Analyze data in a data warehouse

In Microsoft Fabric, a data warehouse provides a relational database for large-scale analytics. Unlike the default read-only SQL endpoint for tables defined in a lakehouse, a data warehouse provides full SQL semantics; including the ability to insert, update, and delete data in the tables.

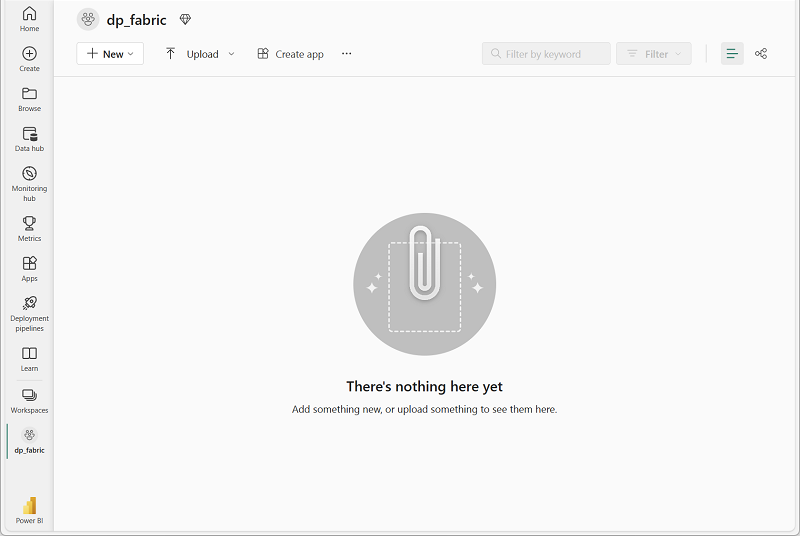
This lab will take approximately **30** minutes to complete.

**Note**: You’ll need a Microsoft Fabric license to complete this exercise. See [Getting started with Fabric](https://learn.microsoft.com/fabric/get-started/fabric-trial) for details of how to enable a free Fabric trial license. You will need a Microsoft *school* or *work* account to do this. If you don’t have one, you can [sign up for a trial of Microsoft Office 365 E3 or higher](https://www.microsoft.com/microsoft-365/business/compare-more-office-365-for-business-plans).

Create a workspace

Before working with data in Fabric, create a workspace with the Fabric trial enabled.

1. Sign into [Microsoft Fabric](https://app.fabric.microsoft.com/) at https://app.fabric.microsoft.com and select **Power BI**.
2. In the menu bar on the left, select **Workspaces** (the icon looks similar to 🗇).
3. Create a new workspace with a name of your choice, selecting a licensing mode that includes Fabric capacity (*Trial*, *Premium*, or *Fabric*).
4. When your new workspace opens, it should be empty, as shown here:

[](https://microsoftlearning.github.io/mslearn-fabric/Instructions/Labs/Images/new-workspace.png)

Create a data warehouse

Now that you have a workspace, it’s time to switch to the *Data Warehouse* experience in the portal and create a data warehouse.

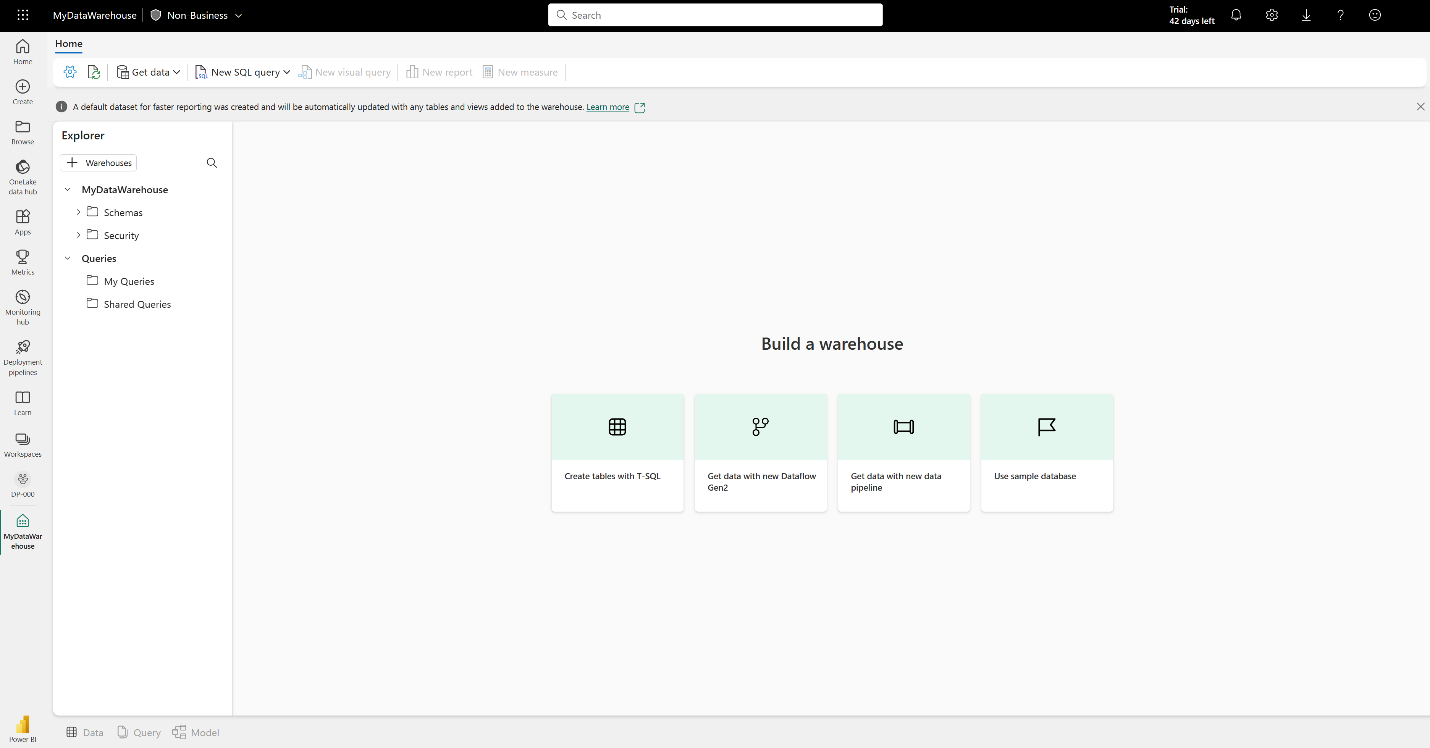
1. At the bottom left of the Power BI portal, switch to the **Data Warehouse** experience.

The Data Warehouse home page includes a shortcut to create a new warehouse:

**Note**: If you don’t see the an icon in the bottom left corner, Fabric isn’t enabled for your tenant. Fabric is enabled in the admin portal for tenants that have a Power BI Premium subscription.

1. In the **Data Warehouse** home page, create a new **Warehouse** with a name of your choice.

After a minute or so, a new warehouse will be created:

[](https://microsoftlearning.github.io/mslearn-fabric/Instructions/Labs/Images/new-data-warehouse.png)

Create tables and insert data

A warehouse is a relational database in which you can define tables and other objects.

1. In your new warehouse, select the **Create tables with T-SQL** tile, and replace the default SQL code with the following CREATE TABLE statement:

SqlCopy

CREATE TABLE dbo.DimProduct

(

ProductKey INTEGER NOT NULL,

ProductAltKey VARCHAR(25) NULL,

ProductName VARCHAR(50) NOT NULL,

Category VARCHAR(50) NULL,

ListPrice DECIMAL(5,2) NULL

);

GO

1. Use the **▷ Run** button to run the SQL script, which creates a new table named **DimProduct** in the **dbo** schema of the data warehouse.
2. Use the **Refresh** button on the toolbar to refresh the view. Then, in the **Explorer** pane, expand **Schemas** > **dbo** > **Tables** and verify that the **DimProduct** table has been created.
3. On the **Home** menu tab, use the **New SQL Query** button to create a new query, and enter the following INSERT statement:

SqlCopy

INSERT INTO dbo.DimProduct

VALUES

(1, 'RING1', 'Bicycle bell', 'Accessories', 5.99),

(2, 'BRITE1', 'Front light', 'Accessories', 15.49),

(3, 'BRITE2', 'Rear light', 'Accessories', 15.49);

GO

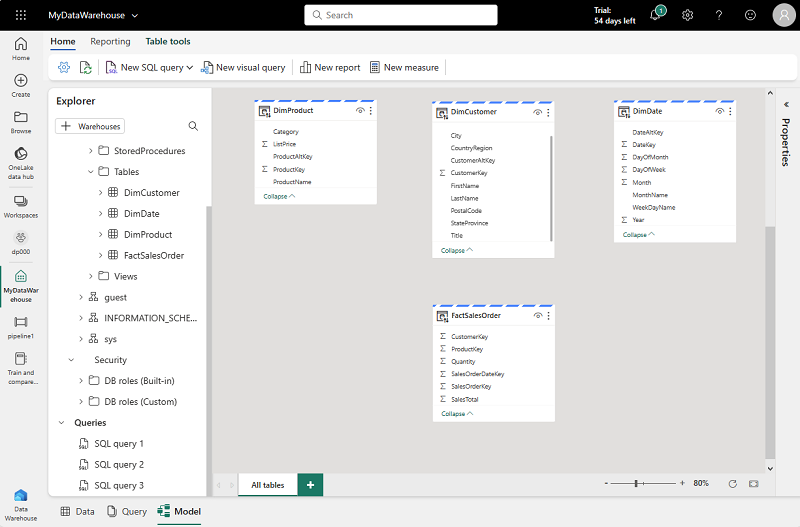
1. Run the new query to insert three rows into the **DimProduct** table.
2. When the query has finished, select the **Data** tab at the bottom of the page in the data warehouse. In the **Explorer** pane, select the **DimProduct** table and verify that the three rows have been added to the table.
3. On the **Home** menu tab, use the **New SQL Query** button to create a new query. Then copy and paste the Transact-SQL code from <https://raw.githubusercontent.com/MicrosoftLearning/dp-data/main/create-dw.txt> into the new query pane.
4. Run the query, which creates a simple data warehouse schema and loads some data. The script should take around 30 seconds to run.
5. Use the **Refresh** button on the toolbar to refresh the view. Then in the **Explorer** pane, verify that the **dbo** schema in the data warehouse now contains the following four tables:
   * **DimCustomer**
   * **DimDate**
   * **DimProduct**
   * **FactSalesOrder**

**Tip**: If the schema takes a while to load, just refresh the browser page.

Define a data model

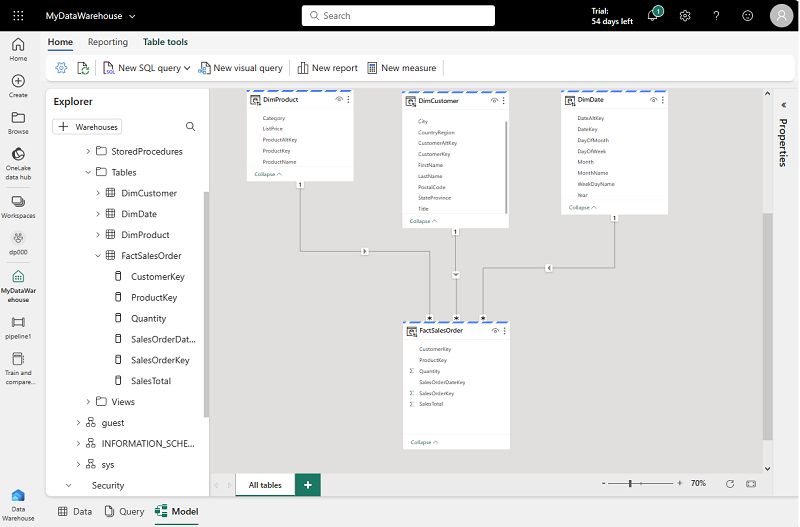
A relational data warehouse typically consists of *fact* and *dimension* tables. The fact tables contain numeric measures you can aggregate to analyze business performance (for example, sales revenue), and the dimension tables contain attributes of the entities by which you can aggregate the data (for example, product, customer, or time). In a Microsoft Fabric data warehouse, you can use these keys to define a data model that encapsulates the relationships between the tables.

1. At the bottom of the page in the data warehouse, select the **Model** tab.
2. In the model pane, rearrange the tables in your data warehouse so that the **FactSalesOrder** table is in the middle, like this:

[](https://microsoftlearning.github.io/mslearn-fabric/Instructions/Labs/Images/model-dw.png)

1. Drag the **ProductKey** field from the **FactSalesOrder** table and drop it on the **ProductKey** field in the **DimProduct** table. Then confirm the following relationship details:
   * **Table 1**: FactSalesOrder
   * **Column**: ProductKey
   * **Table 2**: DimProduct
   * **Column**: ProductKey
   * **Cardinality**: Many to one (\*:1)
   * **Cross filter direction**: Single
   * **Make this relationship active**: Selected
   * **Assume referential integrity**: Unselected
2. Repeat the process to create many-to-one relationships between the following tables:
   * **FactOrderSales.CustomerKey** → **DimCustomer.CustomerKey**
   * **FactOrderSales.SalesOrderDateKey** → **DimDate.DateKey**

When all of the relationships have been defined, the model should look like this:

[](https://microsoftlearning.github.io/mslearn-fabric/Instructions/Labs/Images/dw-relationships.png)

Query data warehouse tables

Since the data warehouse is a relational database, you can use SQL to query its tables.

Query fact and dimension tables

Most queries in a relational data warehouse involve aggregating and grouping data (using aggregate functions and GROUP BY clauses) across related tables (using JOIN clauses).

1. Create a new SQL Query, and run the following code:

SqlCopy

SELECT d.[Year] AS CalendarYear,

d.[Month] AS MonthOfYear,

d.MonthName AS MonthName,

SUM(so.SalesTotal) AS SalesRevenue

FROM FactSalesOrder AS so

JOIN DimDate AS d ON so.SalesOrderDateKey = d.DateKey

GROUP BY d.[Year], d.[Month], d.MonthName

ORDER BY CalendarYear, MonthOfYear;

Note that the attributes in the time dimension enable you to aggregate the measures in the fact table at multiple hierarchical levels - in this case, year and month. This is a common pattern in data warehouses.

1. Modify the query as follows to add a second dimension to the aggregation.

SqlCopy

SELECT d.[Year] AS CalendarYear,

d.[Month] AS MonthOfYear,

d.MonthName AS MonthName,

c.CountryRegion AS SalesRegion,

SUM(so.SalesTotal) AS SalesRevenue

FROM FactSalesOrder AS so

JOIN DimDate AS d ON so.SalesOrderDateKey = d.DateKey

JOIN DimCustomer AS c ON so.CustomerKey = c.CustomerKey

GROUP BY d.[Year], d.[Month], d.MonthName, c.CountryRegion

ORDER BY CalendarYear, MonthOfYear, SalesRegion;

1. Run the modified query and review the results, which now include sales revenue aggregated by year, month, and sales region.

Create a view

A data warehouse in Microsoft Fabric has many of the same capabilities you may be used to in relational databases. For example, you can create database objects like *views* and *stored procedures* to encapsulate SQL logic.

1. Modify the query you created previously as follows to create a view (note that you need to remove the ORDER BY clause to create a view).

SqlCopy

CREATE VIEW vSalesByRegion

AS

SELECT d.[Year] AS CalendarYear,

d.[Month] AS MonthOfYear,

d.MonthName AS MonthName,

c.CountryRegion AS SalesRegion,

SUM(so.SalesTotal) AS SalesRevenue

FROM FactSalesOrder AS so

JOIN DimDate AS d ON so.SalesOrderDateKey = d.DateKey

JOIN DimCustomer AS c ON so.CustomerKey = c.CustomerKey

GROUP BY d.[Year], d.[Month], d.MonthName, c.CountryRegion;

1. Run the query to create the view. Then refresh the data warehouse schema and verify that the new view is listed in the **Explorer** pane.
2. Create a new SQL query and run the following SELECT statement:

CodeCopy

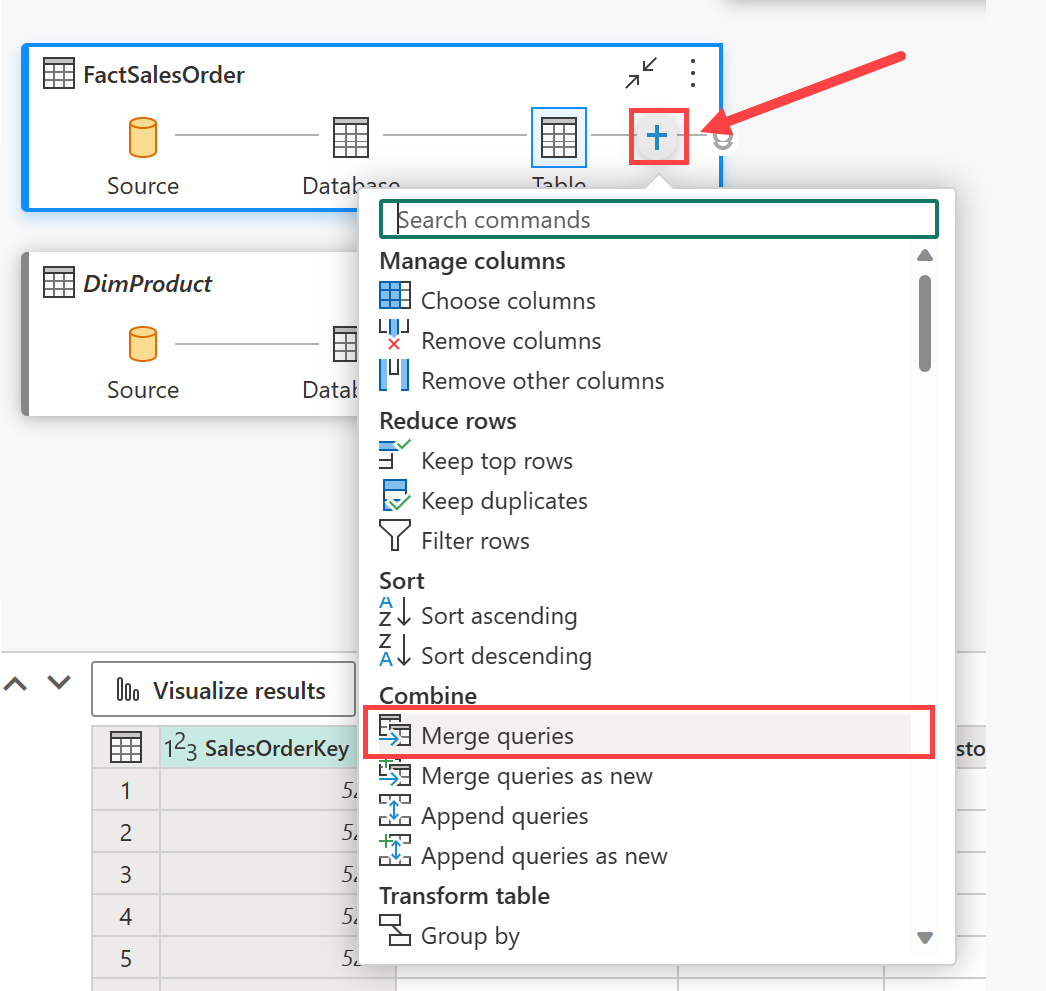
SELECT CalendarYear, MonthName, SalesRegion, SalesRevenue

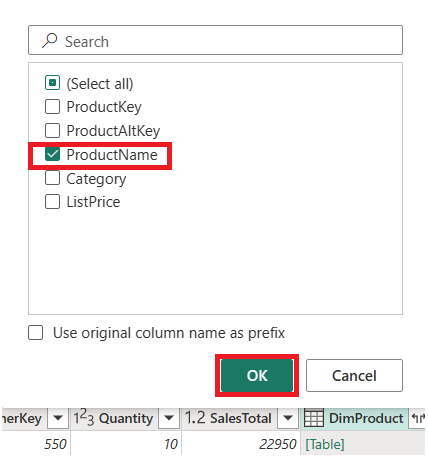
FROM vSalesByRegion

ORDER BY CalendarYear, MonthOfYear, SalesRegion;

Create a visual query

Instead of writing SQL code, you can use the graphical query designer to query the tables in your data warehouse. This experience is similar to Power Query online, where you can create data transformation steps with no code. For more complex tasks, you can use Power Query’s M (Mashup) language.

1. On the **Home** menu, select **New visual query**.
2. Drag **FactSalesOrder** onto the **canvas**. Notice that a preview of the table is displayed in the **Preview** pane below.
3. Drag **DimProduct** onto the **canvas**. We now have two tables in our query.
4. Use the **(+)** button on the **FactSalesOrder** table on the canvas to **Merge queries**. [](https://microsoftlearning.github.io/mslearn-fabric/Instructions/Labs/Images/visual-query-merge.png)
5. In the **Merge queries** window, select **DimProduct** as the right table for merge. Select **ProductKey** in both queries, leave the default **Left outer** join type, and click **OK**.
6. In the **Preview**, note that the new **DimProduct** column has been added to the FactSalesOrder table. Expand the column by clicking the arrow to the right of the column name. Select **ProductName** and click **OK**.

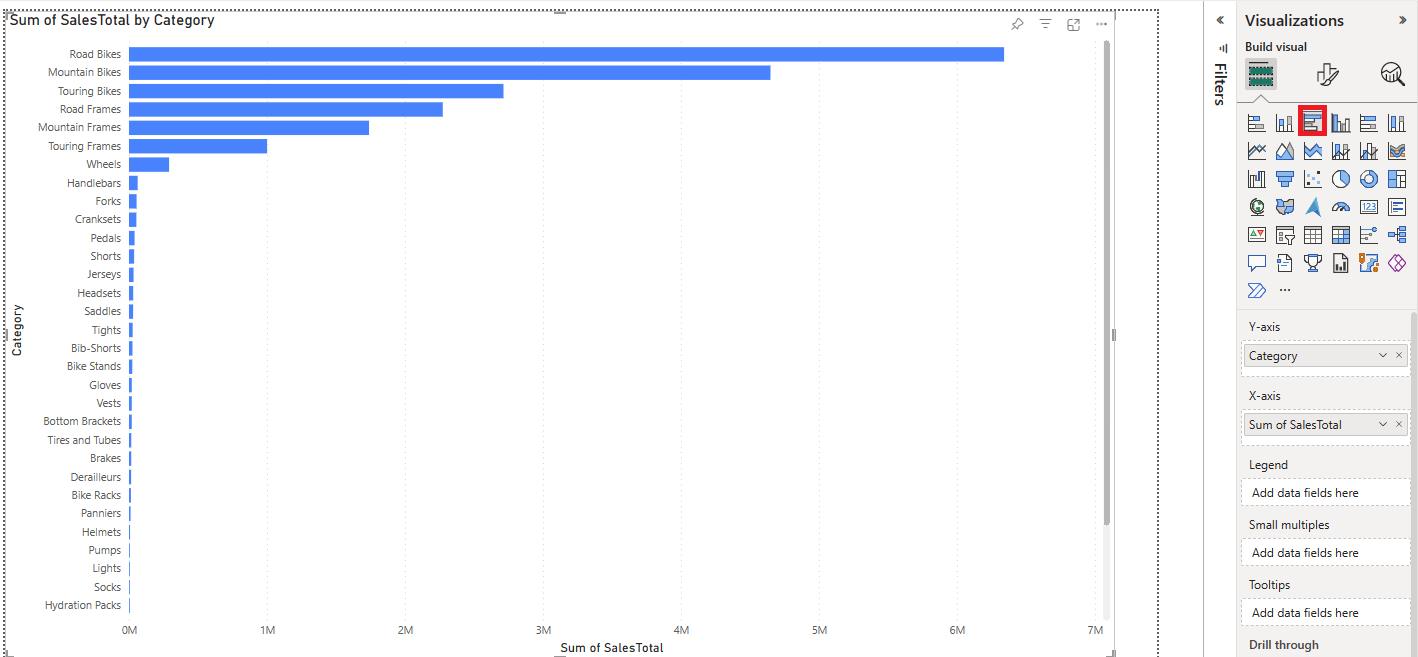
[](https://microsoftlearning.github.io/mslearn-fabric/Instructions/Labs/Images/visual-query-preview.png)

1. If you’re interested in looking at data for a single product, per a manager request, you can now use the **ProductName** column to filter the data in the query. Filter the **ProductName** column to look at **Cable Lock** data only.
2. From here, you can analyze the results of this single query by selecting **Visualize results** or **Open in Excel**. You can now see exactly what the manager was asking for, so we don’t need to analyze the results further.

Visualize your data

You can easily visualize the data in either a single query, or in your data warehouse. Before you visualize, hide columns and/or tables that aren’t friendly to report designers.

1. In the **Explorer** pane, select the **Model** view.
2. Hide the following columns in your Fact and Dimension tables that are not necessary to create a report. Note that this does not remove the columns from the model, it simply hides them from view on the report canvas.
   1. FactSalesOrder
      * **SalesOrderDateKey**
      * **CustomerKey**
      * **ProductKey**
   2. DimCustomer
      * **CustomerKey**
      * **CustomerAltKey**
   3. DimDate
      * **DateKey**
      * **DateAltKey**
   4. DimProduct
      * **ProductKey**
      * **ProductAltKey**
3. Now you’re ready to build a report and make this dataset available to others. On the Home menu, select **New report**. This will open a new window, where you can create a Power BI report.
4. In the **Data** pane, expand **FactSalesOrder**. Note that the columns you hid are no longer visible.
5. Select **SalesTotal**. This will add the column to the **Report canvas**. Because the column is a numeric value, the default visual is a **column chart**.
6. Ensure that the column chart on the canvas is active (with a gray border and handles), and then select **Category** from the **DimProduct** table to add a category to your column chart.
7. In the **Visualizations** pane, change the chart type from a column chart to a **clustered bar chart**. Then resize the chart as necessary to ensure that the categories are readable.

[](https://microsoftlearning.github.io/mslearn-fabric/Instructions/Labs/Images/visualizations-pane.png)

1. In the **Visualizations** pane, select the **Format your visual** tab and in the **General** sub-tab, in the **Title** section, change the **Text** to **Total Sales by Category**.
2. In the **File** menu, select **Save**. Then save the report as **Sales Report** in the workspace you created previously.
3. In the menu hub on the left, navigate back to the workspace. Notice that you now have three items saved in your workspace: your data warehouse, its default dataset, and the report you created.

[](https://microsoftlearning.github.io/mslearn-fabric/Instructions/Labs/Images/workspace-items.png)

Clean up resources

In this exercise, you have created a data warehouse that contains multiple tables. You used SQL to insert data into the tables and query them. and also used the visual query tool. Finally, you enhanced the data model for the data warehouse’s default dataset and used it as the source for a report.

If you’ve finished exploring your data warehouse, you can delete the workspace you created for this exercise.

1. In the bar on the left, select the icon for your workspace to view all of the items it contains.
2. In the **…** menu on the toolbar, select **Workspace settings**.
3. In the **Other** section, select **Remove this workspace**.

Table cloning

Ci/cd

**Create a table clone within the same schema in a warehouse**

1. In the Fabric portal, from the ribbon, select **New SQL query**.



1. In the query editor, paste the following code to create clones of the dbo.dimension\_city and dbo.fact\_sale tables.

SQLCopy

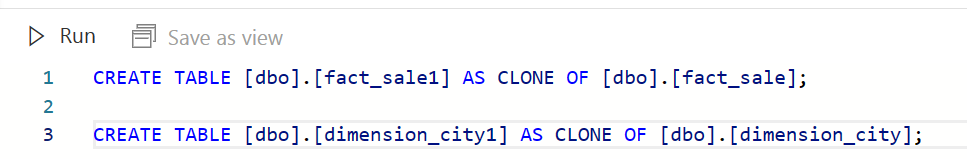
--Create a clone of the dbo.dimension\_city table.

CREATE TABLE [dbo].[dimension\_city1] AS CLONE OF [dbo].[dimension\_city];

--Create a clone of the dbo.fact\_sale table.

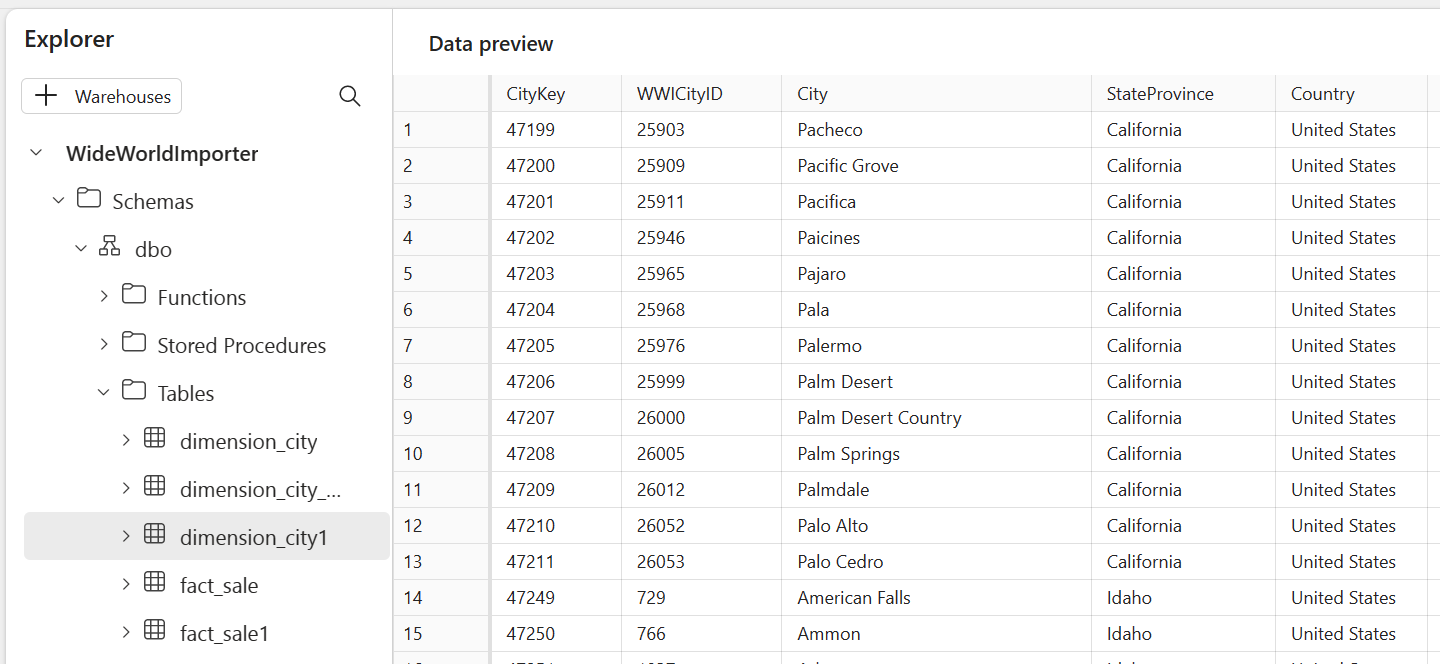
CREATE TABLE [dbo].[fact\_sale1] AS CLONE OF [dbo].[fact\_sale];

1. Select **Run** to execute the query. The query takes a few seconds to execute.

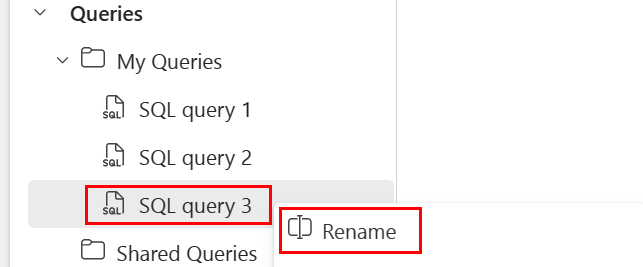


After the query is completed, the table clones dimension\_city1 and fact\_sale1 have been created.

1. Load the data preview to validate the data loaded successfully by selecting on the dimension\_city1 table in the **Explorer**.



1. Rename the query for reference later. Right-click on **SQL query 3** in the **Explorer** and select **Rename**.



1. Type Clone Table to change the name of the query.
2. Press **Enter** on the keyboard or select anywhere outside the tab to save the change.

**Create a table clone across schemas within the same warehouse**

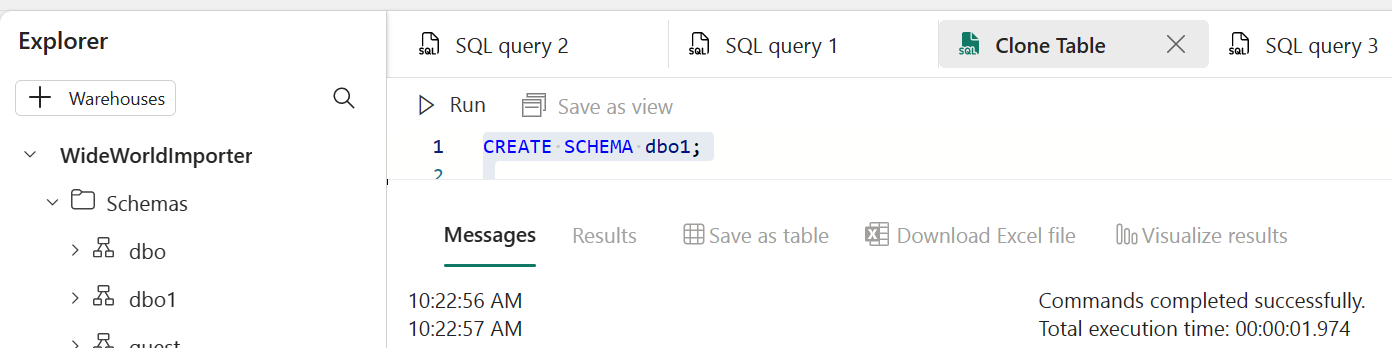
1. From the ribbon, select **New SQL query**.



1. Create a new schema within the WideWorldImporter warehouse named dbo1. Copy, paste, and run the following T-SQL code:

SQLCopy

CREATE SCHEMA dbo1;



1. In the query editor, paste the following code to create clones of the dbo.dimension\_city and dbo.fact\_sale tables in the dbo1 schema.

SQLCopy

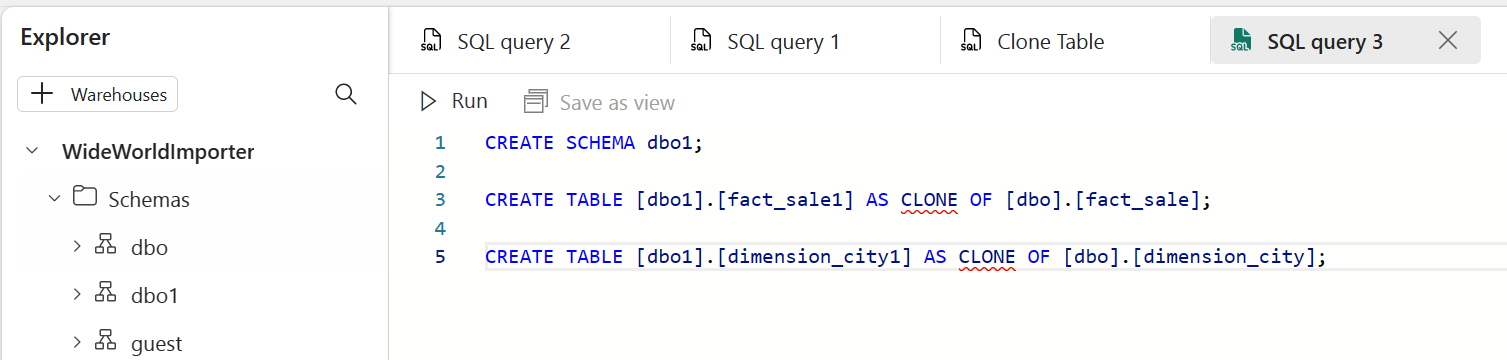
--Create a clone of the dbo.dimension\_city table in the dbo1 schema.

CREATE TABLE [dbo1].[dimension\_city1] AS CLONE OF [dbo].[dimension\_city];

--Create a clone of the dbo.fact\_sale table in the dbo1 schema.

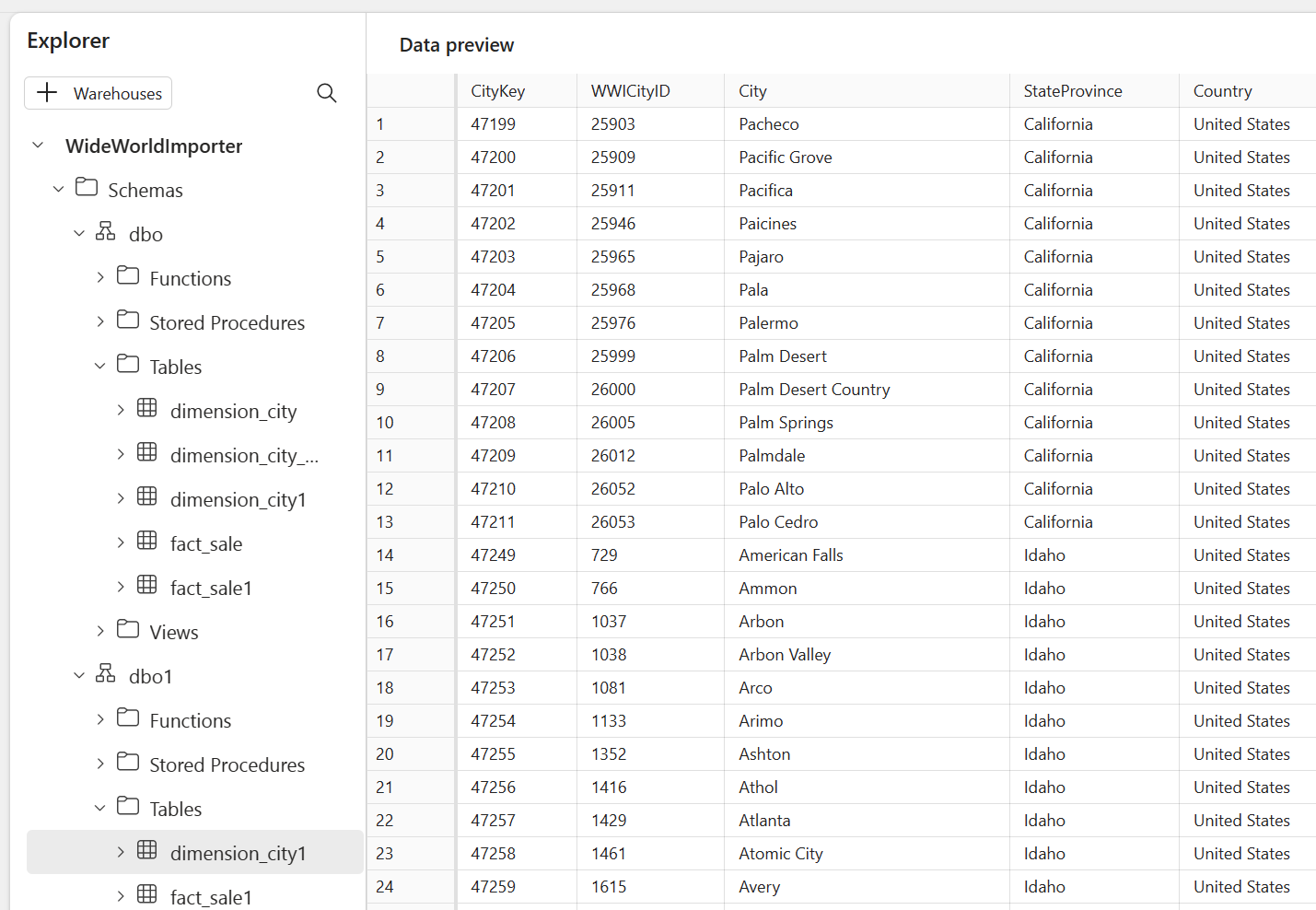
CREATE TABLE [dbo1].[fact\_sale1] AS CLONE OF [dbo].[fact\_sale];

1. Select **Run** to execute the query. The query takes a few seconds to execute.

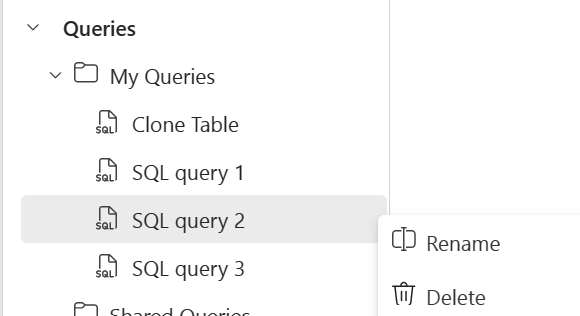


After the query is completed, clones dimension\_city1 and fact\_sale1 are created in the dbo1 schema.

1. Load the data preview to validate the data loaded successfully by selecting on the dimension\_city1 table under dbo1 schema in the **Explorer**.

[](https://learn.microsoft.com/en-us/fabric/data-warehouse/media/tutorial-clone-table/explorer-select-table-schema.png#lightbox)

1. Rename the query for reference later. Right-click on **SQL query 2** in the **Explorer** and select **Rename**.



1. Type Clone Table in another schema to change the name of the query.
2. Press **Enter** on the keyboard or select anywhere outside the tab to save the change.

**Next steps**

# Query the SQL Endpoint or Warehouse in Microsoft Fabric

[Query the SQL Endpoint or Warehouse - Microsoft Fabric | Microsoft Learn](https://learn.microsoft.com/en-us/fabric/data-warehouse/query-warehouse)

[Data warehouse tutorial - analyze data with a notebook - Microsoft Fabric | Microsoft Learn](https://learn.microsoft.com/en-us/fabric/data-warehouse/tutorial-analyze-data-notebook)

# Organize gold, bronze and gold

[Metadata Driven Pipelines for Microsoft Fabric – Part 2, Data Warehouse Style - Microsoft Community Hub](https://techcommunity.microsoft.com/t5/fasttrack-for-azure/metadata-driven-pipelines-for-microsoft-fabric-part-2-data/ba-p/3906749#:~:text=from%20delta.tables%20import%20%2A%20from%20pyspark.sql.functions%20import%20%2A,%3D%20%7B%22maxdate%22%3A%20maxdate_str%2C%20%22numInserted%22%3A%20numInserted%2C%20%22numUpdated%22%3A%20numUpdated%7D%20mssparkutils.notebook.exit%28str%28result%29%29)

# CI CD tutorial Fabric

# Tutorial: Lifecycle management in Fabric

* Article
* 08/02/2023
* 1 contributor

Feedback

## In this article

1. [Prerequisites](https://learn.microsoft.com/en-us/fabric/cicd/cicd-tutorial#prerequisites)
2. [Step 1: Create a Premium workspace](https://learn.microsoft.com/en-us/fabric/cicd/cicd-tutorial#step-1-create-a-premium-workspace)
3. [Step 2: Load content into the workspace](https://learn.microsoft.com/en-us/fabric/cicd/cicd-tutorial#step-2-load-content-into-the-workspace)
4. [Step 3: Connect the team's development workspace to git](https://learn.microsoft.com/en-us/fabric/cicd/cicd-tutorial#step-3-connect-the-teams-development-workspace-to-git)

Show 11 more

This tutorial takes you through the whole process of loading data into your workspace, and using deployment pipelines together with Git integration to collaborate with others in the development, testing, and publication of your data and reports.

## Prerequisites

Before you start, make sure of the following:

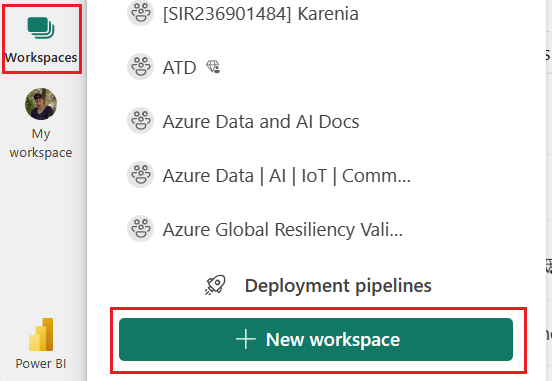
* If you don't have Fabric enabled yet, ask your admin to [enable Fabric for your organization](https://learn.microsoft.com/en-us/fabric/admin/fabric-switch).
* If you aren't signed up yet, [sign up for a free trial](https://learn.microsoft.com/en-us/fabric/get-started/fabric-trial).
* Access to an Azure Git repo. If you don't have one, see [Set up a Git repository](https://learn.microsoft.com/en-us/devops/develop/git/set-up-a-git-repository) for information on creating one.
* Download the [FoodSales.pbix](https://github.com/microsoft/fabric-samples/blob/main/docs-samples/cicd/FoodSales.pbix) file into a Git repo that you can edit. This is the sample file we use in this tutorial. Alternatively, you can use your own dataset and report, if you prefer.

If you already have admin rights to a workspace with data, you can skip to [step 3](https://learn.microsoft.com/en-us/fabric/cicd/cicd-tutorial#step-3-connect-the-teams-development-workspace-to-git).

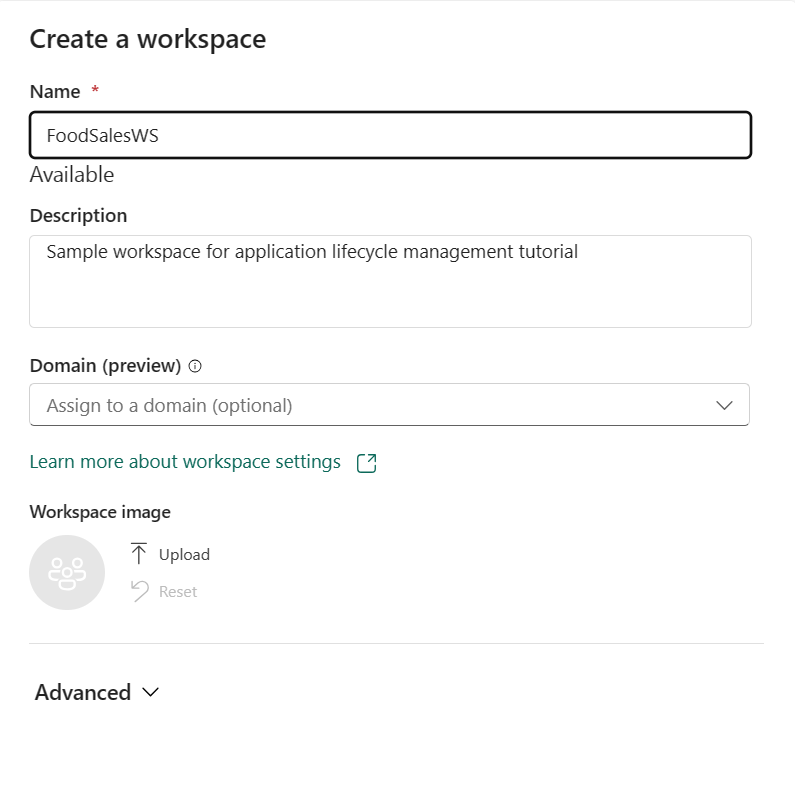
## Step 1: Create a Premium workspace

To create a new workspace and assign it a license:

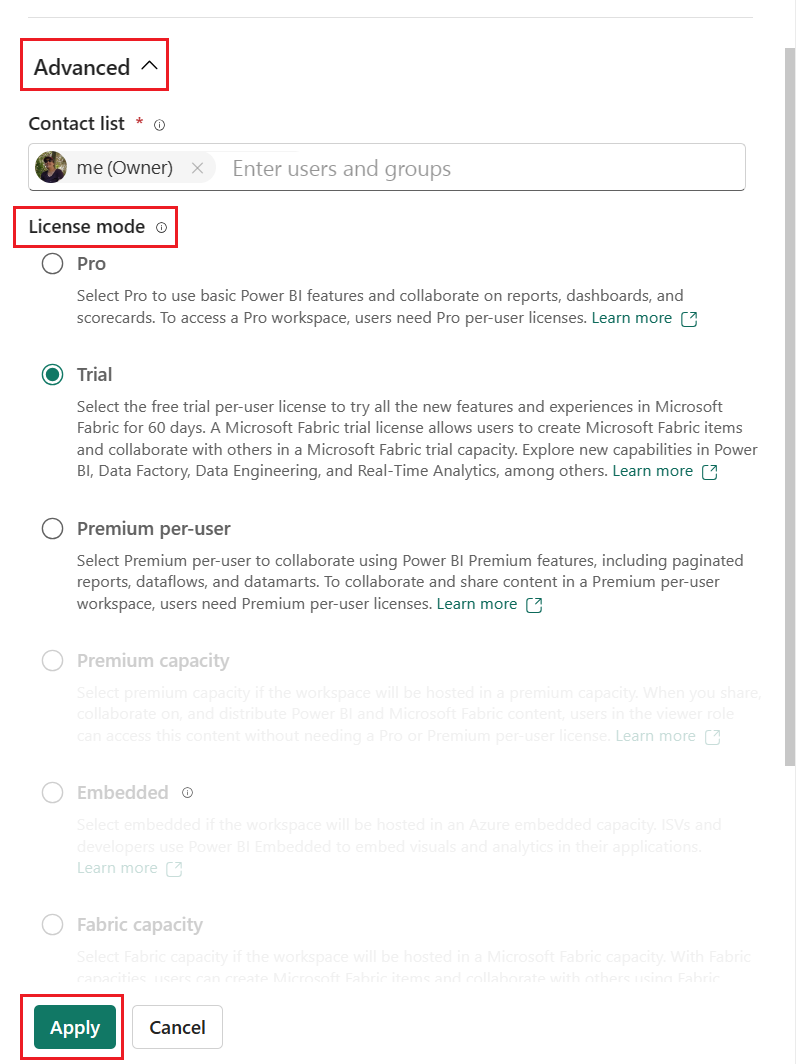
1. From the left navigation bar, select **Workspaces > + New workspace**.



1. Name the workspace **FoodSalesWS**.
2. (Optional) Add a description.



1. Expand the **Advanced** section to reveal **License mode**.
2. Select either **Trial** or **Premium capacity**.



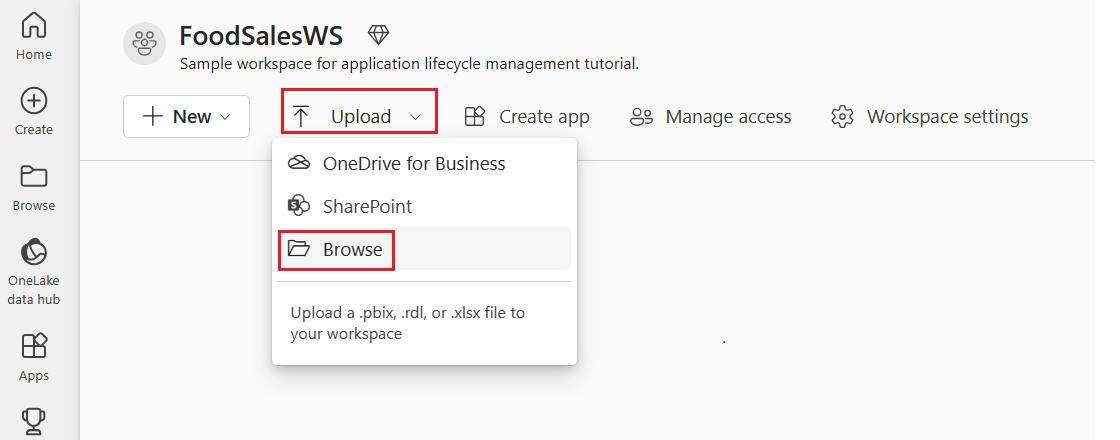
1. Select **Apply**.

For more on creating a workspace, see [Create a workspace](https://learn.microsoft.com/en-us/power-bi/collaborate-share/service-create-the-new-workspaces).

## Step 2: Load content into the workspace

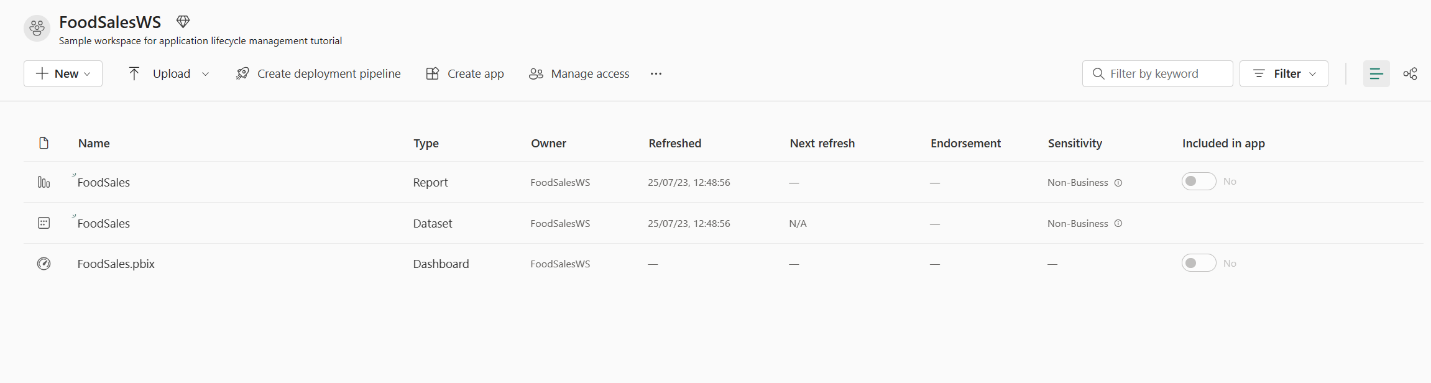
You can upload content from OneDrive, SharePoint, or a local file. In this tutorial, we load a .pbix file.

1. From the top menu bar, select **Upload > Browse**.



1. Browse to the location of the **FoodSales.pbix** file you [downloaded earlier](https://learn.microsoft.com/en-us/fabric/cicd/cicd-tutorial#prerequisites), or load your own sample dataset and report.

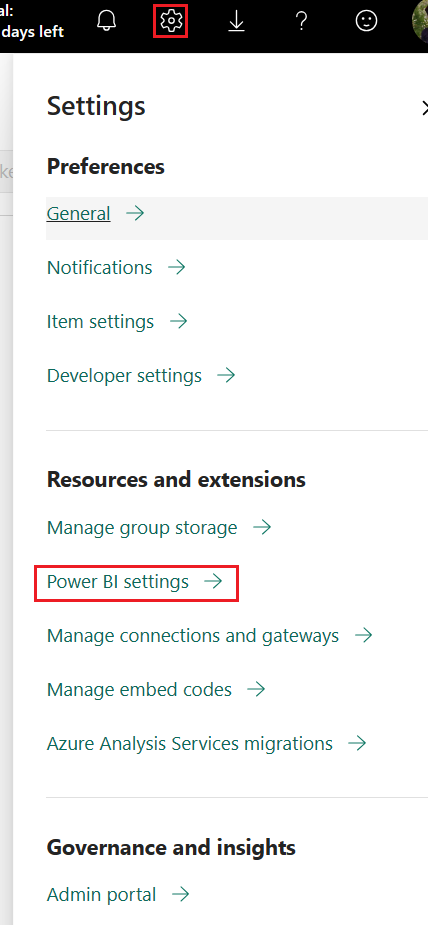
You now have a workspace with content in it for you and your team to work on.



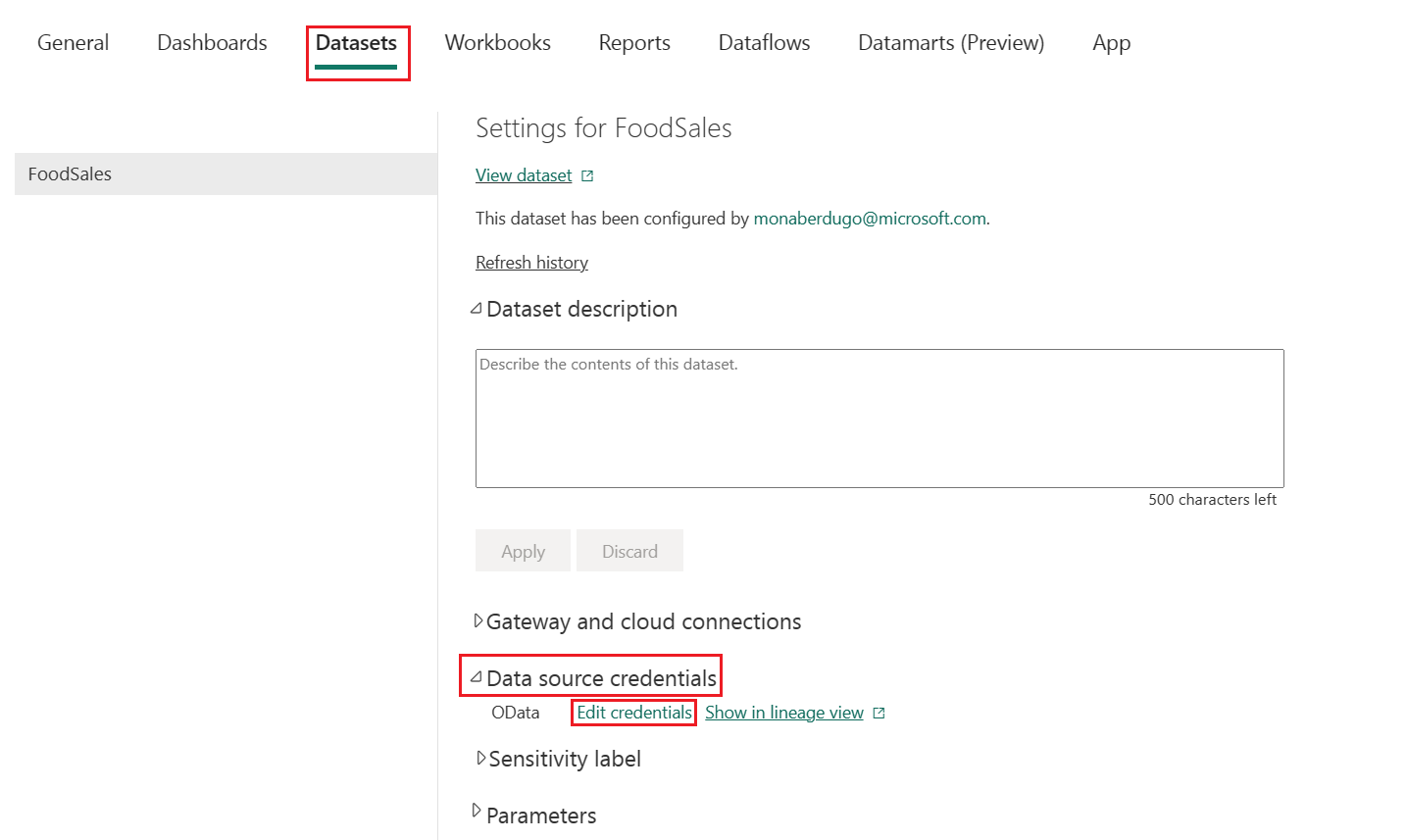
### Edit credentials - first time only

Before you create a deployment pipeline, you need to set the credentials. This step only needs to be done once. After your credentials are set, you won't have to set them again.

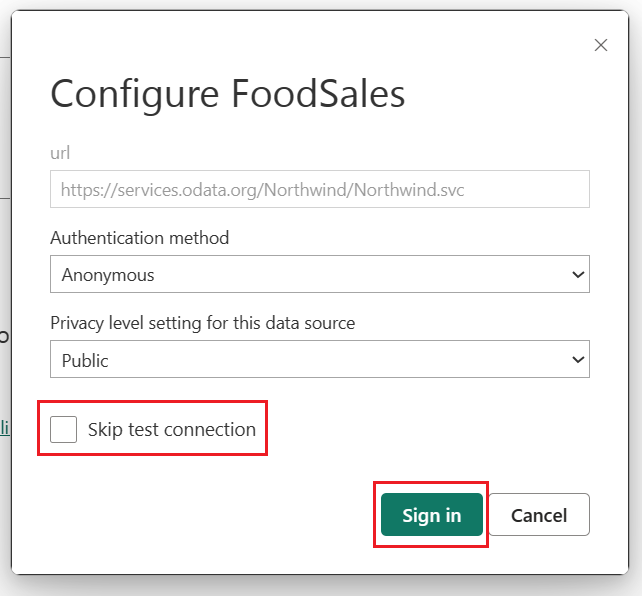
1. Go to **Settings > Power BI settings**.



1. Select **Datasets > Data source credentials > Edit credentials**.



1. Set the **Authentication** method to Anonymous, the **Privacy level** to Public, and uncheck the **Skip test connection** box.



1. Select **Sign in**. The connection is tested and credentials set. This is a one time action. You won't have to edit the credentials again for this dataset.

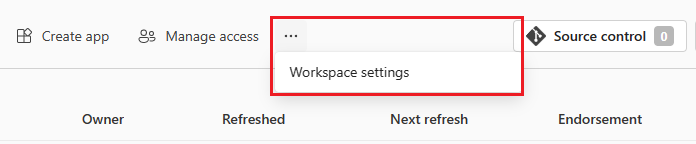
You can now create a deployment pipeline.

## Step 3: Connect the team's development workspace to git

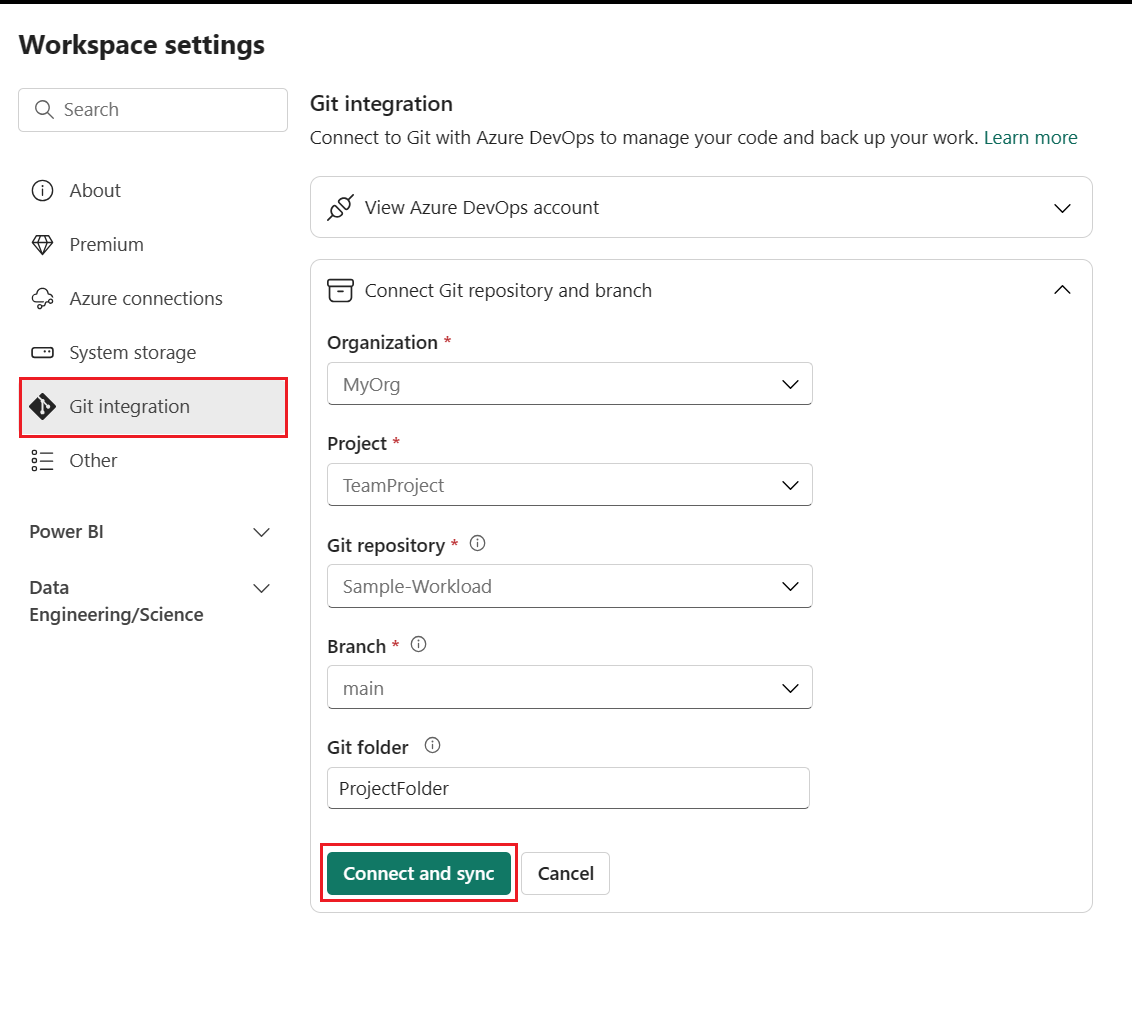
This workspace is shared by the entire team and each member of the team can edit it. By connecting this workspace to git, you can keep track of all the changes and revert back to previous versions if necessary. When all the changes are merged into this shared branch, the workspace is deployed to production using the deployment pipeline.  
Read more about version control with Git in [Introduction to Git integration](https://learn.microsoft.com/en-us/fabric/cicd/git-integration/intro-to-git-integration).

Let's connect this workspace to the main branch of your Azure repo so all team members can edit it and create pull requests.

1. Select the ellipsis (three dots) then **Workspace settings**.

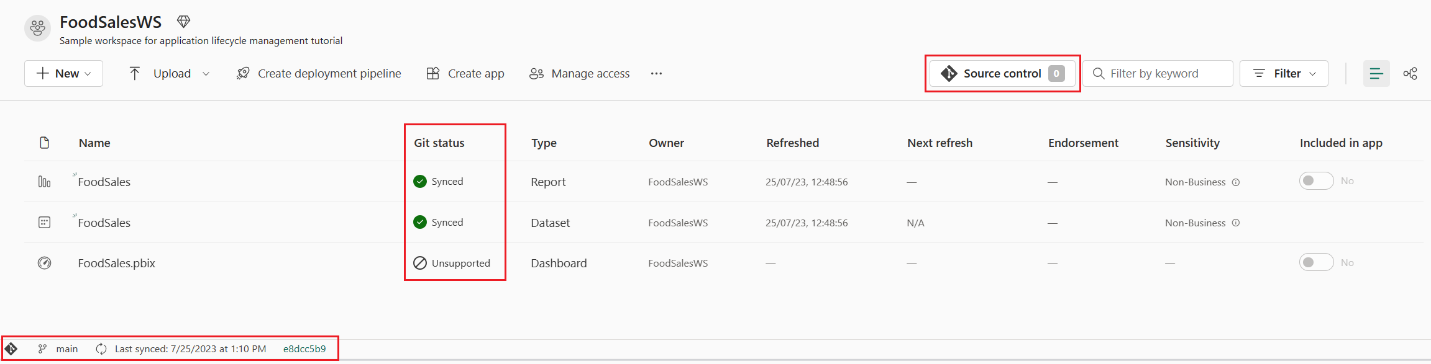


1. Select **Git integration**. You’re automatically signed into the Azure Repos account registered to the Azure AD user signed into the workspace.
2. From the dropdown menu, specify the following details about the branch you want to connect to:
   * [Organization](https://learn.microsoft.com/en-us/azure/devops/user-guide/plan-your-azure-devops-org-structure)
   * [Project](https://learn.microsoft.com/en-us/azure/devops/user-guide/plan-your-azure-devops-org-structure#how-many-projects-do-you-need)
   * [Git repository](https://learn.microsoft.com/en-us/azure/devops/user-guide/plan-your-azure-devops-org-structure#structure-repos-and-version-control-within-a-project)
   * Select main (or master) branch.
   * Type the name of folder in the repo where the .pbix file located. This is the folder that will be synced with the workspace.



1. Select **Connect and sync**.

After you connect, the Workspace displays information about source control that allows you to view the connected branch, the status of each item in the branch and the time of the last sync. The Source control icon shows 0 because the items in the workspace Git repo are identical.



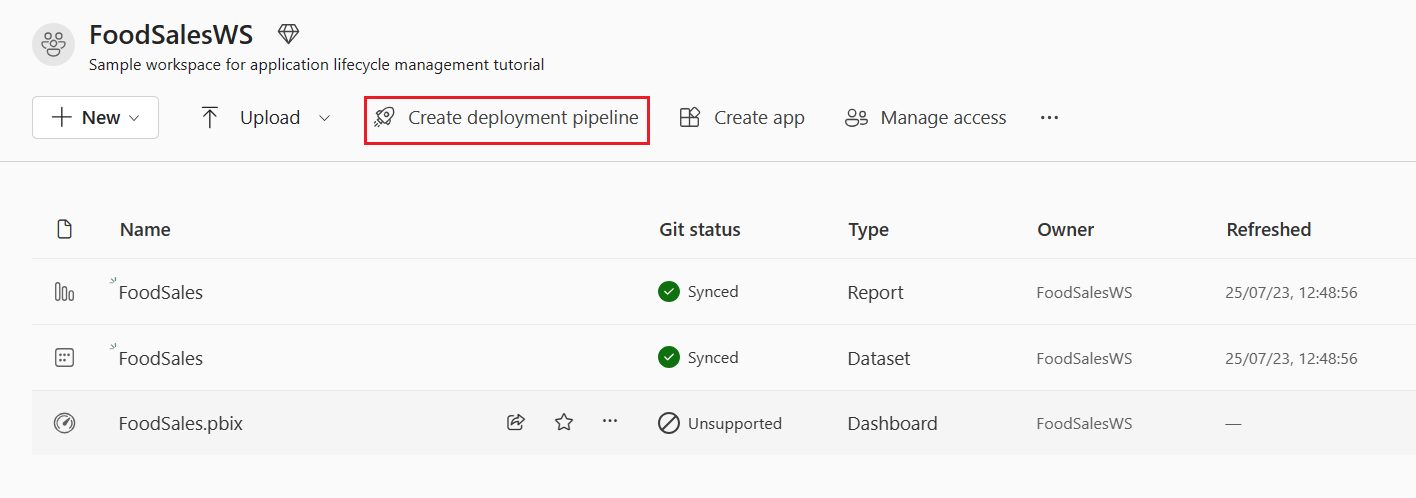
Now the workspace is synced with the main branch of your Git repo making it easy to keep track of changes.

For more information about connecting to git, see [Connect a workspace to an Azure repo](https://learn.microsoft.com/en-us/fabric/cicd/git-integration/git-get-started#connect-a-workspace-to-an-azure-repo).

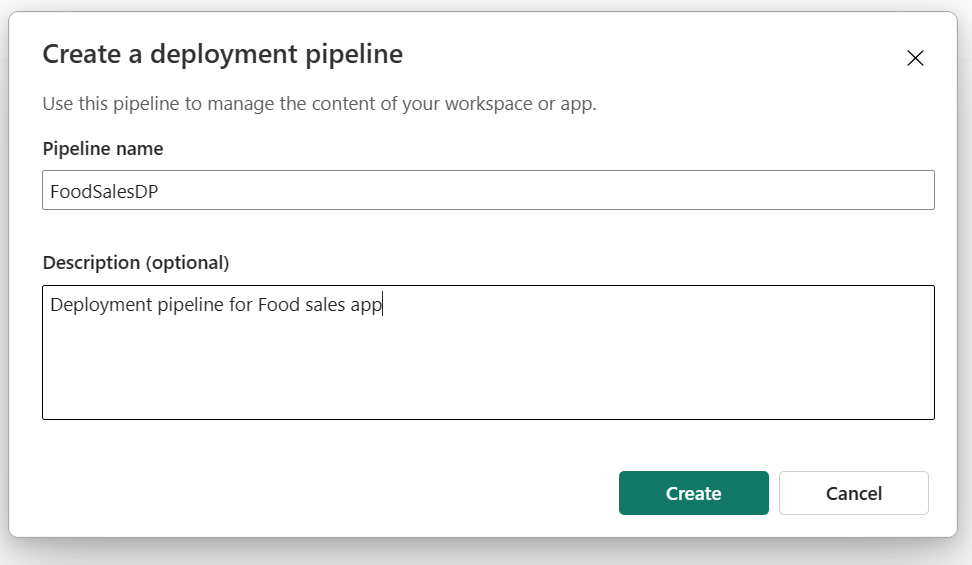
## Step 4: Create a deployment pipeline

In order to share this workspace with others and use it for various stages of testing and development, we need to create a deployment pipeline. You can read about how deployment pipelines work in [Introduction to deployment pipelines](https://learn.microsoft.com/en-us/fabric/cicd/deployment-pipelines/intro-to-deployment-pipelines). To create a deployment pipeline and assign the workspace to the development stage, do the following:

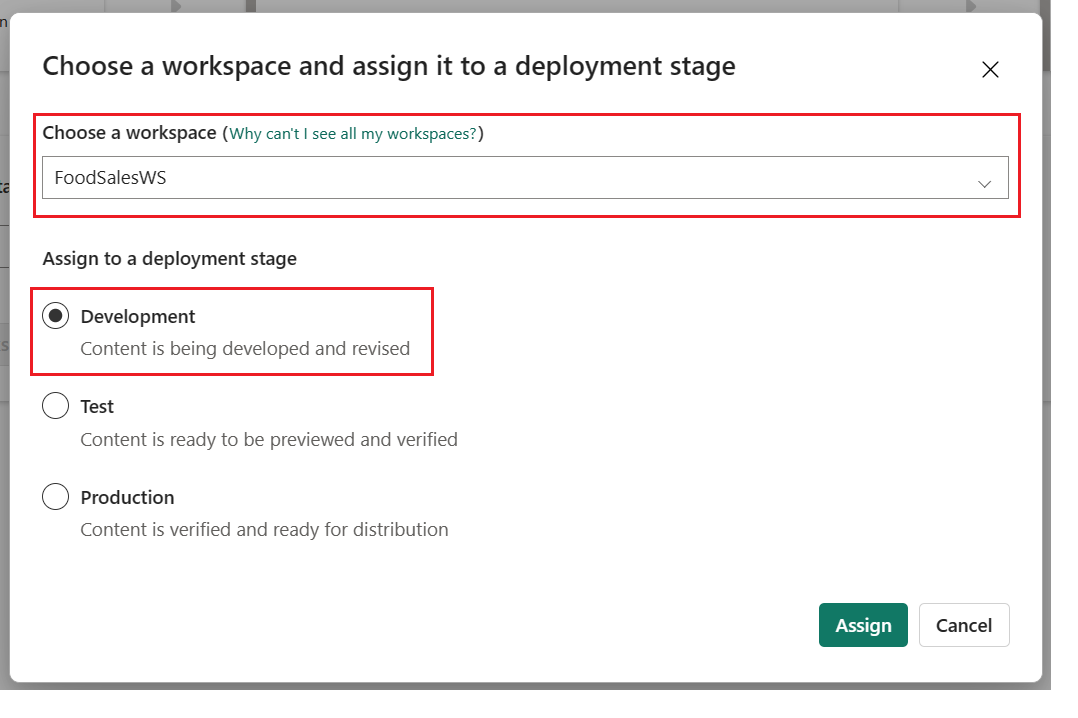
1. From the workspace home page, select **Create deployment pipeline**.



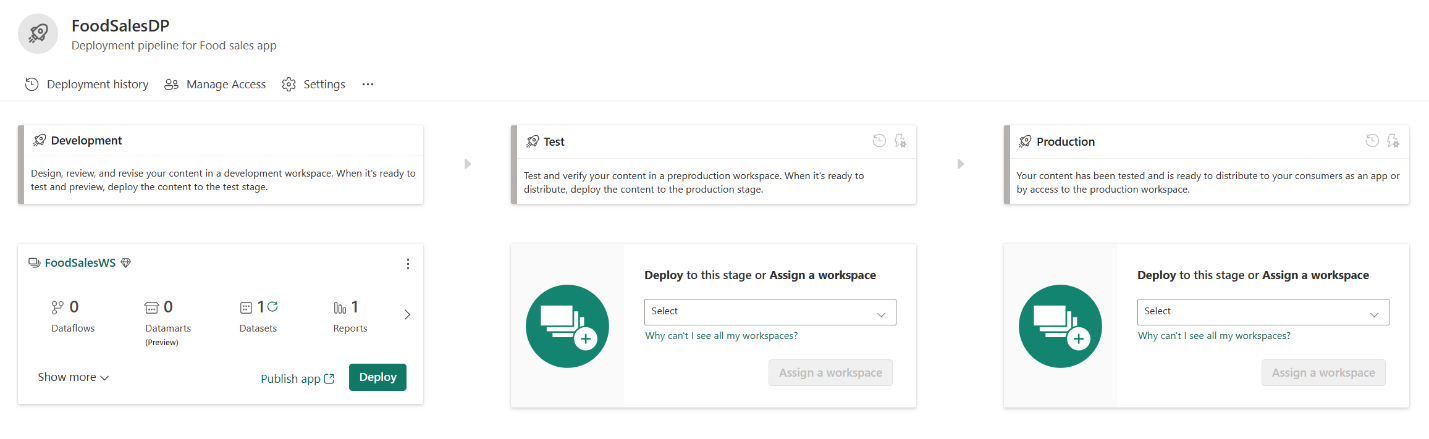
1. Name your pipeline FoodSalesDP, give it a description (optional) and select **Create**.



1. Assign the FoodSalesWS workspace to the Development stage.



The development stage of the deployment pipeline shows one dataset, one report, and one dashboard. The other stages are empty.

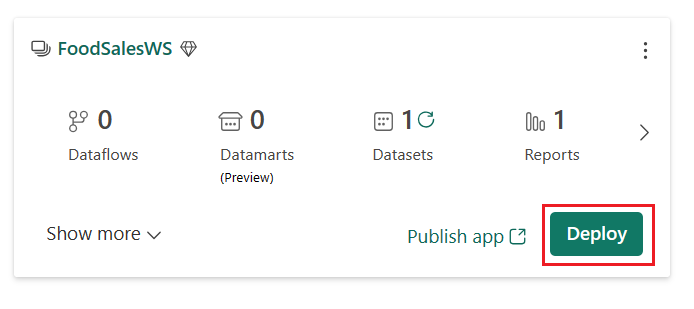


You can read more about creating deployment pipelines in [Deployment pipelines overview](https://learn.microsoft.com/en-us/fabric/cicd/deployment-pipelines/assign-pipeline).

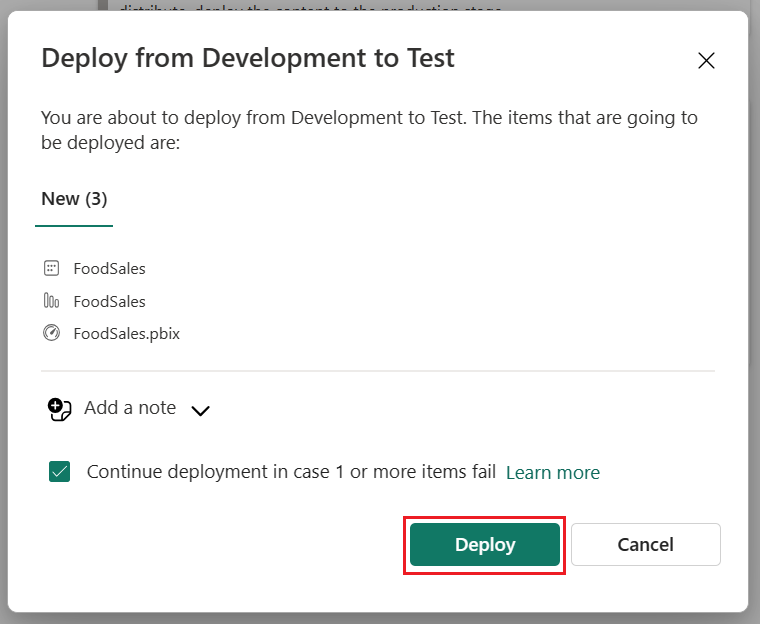
## Step 5: Deploy content to other stages

Now, deploy the content to the other stages of the pipeline.

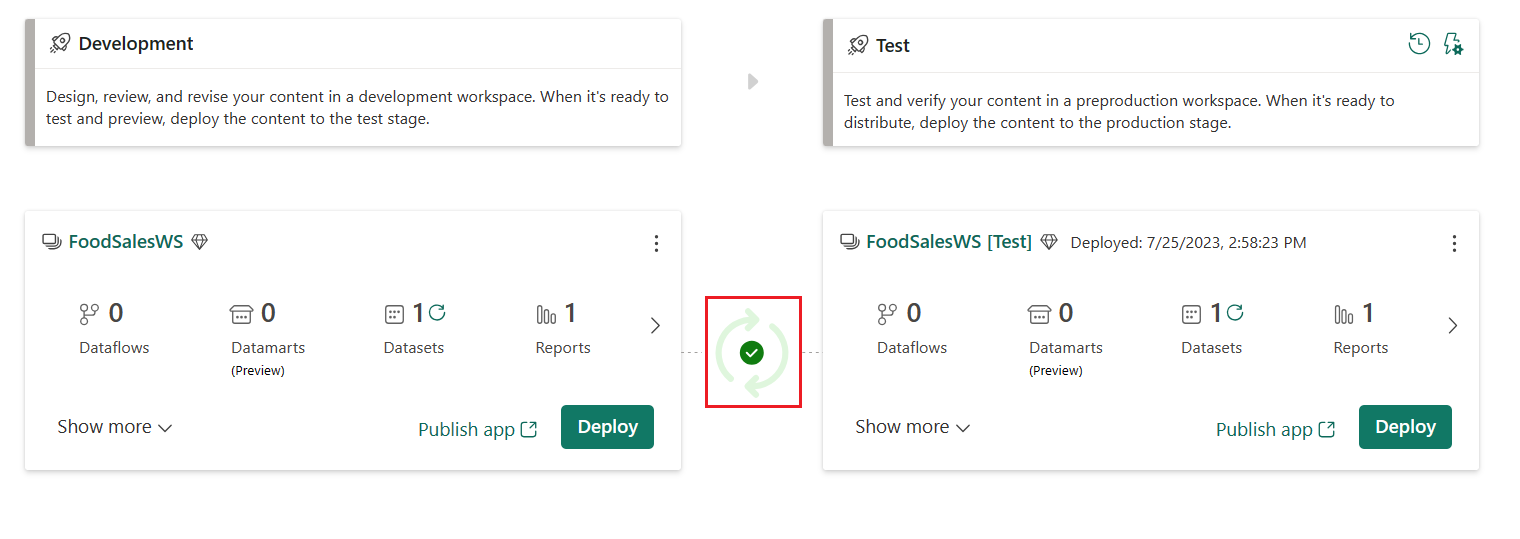
1. From the development stage of the deployment content view, select **Deploy**.



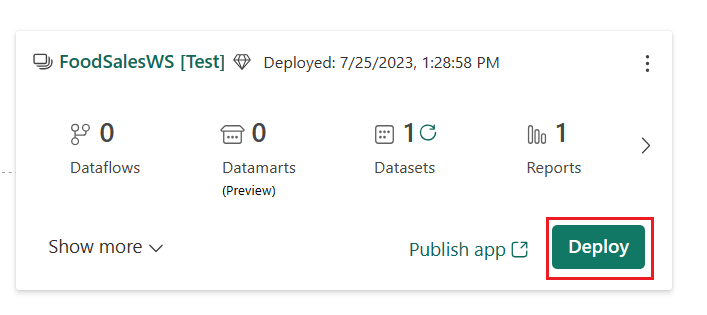
1. Confirm that you want to deploy the content to the test stage.



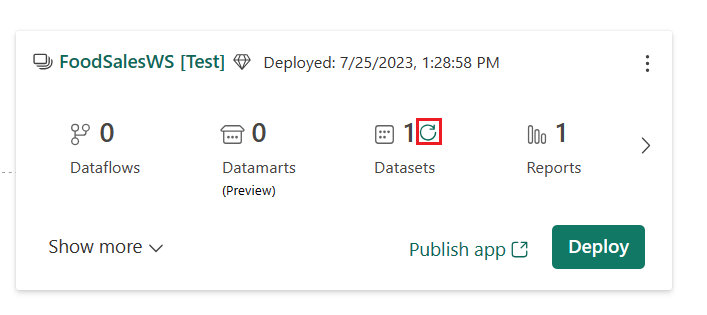
Notice the content of two stages are identical, since you deployed the entire content of the pipeline. This is indicated by the green check icon.



1. Deploy the content from the test stage to the production stage.



1. To refresh the dataset in any stage, select the refresh button next to the datasets icon in the summary card of each stage.



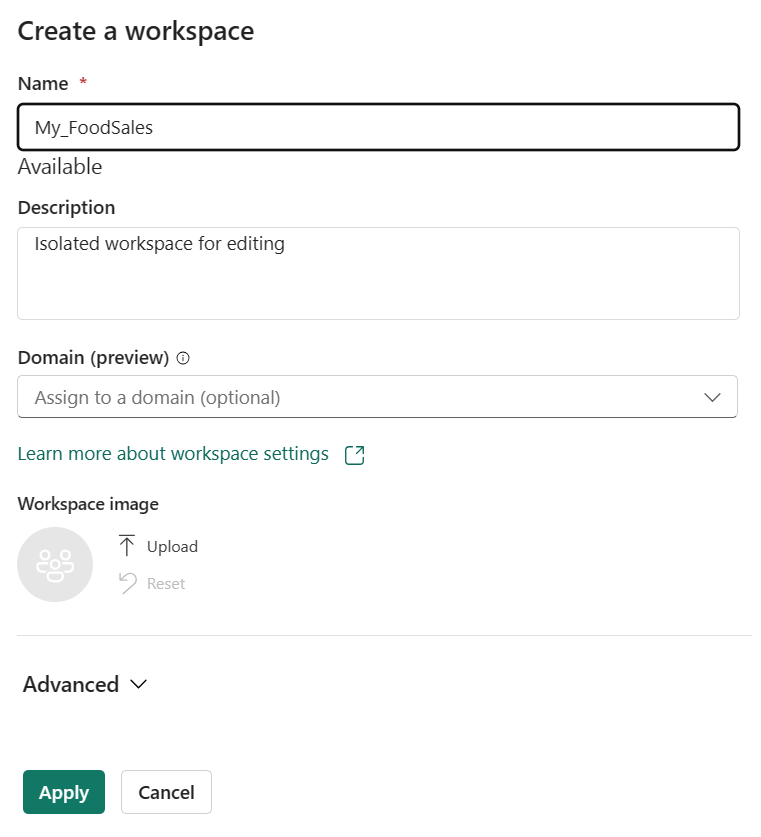
This deployment pipeline is shared by the entire team. Each team member can edit the dataset and report in the development stage. When the team is ready to test the changes, they deploy the content to the test stage. When the team is ready to release the changes to production, they deploy the content to the production stage.

For more information on deploying content, see [Deploy content](https://learn.microsoft.com/en-us/fabric/cicd/deployment-pipelines/deploy-content).

## Step 6: Create an isolated workspace

In order to edit the workspace without interfering with other team members' changes, each team member creates their own isolated workspace to work in until they're ready to share their changes with the team.

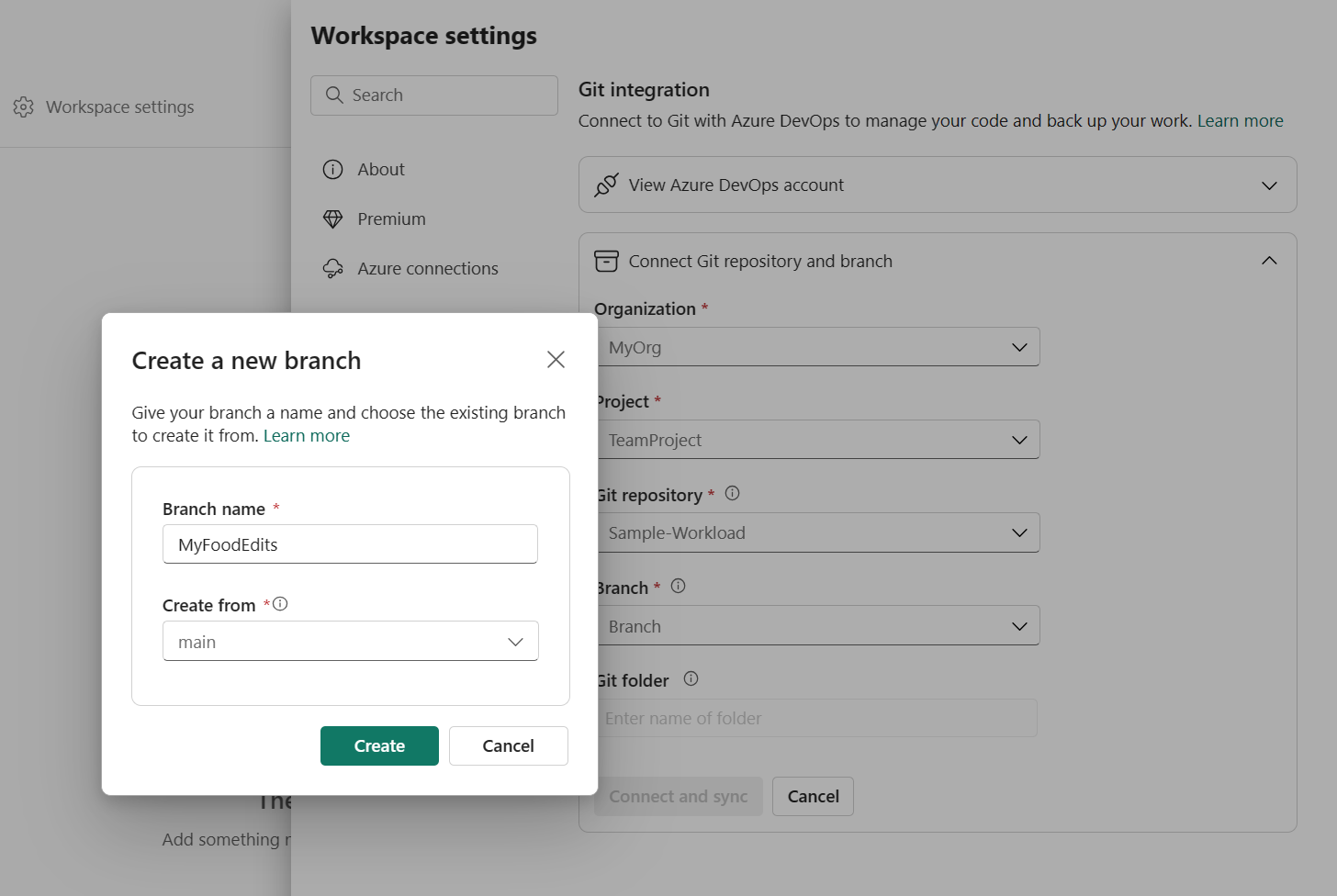
1. Create a new workspace like you did in [Step 1](https://learn.microsoft.com/en-us/fabric/cicd/cicd-tutorial#step-1-create-a-premium-workspace).



1. Connect this new workspace to a new branch of the Git repo:

From the dropdown menu, specify the following details about the branch you want to connect to:

* + [Organization](https://learn.microsoft.com/en-us/azure/devops/user-guide/plan-your-azure-devops-org-structure)
  + [Project](https://learn.microsoft.com/en-us/azure/devops/user-guide/plan-your-azure-devops-org-structure#how-many-projects-do-you-need)
  + [Git repository](https://learn.microsoft.com/en-us/azure/devops/user-guide/plan-your-azure-devops-org-structure#structure-repos-and-version-control-within-a-project)
  + Select **+ New Branch** to create a new branch.
  + Name the new branch MyFoods\_FeatureBranch, branch it from main (or master), and Select **Create**.
  + The folder in the repo where the .pbix file located.



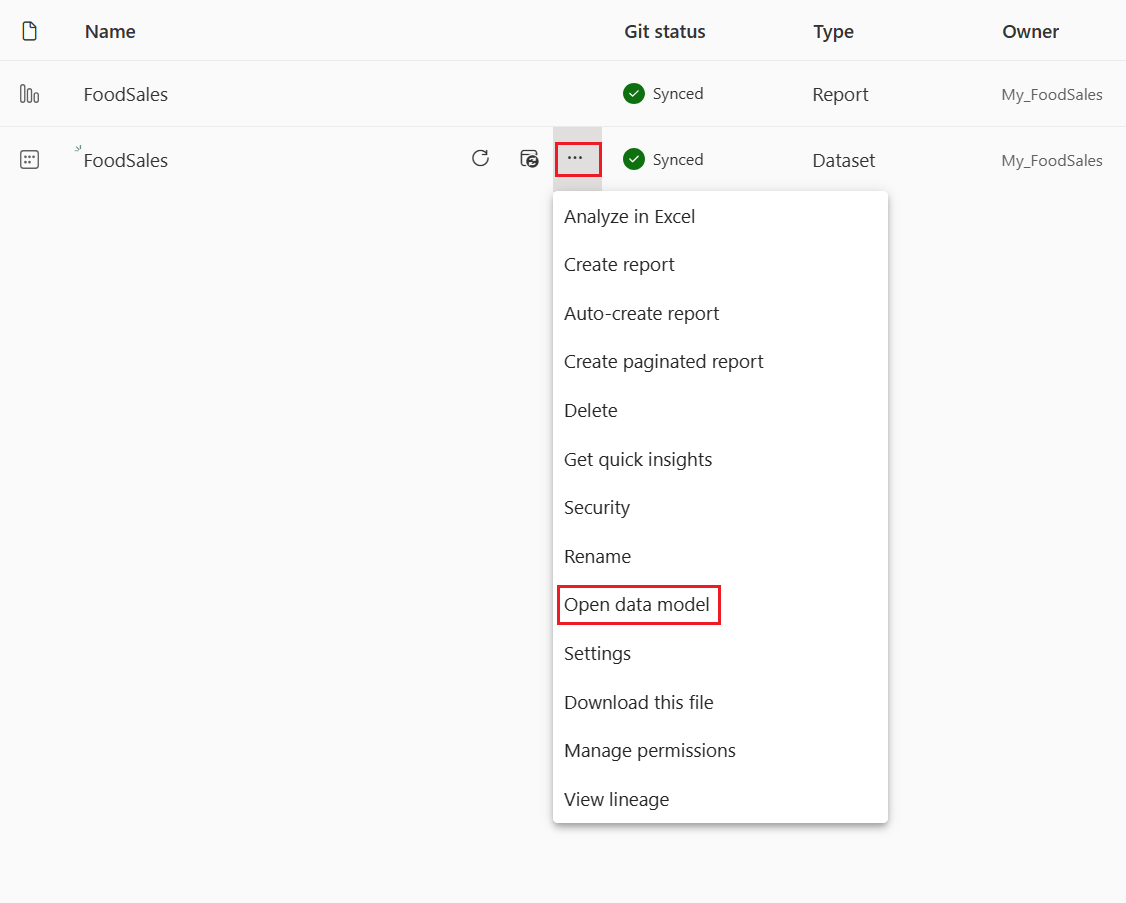
1. Select **Connect and sync**.

The new workspace now contains the content of the Git repo folder. Notice it doesn't contain the .pbix file. Since .pbix files are unsupported, this file wasn't copied to the Git repo when we synced.  
This is the workspace you use to make changes to the dataset and report until you're ready to share them with your team.

## Step 7: Edit the workspace

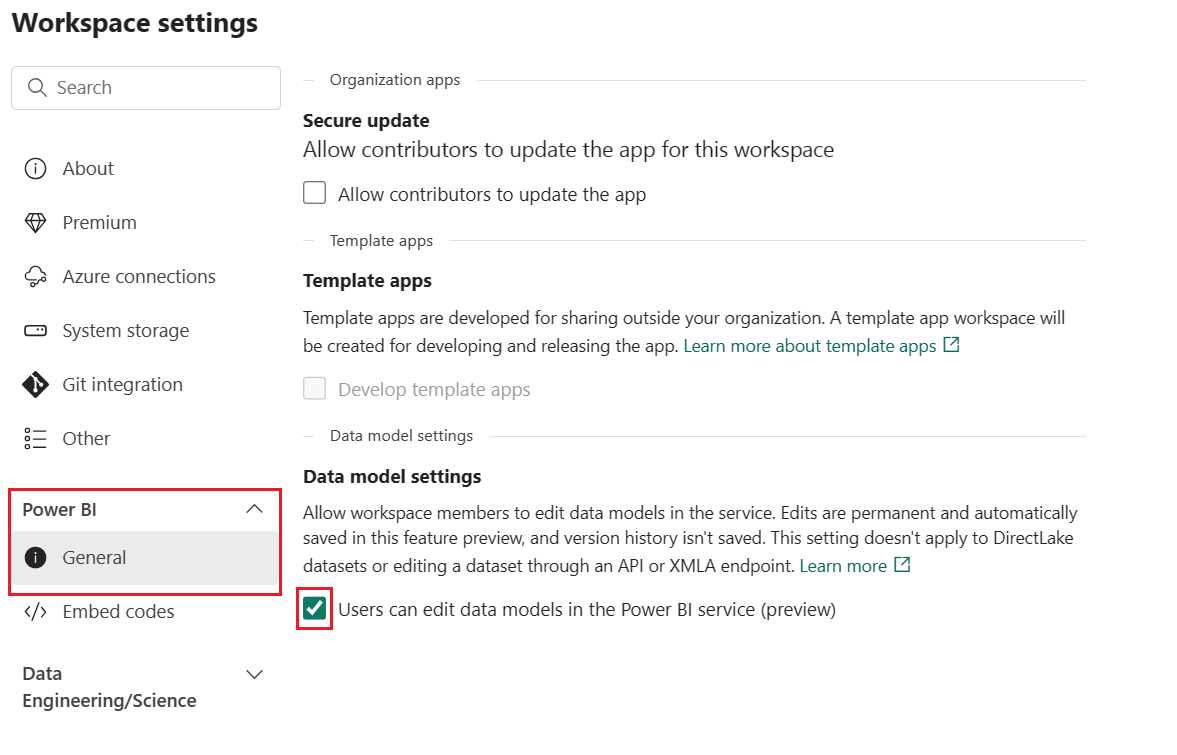
Make changes to the workspace by creating, deleting, or editing an item. In this tutorial, we change the format of a dataset column. You can edit the workspace in [Power BI Desktop](https://learn.microsoft.com/en-us/power-bi/fundamentals/desktop-what-is-desktop) or [data model](https://learn.microsoft.com/en-us/power-bi/transform-model/service-edit-data-models). In this tutorial, we edit the workspace from the data model.

1. From the dataset workspace, select the dataset ellipsis (three dots) > **Open data model**.

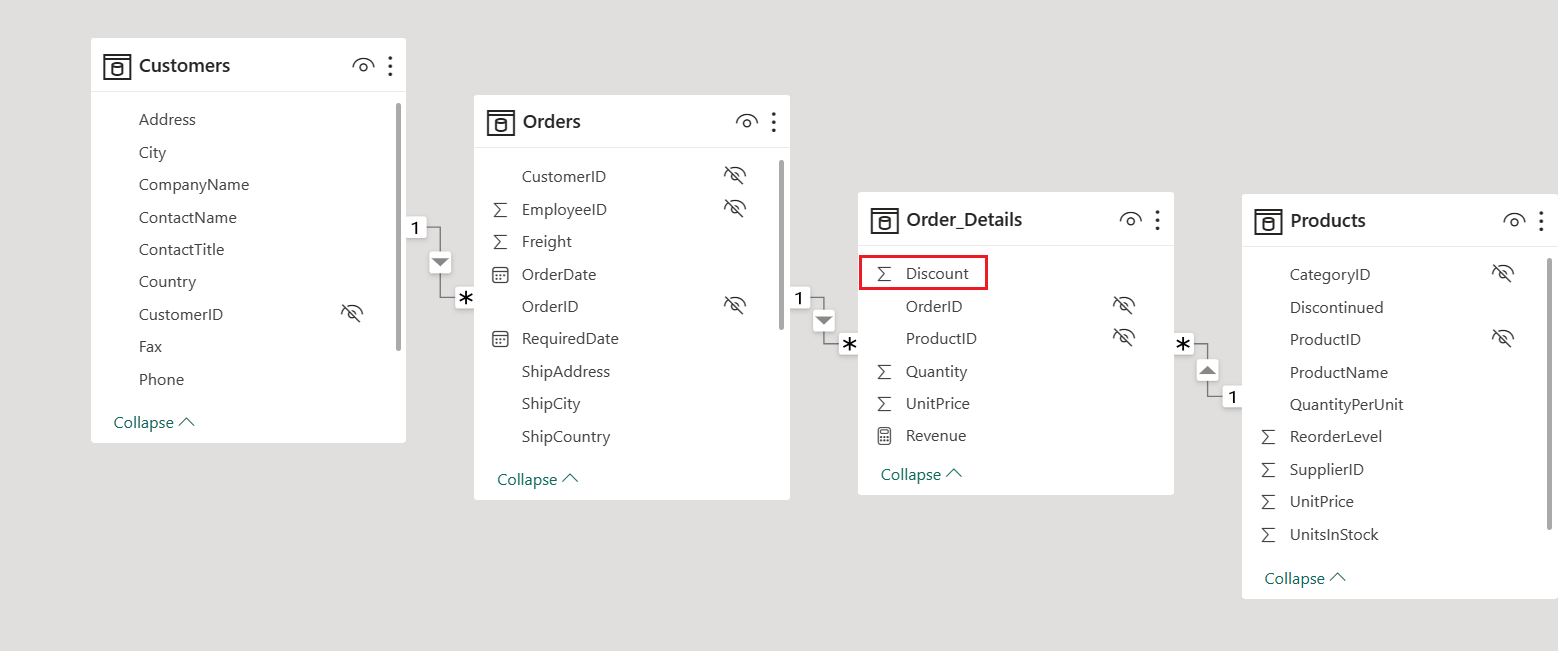


**Note**

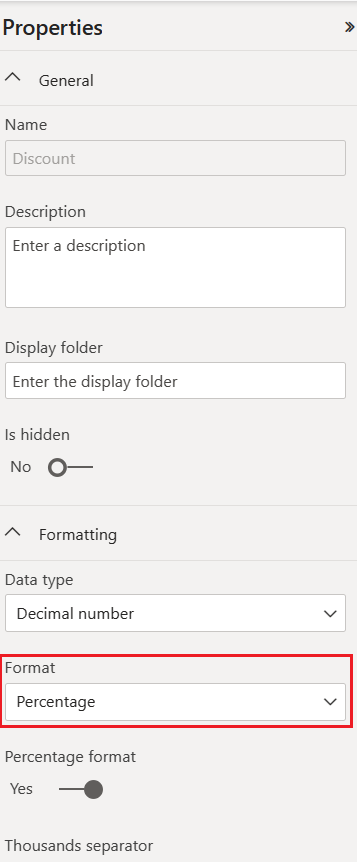
If **Open data model** is disabled, go to **Workspace settings > Power BI > General** and enable **Data model settings**.



1. From the **Order\_details** table, select **Discount**.



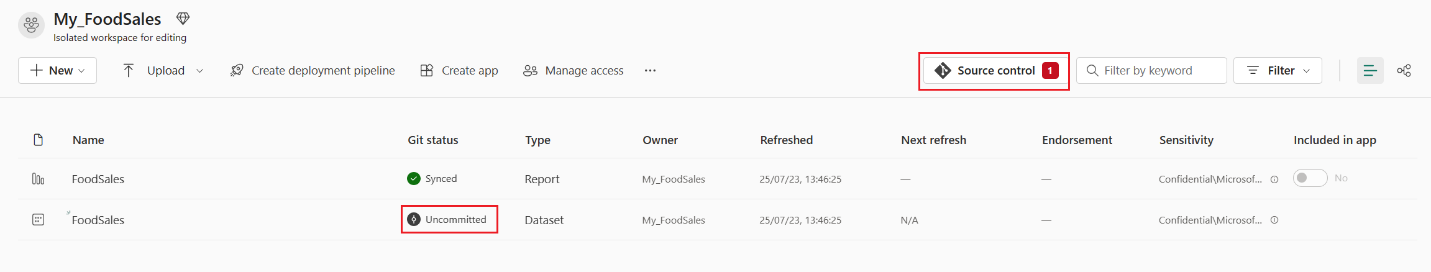
1. From the **Properties** pane, change the **Format** from General to Percentage.



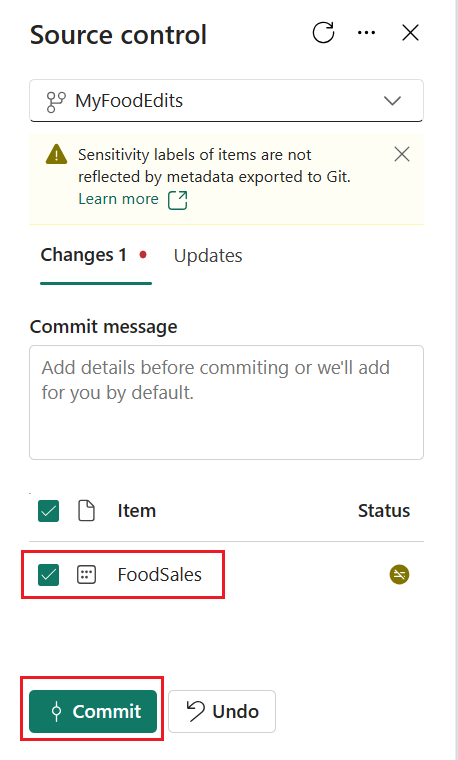
## Step 8: Commit changes

To commit this change from the workspace into the Git branch, go back to the workspace home page.

The source control icon now shows 1 because one item in the workspace was changed but not committed to the Git repo. The FoodSales dataset shows a status of Uncommitted.



1. Select the source control icon to view the changed items in the Git repo. The dataset shows a status of Modified.
2. Select the item to commit and add an optional message.
3. Select **Commit**.

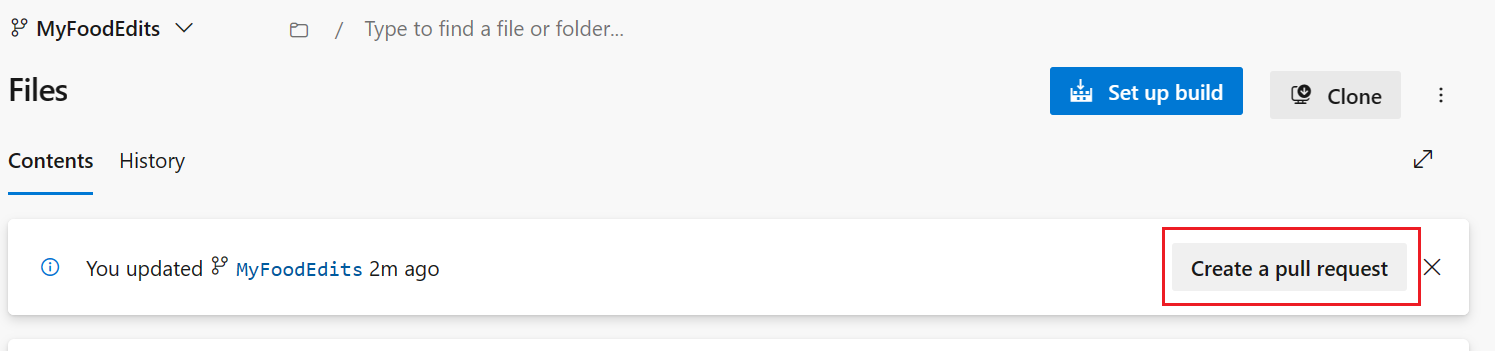


The Git status of the dataset changes to Synced and the workspace and Git repo are in sync.

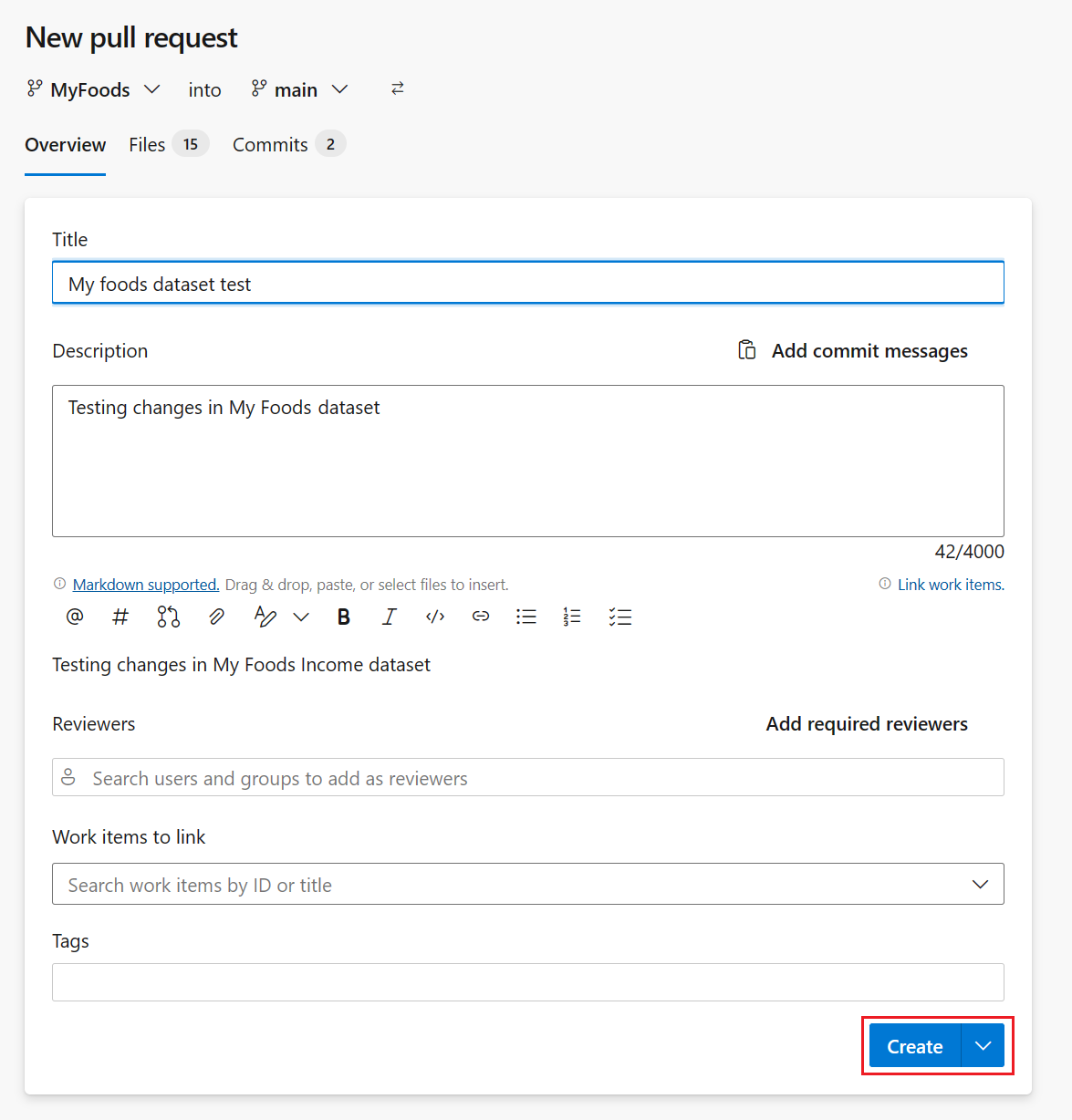
## Step 9: Create PR and merge

In the Git repo, [create a pull request](https://learn.microsoft.com/en-us/azure/devops/repos/git/pull-requests#create-a-pull-request) to merge the MyFoods branch with the main branch.

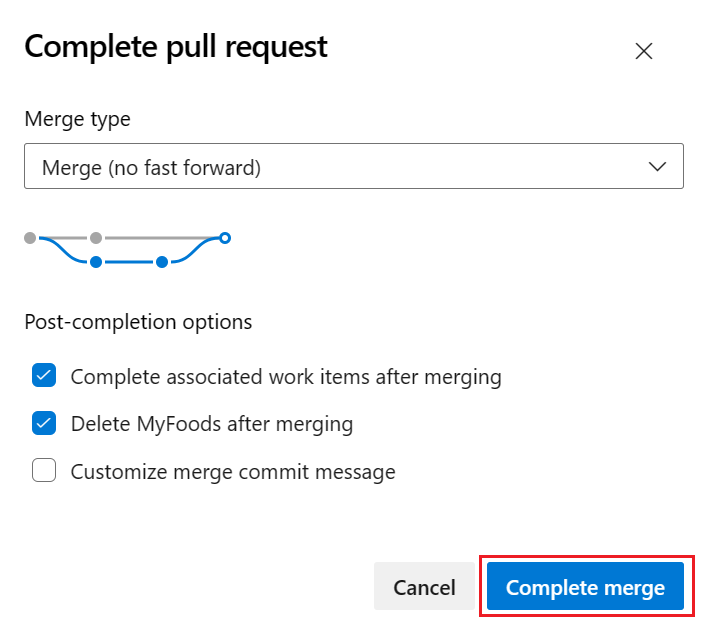
1. Select **Create a pull request**.



1. Provide a title, description, and any other information you want for the pull request. Then select **Create**.

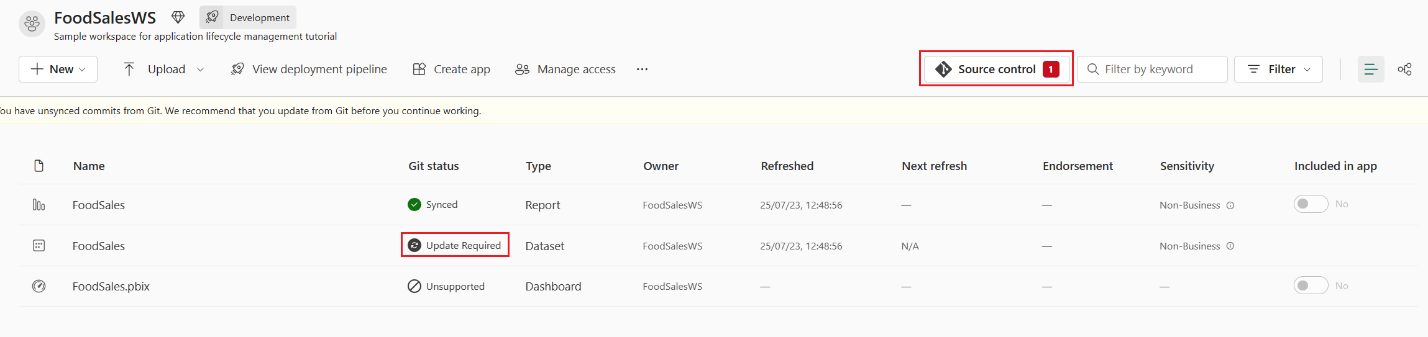


1. [Merge the pull request](https://learn.microsoft.com/en-us/azure/devops/repos/git/complete-pull-requests#complete-a-pull-request).

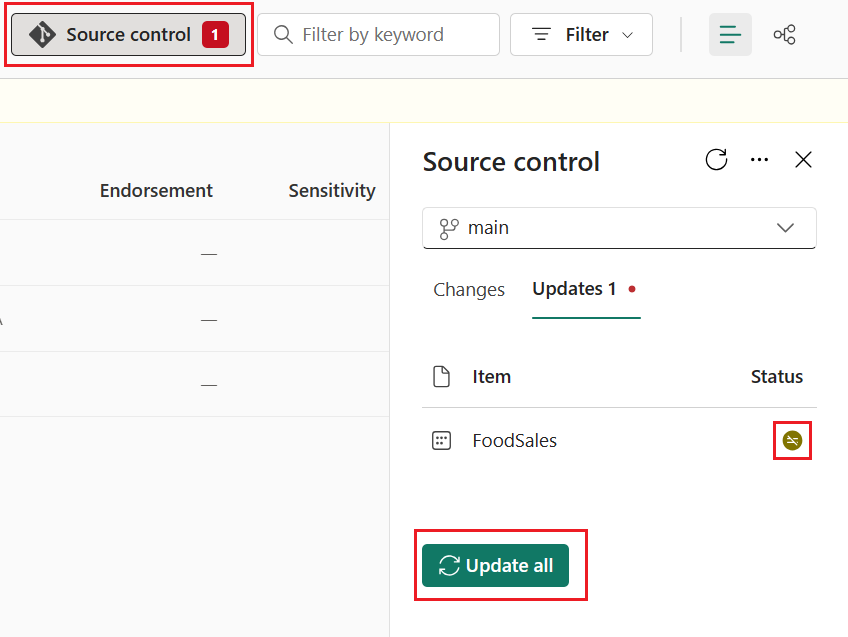


## Step 10: Update shared workspace

Go back to the shared workspace connected to the dev stage of the deployment pipeline (the one we created in [Step 1](https://learn.microsoft.com/en-us/fabric/cicd/cicd-tutorial#step-1-create-a-premium-workspace)) and refresh the page.  
The source control icon now shows 1 because one item in the Git repo was changed and is different from the items in the FoodSales workspace. The FoodSales dataset shows a status of Update required.



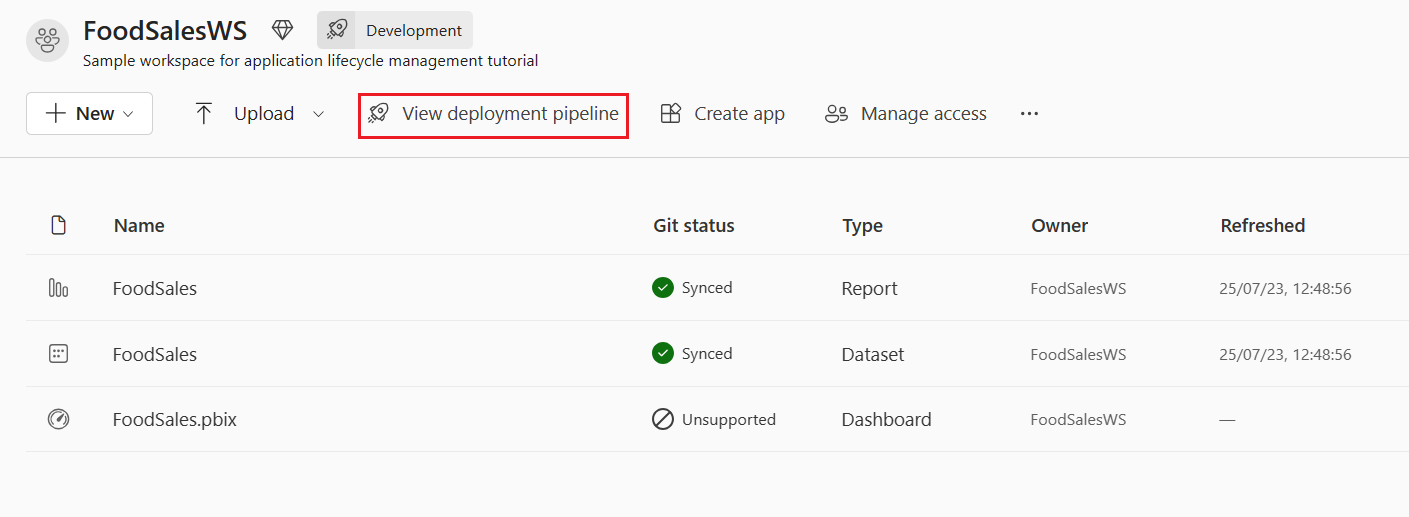
1. Select the source control icon to view the changed items in the Git repo. The dataset shows a status of Modified.
2. Select **Update all**.



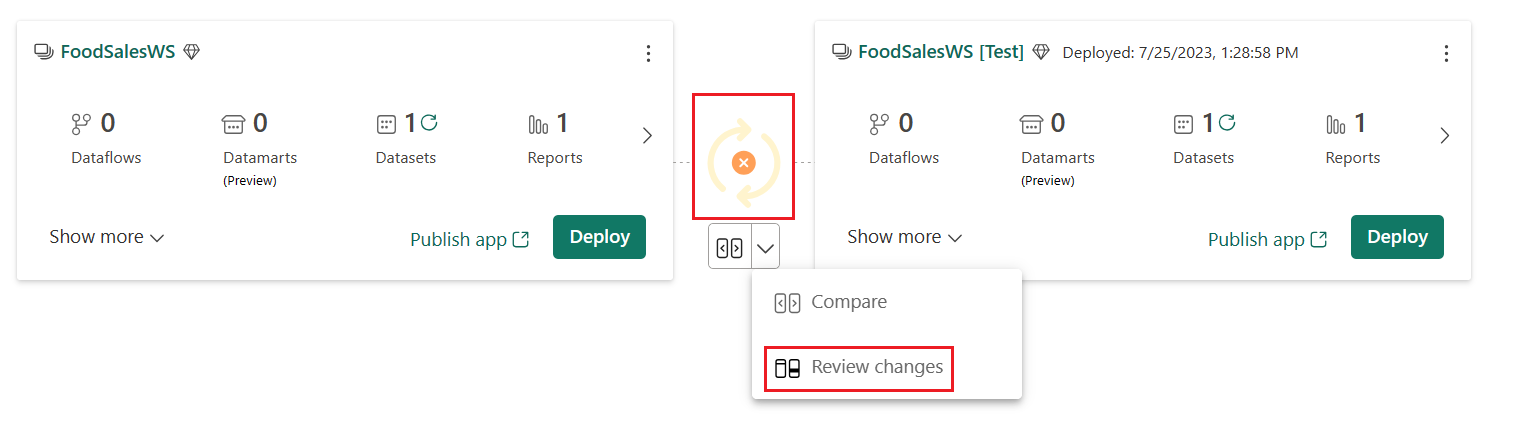
The Git status of the dataset changes to Synced and the workspace is synced with the main Git branch.

## Step 11: Compare stages in deployment pipeline

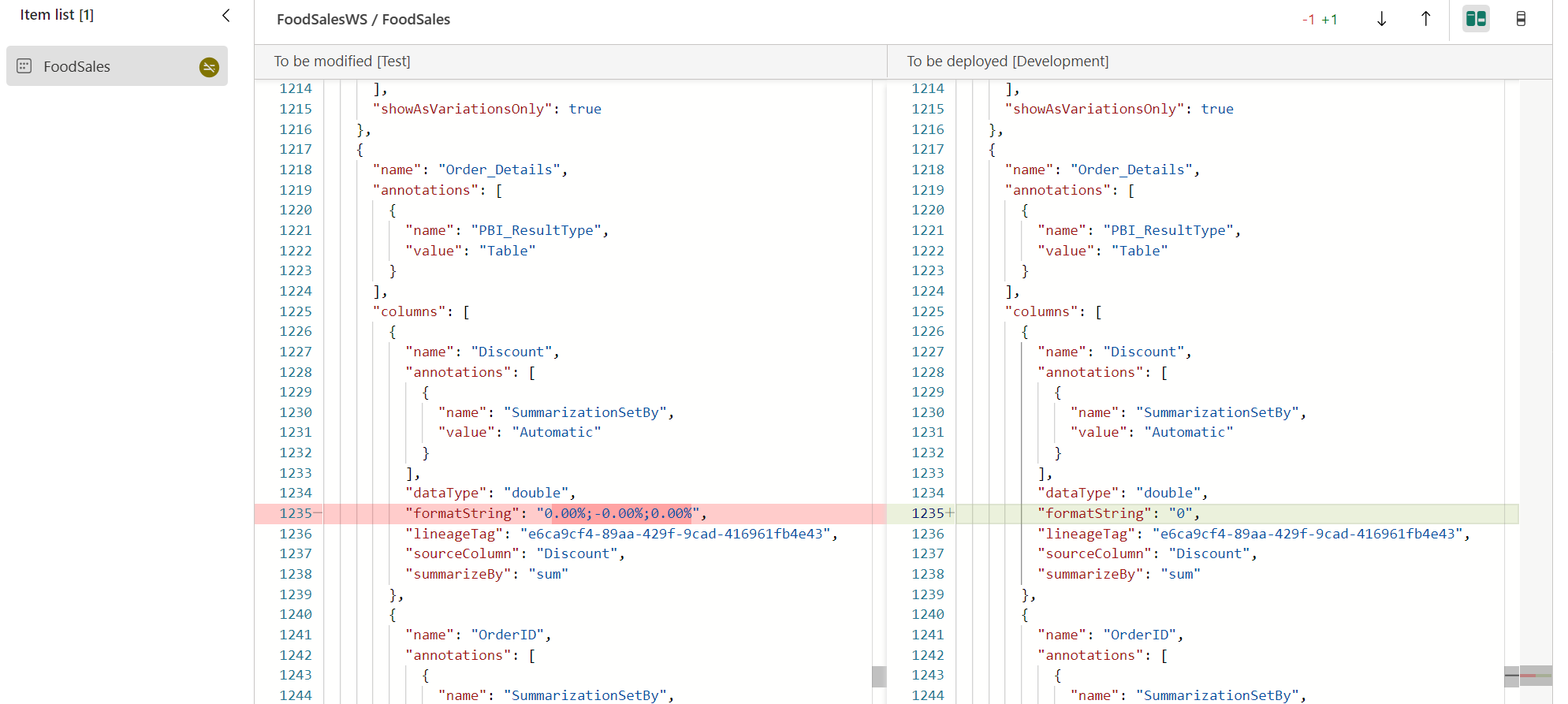
1. Select **View deployment pipelines** to compare the content in the development stage with the content in the test stage.



Notice the orange X icon between the stages indicating that changes were made to the content in one of the stages since the last deployment.



1. Select the down arrow > **Review Changes** to view the changes. The **Change Review** screen shows the difference between the datasets in the two stages.



1. Review the changes and close the window.

For more information about comparing stages in a deployment pipeline, see [Compare stages in a deployment pipeline](https://learn.microsoft.com/en-us/fabric/cicd/deployment-pipelines/compare-pipeline-content).

## Step 12: Deploy to test stage

When you’re satisfied with the changes, deploy the changes to the test and/or production stages using the same process you used in [Step 5](https://learn.microsoft.com/en-us/fabric/cicd/cicd-tutorial#step-5-deploy-content-to-other-stages).

## Summary

In this tutorial, you learned how to use deployment pipelines along with Git integration to manage the lifecycle of an app, report, or other content in a workspace.  
In particular, you learned how to:

* Setup workspaces and add content for managing their lifecycle in Fabric.
* Apply Git best practices to work alone and collaborate with teammates on changes.
* Combine Git and deployment pipelines for an efficient end to end release process.

## Next steps

[Manage Git branches](https://learn.microsoft.com/en-us/fabric/cicd/git-integration/manage-branches)