

Lab 02 – Building a Copy Data Pipeline

You will need:

- Access to our AdventureWorks Database (details below)
- An Azure subscription with a Data Factory & ADLS Gen 2 (from Lab 1)

Lab 02.A – Create the Linked Services & Datasets

First things first, we'll get you linked up to our public copy of AdventureWorks. This is the OLTP version of the database, stored as an Azure SQLDB. We restrict the firewall to our events, but it's very easy to create your own version. There are some instructions for that as a separate lab!

1. Open up the Data Factory we created in Lab 1 and navigate to the connections tab. Click the new button to add a new linked service.



2. We've stored AdventureWorks as a SQL DB, so select that from the list and click continue.



3. Give the linked service a sensible name – we use the pattern LS_TYPE_NAME, so LS_SQL_Adventureworks.

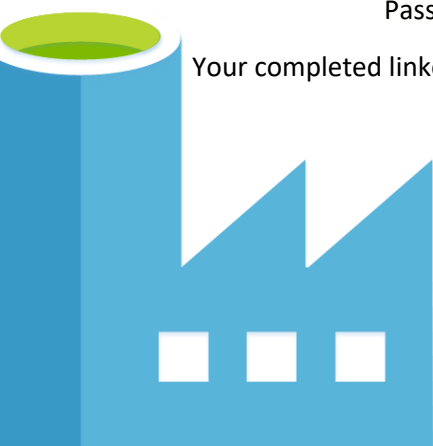
The connection details to the shared database are as follows:

Server: teachmeadf.database.windows.net

Username: [ADF](#)

Password: [@zureDataFactory](#)

Your completed linked service should look something like:



Lab 02 – Building a Copy Data Pipeline

Connect via integration runtime ^{*} ⓘ

AutoResolveIntegrationRuntime ▼

Connection String Azure Key Vault

Account selection method ⓘ

☐ From Azure subscription ☒ Enter manually

Fully qualified domain name ^{*}

teachmeadf.database.windows.net

Database name ^{*}

Adventureworks

Authentication type ^{*}

SQL Authentication ▼

User name ^{*}

ADF


Add dynamic content [Alt+P]




Password Azure Key Vault

Password ^{*}

.....

4. Test the connection, ensure it succeeds then click save. You should now see a completed Linked Service ready to be used.


LS_SQL_AdventureWorks

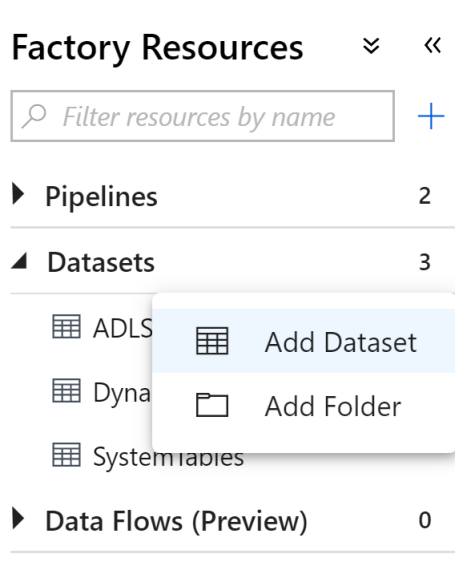




However, this is simply a connection to the server. To perform a copy, we need a dataset that tells us which table to copy.

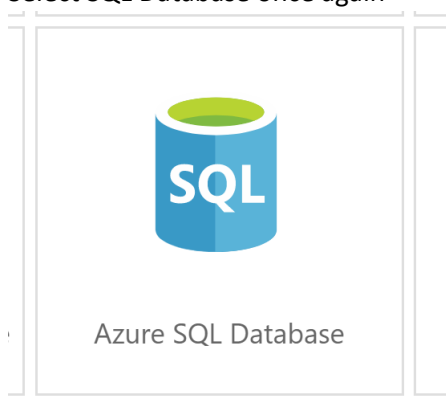
5. Expand the Datasets menu and click the “Add Dataset” button:



Lab 02 – Building a Copy Data Pipeline



6. Select SQL Database once again



7. Complete the properties, select a table from adventureworks and click Continue

Name

Linked service ^{*}

[Edit Connection](#)

Table

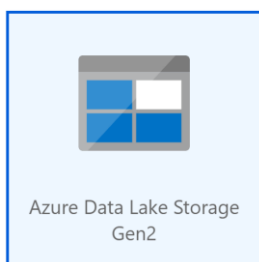
☐ Edit

Import schema
☒ From connection/store ☐ None

And we now have a dataset created, pointing at a specific table within Adventureworks.

Lab 02 – Building a Copy Data Pipeline

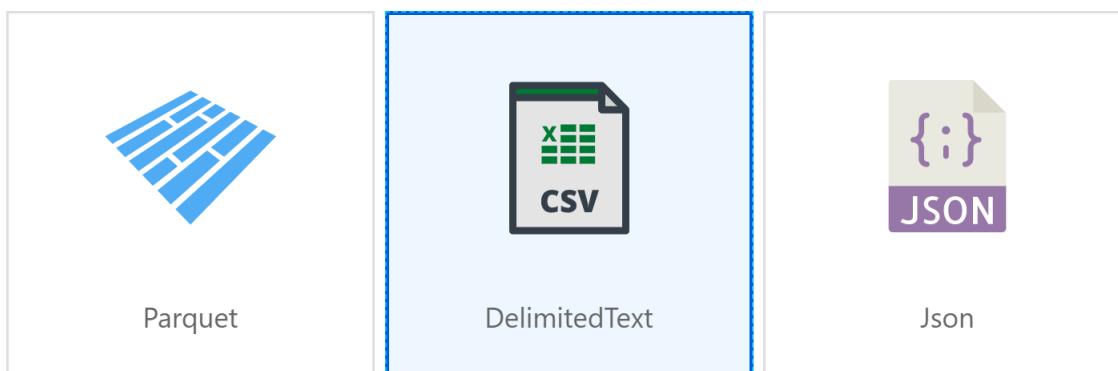
8. We need to do the same for the destination. Click and add a new Dataset, this time choosing “Data Lake Store Gen 2” as the type.



9. For flat formats, we also choose the file type. For now, select CSV:

← Select Format

Choose the format type of your data



10. Name the new dataset and select the ADLS Gen 2 linked service that we created in Lab 1. This will then open up file path – you can use the “browse” button to navigate your lake and select a destination file path

Lab 02 – Building a Copy Data Pipeline

Name
SalesOrderHeader

Linked service *
ADLS_MDWLake

[Edit Connection](#)

File path
lakeroot / RAW/Public/Adventure / File [Browse](#)

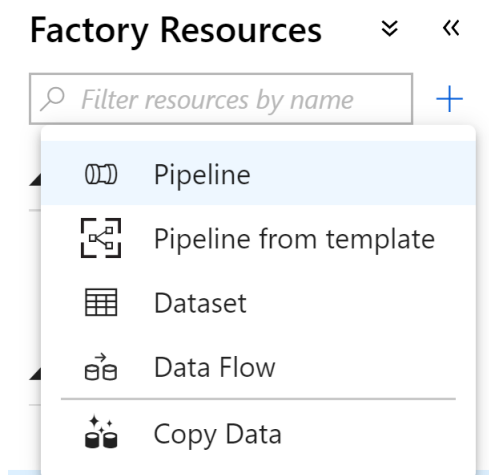
First row as header ☒

Import schema
☒ From connection/store
 ☐ From local file
 ☐ None

Lab 02.B – Create a Data Factory Pipeline

We now have our data lake ready to receive data, and a SQL database we want to get data from. Now we can create the data factory pipeline to extract the data from a table and land it into our lake.

1. Click on the “create pipeline” button to create a new data factory pipeline



This creates a blank pipeline that we can start working with.

2. Give the pipeline a reasonable name:



Lab 02 – Building a Copy Data Pipeline

The screenshot shows the 'General' tab of a new pipeline in Azure Data Factory. The pipeline is named 'PL_Copy_SalesOrder'. The 'Description' field is empty. The 'General' tab is selected, and the 'Parameters', 'Variables', and 'Output' tabs are visible in the background.

- Open the “Move and Transform” activity menu and drag a “Copy Data” activity onto our pipeline canvas:

The screenshot shows the 'Activities' pane on the left, which is expanded to show the 'Move & Transform' category. The 'Copy Data' activity is highlighted with a dashed blue border. On the right, the 'Copy Data' activity is being added to the pipeline canvas, indicated by a red circle around the activity icon. The activity is named 'Copy Data1'.

- Give the activity a sensible name. Then click “Source” tab to set up where we’re copying data from. Just select the SQL dataset we created earlier.

The screenshot shows the 'Source' tab of the 'Copy Data' activity. The 'Source dataset' is set to 'DS_SQL_SalesOrderHeader'. The 'Use query' section has the 'Table' radio button selected. The 'Sink' tab is also visible in the background.

- Next click on “Sink” to set up the destination using the CSV dataset:

The screenshot shows the 'Sink' tab of the 'Copy Data' activity. The 'Sink dataset' is set to 'SalesOrderHeader'. The 'Copy behavior' is set to 'None'. The 'Source' tab is also visible in the background.

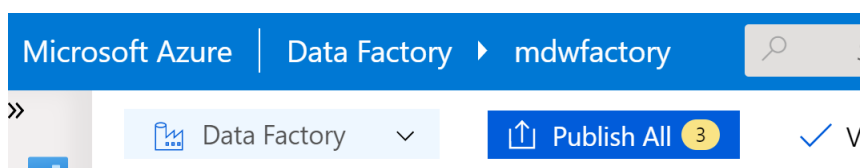
Lab 02 – Building a Copy Data Pipeline

That's everything set up for our very basic copy pipeline. We have told it where to get the data from, and how to connect. We have configured the destination and where it should place it, and we have created a workflow that to perform that work. Now, like any good developer, we just need to test it.

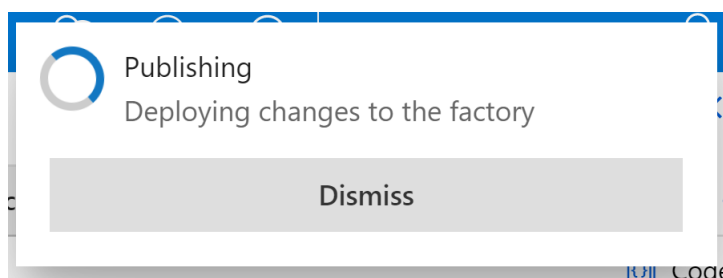
Lab 02.C – Trigger & Review Our Pipeline

So far, the changes we've made have not yet been deployed. We want to try out the pipeline in the real world, so let's publish our changes to our development data factory.

1. Click the "Publish" button to deploy our changes:

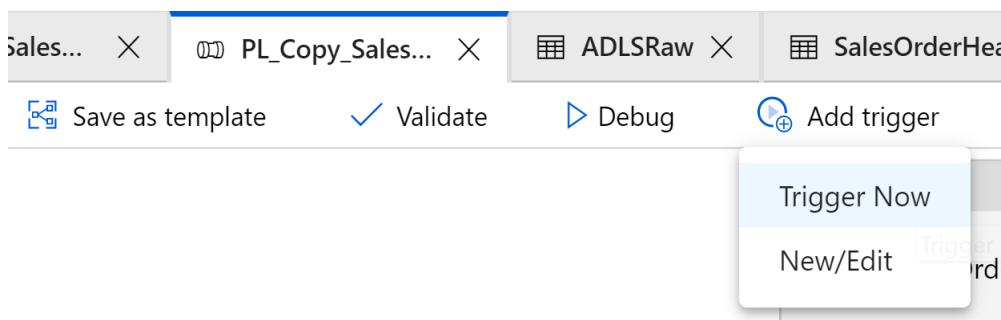


It'll take a moment to deploy your changes:

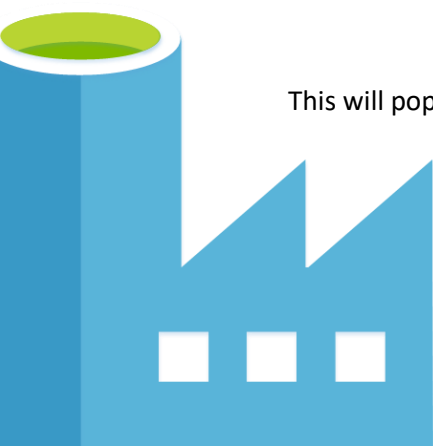


This is a useful step as it will also validate your configuration and make sure nothing has been done incorrectly.

2. Next, let's test out the pipeline and see if it works. Within our pipeline, click on the "Add Trigger" button and select "Trigger Now" to perform a manual execution of the pipeline.



This will pop up a confirmation screen:



Lab 02 – Building a Copy Data Pipeline

Pipeline Run



Trigger pipeline now using last published configuration.

Parameters

NAME	TYPE	VALUE
No records found		

Cancel

Finish

- Click Finish to complete the trigger.
- Finally, we can click on the “Monitor” button and go and see our executing pipeline. After a minute or so, you should see a successful pipeline like this one:

The screenshot shows the 'Pipeline Runs' tab in the Azure Data Factory portal. The interface includes a sidebar with navigation icons, a top navigation bar with links to Dashboards, Pipeline Runs, Trigger Runs, Integration Runtimes, and Alerts & Metrics. Below the navigation bar, there are buttons for 'Run', 'Cancel options', and 'Refresh'. A filter section shows 'Last 24 Hours' with a date range of '06/22/2019 9:44 PM - 06/23/2019 9:44 PM' and a 'Time Zone' dropdown set to '(UTC+00:00) Dublin, Edinburgh, Li...'. A toggle switch for 'View All Rerun History' is also present. The main table displays a list of pipeline runs with columns: Pipeline Name, Actions, Run Start, Duration, Triggered By, Status, and Parameters. One run is visible: 'PL_Copy_SalesOrder' with a status of 'Succeeded'.

Pipeline Name	Actions	Run Start	Duration	Triggered By	Status	Parameters
PL_Copy_SalesOrder		06/23/2019, 10:08:16 PM	00:00:09	Manual trigger	✓ Succeeded	

You might also want to go and check your data lake store, where you should see a file ready and waiting!

