

Prerequisites for Connecting to Microsoft Power BI Cloud

1. Open Access to Power BI Endpoints

Integration with Power BI requires opening the following endpoints:

- **app.powerbi.com** (port 443)
- ***.windows.net** (port 443)
- **powerbi.microsoft.com** (port 443)
- **dc.services.visualstudio.com** (port 443)
- **login.microsoftonline.com** (port 443)
- ***.office.com** (port 443)
- **powerapp.com** (port 443)
- **powerapps.com** (port 443)

These endpoints must be accessible from either the Metric Insights application server, or the Remote Data Collector host.

 [Release 6.x]: 'Remote Data Collector' renamed to 'Remote Data Processor'

 [Release 6.4.0]: The data from Power BI can be fetched via DAX API without using an RDP. Note that this method of fetching data has limitations of 100,000 rows or 1,000,000 values per request. This setting is controlled by the [OleDB data parameter](#) in Microsoft Power BI Cloud plugin connection profile.

- See [details on DAX API limitations for fetched data](#)

There are two authentication types:

- Username/Password
- OAuth

If you are using OAuth, make sure the endpoints are open and proceed with configuring [Microsoft Power BI OAuth in Azure AD](#). After OAuth is configured, skip to [Step 7](#) if you will be using an RDP or skip to [Step 9](#) if you will not be using an RDP.

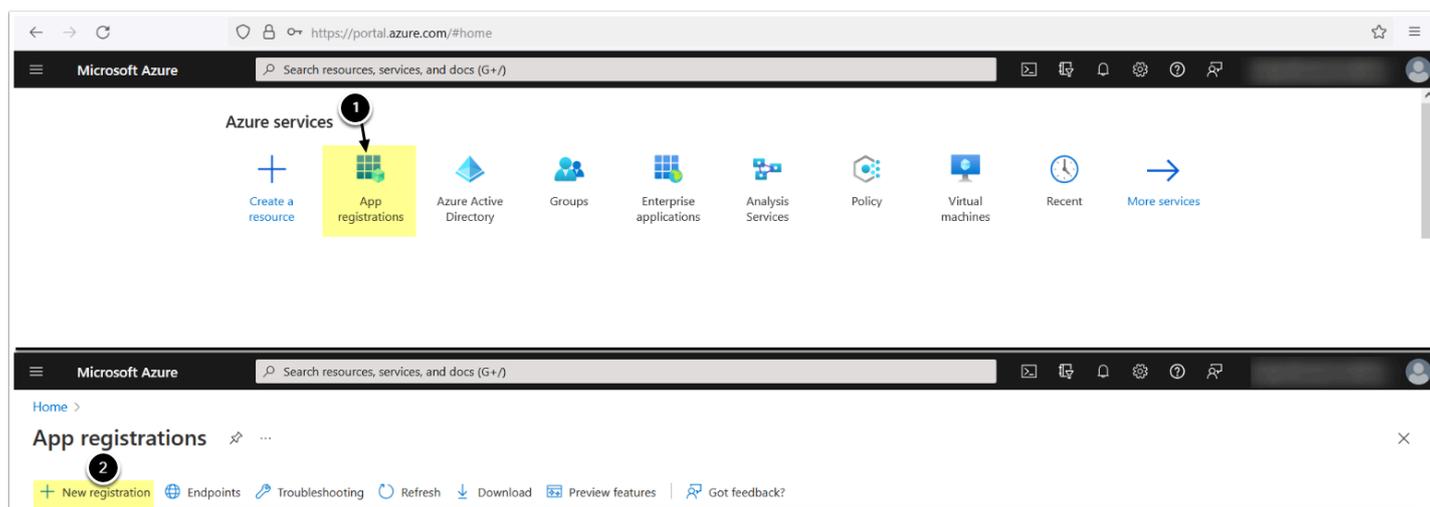
If your primary aim is to pull *images* from Power BI, and DAX API limitations for data fetching are suitable for you, then the endpoints must be accessible from the Metric Insights server. Proceed with the steps described in the article **skipping** the Steps 7 and 8.

! The integration is taking place directly from the application server.

If you wish to get *data only* (or both data and images) without limitations for data fetching, then a Remote Data Processor (RDP) must be deployed. The RDP must be installed on a *Windows* machine that can access the Power BI endpoints listed above.

! The integration is taking place on the [RDP host machine](#) (Windows).

2. Access Microsoft Azure Portal



1. In [Azure Portal](#), access **App registrations**
2. [**+ New registration**]

3. Enter App Info

! **Note:** Admin access is required for the user/service account who is setting up the Azure App. The user role in Azure should be "Application Administrator"

Microsoft Azure

Home > App registrations > Register an application

* Name
The user-facing display name for this application (this can be changed later).

1 Microsoft Power BI Documentation Example ✓

Supported account types
Who can use this application or access this API?

2 Accounts in this organizational directory only (mi only - Single tenant)
 Accounts in any organizational directory (Any Azure AD directory - Multitenant)
 Accounts in any organizational directory (Any Azure AD directory - Multitenant) and personal Microsoft accounts (e.g. Skype, Xbox)
 Personal Microsoft accounts only

Help me choose...

Redirect URI (optional)
We'll return the authentication response to this URI after successfully authenticating the user. Providing this now is optional and it can be changed later, but a value is required for most authentication scenarios.

3 Web https://<MI hostname>.com/editor/service/validatepowerbioauth ✓

By proceeding, you agree to the Microsoft Platform Policies

4 Register

1. Enter the App's **Name**
2. **Supported account types:** "Accounts in this organizational directory only (<directory name> only - Single tenant)"
3. **Redirect URI:** select **Web**, enter `https://<MI hostname>.com/editor/service/validatepowerbioauth`
4. **[Register]**

4. Enable Public Client Flows

! Once the application has been registered, **please enable Public Client Flows under the Authentication tab. This step is only required when using Password Authentication - it is not required for OAuth.** Public Flows are what allows third party apps (like MI) to get a token using username and password (enabling Metric Insights to get a token by itself).

If this is not enabled, the only way to get a token is via user interaction in a browser and there is no existing functionality using Password auth to get a token via user interaction in MI. This is the only option for native applications like MI

Who can use this application or access this API?

Accounts in this organizational directory only (mi only - Single tenant)
 Accounts in any organizational directory (Any Azure AD directory - Multitenant)

[Help me decide...](#)

⚠ Due to temporary differences in supported functionality, we don't recommend enabling personal Microsoft accounts for an existing registration. If you need to enable personal accounts, you can do so using the manifest editor. [Learn more about these restrictions.](#)

Advanced settings

Allow public client flows ⓘ

Enable the following mobile and desktop flows:

Yes No

- App collects plaintext password (Resource Owner Password Credential Flow) [Learn more](#)
- No keyboard (Device Code Flow) [Learn more](#)
- SSO for domain-joined Windows (Windows Integrated Auth Flow) [Learn more](#)

App instance property lock ⓘ

Configure the application instance modification lock. [Learn more](#) [Configure](#)

5. Add Power BI Service Permissions

The screenshot shows the 'Request API permissions' dialog in the Microsoft Azure portal. The 'Power BI Service' is selected as the API to request permissions for. The 'What type of permissions does your application require?' section shows 'Delegated permissions' selected. In the 'Select permissions' list, the 'App.Read.All' permission is checked, and the 'Admin consent required' column shows 'No'.

Note: Delegated permissions allow the application to access the API as the signed-in user.

1. Access *API Permissions* tab
2. **[+ Add a permission]**
3. **[Power BI Service]**
4. **[Delegated permissions]**
5. Enable the following permissions to get a list of all Power BI Apps, Dashboards, Datasets, Reports, and Workspaces respectively:
 - **App:**
 - App.Read.All
 - **Dashboard:**
 - Dashboard.Read.All
 - **Dataset:**
 - Dataset.Read.All
 - **Report:**
 - Report.Read.All
 - **Workspace:**
 - Workspace.Read.All
6. **[Add permissions]**

6. Grant Admin Consent

The screenshot shows the Microsoft Azure portal interface. At the top, there's a search bar and navigation icons. The main content area is titled 'Microsoft Power BI Documentation Example | API permissions'. A modal dialog box is open, titled 'Grant admin consent confirmation.' with the text: 'Do you want to grant consent for the requested permissions for all accounts in mi? This will update any existing admin consent records this application already has to match what is listed below.' There are 'Yes' and 'No' buttons, with 'Yes' highlighted. Below the dialog, the 'Configured permissions' section is visible, showing a table of permissions. A yellow highlight is on the 'Grant admin consent for mi' button, and a red circle with the number '1' is around it. Another red circle with the number '2' is around the 'Yes' button in the dialog.

API / Permissions name	Type	Description	Admin consent requ...	Status
Microsoft Graph (1)				
User.Read	Delegated	Sign in and read user profile	No	...
Power BI Service (5)				
App.Read.All	Delegated	View all Power BI apps	No	...
Dashboard.Read.All	Delegated	View all dashboards	No	...
Dataset.Read.All	Delegated	View all datasets	No	...
Report.Read.All	Delegated	View all reports	No	...

1. [Grant admin consent for <directory name>]
2. [Yes]

7. Install the Remote Data Processor

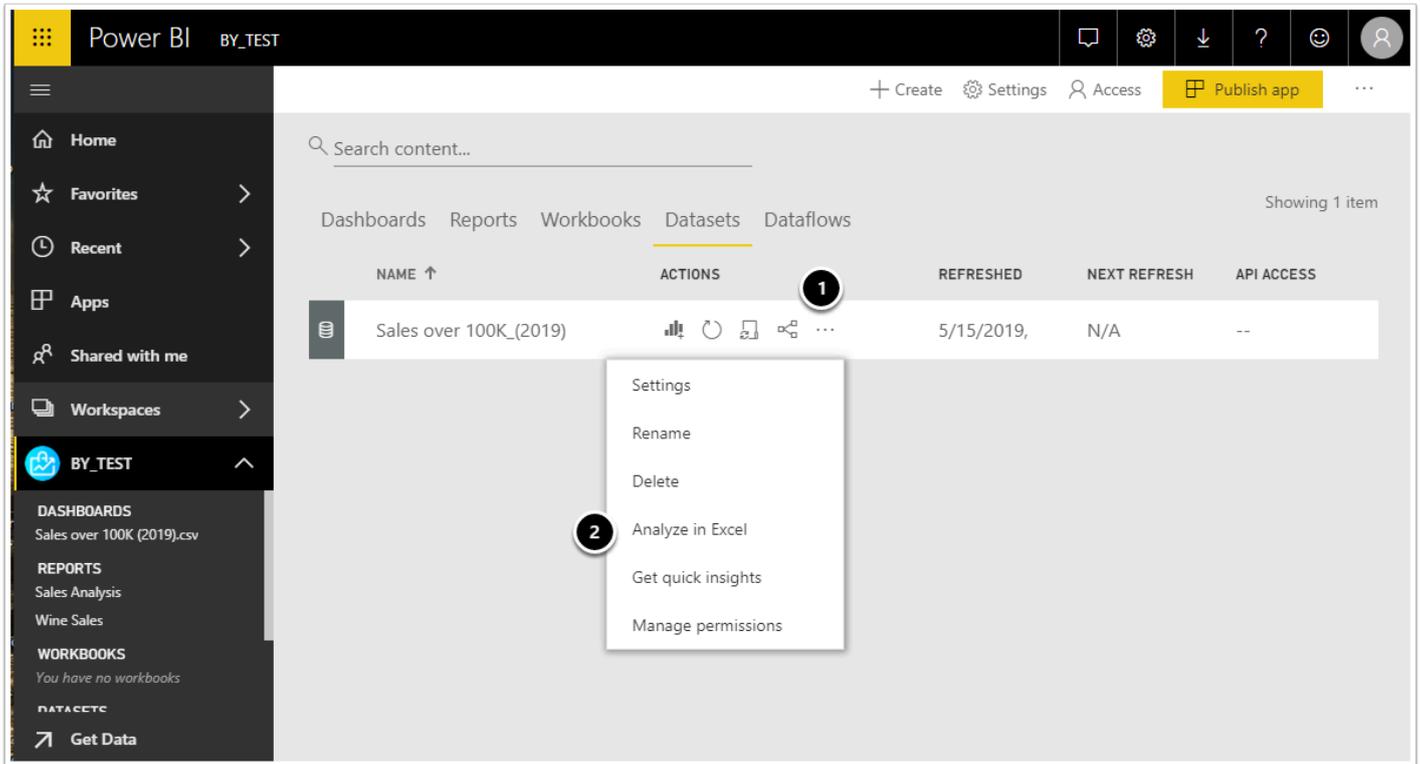
For steps on how to install the RDP, refer to [Configure a Remote Data Processor](#).

-  If you are using AD, you may want to run the RDP service as an AD service account:
- Access **Services > Metric Insights Data Processor Daemon > Properties > Log On**, select AD service account

8. .NET and Excel Adapter Must Be Installed on the RDP Host Machine

 **Note:**

- RDP requires *.NET Framework v4.5* or higher.
- Microsoft Power BI Datasets can only be fetched by the Metric Insights plugin with the corresponding *Excel adapter (OLEDB adapter)*.



i When you select **Analyze in Excel** from the *Ellipses menu (the ...)* associated with a dataset or report, Power BI creates an `.msi` file and downloads it from the browser to your computer. Open the downloaded `SQL_AS_OLEDB.msi` file and install the adapter.

[Learn more about Analyze in Excel.](#)

To download the Excel adapter:

1. Go to the **Ellipses menu**
2. Click **Analyze in Excel**

Install the adapter tool as prompted.

9. Create a Power BI Cloud Connection Profile in Metric Insights

Now that you have completed the prerequisites, you are ready to create a connection profile in Metric Insights.

- Follow the steps in [Establish Connectivity to Microsoft Power BI Cloud](#) article to complete the integration.