# Using TOM to Create a DirectLake Dataset

This repository contains C# console application named **TOM\_CreateFabricDataset** which demonstrates how to create a DirectLake data model for Fabric and Power BI using the Tabular Object Model (TOM). This repository also contains a Fabric notebook named **CreateLakehouseTables.ipynb** with Python code which must be used to create tables in a Fabric Lakehouse that will be used as the underlying datasource for the DirectLake data model.

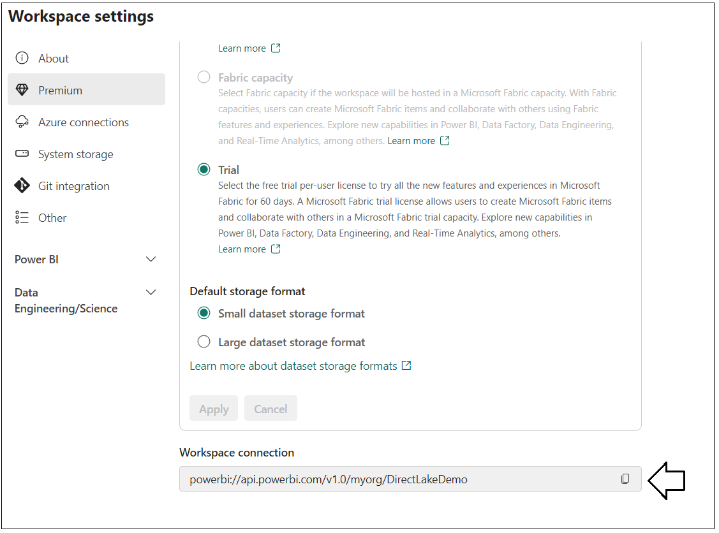
Here are the high-level steps to completing this demonstration:

* Create workspace associated with Fabric capacity
* Create a new Lakehouse in the new workspace
* Create Lakehouse tables using a pre-provided Fabric notebook
* Run the custom C# application to create DirectLake data model using TOM

## Create workspace associated with Fabric capacity

Create a new workspace with a name such as **DirectLakeDemo**. Make sure the workspace is associated with a Premium capacity or a trial capacity with Fabric capabilities.

Get URL to Workspace Connection

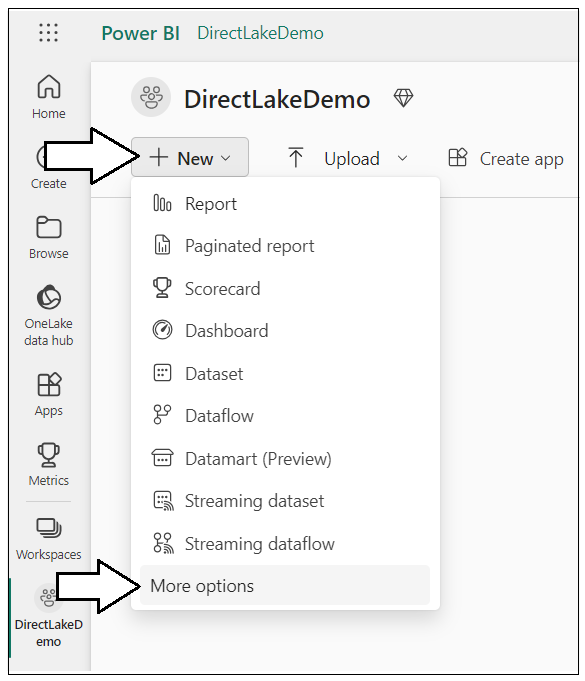


The new

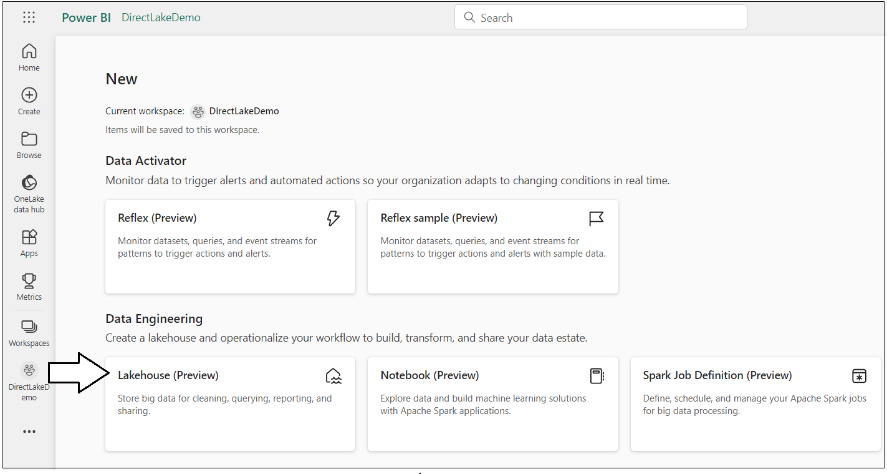
powerbi://api.powerbi.com/v1.0/myorg/DirectLakeDemo

## Create a new Lakehouse in the new workspace

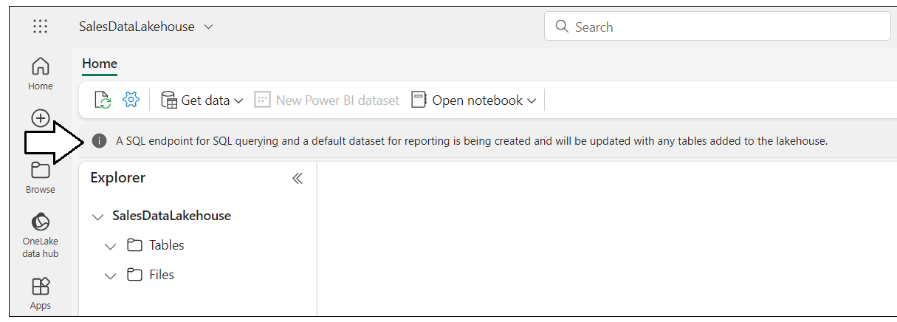
Inside the new workspace, create a new Lakehouse named **SalesDataLakehouse**.



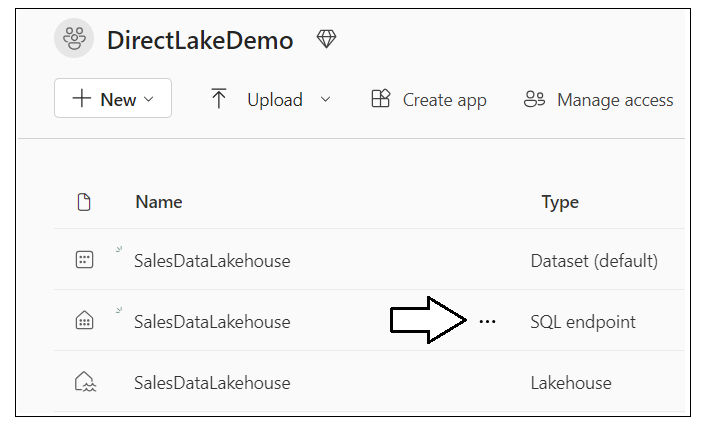
Ssss



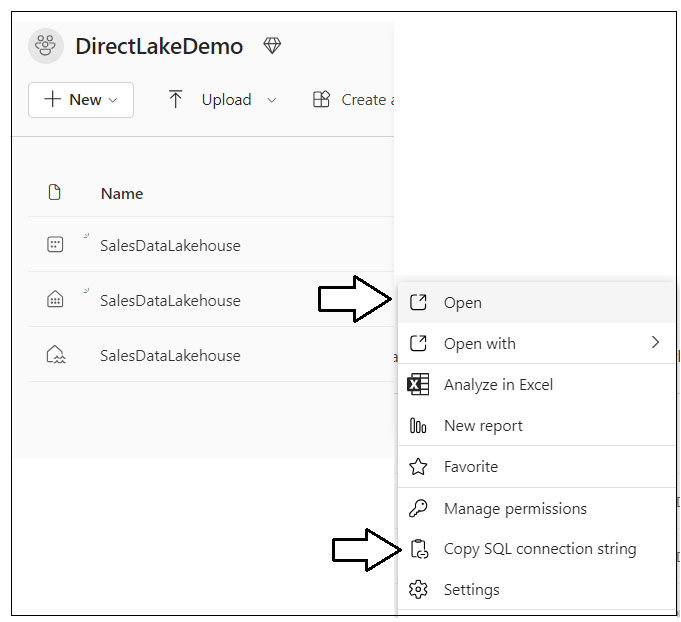
Sss



Ssss



Sssssss



Get Lakehouse SQL Endpoint

A screenshot of a computer

Description automatically generated

Ssss

5lcsgl3vll3edero2m4sge7gdu-nya26urqtgsejoagwutwdoogl4**.datawarehouse.pbidedicated.windows.net**

## Create Lakehouse tables using a pre-provided Fabric notebook

* Upload Python notebook named **CreateLakehouseTables.ipynb**
* Associate notebook named **CreateLakehouseTables.ipynb** with Lakehouse
* Execute code in notebook to copy CSV files from GitHib repository into Lakehouse file system
* Execute code in notebook to load CSV files and convert then into delta tables for bronze zone
* Execute code in notebook to load bronze tables and reshape/transform data in into delta tables for silver layer
* Execute code in notebook to generate calendar table for silver layer

## Run the custom C# application to create DirectLake data model using TOM

* Create Azure AD application
  1. Create a native/public application with redirect URI of <http://localhost>
  2. Record Application ID for use in console application.
* Download C# console application source code and open project in Visual Studio 2022
* Open **AppSettings.cs** and updae the following:
  1. ApplicationID of Azure AD application
  2. Workspace Connection
* SQL Endpoint
  1. Lakehouse Name
  2. UserID and Password to prevent interactive login
  3. Save changes
* Run application
  1. It should run without error
  2. When done, verify you can see new data model and use it to create new report