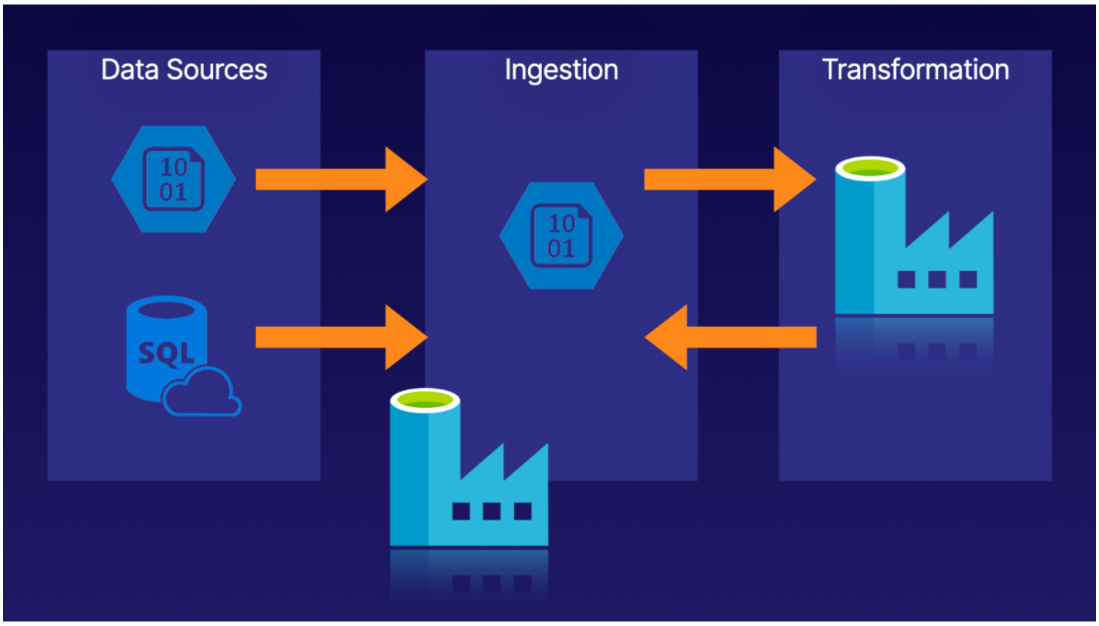
# Developing a Pipeline in Azure Data Factory

## Description

Azure Data Factory is a core service for any Azure cloud project. It is an orchestration service responsible for the movement and automation of data into and throughout the Azure cloud. In this lab, we will learn how to connect data sources and create a data pipeline that will move data in Azure.

In this scenario, you work for a company selling various items online all around the United States. You have been asked to research a way to combine sales data with other data sources, such as parking lot information to highlight potential new brick-and-mortar stores. In order to do this, you are going to provide a presentation on leveraging cloud services (specifically Data Factory) to automate this research.

https://docs.microsoft.com/en-us/learn/modules/data-integration-azure-data-factory/
https://docs.microsoft.com/en-us/learn/modules/orchestrate-data-movement-transformation-azure-data-factory/
https://docs.microsoft.com/en-us/learn/modules/receive-data-with-azure-data-share-transforming-with-azure-data-factory/
https://docs.microsoft.com/en-us/learn/modules/create-production-workloads-azure-databricks-azure-data-factory/



**Prepare the Environment**

**Create a Data Factory Instance**

1. In the search bar at the top, enter "data factory".
2. Select Data factories in the list of search results.
3. Click Create data factory.
4. Set the following values:

- Resource group: Select the lab-provided resource group
- Region: (US) West US 2
- Name: Enter a globally unique name (e.g., demo-<TODAY'S DATE-YOUR INITIALS>)
- Version: V2

5. Click Next: Git configuration.
6. Select the checkbox for Configure Git later.
7. Click Review + create > Create.

**Create a SQL Database**

1. In the search bar at the top, enter "SQL database".
2. Select SQL databases in the list of results.
3. Click Create SQL database.
4. Set the following values:

- Resource group: Select the lab-provided resource group
- Database name: sqldatabase
- Server: Click Create new
    - Server name: Enter a globally unique name (e.g., sqldatabase<TODAY'S DATE YOUR INITIALS>)
    - Server admin login: Enter a unique login name (e.g., labadmin)
    - Password and Confirm password: Enter a password of your choice
    - Location: (US) East US
    - Click OK.
- Want to use SQL elastic pool?: No
- Compute + storage:
    - Click Configure database.

    - Click Looking for basic, standard, premium?.
    - Click Basic.
    - Ensure it's set to 5 (Basic) DTUs at 2 GB.
    - Click Apply.
5. Click Next: Networking.
6. Set the following values:

- Connectivity method: Public endpoint
- Allow Azure services and resources to access this server: Yes
- Add current client IP address: Yes

7. Click Next: Additional settings.
8. Set the following values:

- Use existing data: Sample
- Enable Azure Defender for SQL: Not now

9. Click Next: Tags.
10. Click Review + create > Create.

## Create a Storage Account and Containers

1. In the search bar at the top, enter "storage account".
2. Select Storage accounts in the list of results.
3. Click Create storage account.
4. Set the following values:

- Resource group: Select the lab-provided resource group
- Storage account name: Enter a globally unique name (e.g., blobstorage<TODAY'S DATE YOUR INITIALS>)
- Location: (US) West US 2
- Performance: Standard
- Account kind: StorageV2 (general purpose v2)
- Replication: Read-access geo-redundant storage (RA-GRS)

5. We won't change any other default settings, so click Review + create > Create.
6. Once the storage account creation is done, click Go to resource.
7. Click Containers.
8. Click + Container.
9. In the New container pane that appears:

- Enter a Name of "raw".
- Set Public access level to Container (anonymous read access for containers and blobs).
- Click Create.

10. Click + Container again.
11. In the New container pane that appears:

- Enter a Name of "curated".
- Set Public access level to Container (anonymous read access for containers and blobs).
- Click Create.

## Create and Connect Datasets

### Create AzureSqlTable1 Dataset

1. Click the hamburger menu icon in the upper left corner.
2. Click All resources.
3. Click the data factory instance you created earlier.
4. Click Author & Monitor. This will open in a new browser tab.
5. Click the pencil icon in the left-hand menu.
6. Click the plus sign next to the filter resources box.
7. Select Dataset in the dropdown.
8. In the search box in the New dataset pane that appears, enter "sql".
9. Select Azure SQL Database in the search results.
10. Click Continue.
11. In the Set properties pane, for Linked service, select + New.
12. In the New linked service (Azure SQL Database), set the following values:

- Connect via integration runtime: AutoResolveIntegrationRuntime
- Account selection method: From Azure subscription
- Azure subscription: Select the lab-provided subscription
- Server name: Select the server you created earlier
- Database name: sqldatabase
- Authentication type: SQL authentication
- User name: Enter the admin login user name you created earlier
- Password: Enter the password you created earlier

- Click Create.

13. Back in the Set properties pane, for Table name, select SalesLT.Address.
14. Click OK.
15. In the Connection section, next to Table, click Preview data. This will bring up a window showing us the data pulled in to our SQL database table.
16. Close the Preview data window.

## Create raw Dataset

1. Click the plus sign next to the filter resources box.
2. Select Dataset in the dropdown.
3. In the New dataset pane, select Azure Blob Storage.
4. Click Continue.
5. Select DelimitedText (CSV), and click Continue.
6. In the Set properties pane, for Linked service, select + New.
7. In the New linked service (Azure Blob Storage), set the following values:

- Connect via integration runtime: AutoResolveIntegrationRuntime
- Authentication method: Account key
- Account selection method: From Azure subscription
- Azure subscription: Select the lab-provided subscription
- Storage account name: Select the storage account you created earlier
- Test connection: To linked service
- Click Create.

8. Back in the Set properties pane, set the following values:

- Name: raw
- File path:

- Container: raw

9. Click OK.

## Create curated Dataset

1. Click the plus sign next to the filter resources box.
2. Select Dataset in the dropdown.
3. In the New dataset pane, select Azure Blob Storage.
4. Click Continue.

5. Select DelimitedText (CSV), and click Continue.
6. In the Set properties pane, set the following values:

- Name: curated
- Linked service: AzureBlobStorage1
- File path:

- Container: curated

7. Click OK.

## Download CSV File and Upload to Blob Storage Account Container

1. In a new browser tab, navigate to the following URL: https://opendata.hawaii.gov/dataset/oahu-state-public-parking-lots/resource/2c1446fc-caef-4eab-b641-35d2ea0c41eb
2. Click Download.
3. Back in the Azure portal, navigate to your blob storage account.
4. Click Containers.
5. Click raw.
6. Click Upload.
7. In the Upload blob pane, navigate to the oahu-state-public-parking-lots-csv.csv file you just downloaded and click Upload.

### Create oahu state public parking Dataset

1. Back in the Data Factory, click the plus sign next to the filter resources box.
2. Select Dataset in the dropdown.
3. In the New dataset pane, select Azure Blob Storage.
4. Click Continue.
5. Select DelimitedText (CSV), and click Continue.
6. In the Set properties pane, set the following values:

- Name: oahu state public parking
- Linked service: AzureBlobStorage1
- File path:
    - Container: raw
    - File: oahu-state-public-parking-lots-csv.csv

- First row as header: Check the box

7. Click OK.

## Create the Copy Steps of Our Pipeline

1. Click the plus sign next to the filter resources box.
2. Select Pipeline in the dropdown.
3. Expand Move & transform.
4. Left-click Copy data and drag it onto the canvas.
5. In the General section below the canvas, change its name to "**sqldbingest**".
6. Click the Source tab.
7. For Source dataset, select AzureSqlTable1.
8. Click Preview data to ensure it's the correct data.
9. Close out of the Preview data window.
10. Click the Sink tab.
11. For Sink dataset, select raw.
12. Set Copy behavior to Preserve hierarchy.
13. Change File extension to ".csv".
14. Click Validate at the top of the canvas. It should then be validated without any errors.
15. Click Publish all at the top.
16. We'll see an error regarding the name "oahu state public parking".
17. Expand Datasets in the left-hand menu.
18. Select oahu state public parking.
19. In the Properties pane on the right, remove all the spaces in its name (making it **oahustatepublicparking**).
20. Click pipeline1 in the left-hand menu.
21. Click Publish all. (This time, there shouldn't be an errors.)
22. Click Publish.
23. Click Add trigger > Trigger now.
24. Click OK.
25. In the Azure portal, navigate to the raw container in your blob storage account.
26. Refresh the container so you see the **SalesLT.Address.csv** file.

## Use Data Flow to Combine Data from Our Copied File and CSV File

### Create a Data Flow and Data Sources

1. Back in Data Factory, click the pencil icon in the left-hand menu.
2. Click the plus sign next to the filter resources box.
3. Select Data flow.
4. Click through and close out of any intro or tutorial dialogs that appear.
5. Click the Add Source box on the canvas.
6. Click through and close out of any intro or tutorial dialogs that appear.
7. In the Source settings section below the canvas, set the following values:

- Output stream name: oahu
- Source type: Dataset
- Dataset: oahustatepublicparking

8. At the top of the page, click the toggle to enable Data flow debug.
9. Click OK in the confirmation dialog.
10. Click the Add Source box on the canvas.
11. In the Source settings section below the canvas, set the following values:

- Output stream name: curatedsql
- Source type: Dataset
- Dataset:

- Click New.
- In the New dataset pane, select Azure Blob Storage.
- Click Continue.
- Select DelimitedText (CSV), and click Continue.
- In the Set properties pane, set the following values:
    - Name: oahu
    - Linked service: AzureBlobStorage1
    - File path:
        - Container: raw
        - File: oahu-state-public-parking-lots-csv.csv
    - First row as header: Check the box
- Click OK.

12. Click the Data preview tab.

### Combine oahu and curatedsql Data Streams

1. Click the plus sign on the lower right corner of the oahu source box on the canvas.
2. Click Union in the menu that appears.
3. In the Union settings section below the canvas, set Streams to curatedsql.
4. Click the Optimize tab.
5. Set Partition option to Single partition.
6. Click the plus sign on the lower right corner of the Union1 box on the canvas.
7. Click Sink in the menu that appears.
8. Click Finish in the dialog.
9. In the Sink section below the canvas, set Dataset to curated.
10. Click the Settings tab.
11. Set File name option to Output to single file.
12. In the Output to single file box, enter "results".
13. Click the Optimize tab.
14. Set Partition option to Single partition.
15. Click the oahu source box on the canvas.
16. Click the Data preview tab.
17. Click Refresh. We should see all our data listed.
18. Click the Source settings tab.
19. Next to Dataset, click Open.
20. Check the box for First row as header.
21. Click dataflow1 in the left-hand menu.
22. Click the Data preview tab.
23. Click Refresh. This time, the text previously listed in the first row should now be the header names instead.

- You may have to scroll to the right to see the data.

24. Click Validate at the top.
25. Click pipeline1 in the left-hand menu.
26. From the list of factory resources in the left pane, click and drag dataflow1 onto the canvas.
27. Click the sqldbingest box.

28. Click the plus sign and arrow icon on the lower right corner of the box.
29. In the Add activity on dialog, choose Completion. This will add a blue square to the sqldbingest box.
30. Click the blue square and drag the arrow that appears to the dataflow1 box.

## Publish the Pipeline

1. Click the dataflow1 box on the canvas.
2. Click Validate at the top.
3. Click Debug at the top.
4. Click the monitor icon (beneath the pencil icon) in the far left-hand menu.
5. In the Debug section, we should see it's succeeded.
6. Click the pencil icon in the far left-hand menu.
7. Click Publish all at the top.
8. In the Publish all pane, click Publish.
9. In the Azure portal, navigate to the curated container in your blob storage account.
10. Click the listed results file. It will be a combination of all the results from those two files.

### Conclusion

Congratulations on successfully completing this hands-on lab!