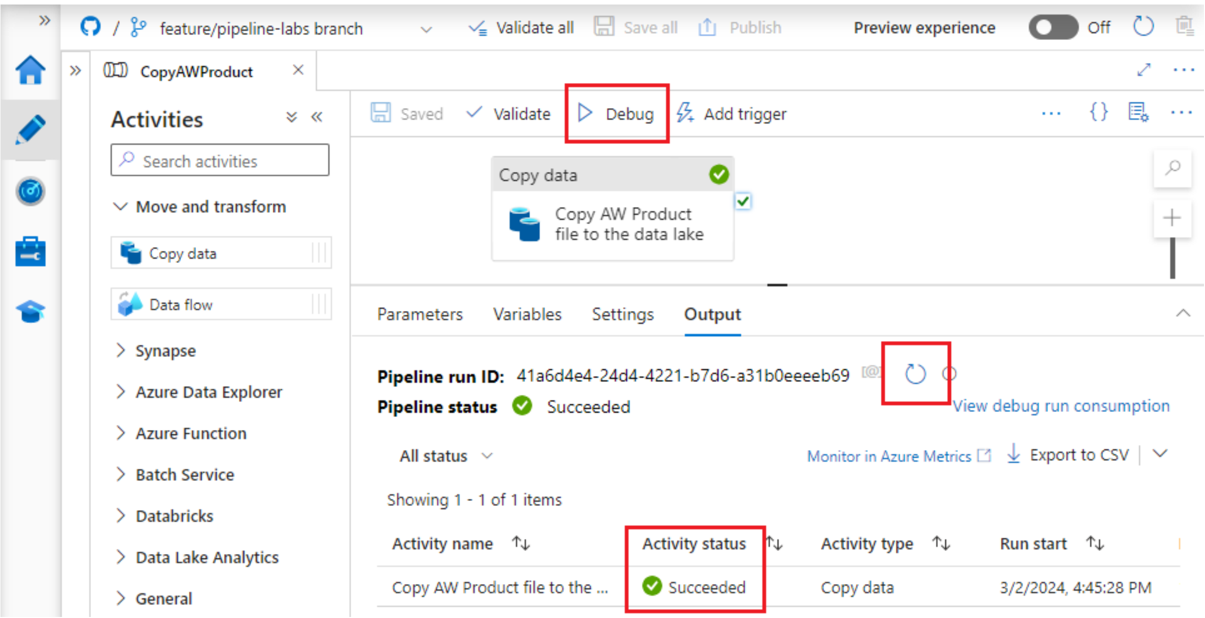
6. Finally, check your pipeline configuration by clicking the “Validate” button above the pipeline design surface (indicated in the screenshot above) and save your changes.

Lab 2.4 – Debug and test the pipeline

You can test your pipeline’s execution and outcome by running it in “Debug” mode in ADF Studio.

1. Click “Debug” above the pipeline canvas. The pipeline’s “Output” pane appears below the canvas.

2. The “Output” pane contains a row for each of the pipeline’s activity executions – in this case just one, for the Copy activity. The row shows the execution’s current status. While the pipeline is running, you can get status updates using the “Refresh” button.



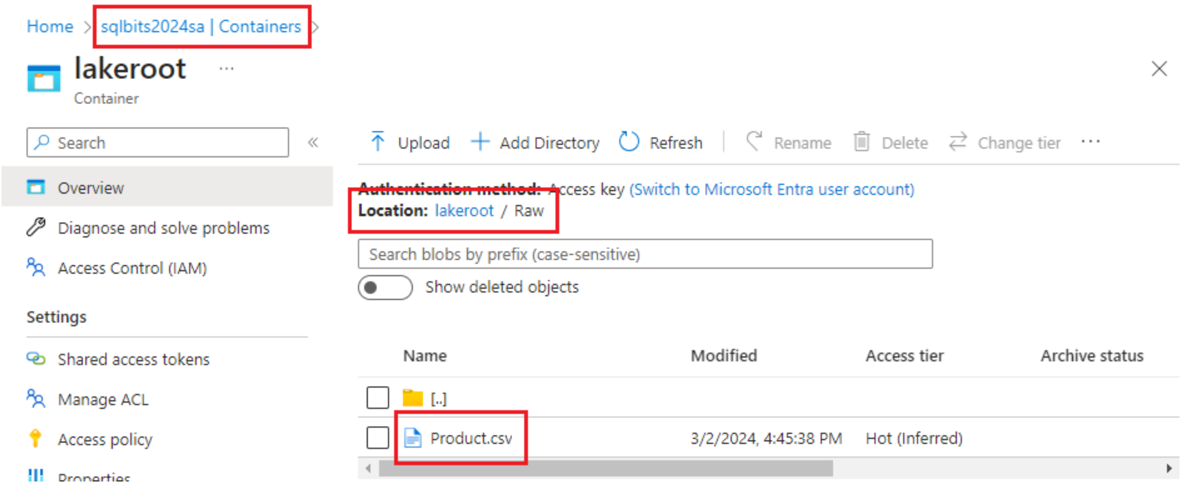
Pipeline run ID: 41a6d4e4-24d4-4221-b7d6-a31b0eeeeb69

Pipeline status ✓ Succeeded

Showing 1 - 1 of 1 items

Activity name	Activity status	Activity type	Run start
Copy AW Product file to the ...	✓ Succeeded	Copy data	3/2/2024, 4:45:28 PM

3. “Debug” runs your pipeline without publishing it to the data factory instance, but its effect is just the same – it has the same external dependencies, so has real effects on external resources. Open the “Raw” folder in the Azure portal and you will see the newly-copied file “Product.csv”.



Home > sqlbits2024sa | Containers >

lakeroot ···
Container

Search «

Overview
Diagnose and solve problems
Access Control (IAM)

Settings

Shared access tokens
Manage ACL
Access policy
Properties

Upload + Add Directory Refresh | Rename Delete Change tier ···

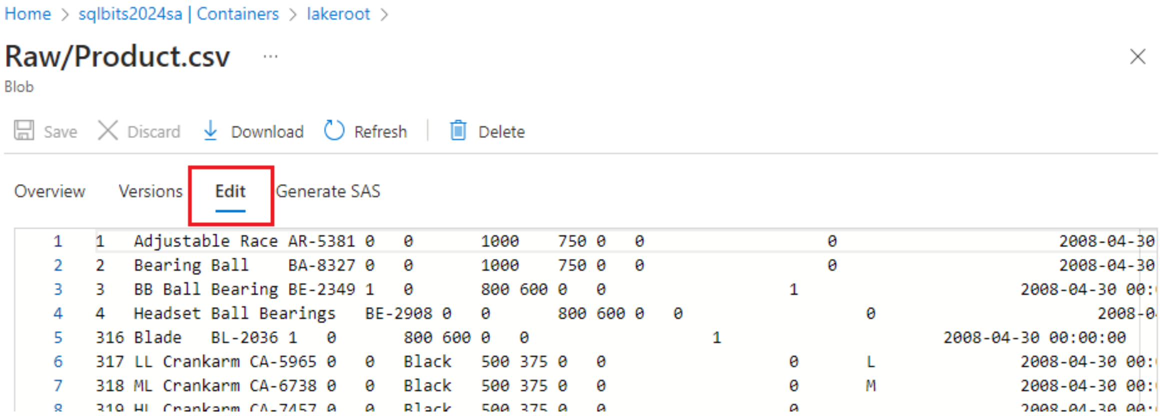
Authentication method: Access key (Switch to Microsoft Entra user account)
Location: lakeroot / Raw

Search blobs by prefix (case-sensitive)

Show deleted objects

	Name	Modified	Access tier	Archive status
□	[..]			
□	Product.csv	3/2/2024, 4:45:38 PM	Hot (Inferred)	

4. To inspect its contents, click the filename to open the “Blob” blade, then select the “Edit” tab.



Home > sqlbits2024sa | Containers > lakeroot >

Raw/Product.csv ···
Blob

Save Discard Download Refresh | Delete

Overview Versions Edit Generate SAS

```
1   Adjustable Race AR-5381 0   0       1000    750 0   0                   0                       2008-04-30
2   Bearing Ball    BA-8327 0   0       1000    750 0   0                   0                       2008-04-30
3   BB Ball Bearing BE-2349 1   0       800 600 0   0                   1                       2008-04-30 00:
4   Headset Ball Bearings   BE-2908 0   0       800 600 0   0                   0                       2008-0
316 Blade   BL-2036 1   0       800 600 0   0                   1                       2008-04-30 00:00:00
317 LL Crankarm CA-5965 0   0   Black   500 375 0   0                   0       L               2008-04-30 00:
318 ML Crankarm CA-6738 0   0   Black   500 375 0   0                   0       M               2008-04-30 00:
319 HL Crankarm CA-7457 0   0   Black   500 375 0   0                   0                       2008-04-30 00:
```

Notice that although the file has a “.csv” extension, it is not comma-separated – fields in the file are separated by tabs instead. This will be important in Lab 4.

Recap

In Lab 2 you:

- created a linked service to connect to an external web source (GitHub)
- created datasets to represent a data file in the source and in the data lake
- created a pipeline to copy the Products.csv file from GitHub and into your data lake, using the new datasets and linked services
- ran the pipeline in debug mode and inspected its results.