

Something from an old presentation on the subject

Seamless connection to Fabric data

Microsoft Fabric

1. Create Lakehouse
2. Upload .csv
3. Load as (Delta) Table
4. Add AML compute instance workspace Viewer/Contributor

The screenshot shows the Microsoft Fabric Lakehouse interface. The URL is <https://onlake.dfs.fabric.microsoft.com/jlaWorkspace1/jlaLakehouse.Lakehouse/Tables/churn>. The interface displays a table named 'churn' with the following data:

	RowNumber	Customerid	Surname	CreditScore	Geography
1	1489	15625824	Kornilova	596	Spain
2	2873	15671591	Castiglione	439	Spain
3	3366	15764431	Chinwenma	671	Spain
4	4167	15654562	Ma	850	Spain
5	6280	15608338	Chienam	757	Spain
6	1255	15610383	Dumetolisa	628	France
7	2125	15603851	Galkin	704	France
8	2463	15704442	Fleming	672	France
9	2500	15634974	Seppelt	614	France
10	2542	15679770	Smith	611	France
11	3153	15724161	Sutton	644	France
12	3842	15750778	Ponomarev	653	France
13	5138	15601594	Ifeanacho	698	France
14	5387	15668283	Gardiner	642	France
15	6876	15665283	Brookes	610	France
16	7258	15648681	Voronoff	747	France
17	7725	15673591	Oluchukwu	842	France
18	7730	15681007	Yen	850	France
19	8591	15633194	Osborne	771	France
20	8684	15599329	Christopher	697	France

URL: <https://onlake.dfs.fabric.microsoft.com/jlaWorkspace1/jlaLakehouse.Lakehouse/Tables/churn>
ABFS path: `abfss://jlaWorkspace1@onlake.dfs.fabric.microsoft.com/jlaLakehouse.Lakehouse/Tables/churn`

Azure Machine Learning

1. Create Compute Instance – assign it managed identity
2. Create Notebook
 1. Reference delta table using abfss path

The screenshot shows the Azure Machine Learning Studio interface. The notebook is titled 'Contoso > jlaaml > Notebooks'. The code in the notebook is as follows:

```
1 !pip install mltable
[12] ✓ 4 sec

...
Requirement already satisfied: mltable in /anaconda/envs/azureml_py38/lib/python3.10/site-packages (1.6.1)
Requirement already satisfied: azureml-dataprep[parquet]<5.2.0a,>=5.1.0a in /anaconda/envs/azureml_py38/lib/python3.10/site-packages (from mltable) (5.1.6)
Requirement already satisfied: jsonschema<5.0.0,>=4.0.0 in /anaconda/envs/azureml_py38/lib/python3.10/site-packages (from mltable) (4.7.0)

1 from mltable import from_delta_lake
2
3 url = "abfss://jlaWorkspace1@onlake.dfs.fabric.microsoft.com/jlaLakehouse.Lakehouse/Tables/churn"
4 df2 = from_delta_lake(url).to_pandas_dataframe()
[4] ✓

Resolving access token for scope "https://storage.azure.com/.default" using identity of type "MANAGED".
Getting data access token with Assigned Identity (client_id=clientid) and endpoint type based on configuration

1 df2.head(5)
[11] ✓ <1 sec
```

The output of the code shows a table with the following data:

RowNumber	Customerid	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited	
0	1489	15625824	Kornilova	596	Spain	Male	30	6	121345.88	4	1	0	41921.75	1
1	2873	15671591	Castiglione	439	Spain	Male	52	3	96196.24	4	1	0	198874.52	1
2	3366	15764431	Chinwenma	671	Spain	Female	34	5	130929.02	4	1	1	28238.25	1
3	4167	15654562	Ma	850	Spain	Female	45	5	174088.30	4	1	0	5669.31	1
4	6280	15608338	Chienam	757	Spain	Female	55	9	117294.12	4	1	0	94187.47	1

There is no need to register onlake as datastore

Prove it's not needed:

It's not necessary to register the datastore to reference fabric tables

The screenshot shows the Azure AI Machine Learning Studio interface. The left sidebar contains navigation options: All workspaces, Home, Model catalog, Authoring, Notebooks, Automated ML, Designer, Prompt flow, Assets, Data, Jobs, Components, Pipelines, Environments, Models, Endpoints, Manage, Compute, Monitoring, Data Labeling, Linked Services, and Connections. The main area displays a Jupyter notebook titled 'jeffrey.ipynb' in the 'Notebooks' section. The notebook is running on 'jlaaml' with 'Python 3.8 - AzureML'. The code in the notebook is as follows:

```
1 !pip install mltable
[12] ✓ 4 sec

...
Requirement already satisfied: mltable in /anaconda/envs/azureml_py38/lib/python3.10/site-packages (1.6.1)
Requirement already satisfied: azureml-dataprep[parquet]<5.2.0a,>=5.1.0a in /anaconda/envs/azureml_py38/lib/python3.10/site-packages (from mltable) (5.1.6)
Requirement already satisfied: jsonschema<5.0.0,>=4.0.0 in /anaconda/envs/azureml_py38/lib/python3.10/site-packages (from mltable) (4.7.0)

1 from mltable import from_delta_lake
2
3 url = "abfss://jlaworkspace1@onelake.dfs.fabric.microsoft.com/jlaLakehouse.Lakehouse/Tables/churn"
4 df2 = from_delta_lake(url).to_pandas_dataframe()
[4] ✓

Resolving access token for scope "https://storage.azure.com/.default" using identity of type "MANAGED".
Getting data access token with Assigned Identity (client_id=clientid) and endpoint type based on configuration

1 df2.head(5)
[11] ✓ <1 sec
```

The output of the code shows a table with 11 columns: RowNumber, CustomerId, Surname, CreditScore, Geography, Gender, Age, Tenure, Balance, NumOfProducts, HasCrCard, IsActiveMember, EstimatedSalary, and Exited. The first five rows of data are displayed.

The screenshot shows the Azure AI Machine Learning Studio interface. The left sidebar contains navigation options: All workspaces, Home, Model catalog, Authoring, Notebooks, Automated ML, Designer, Prompt flow, Assets, Data, Jobs, Components, Pipelines, Environments, Models, Endpoints, Manage, Compute, Monitoring, Data Labeling, Linked Services, and Connections. The main area displays the 'Data' section. The 'Datastores' tab is selected, showing a list of datastores. The table below shows the details of the datastores.

Name	Type	Storage name
workspaceworkingdirectory	Azure file share	jlaaml2660487754
workspacefilestore	Azure file share	jlaaml2660487754
workspaceartifactstore	Azure Blob Storage	jlaaml2660487754
workspaceblobstore (Default)	Azure Blob Storage	jlaaml2660487754

The screenshot shows the Azure AI Machine Learning Studio interface. The left sidebar contains navigation options: All workspaces, Home, Model catalog, Authoring, Notebooks, Automated ML, Designer, Prompt flow, Assets, Data, Jobs, Components, Pipelines, Environments, Models, Endpoints, Manage, Compute, Monitoring, Data Labeling, Linked Services, and Connections. The main area displays the 'Connections' section. The 'Connect' button is highlighted. The table below shows the details of the connections.

Name	Type	Target
azureml_globaldatasets	Azure Blob Storage	https://mmstorageswedencentral.blob.core.wi...
workspaceartifactstore	Azure Blob Storage	https://jlaaml2660487754.blob.core.windows.n...
workspaceblobstore	Azure Blob Storage	https://jlaaml2660487754.blob.core.windows.n...

Load specific snapshot of delta table as mltable

The screenshot shows the Azure AI Machine Learning Studio interface. The notebook is titled 'jeffrey.jpynb' and is running on a compute instance named 'jaCompute1'. The code in the notebook is as follows:

```
1 import mltable
2
3 # create an MLTable from the data files
4 tbl = mltable.from_delta_lake("abfss://jlaworkspace@onelake.dfs.fabric.microsoft.com/jlaLakehouse.Lakehouse/Tables/churn", timestamp_as_of="2022-10-01T00:00:00Z")
5
6 # show the first 5 records
7 tbl.show(5)
```

The output of the code is a table with 12 columns: RowNumber, CustomerId, Surname, CreditScore, Geography, Gender, Age, Tenure, Balance, NumOfProducts, HasCrCard, IsActiveMember, EstimatedSalary, and Exited. The first 5 rows are displayed:

RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
0	1489	15625824	Kornilova	Spain	Male	30	6	121345.88	4	1	0	41921.75	1
1	2873	15671591	Castiglione	Spain	Male	52	3	96196.24	4	1	0	198874.52	1
2	3366	15764431	Chinwenma	Spain	Female	34	5	130929.02	4	1	1	28238.25	1
3	4167	15654562	Ma	Spain	Female	45	5	174088.30	4	1	0	5669.31	1
4	6280	15608338	Chienenam	Spain	Female	55	9	117294.12	4	1	0	94187.47	1

The code then saves the data loading steps in an MLTable file:

```
1 # save the data loading steps in an MLTable file
2 tbl.save("./covid")
```

The output of this code is a JSON object representing the MLTable file:

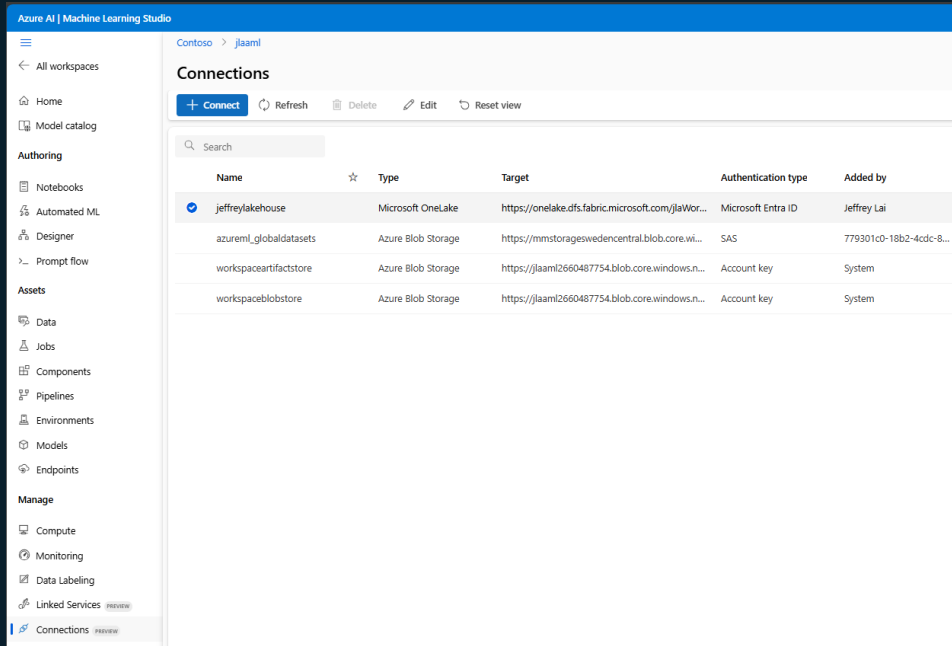
```
{
  "paths": [
    {
      "folder": "abfss://jlaworkspace@onelake.dfs.fabric.microsoft.com/jlaLakehouse.Lakehouse/Tables/churn"
    }
  ],
  "transformations": [
    {
      "read_delta_lake": {
        "include_path_column": false,
        "timestamp_as_of": "2022-10-01T00:00:00Z"
      }
    }
  ],
  "type": "mltable"
}
```

The code then reads the MLTable file and prints its contents:

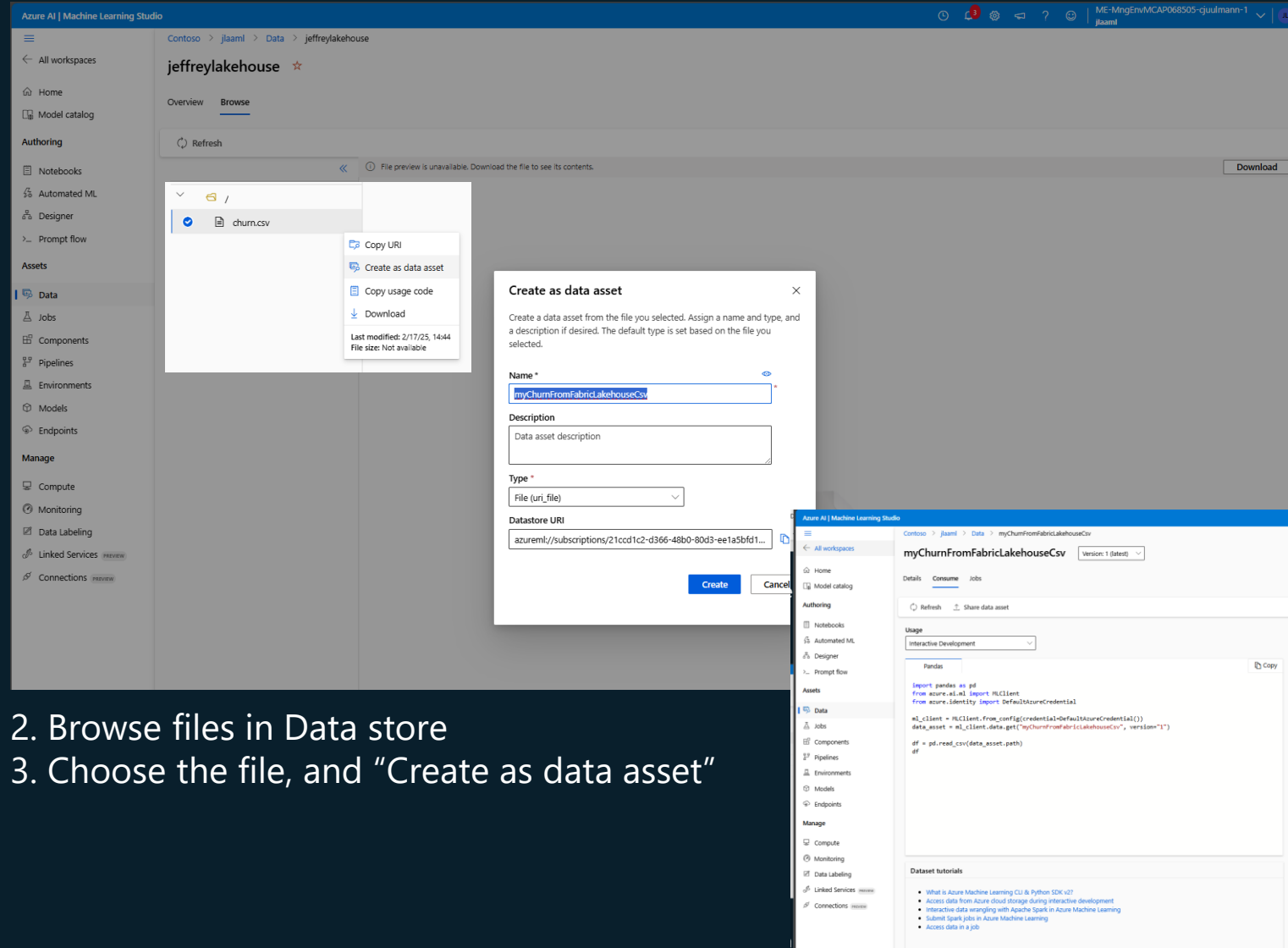
```
1 with open("./covid/MLTable", "r") as f:
2   print(f.read())
```

The output of this code is the same JSON object as above.

Creating Data Assets from files inside of Fabric Lakehouse



1. Register Lakehouse as Data store



2. Browse files in Data store

3. Choose the file, and "Create as data asset"