IT Real-Time training that work for your career.

PROVIDED TRAINING FOR THOUSANDS OF STUDENTS.

SUBJECT, MATERIAL & VIDEOS

VIDEOS PROVIDED

POWER BI MATERIAL

- Power BI Embed
- Power BI Data Flow and Self Service Operation
- Azure Active Directory
- Power BI App Registrations and Developer

Trainings:

CLASS ROOM

ONLINE

FAST TRACK

ONE ON ONE

PROJECT TRAINING

Address:

Flat No: 506/B

Nilgiri Block

Aditya Enclave

Mytrivanam Area

Hyderabad.

Website & Blog

www.vinaytechhouse.com

www.msbivinay.blogspot.in

Contact Information

+91 9573168449

040 66638869

MSBI

IS, AS, RS & MDS

POWER BI

SERVER, DESKTOP & DAX

Sri VINAY TechHouse

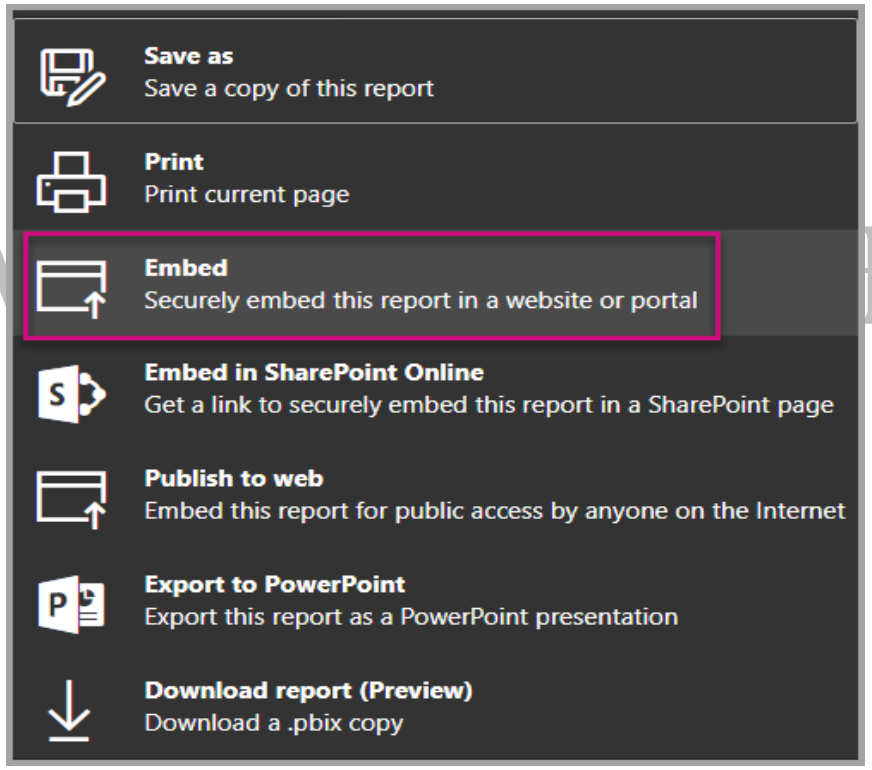
POWER BI EMBEDDING INTO WEBSITES, BLOGS, AND PORTALS

With the new **Embed** option for Power BI reports, **you can easily and securely embed reports in internal web portals.** These portals can be **cloud-based** or **hosted on-premises**, **such as SharePoint 2019**. **Embedded reports respect all item permissions and data security through row-level security (RLS). They provide no-code embedding into any portal that accepts a URL or iFrame.**

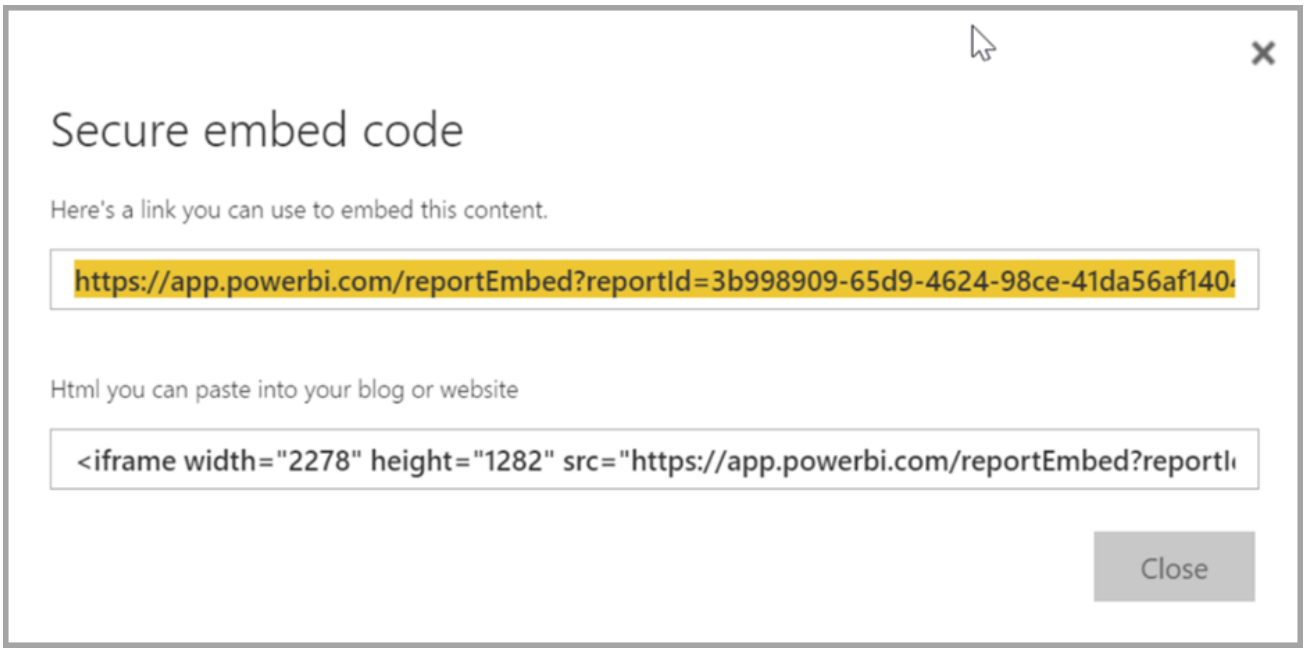
The **Embed** option supports **URL Filters and URL settings.** It allows you to integrate with portals using **a low-code approach requiring only basic HTML and JavaScript knowledge.**

How to Embed Power BI reports into portals

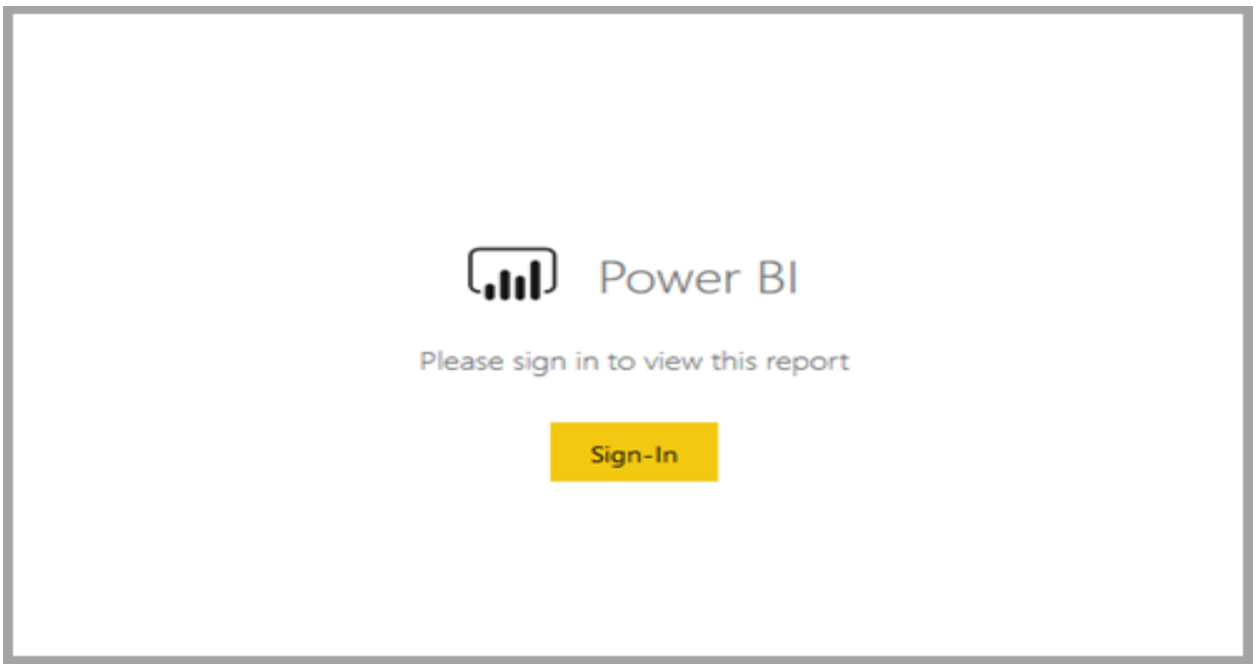
1. The new **Embed** option is available on the **File** menu for reports in the Power BI service.



2. Select the **Embed** option to open a dialog that **provides a link and an iFrame** you can use to embed the report securely.

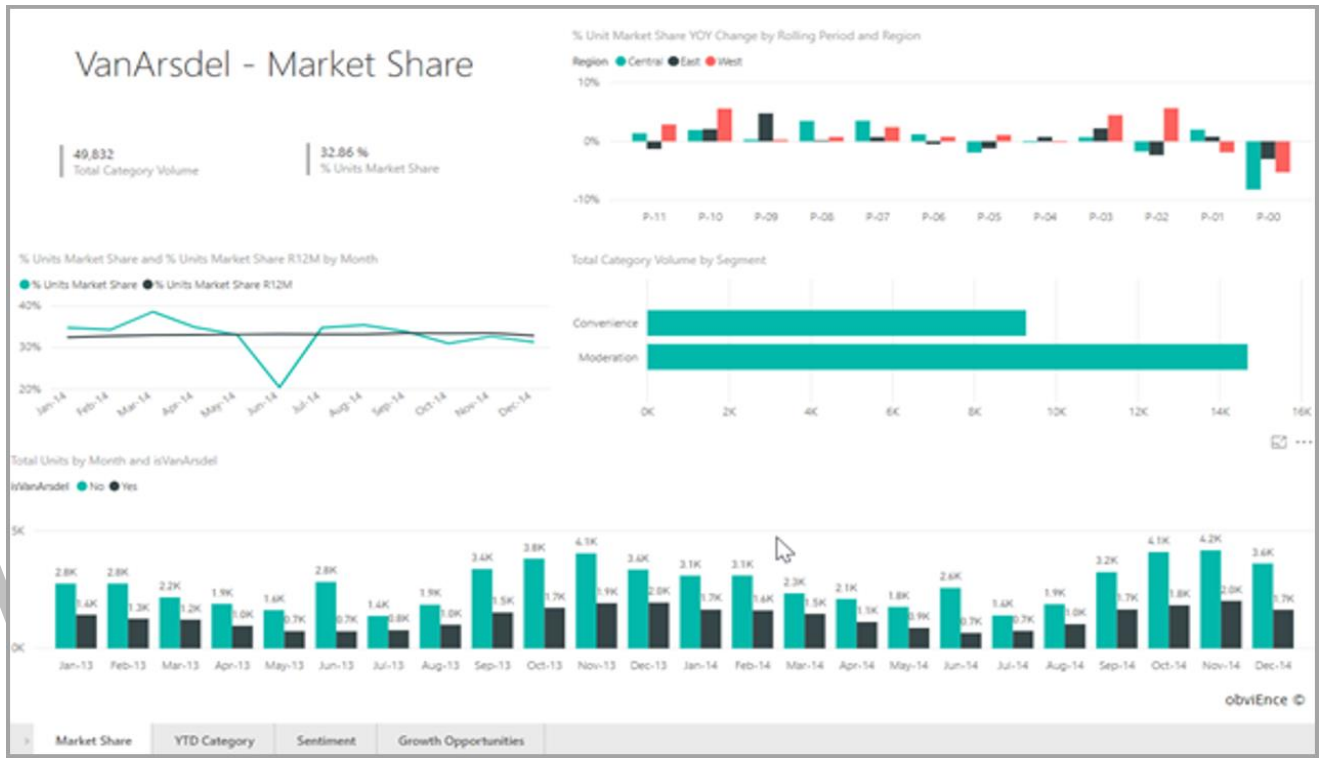


3. Whether a user opens a report URL directly, or one embedded in a web portal, **report access requires authentication.** The following screen appears if a user has not signed-in to Power BI in their browser session. When they select **Sign-In**, a new browser window or tab could open. **Have them check for pop-up blockers if they don't get prompted to sign in.**



4. After the user has signed in, the report opens, showing the data and allowing page navigation and filter setting. **Only users who have view permission can see**

the report in Power BI. All row-level security (RLS) rules are also applied. Lastly, the user needs to be correctly licensed – either they need a Power BI Pro license, or the report must be in a workspace that is in a Power BI Premium capacity. The user needs to sign in each time they open a new browser window. However, once signed in, other reports load automatically.



5. When using an iFrame, you **may need to edit the** height **and** width to have it fit in your portal's web page.

```
<iframe width="1080" height="760"
src="https://app.powerbi.com/reportEmbed?reportId=3b998909
-65d9-4624-98ce-41da56af1404&autoAuth=true"
frameborder="0" allowFullScreen="true"></iframe>
```

Granting report access

The **Embed** option doesn't automatically permit users to view the report. **View permissions are set in the Power BI service.**

In the Power BI service, **you can share embedded reports with users requiring access.** If you're using an Office 365 Group, you can list the user as an app workspace member.

Licensing

To view the embedded report, users need either a Power BI Pro license or the content needs to be in a workspace that's in a Power BI Premium capacity (EM or P SKU).

Customize your embed experience using URL settings

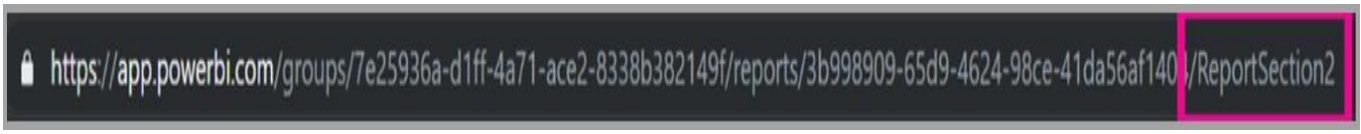
You can customize the user experience using the embed URL's input settings. In the provided iFrame, **you can update the URL's** src **settings.**

Property	Description
pageName	You can use the ***pageName query string parameter to set which report page to open***. You can find this value at the report URL's end when viewing a report in the Power BI service, as shown below.
URL Filters	**You can use URL Filters in the embed URL you received from the Power BI UI to filter the embed content.** This way you can build low-code integrations with only basic HTML and JavaScript experience.

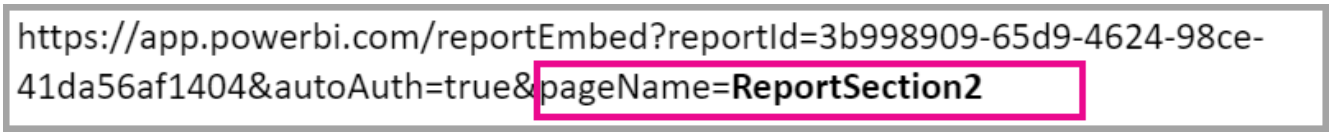
Set which page opens for an embedded report

You can find the **pageName** value at the report URL's end when viewing a report in the Power BI service.

1. Open the report from the Power BI service in your web browser, and then copy the address bar URL.



2. Append the **pageName** setting to the URL.



Filter report content using URL filters

You can use URL Filters to provide different report views. For example, the URL below filters the report to show data for the Energy industry.

Using the combination of pageName **and URL Filters can be powerful.** You can build experiences using basic HTML and JavaScript.

For example, here's a button you can add to an HTML page:

HTML Copy

```
<button class="textLarge" onclick='show("ReportSection", "Energy");' style="display: inline-block;">Show Energy</button>
```

When selected, **the button calls a function to update the iFrame with an updated URL**, which includes the Energy industry filter.

JavaScript Copy

```
function show(pageName, filterValue)

{

var newUrl = baseUrl + "&pageName=" + pageName;

if(null != filterValue && "" != filterValue)

{

newUrl += "&$filter=Industries/Industry eq '" + filterValue + "'";

}

//Assumes there's an iFrame on the page with id="iFrame"

var report = document.getElementById("iFrame")

report.src = newUrl;

}
```



```
https://app.powerbi.com/reportEmbed?reportId=3b998909-65d9-4624-98ce-41da56af1404&autoAuth=true&pageName=ReportSection&$filter=Industries/Industry eq 'Energy'
```

You can add as many buttons as you'd like to create a low-code custom experience.

Considerations and limitations

- **Doesn't support external guest users** with Azure business to business (B2B).
- Secure **embed works for reports published to the Power BI service**.
- The user **needs to sign in to view the report whenever they open a new browser window**.
- **Some browsers require you to refresh the page after sign-in, especially when using InPrivate or Incognito modes.**
- To achieve **a single sign-on experience, use the Embed in SharePoint Online option**, or build a custom integration using the user owns data embedding method.
- The automatic authentication capability provided with the Embed **option doesn't work with the Power BI JavaScript API**. For the Power BI JavaScript API, use the user owns data embedding method.
- The **authentication token lifetime is controlled based on your AAD settings**. When the authentication token expires, the user will need to refresh their browser to get an updated authentication token. The default lifetime is one hour, but it could be shorter or longer in your organization.

Trainer's practice on his own blog [Notes@ classroom]

Identify a report and its URL or iFrame

Go to Power BI Service→ Report→File menu→ Embed→ Copy the URL or iFrame

Note: Do the customization if required [PageName, height width etc…]

Embed in the Blog:

Open a blog /portal with the respective user account

Open a html page, copy and paste the iframe tag.

Save and preview

It now shows the Power BI Report.

Observation:

a)If there is a Row-Level security, respective data only displayed

b) No RLS, then complete information displayed.

For your practice: Use project number-2 iframe [which is not having RLS]

Additional information:

URL appears like below:

https://app.powerbi.com/reportEmbed?reportId=84cbeba0-c9b4-45d1-81ef-d2d877a7a3ea&autoAuth=true&ctid=e2aa1f92-8ae9-48c0-8aa8-894fb201ce24&config=eyJjbHVzdGVyVXJsIjoiaHR0cHM6Ly93YWJpLWluZGlhLXdlc3QtcmVkaXJlY3QuYW5hbHlzaXMud2luZG93cy5uZXQvIn0%3D

.net / java applications embedding for actions

Ex: In a button click opening a Power BI Report

iFrame appears like below

<iframe width="600" height="300" src="https://app.powerbi.com/reportEmbed?reportId=84cbeba0-c9b4-45d1-81ef-d2d877a7a3ea&autoAuth=true&ctid=e2aa1f92-8ae9-48c0-8aa8-894fb201ce24&config=eyJjbHVzdGVyVXJsIjoiaHR0cHM6Ly93YWJpLWluZGlhLXdlc3QtcmVkaXJlY3QuYW5hbHlzaXMud2luZG93cy5uZXQvIn0%3D" frameborder="0" allowFullScreen="true"></iframe>

--html / Java Script etc...

SELF-SERVICE DATA PREPARATION IN POWER BI

How do we share our dataset and columns?

a)Power BI Template b) Power BI Dataset c) Power BI .pbix file d) Dataflow

What is Self Service Data Preparation?

a)This is advanced than the above and maintains a common data model in the project.

b)This model can be used by any report for reporting operations.

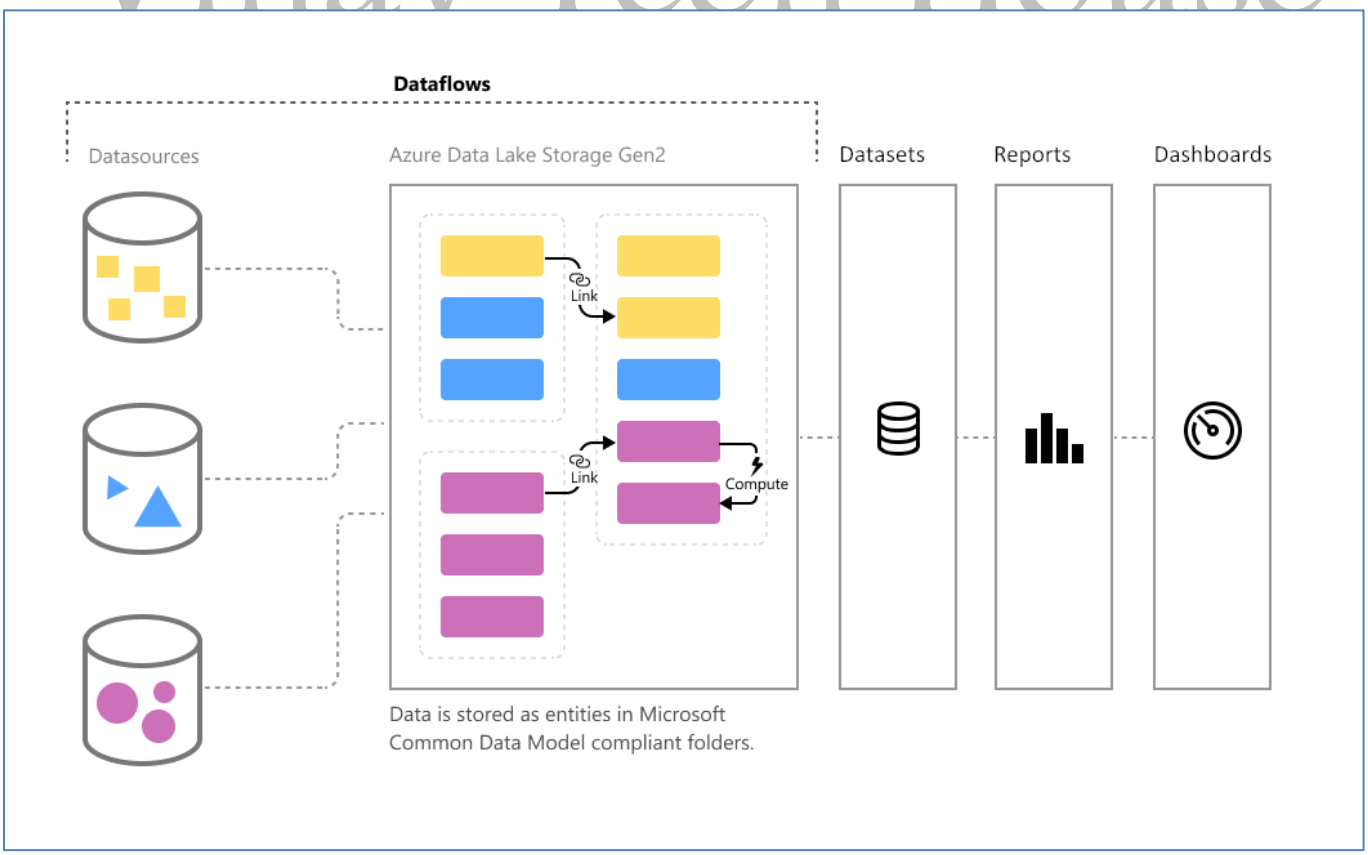
c)This preparation is helpful **when data grows and reports increase.**

Note: In SSRS we call it as Report Model which is .smdl language.

When do we go for it?

As **data volume continues to grow**, so does the challenge of **wrangling that data into well-formed, actionable information**. With **self-service data prep** for **big data** in Power BI, you can go from **data to Power BI insights with just a few clicks**.

By storing business data in the standardized format of the **Common Data Model**, **your BI Pros (or developers) can create apps that generate quick, easy, and automatic visuals and reports.**



What is Data Flow?

A dataflow **is a collection of *entities* (entities are similar to tables) that are created and managed in app workspaces in the Power BI service**. You can **add and edit entities in your dataflow, as well as manage data refresh schedules,** directly from the workspace in which your dataflow was created.

What are the advantages of Data Flows?

With **advanced data preparation available in** Power BI, you can create a collection of data called a dataflow, **which you can then use to connect with business data from various sources, clean the data, transform it, and then load it to Power BI storage.**

Power BI introduces dataflows **to help organizations unify data from disparate sources and prepare it for modeling.** Analysts can easily create dataflows, using familiar, self-service tools. **Dataflows are used to ingest, transform, integrate, and enrich big data by defining data source connections, ETL logic, refresh schedules, and more.**

Dataflows are designed to use the **Common Data Model**, a standardized, modular, extensible collection of data schemas published by Microsoft that are designed to make it easier for you to build, use, and analyze data. With this model, you can go from data sources to Power BI dashboards with nearly zero friction.

Data flow features [brief points]

1.Data is stored as entities in the **Common Data Model** in **Azure Data Lake Storage Gen2.**

2.Dataflows are created and managed in **app workspaces** by using the Power BI service.

3. **You can use dataflows to ingest data from a large and growing set of supported on-premises and cloud- based data sources** including Dynamics 365, Salesforce, Azure SQL Database, Excel, SharePoint, and more.

4. You **can then map data to standard entities in the Common Data Model, modify and extend existing entities, and create custom entities**. Advanced users can create fully customized dataflows, **using a self-service, low- code/no-code, built-in Power Query authoring experience, similar to the Power Query experience** that millions of Power BI Desktop and Excel users already know.

5. **Once you've created a dataflow, you can use Power BI Desktop and the Power BI service to create datasets, reports, dashboards, and apps** that leverage the power of the Common Data Model to drive deep insights into your business activities.

6. Dataflow refresh scheduling is managed directly from the workspace in which your dataflow was created, **just like your datasets.**

Dataflow working process

Here are some examples of how dataflows can work for you:

- **Organizations can map their data to standard entities in the Common Data Model or create their own custom entities. These entities can then be used as building blocks to create reports, dashboards,** and apps that work out of the box, and distribute them to users across their organization.
- Using the extensive collection of Microsoft data connectors, **organizations can connect their own data sources to dataflows, using Power Query to map the data from its origin and bring it into Power BI**. Once that data is imported by a dataflow (and refreshed at a specified frequency), those dataflow entities may be used in the Power BI Desktop application to create compelling reports and dashboards.

Create and use data flows

You must have a paid Power BI account to use dataflows, such as a **Power BI Pro or Power BI Premium account**, but you are not charged separately for using dataflows.

Extend the common data model for your business needs

For organizations **that want to extend the Common Data Model (CDM), dataflows enable business intelligence professionals to customize the standard entities**, or create new ones. This self-service approach to customizing the data model can then be used with dataflows to build apps and Power BI dashboards that are tailored to an organization.

Define dataflows programmatically

You might also want to develop **your own programmatic solutions to create dataflows. With public APIs and the capability to programmatically create custom dataflow definition files (model.json),** you create a custom solution that fits your organization's unique data and analytics needs.

Public APIs allow developers simple and easy ways to interact with Power BI and dataflows.

Extend your capabilities with Azure

Azure Data Lake Storage Gen2 is included with every paid Power BI subscription (10 GB per user, 100 TB per P1 node). So, you can easily get started with self-service data prep on Azure Data Lake.

Power BI can be configured to store dataflow data in your organization's Azure Data Lake Storage Gen2 account. **When Power BI is connected to your Azure subscription, data**

developers and data scientists can leverage powerful Azure products such as Azure Machine Learning, Azure Databricks, Azure Data Factory, and more.

Power BI can also connect to folders with schematized data in the Common Data Model format, which are stored in your organization's Azure Data Lake Storage account. These folders can be created by services such as Azure data services. By connecting to these folders, analysts can work seamlessly with this data in Power BI.

Dataflow capabilities on Power BI Premium

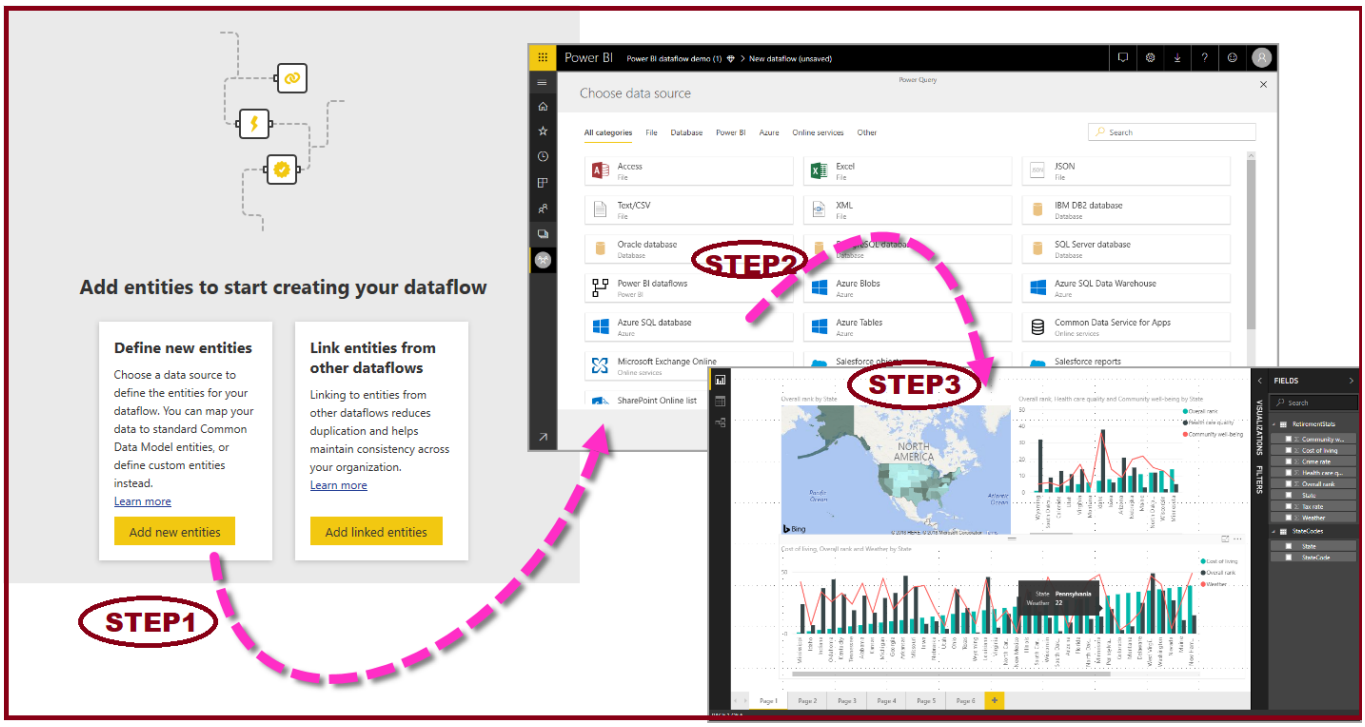
For dataflow features and workloads to operate on a Power BI Premium subscription, the dataflow workload for that Premium capacity must be turned on. The following table describes dataflow features and their capacities when using a Power BI Pro account, and how that compares to using Power BI Premium.

Dataflow capability	Power BI Pro	Power BI Premium
Scheduled refresh	8 per day	48
Total Storage	10 GB/user	100 TB/node
Dataflow Authoring with Power Query Online	+	+
Dataflow Management within Power BI	+	+
Dataflows Data Connector in the Power BI Desktop	+	+
Integration with Azure	+	+
Computed Entities (in-storage transformations via M)		+
New connectors	+	+
Dataflow incremental refresh		+

Dataflow capability	Power BI Pro	Power BI Premium
Running on Power BI Premium capacity / Parallel execution of transforms		+
Dataflow linked entities		+
Standardized Schema / Built-In Support for the Common Data Model	+	+

Can we implement incremental refresh in Pro account?

Dataflow workloads **are not currently available in multi-geo capacities.**



What is entity?

An **entity** is a set of fields that are used to store data, much like a table within a database.

PRACTICAL ON DATA FLOW WITH CDM [COMMON DATA MODEL]

PRIMARY STEPS FOR DATA FLOW USAGE

There are **three primary steps** to using a dataflow:

1. Author the dataflow, using Microsoft tools that are designed to make doing so straightforward
2. Schedule the refresh frequency of the data you want to bring into your dataflow
3. Build the dataset using your dataflow, using Power BI Desktop

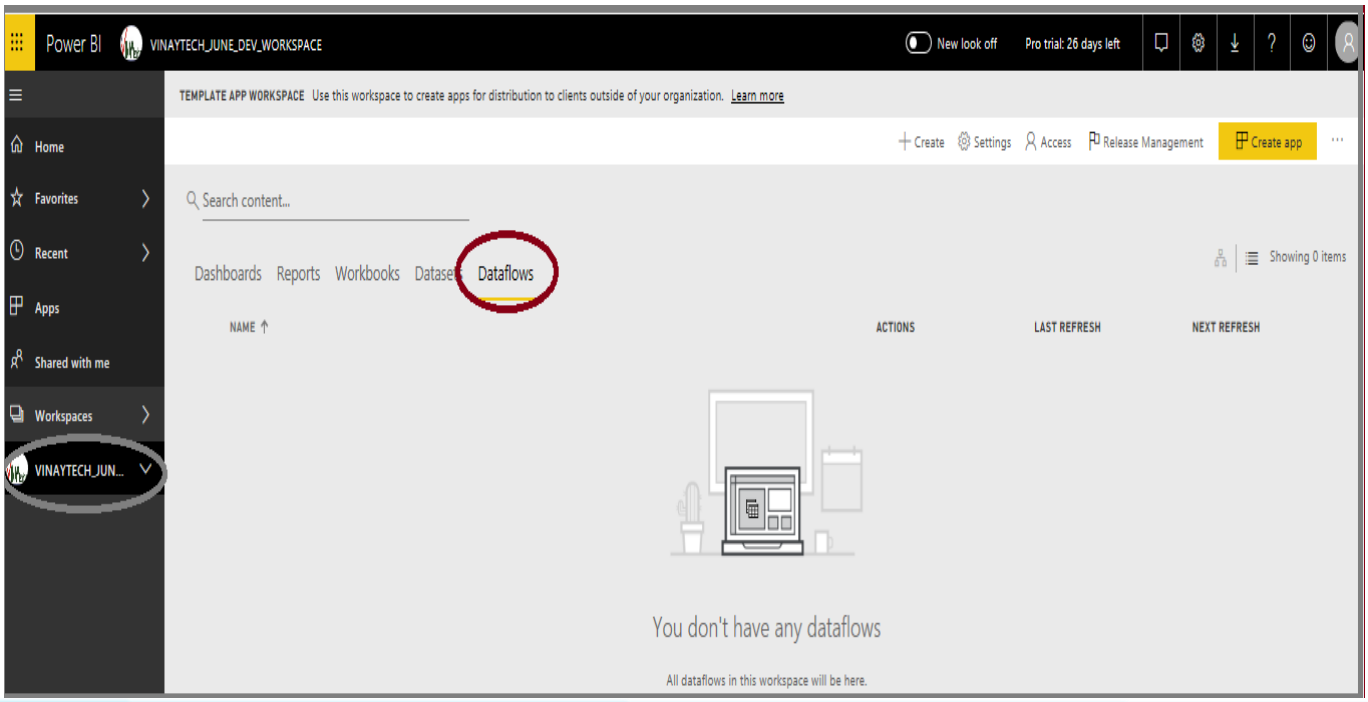
STEP -1: PRE-REQUISITE

A) VALID ACCOUNT [PRO OR PREMIUM] required

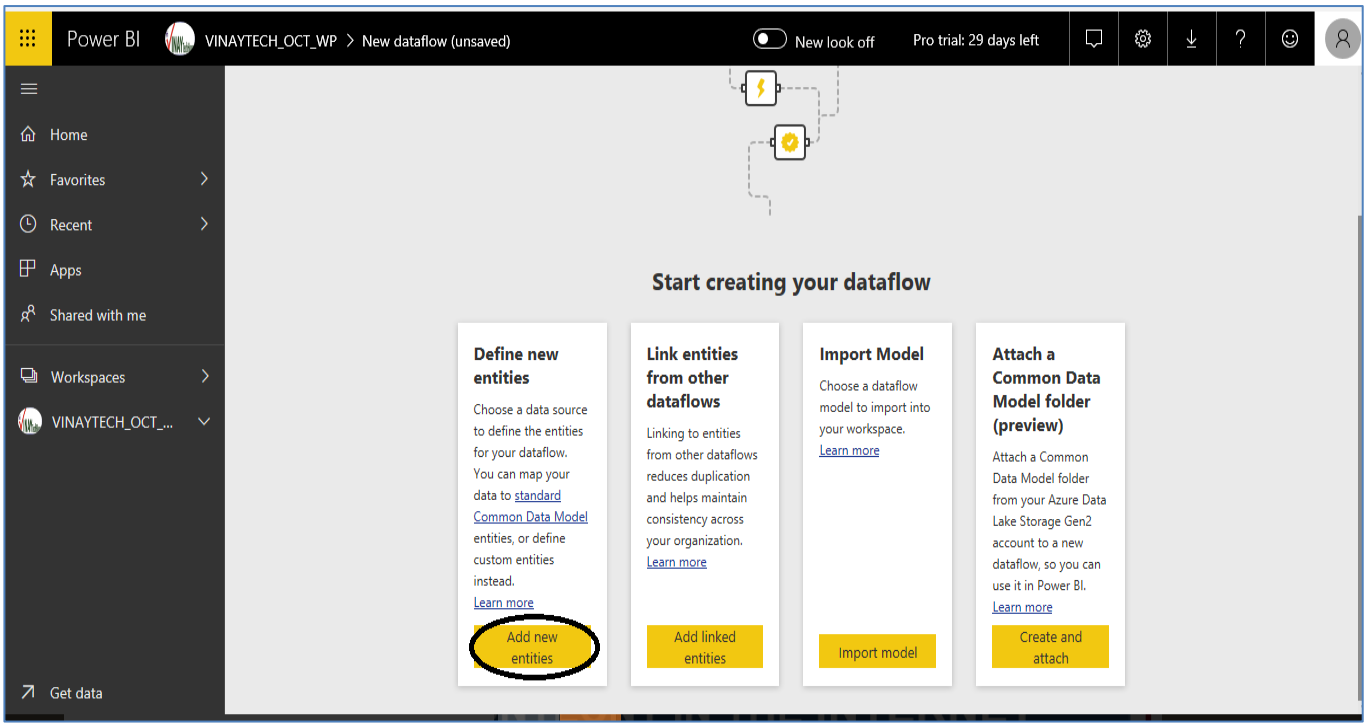
B) A DATA GATEWAY [NOT PERSONAL] WITH VALID AND APPROPRIATE DATA SOURCE IS REQUIRED

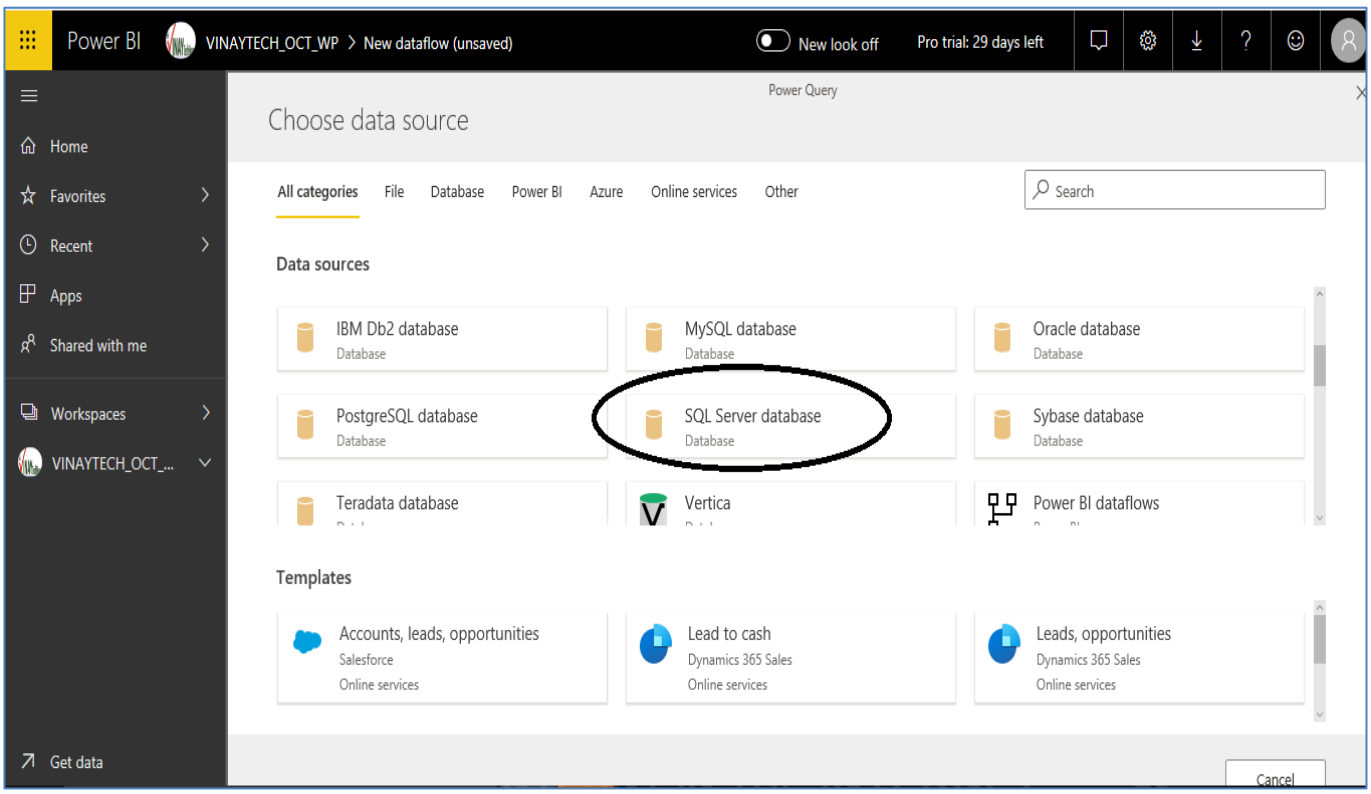
STEP -2: CREATING A DATAFLOW

Launch Power BI service in a browser and then select an **app workspace** (dataflows are not available in *my-workspace* in the Power BI service) from the navigation pane on the left, as shown in the following screen.



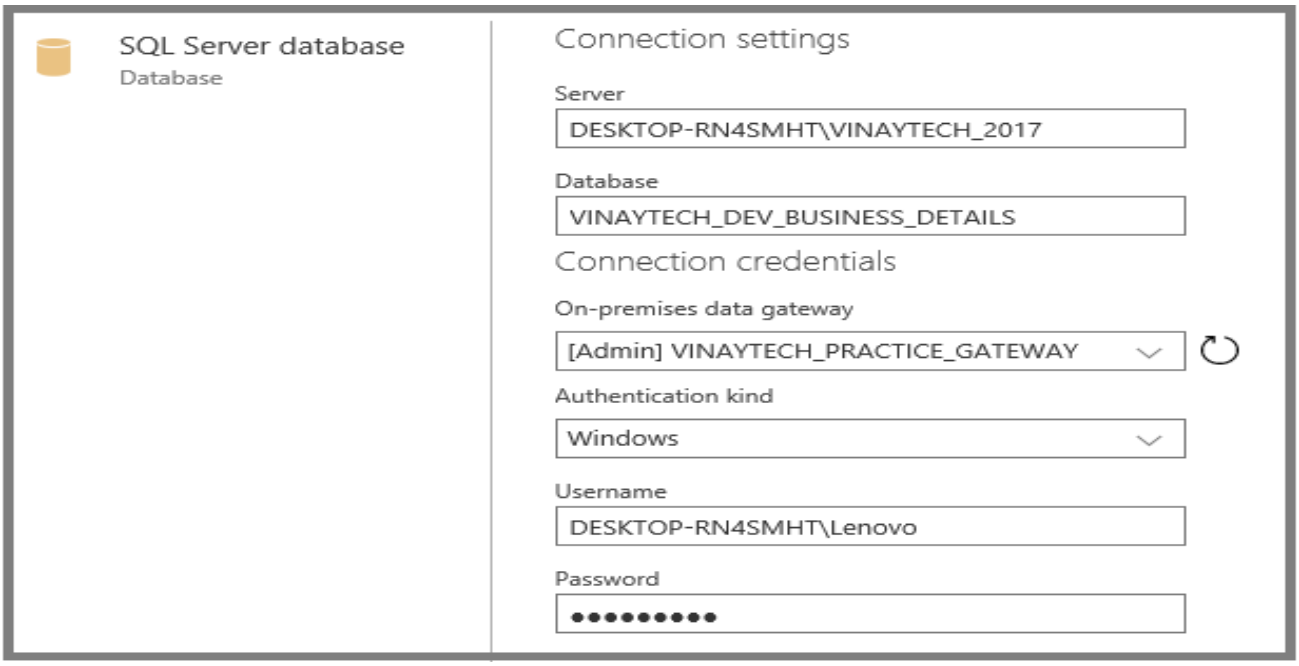
STEP -3: Click Add new entities, choose SQL Server database





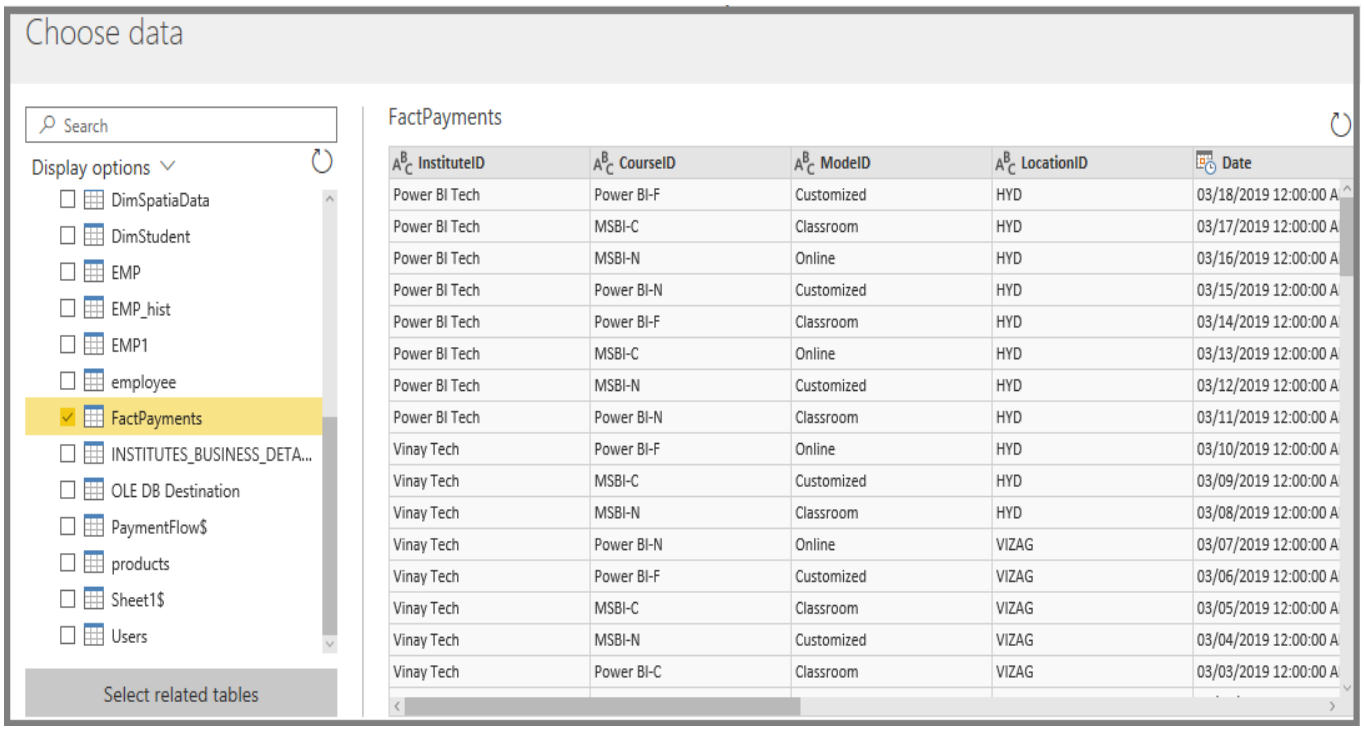
When you select a data source, you're prompted to provide the connection settings, including the account to use when connecting to the data source, as shown in the following image.

STEP -4: Specify the connection string and the gateway to be used



Once connected, you can select which data to use for your entity. When you choose data and a source, Power BI will subsequently reconnect to the data source in order to keep the data in your dataflow refreshed, at the frequency you select later in the setup process.

STEP -5: Choose the tables to be used in the model

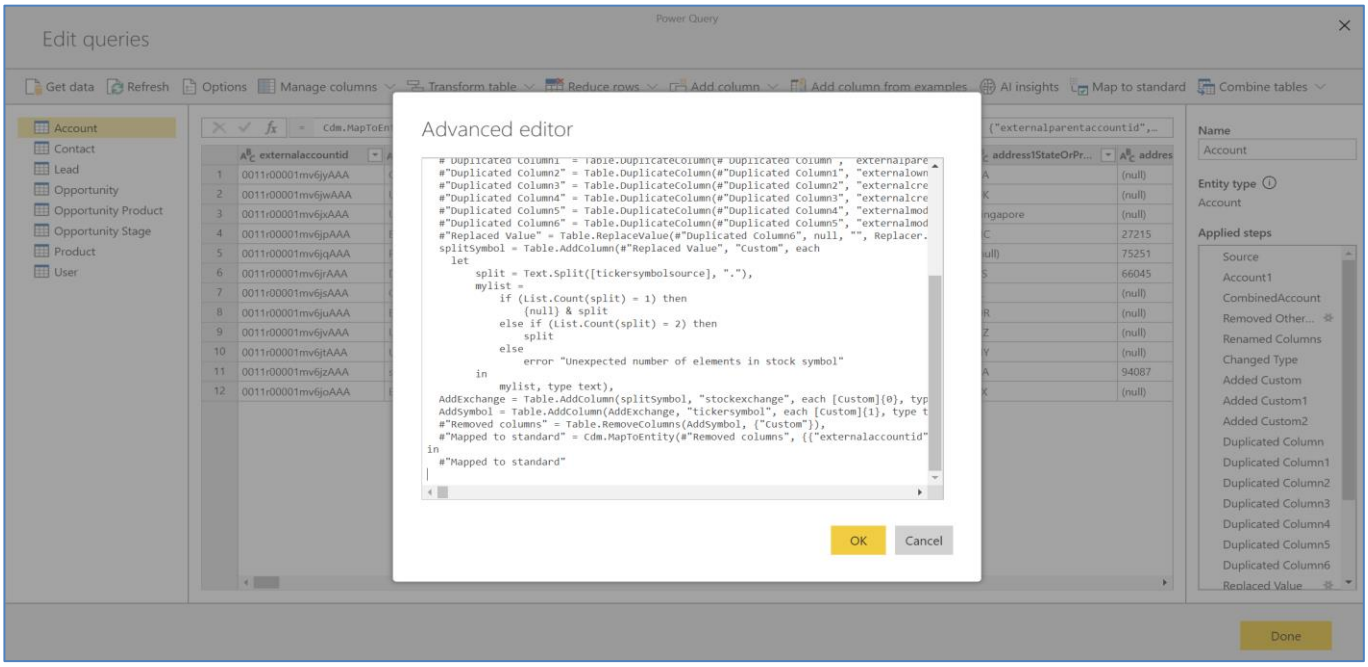


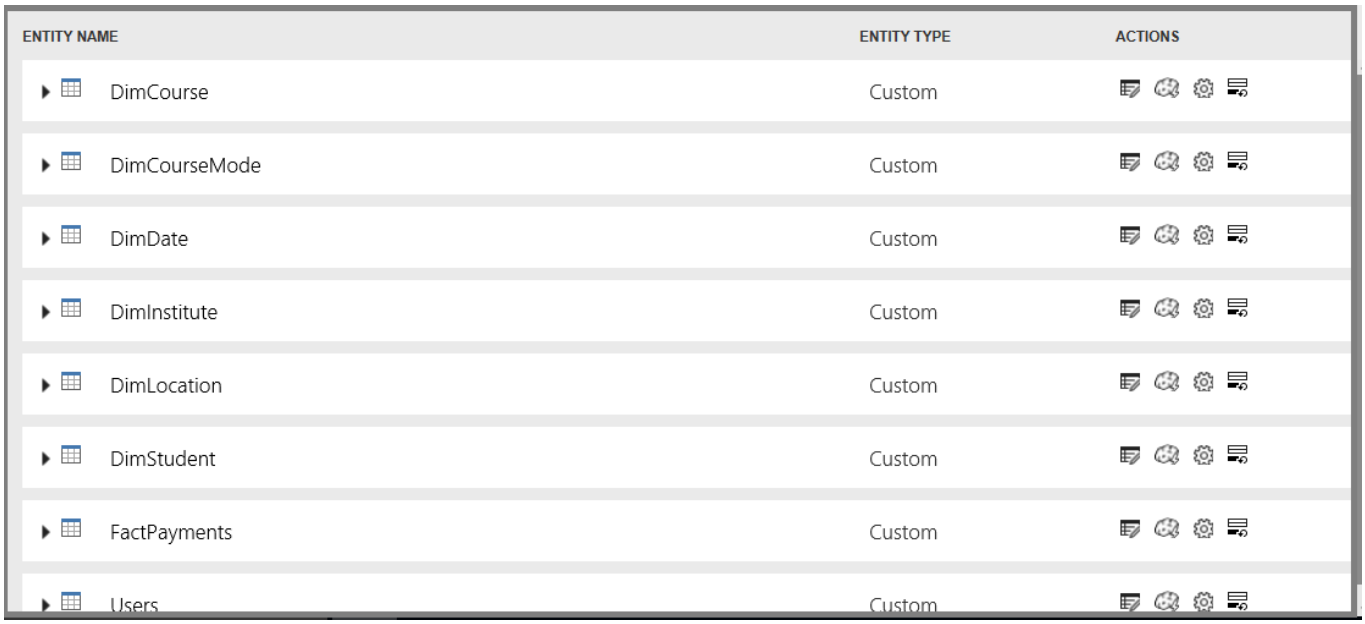
InstituteID	CourseID	ModelID	LocationID	Date
Power BI Tech	Power BI-F	Customized	HYD	03/18/2019 12:00:00 A
Power BI Tech	MSBI-C	Classroom	HYD	03/17/2019 12:00:00 A
Power BI Tech	MSBI-N	Online	HYD	03/16/2019 12:00:00 A
Power BI Tech	Power BI-N	Customized	HYD	03/15/2019 12:00:00 A
Power BI Tech	Power BI-F	Classroom	HYD	03/14/2019 12:00:00 A
Power BI Tech	MSBI-C	Online	HYD	03/13/2019 12:00:00 A
Power BI Tech	MSBI-N	Customized	HYD	03/12/2019 12:00:00 A
Power BI Tech	Power BI-N	Classroom	HYD	03/11/2019 12:00:00 A
Vinay Tech	Power BI-F	Online	HYD	03/10/2019 12:00:00 A
Vinay Tech	MSBI-C	Customized	HYD	03/09/2019 12:00:00 A
Vinay Tech	MSBI-N	Classroom	HYD	03/08/2019 12:00:00 A
Vinay Tech	Power BI-N	Online	VIZAG	03/07/2019 12:00:00 A
Vinay Tech	Power BI-F	Customized	VIZAG	03/06/2019 12:00:00 A
Vinay Tech	MSBI-C	Classroom	VIZAG	03/05/2019 12:00:00 A
Vinay Tech	MSBI-N	Customized	VIZAG	03/04/2019 12:00:00 A
Vinay Tech	Power BI-C	Classroom	VIZAG	03/03/2019 12:00:00 A

Once you select the data for use in the entity, you can use dataflow editor to shape or transform that data into the format necessary for use in your dataflow.

Step -6: Using the dataflow editor to perform various operations [duplicate removal, adding columns, data type conversion etc…]

Once you select which data from your source to use for your entity, you can shape your data selection into a form that works best for your entity, using a Power Query editing experience, similar to the **Power Query Editor** in **Power BI Desktop**.

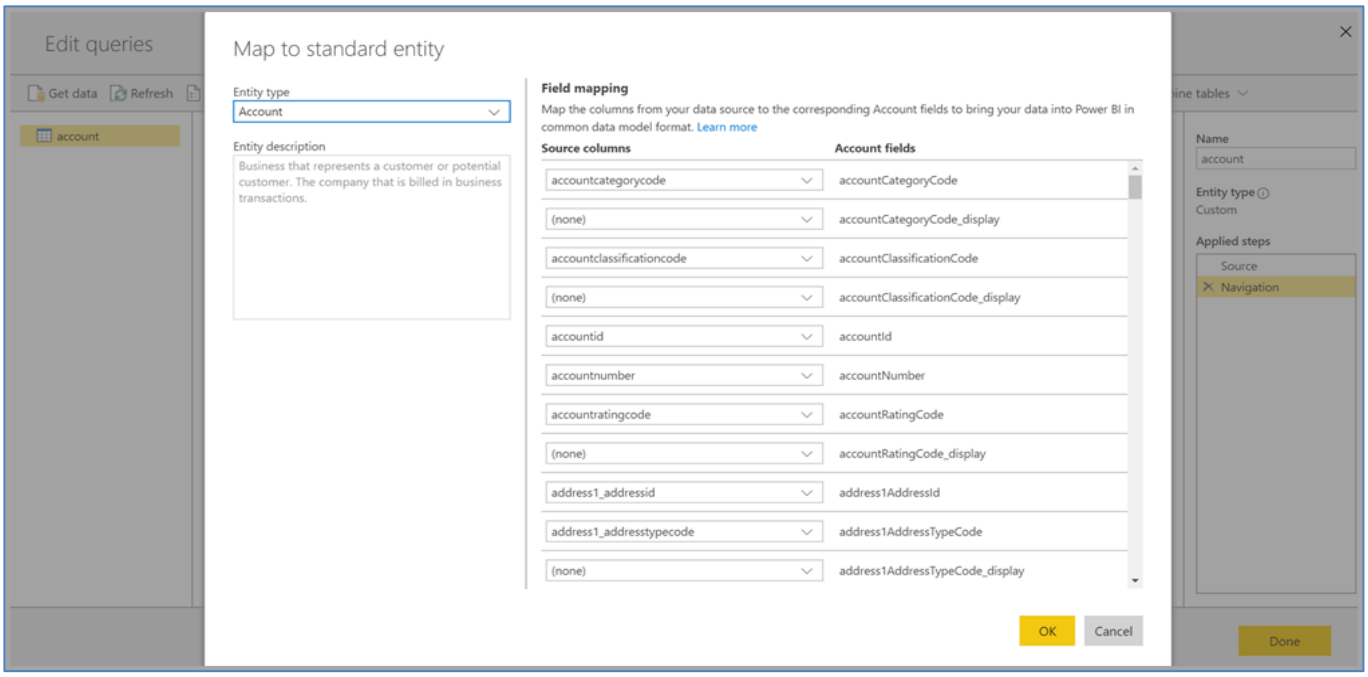




Step -7: [Optional] Dataflows and the Common Data Model (CDM)

Dataflows entities include new tools to easily map your business data to the Common Data Model (Microsoft's standardized schema), enrich it with Microsoft and third-party data, and gain simplified access to machine learning. These new capabilities can be leveraged to provide intelligent and actionable insights into your business data. Once you've completed any transformations in the Edit Queries step, you can map columns from your data source tables to standard entity fields as defined by the Common Data Model. Standard entities have a known schema defined by the common data model.

To leverage the Common Data Model with your dataflow, click on the **Map to Standard** transformation in the **Edit Queries** dialog. In the **Map Entities** screen that appears, you can select the standard entity to which you want to map.



When you map a source column to standard field, the following occurs:

1. The source column takes on the standard field name (the column is renamed if the names are different)
2. The sources column gets the standard field data type

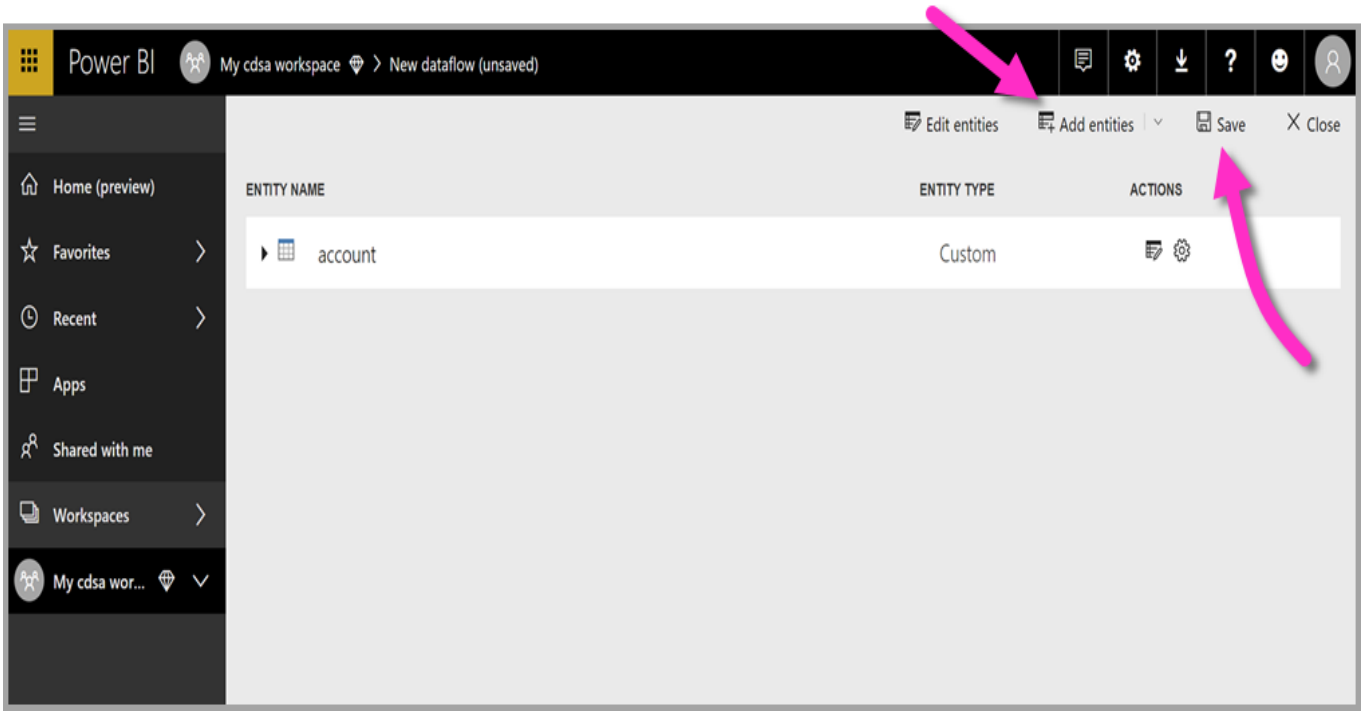
To keep the Common Data Model standard entity, all standard fields that are not mapped **get *Null* values**.

All source columns that are not mapped remain as-is, to ensure that the result of the mapping is a standard entity with custom fields.

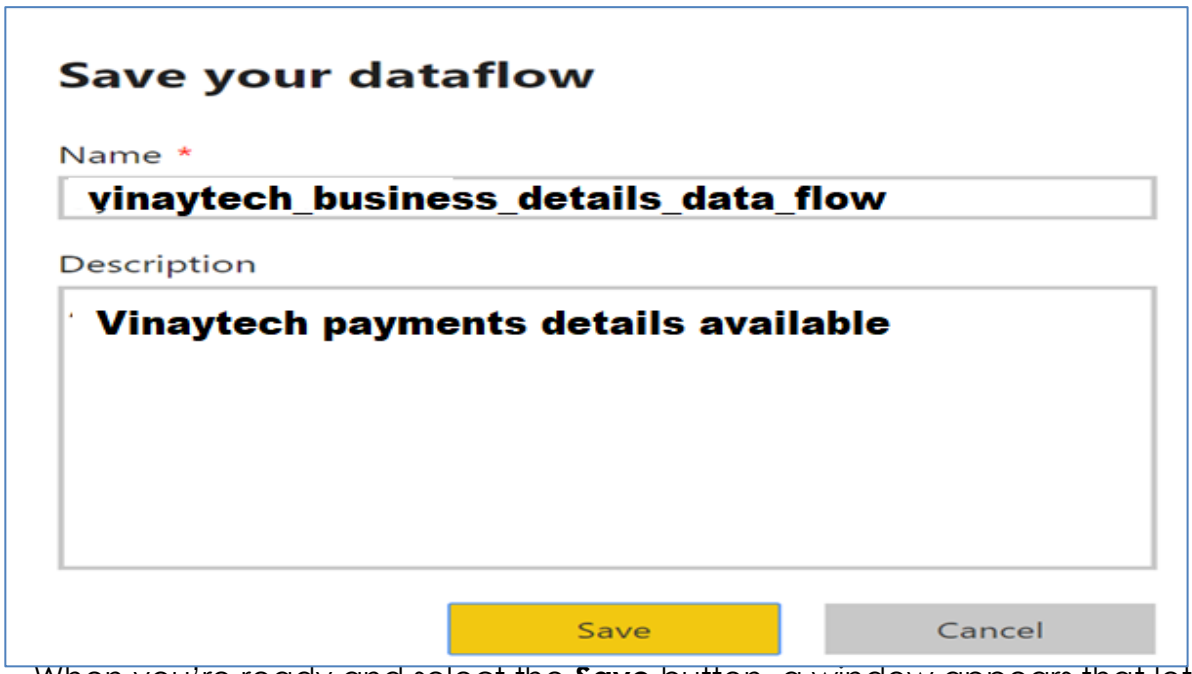
STEP -8: Saving Dataflow, specify a proper name to recognize it easily

Once you've completed your selections and your entity and its data settings are ready to save, you can select **Save** from the menu. Note that you can create multiple entities by selecting the **Add entities** button, and that you can Edit entities to refine the queries and entities you've created.

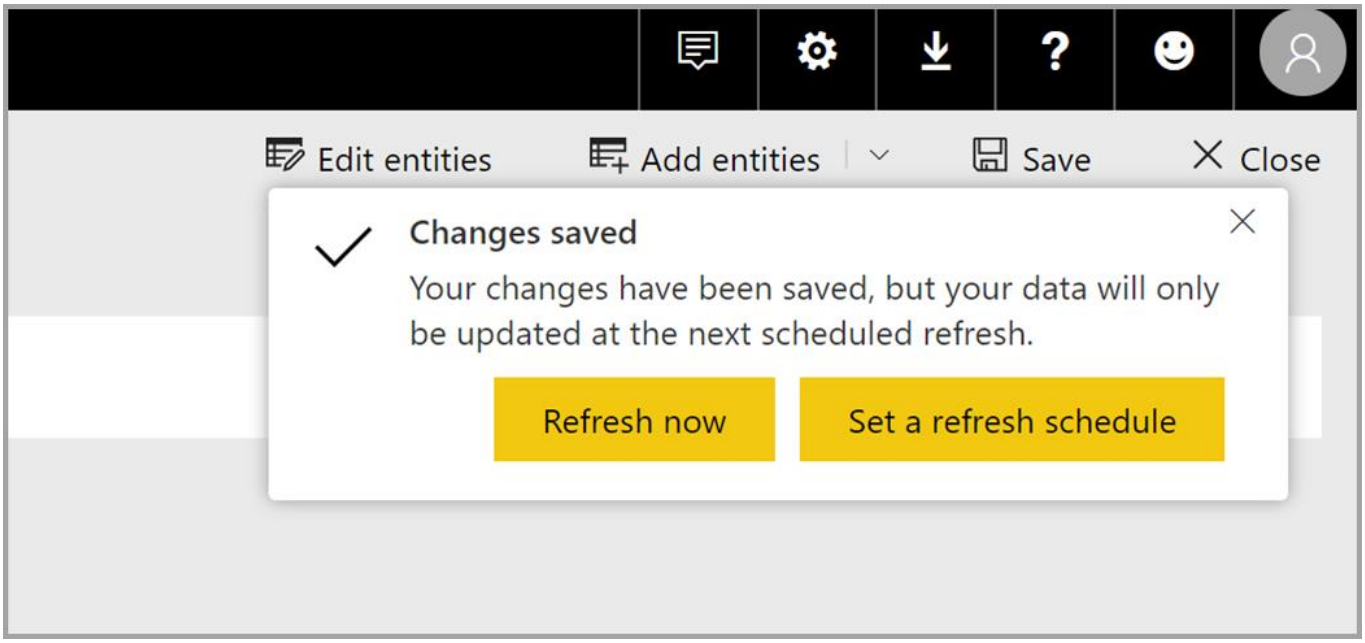
Once you've completed your selections and your entity and its data settings are ready to save, you can select **Save** from the menu. Note that you can create multiple entities by selecting the **Add entities** button, and that you can Edit entities to refine the queries and entities you've created.



When you select **Save**, you're prompted to name your dataflow and provide a description.



When you're ready and select the **Save** button, a window appears that lets you know your **dataflow** has been created.



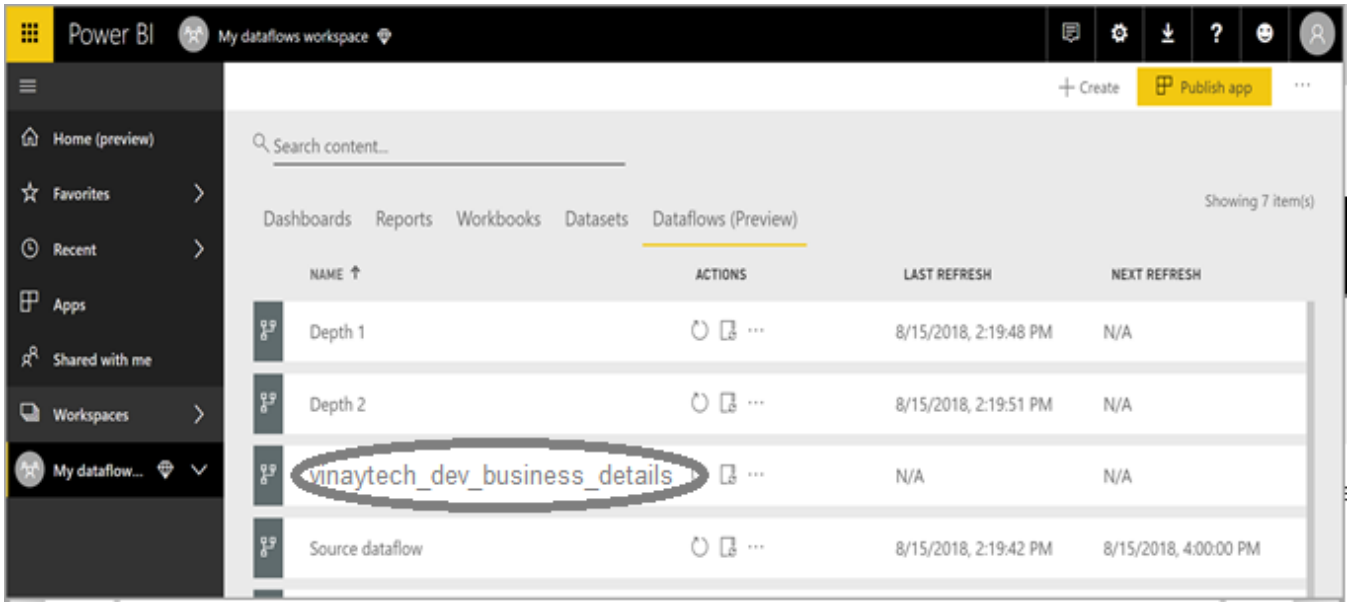
Now you're ready for the next step, which is scheduling the refresh frequency of your data sources.

STEP -9: [Optional] Schedule the refresh frequency

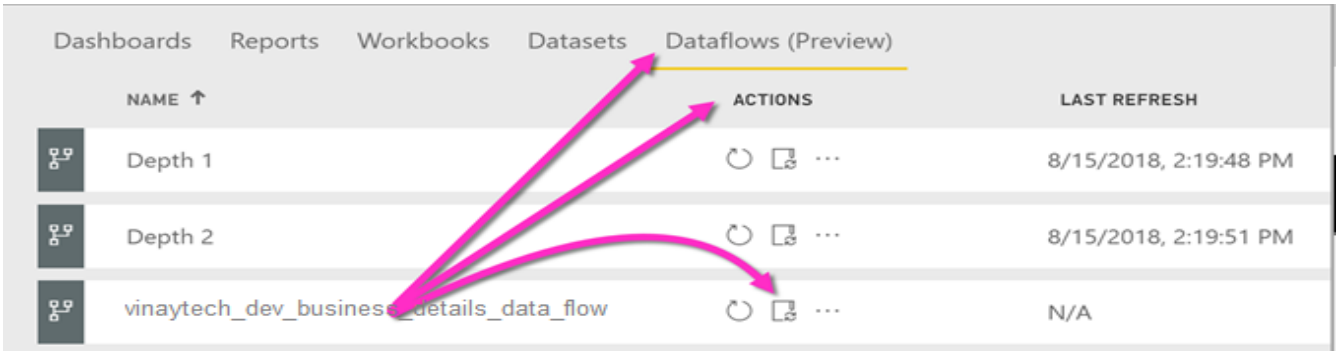
Once your dataflow has been saved, you'll want to schedule the refresh frequency for each of your connected data sources.

Power BI dataflows use the Power BI data refresh process to keep your data up to date. In the **Power BI service**, in the **app workspace** section, there's a collection of areas

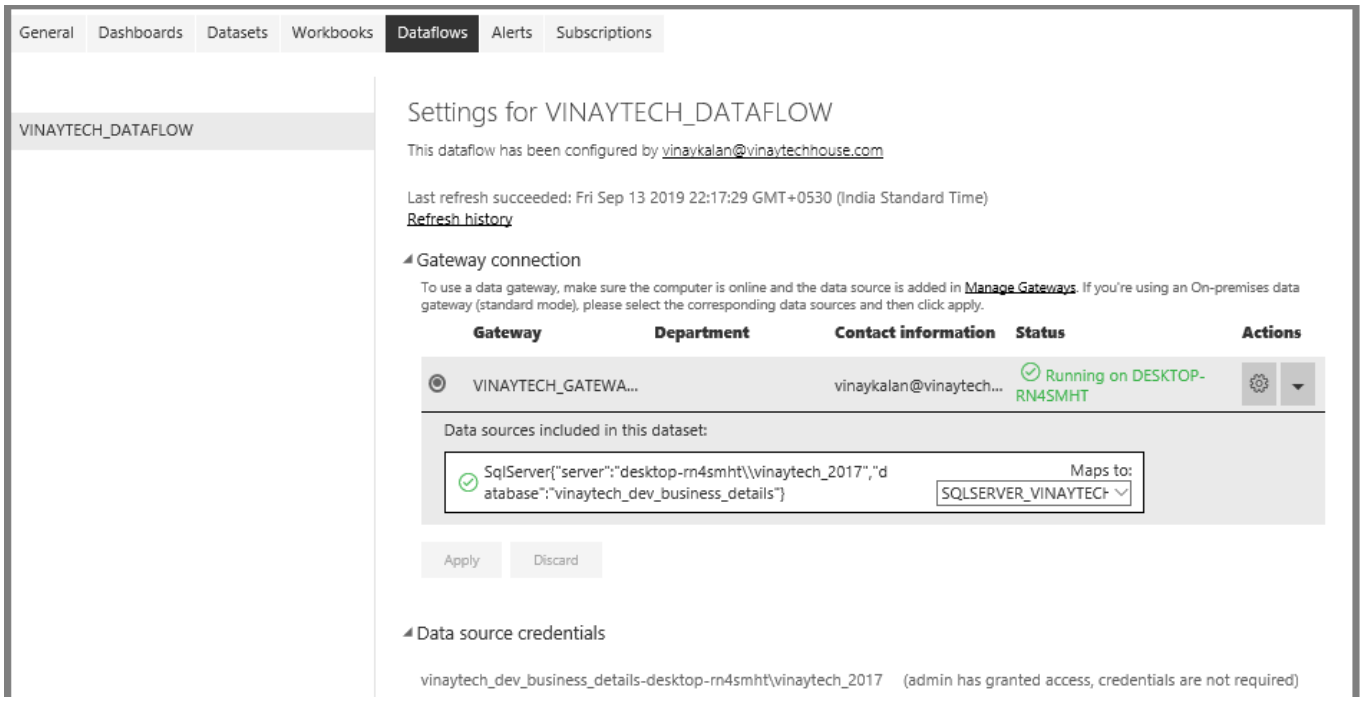
where your information can be listed, including dataflows, as shown in the following image.

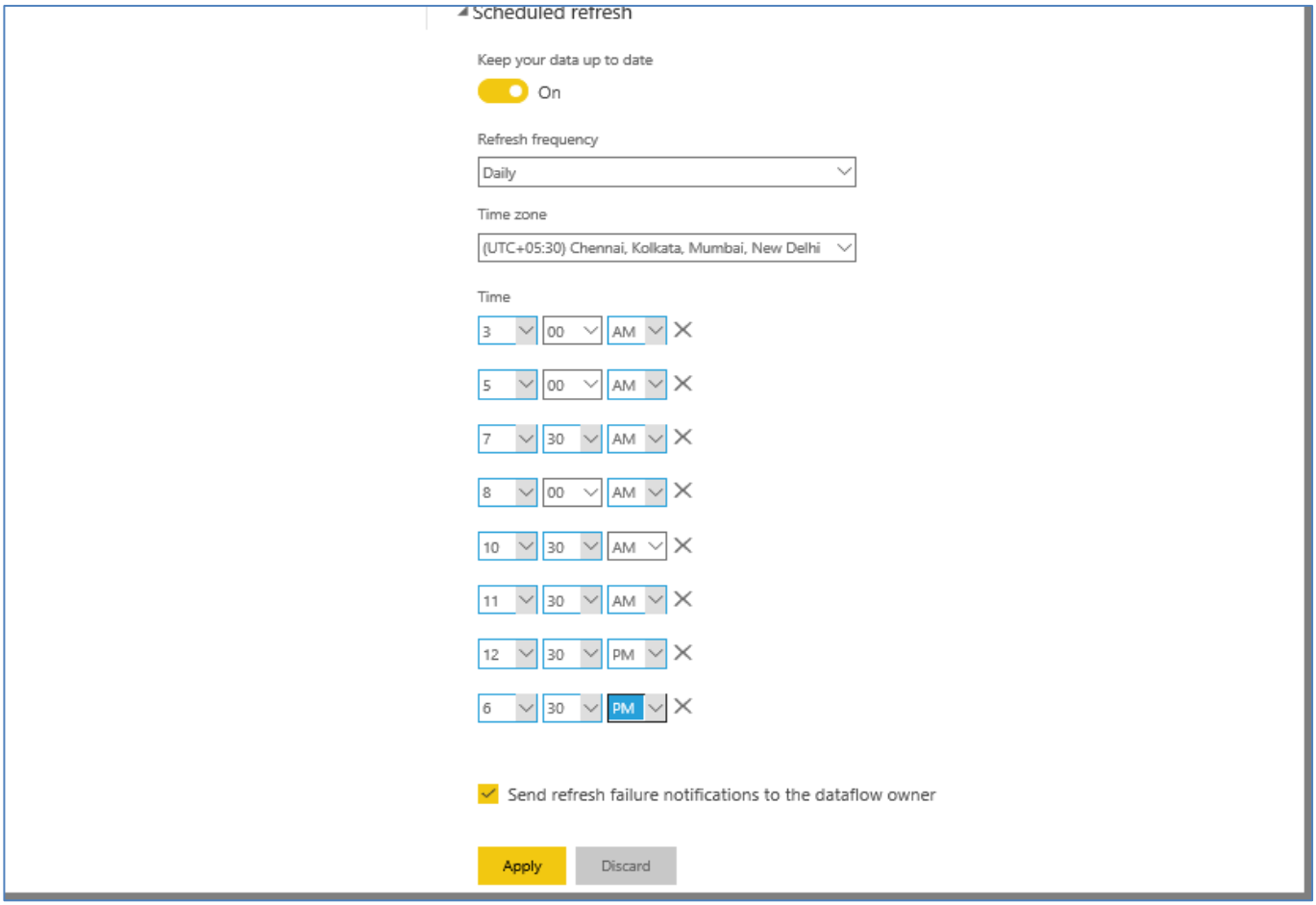


The ***Dynamics dataflow* entry in the previous image** is the dataflow we created in the previous section. To schedule refresh, select the **Schedule refresh** icon, under the **Actions** section, as shown in the following image.



When you select the **Schedule refresh** icon you're taken to the **Schedule refresh** pane, which lets you set the dataflow refresh frequency and time.



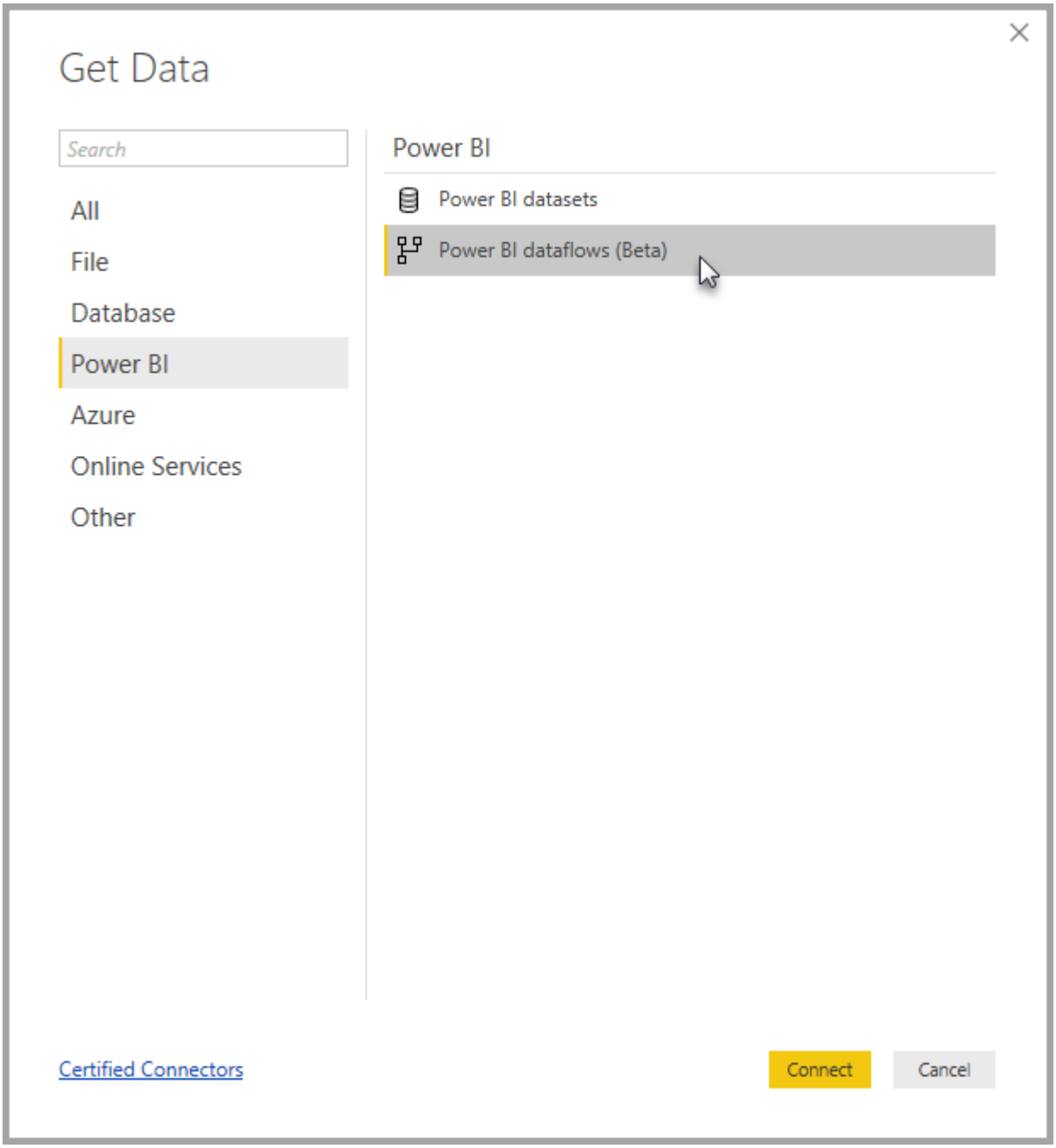


Dataflows behave the same behavior as Power BI datasets in terms of refresh settings.

Step -10: Connect to your dataflow in Power BI Desktop

Once you've created your dataflow and you have scheduled the refresh frequency for each data source that will populate the model, you're ready for the third and final step, which is connecting to your dataflow from within **Power BI Desktop**.

To connect to the dataflow, in Power BI Desktop select **Get Data > Power BI > Power BI dataflows (Beta)** as shown in the following image.



From there, navigate to the **app workspace** where you saved your dataflow, select the dataflow and then select the entities that you created from the list.

You can also use the **search bar**, near the top of the window, to quickly find the name of your dataflow or entities from among many dataflow entities.

When you select the entity and then select the **Load** button, the entities appear in the **Fields** pane in **Power BI Desktop**, and appear and behave just like **tables** from any other dataset.

Additional Practice:

Step -1: Using dataflows stored in Azure Data Lake Storage Gen2

Some organizations may want to use their own storage for creation and management of dataflows. You can integrate dataflows with Azure Data Lake Storage Gen2, if you follow the requirements and manage permissions properly.

Troubleshooting data connections

There may be occasions when connecting to data sources for dataflows run into issues. This section provides troubleshooting tips when such issues arise.

- **Salesforce connector** - Using a trial account for Salesforce with dataflows results in a connection failure with no information provided. To resolve this, use a production Salesforce account or a developer account for testing.
- **SharePoint connector** - Make sure you supply the root address of the SharePoint site, without any subfolders or documents. For example, use link similar to the following: https://microsoft.sharepoint.com/teams/ObjectModel/
- **JSON File connector** - Currently you can connect to a JSON file using basic authentication only. Connecting to a JSON file by providing the credentials within the URL (for example, https://XXXXX.blob.core.windows.net/path/file.json?sv=2019-01-01&si=something&sr=c&sig=123456abcdefg...) is **not** currently supported.
- **Azure SQL Data Warehouse** - Dataflows do not currently support Azure Active Directory (AAD) authentication for Azure SQL Data Warehouse. Use Basic authentication for this scenario.

ACTIVE DIRECTORYAND AZURE ACTIVE DIRECTORY

What is Active Directory?

Active Directory is essentially a database **that helps organize your company's users, computers and more.**

A) It provides authentication and authorization to applications, file services, printers, and other on-premises resources.

B) It uses protocols such as Kerberos and NTLM for authentication and LDAP to query and modify items in the AD databases.

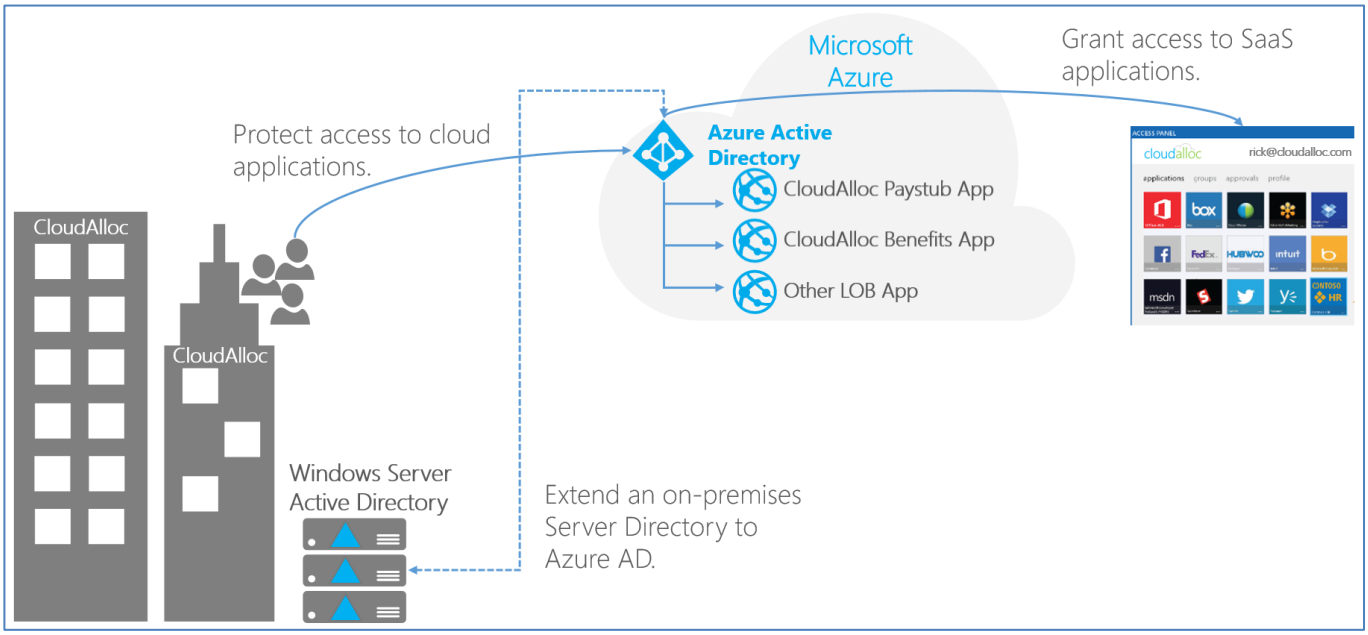
C) First released with Windows 2000 Server edition.

What is Azure Active Directory?

Azure Active Directory (aka **Azure AD**) is a fully managed multi-tenant service from Microsoft that offers identity and access capabilities for applications running in Microsoft **Azure** and for applications running in an on-premises environment. ... **Azure AD** is not a replacement for Windows Server **Active Directory**.

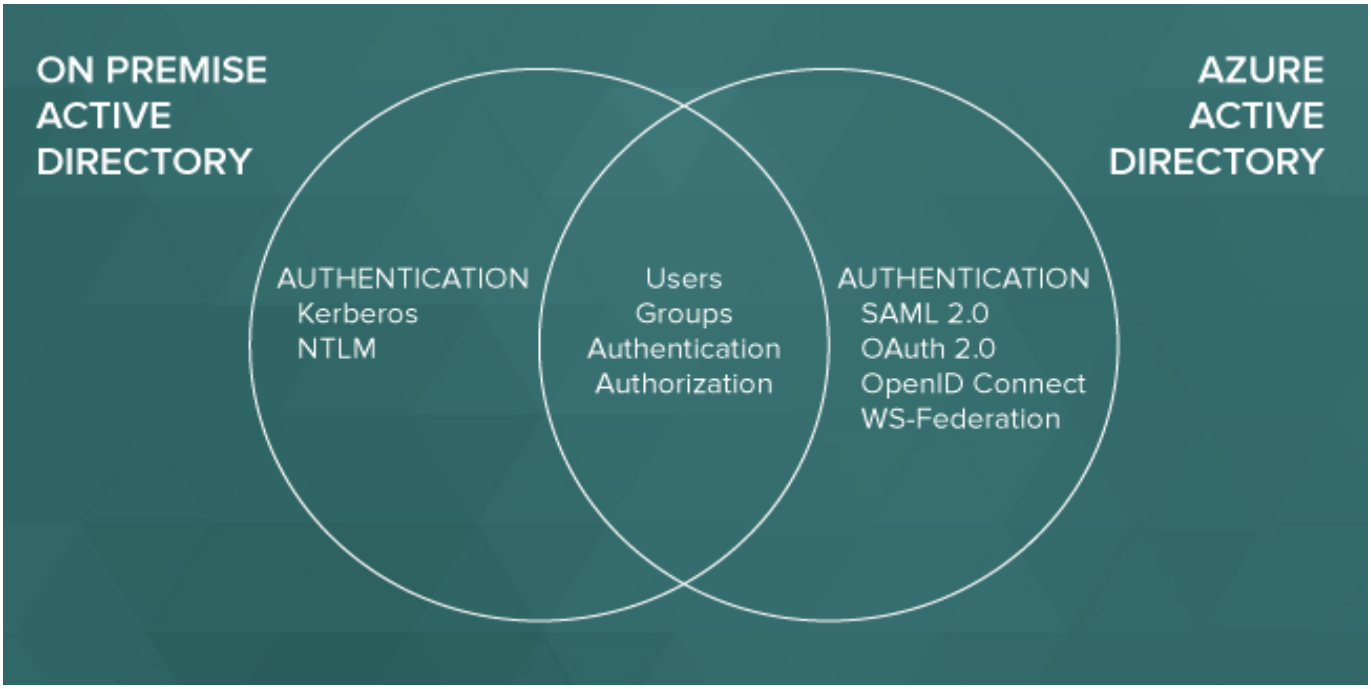
a) Was designed to support web-based services that use **REST (REpresentational State Transfer)** API interfaces for Office 365, Salesforce.com etc.

b) It uses completely different protocols (Goodbye, Kerberos, and NTLM) that work with these services–protocols such as **SAML and OAuth 2.0.**



AZURE ACTIVE DIRECTORY

Normal Directory [Windows Directory]	Azure Active Directory
Wasn't designed to manage web-based services.	Was designed to support web-based services that use REST (REpresentational State Transfer) API



What is oAuth2.0?

OAuth 2.0 is a protocol that allows a user to grant limited access to their resources on one site, to another site, without having to expose their credentials.

Introduction

OAuth 2 is an authorization framework that enables applications to obtain limited access to user accounts on an HTTP service, such as Facebook, GitHub, and DigitalOcean. It works by delegating user authentication to the service that hosts the user account, and authorizing third-party applications to access the user account. OAuth 2 provides authorization flows for web and desktop applications, and mobile devices.

This informational guide is geared towards application developers, and provides an overview of OAuth 2 roles, authorization grant types, use cases, and flows.

Let's get started with OAuth Roles!

OAuth Roles

OAuth defines four roles:

- Resource Owner
- Client
- Resource Server
- Authorization Server

We will detail each role in the following subsections.

Resource Owner: *User*

The resource owner is the *user* who authorizes an *application* to access their account. The application's access to the user's account is limited to the "scope" of the authorization granted (e.g. read or write access).

Resource / Authorization Server: *API*

The resource server hosts the protected user accounts, and the authorization server verifies the identity of the *user* then issues access tokens to the *application*.

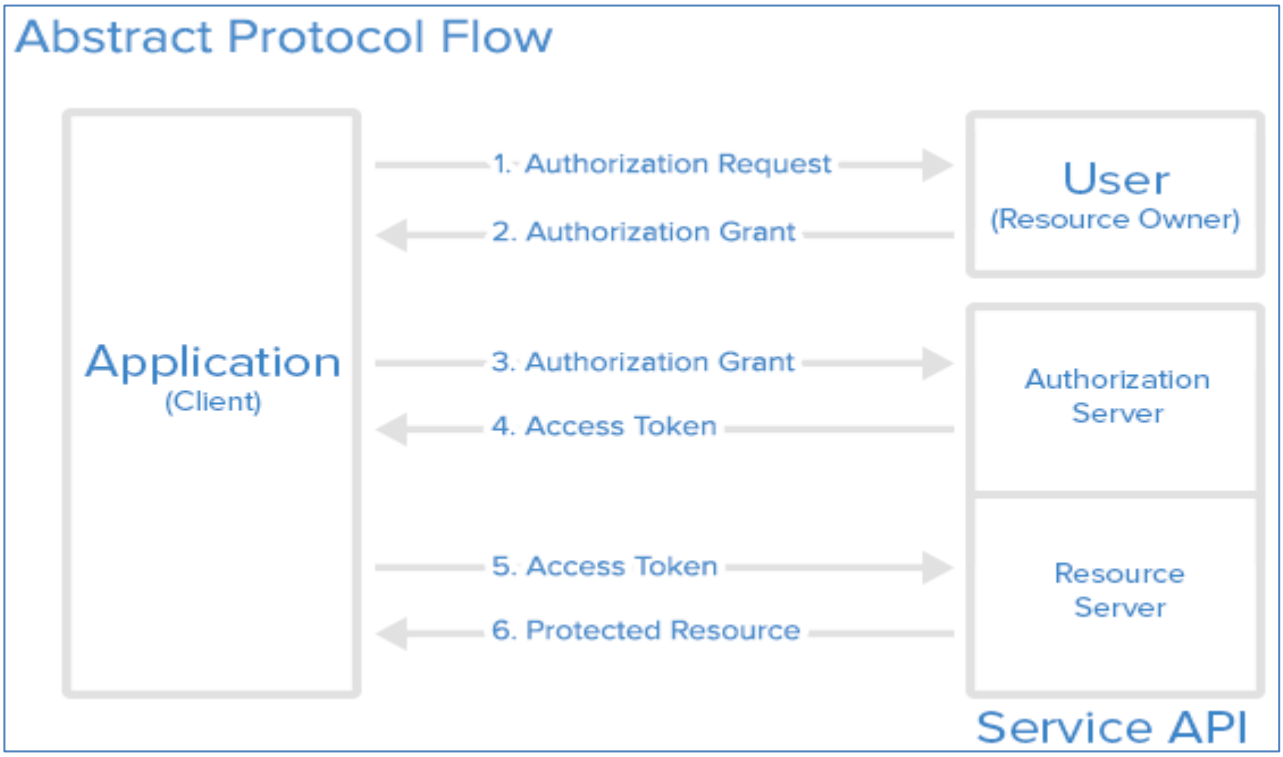
From an application developer's point of view, a service's API fulfills both the resource and authorization server roles. We will refer to both of these roles combined, as the *Service* or *API* role.

Client: *Application*

The client is the *application* that wants to access the *user*'s account. Before it may do so, it must be authorized by the user, and the authorization must be validated by the API.

Abstract Protocol Flow

Now that you have an idea of what the OAuth roles are, let's look at a diagram of how they generally interact with each other:



Here is a more detailed explanation of the steps in the diagram:

1. The *application* requests authorization to access service resources from the *user*
2. If the *user* authorized the request, the *application* receives an authorization grant
3. The *application* requests an access token from the *authorization server* (API) by presenting authentication of its own identity, and the authorization grant
4. If the application identity is authenticated and the authorization grant is valid, the *authorization server* (API) issues an access token to the application. Authorization is complete.
5. The *application* requests the resource from the *resource server* (API) and presents the access token for authentication
6. If the access token is valid, the *resource server* (API) serves the resource to the *application*

The actual flow of this process will differ depending on the authorization grant type in use, but this is the general idea. We will explore different grant types in a later section.

Application Registration

Before using OAuth with your application, you must register your application with the service. This is done through a registration form in the "developer" or "API" portion of the service's website, where you will provide the following information (and probably details about your application):

- Application Name
- Application Website

- Redirect URI or Callback URL

The redirect URI is where the service will redirect the user after they authorize (or deny) your application, and therefore the part of your application that will handle authorization codes or access tokens.

Client ID and Client Secret

Once your application is registered, the service will issue "client credentials" in the form of a *client identifier* and a *client secret*. The Client ID is a publicly exposed string that is used by the service API to identify the application, and is also used to build authorization URLs that are presented to users. The Client Secret is used to authenticate the identity of the application to the service API when the application requests to access a user's account, and must be kept private between the application and the API.

Authorization Grant

In the *Abstract Protocol Flow* above, the first four steps cover obtaining an authorization grant and access token. The authorization grant type depends on the method used by the application to request authorization, and the grant types supported by the API. OAuth 2 defines four grant types, each of which is useful in different cases:

- Authorization Code: used with server-side Applications
- Implicit: used with Mobile Apps or Web Applications (applications that run on the user's device)
- Resource Owner Password Credentials: used with trusted Applications, such as those owned by the service itself
- Client Credentials: used with Applications API access

Power BI Developer Center

What is the role of Power BI Developer Center?

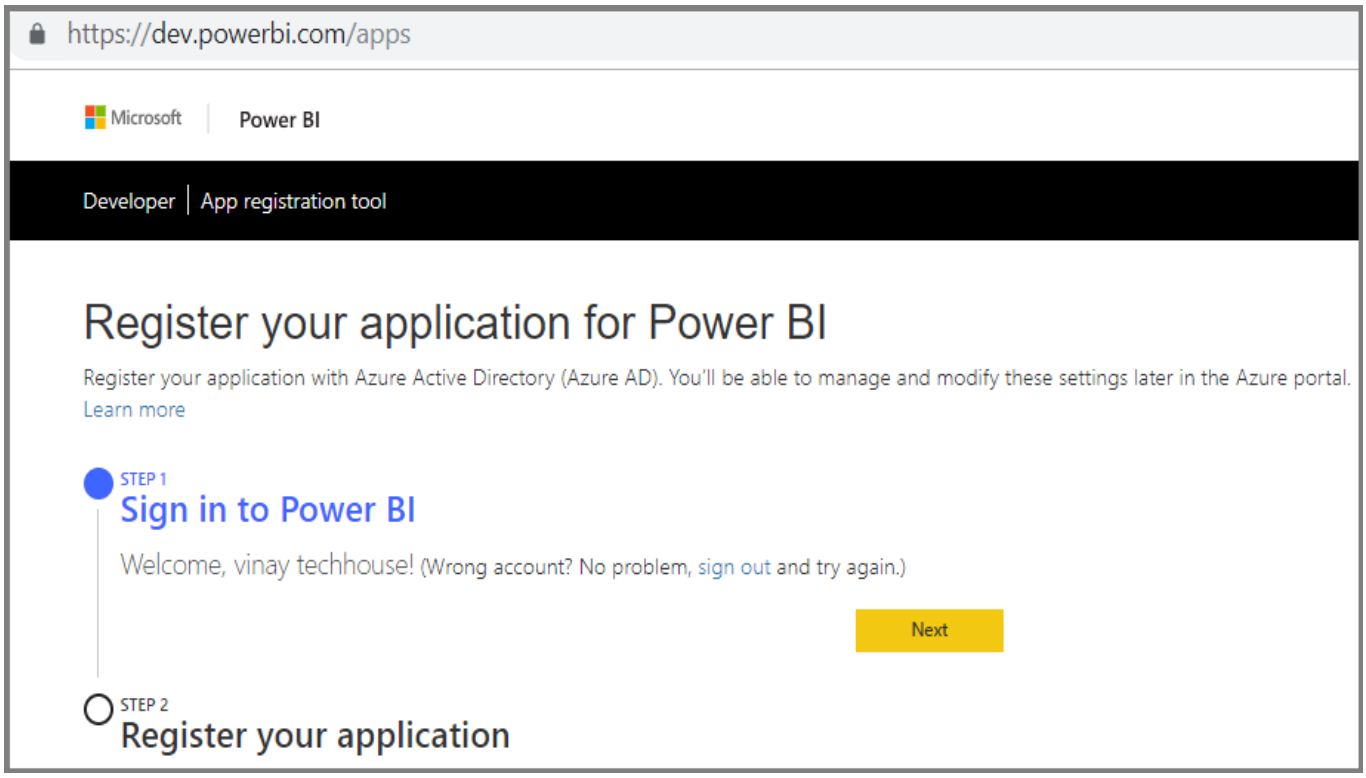
Business intelligence and analytics from Microsoft Power BI can be customized, extended, and embedded in applications **using our comprehensive set of APIs (Application Programming Interface) and fully documented SDK (Software Development Kit) libraries.**

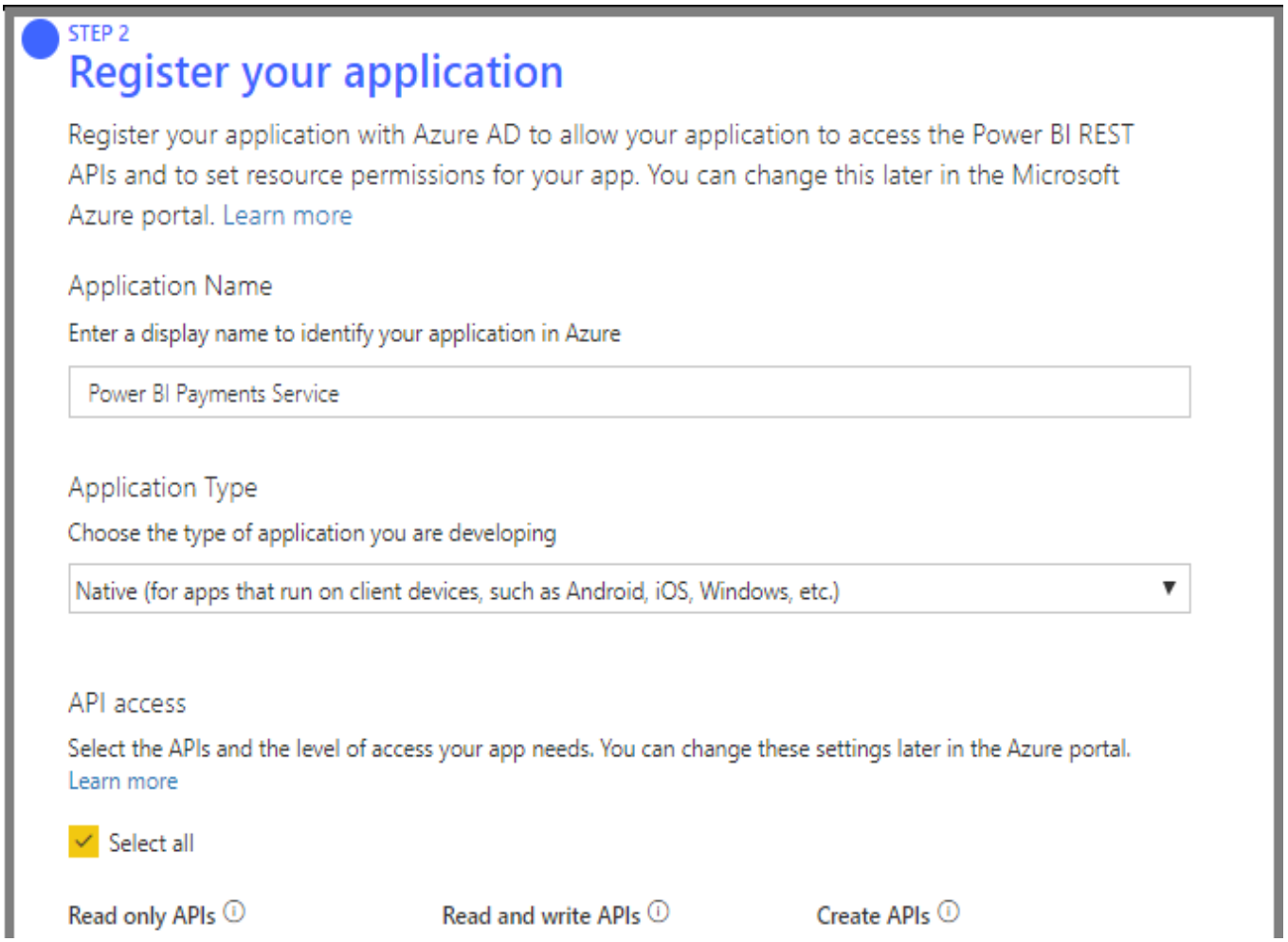
What do you mean by Power BI Embedding?

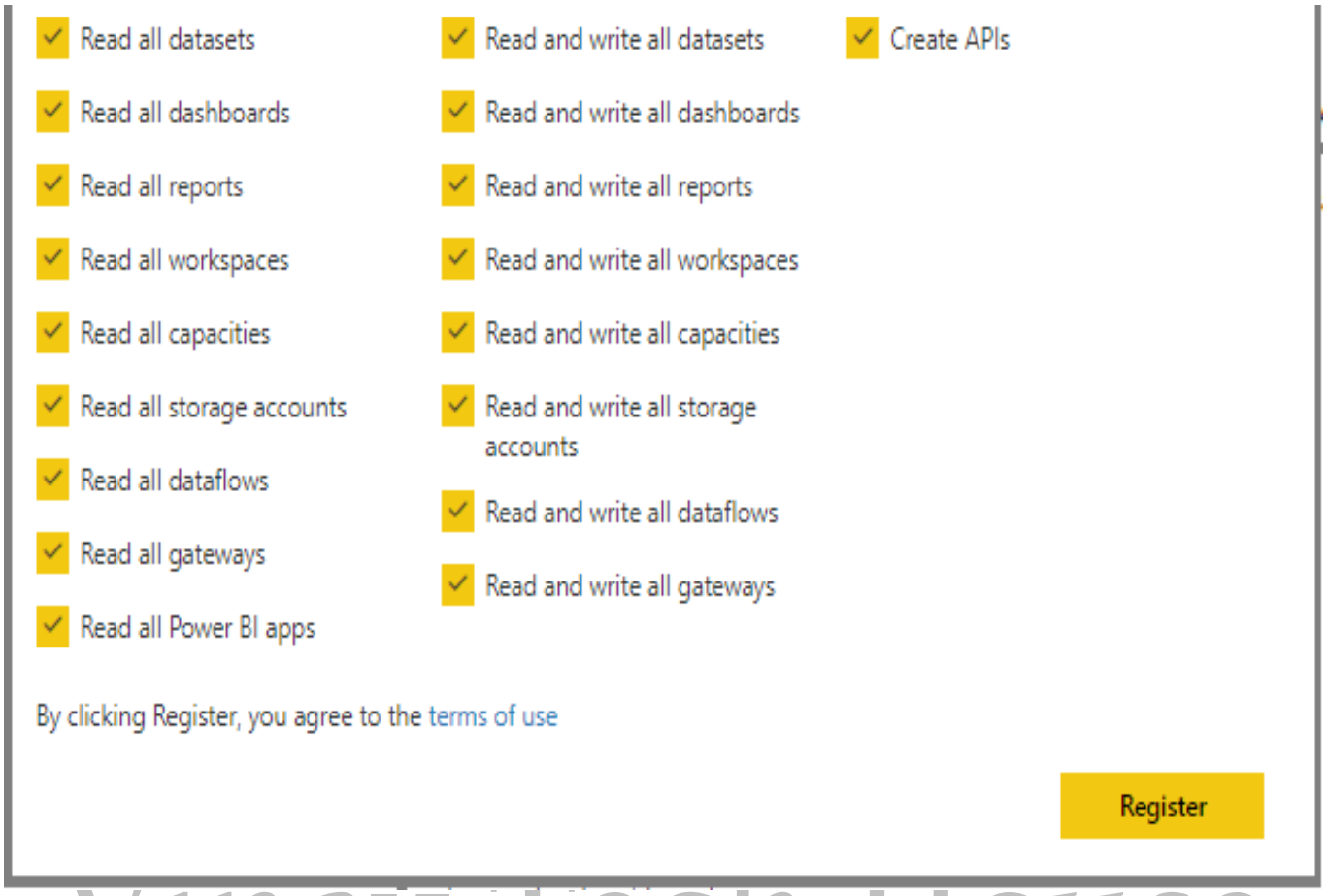
Embed stunning, interactive data visualizations in applications, websites, portals, SharePoint, Microsoft Teams, and more, to showcase your business data in context. Using Power BI REST APIs and the Power BI SDK, you can easily embed interactive reports and dashboards, so your users can enjoy consistent, high-fidelity experiences across devices.

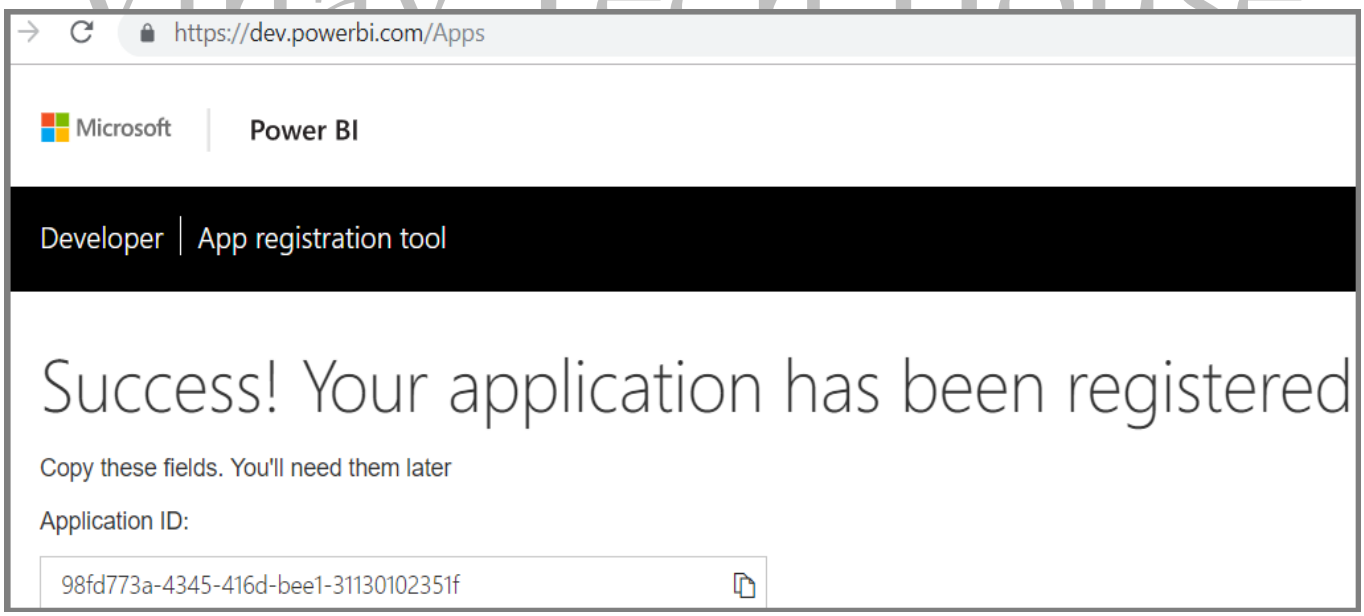
How do we register an app, get client id and provide permissions?

1.Connect to https://dev.powerbi.com/apps

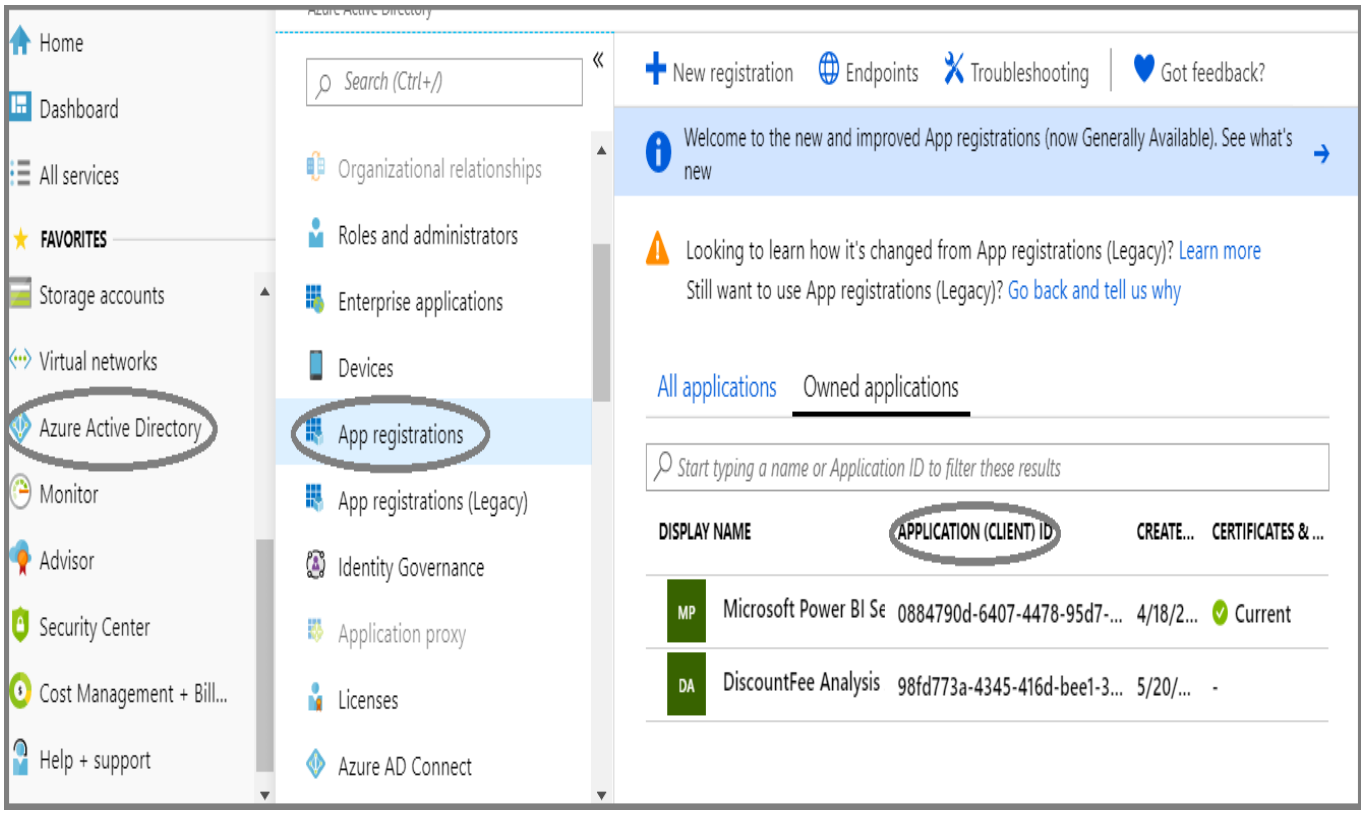

https://dev.powerbi.com/apps
Microsoft
Power BI
Developer
App registration tool
Register your application for Power BI
Register your application with Azure Active Directory (Azure AD). You'll be able to manage and modify these settings later in the Azure portal.
Learn more
STEP 1
Sign in to Power BI
Welcome, vinay techhouse! (Wrong account? No problem, sign out and try again.)
Next
STEP 2
Register your application


STEP 2
Register your application
Register your application with Azure AD to allow your application to access the Power BI REST APIs and to set resource permissions for your app. You can change this later in the Microsoft Azure portal. Learn more
Application Name
Enter a display name to identify your application in Azure
Power BI Payments Service
Application Type
Choose the type of application you are developing
Native (for apps that run on client devices, such as Android, iOS, Windows, etc.)
API access
Select the APIs and the level of access your app needs. You can change these settings later in the Azure portal.
Learn more
Select all
Read only APIs
Read and write APIs
Create APIs

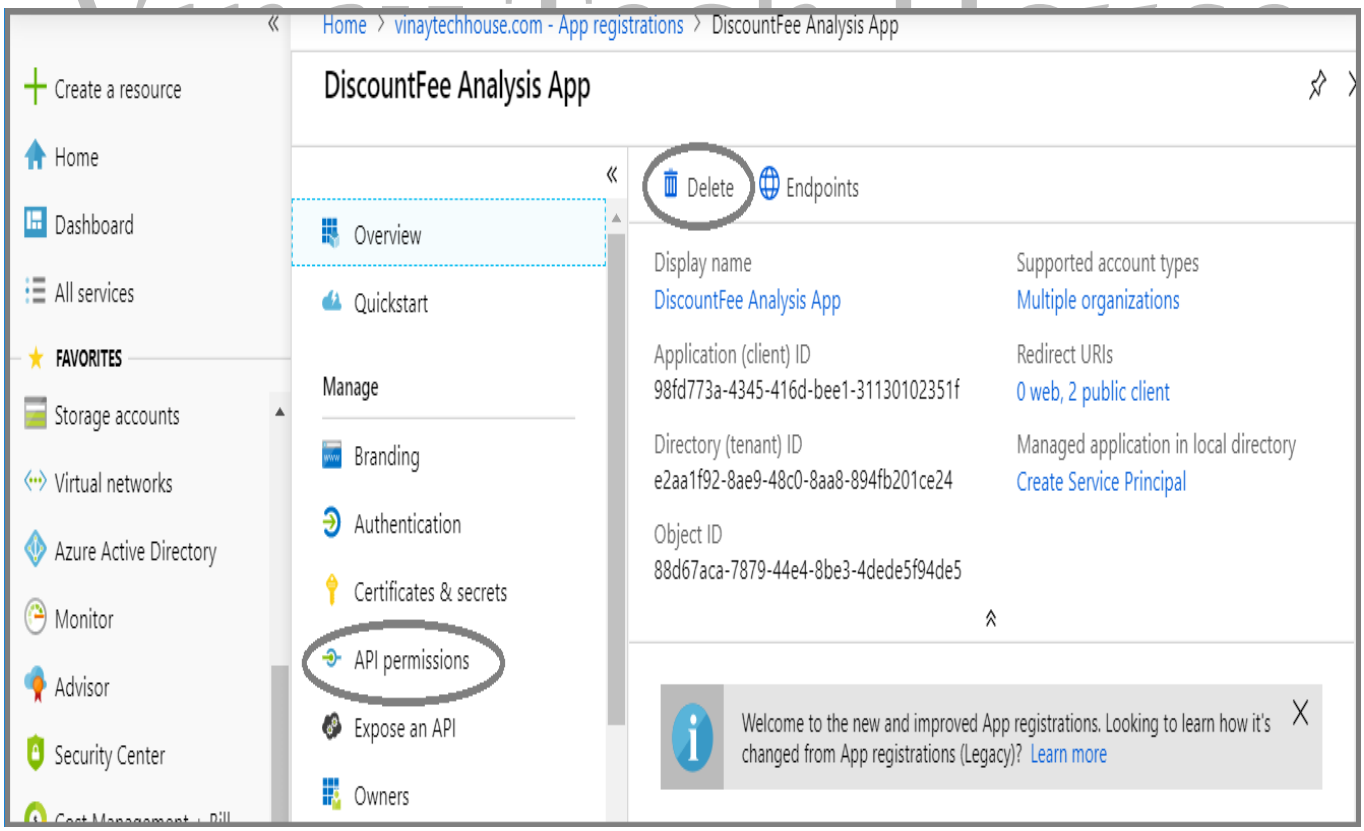




2. Connect to portal.azure.com with the account



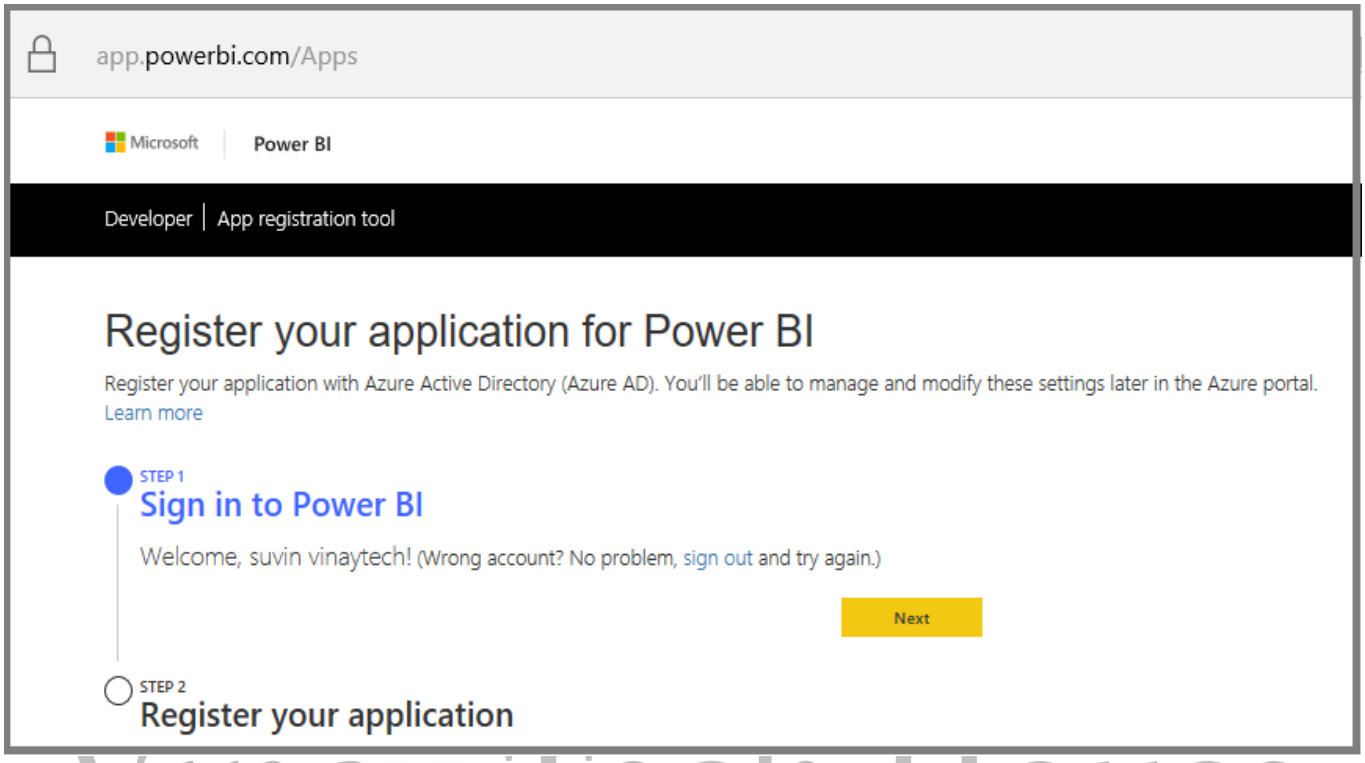
Provide permissions for the applications



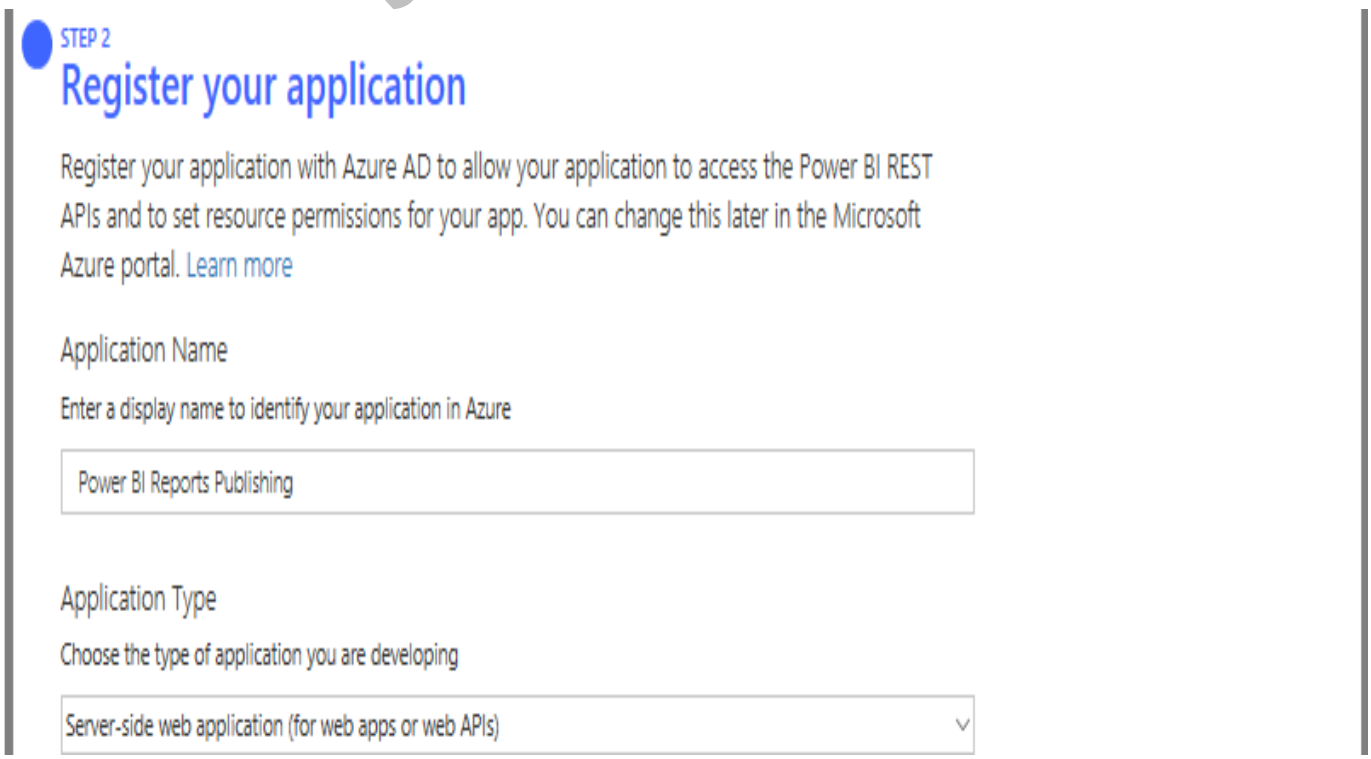
1. You now use the clientID and app permissions for embedding.

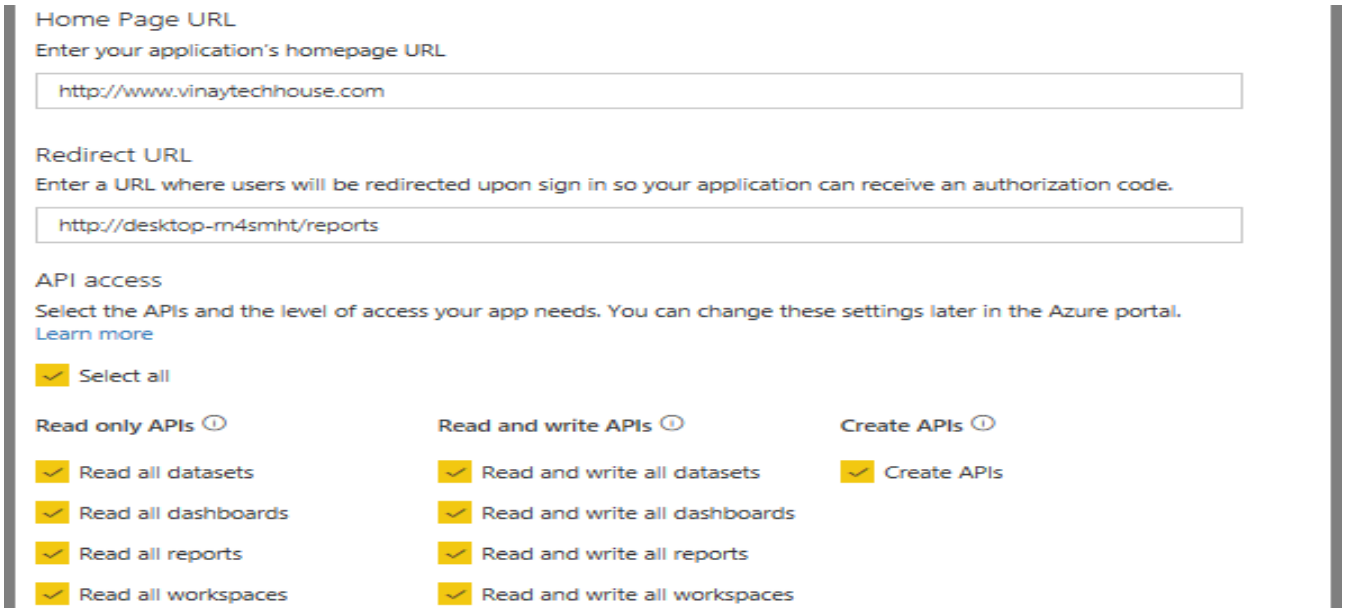
SCENARIO: USING POWER BI REST API PUBLISHING POWER BI REPORTS

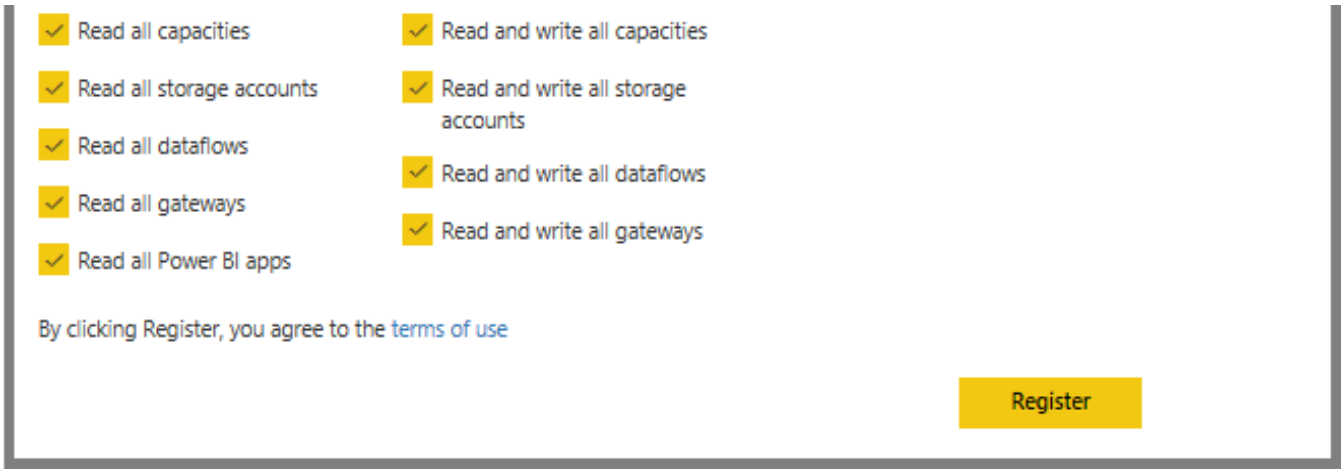
1. Connect to app.powerbi.com/apps



Click Next



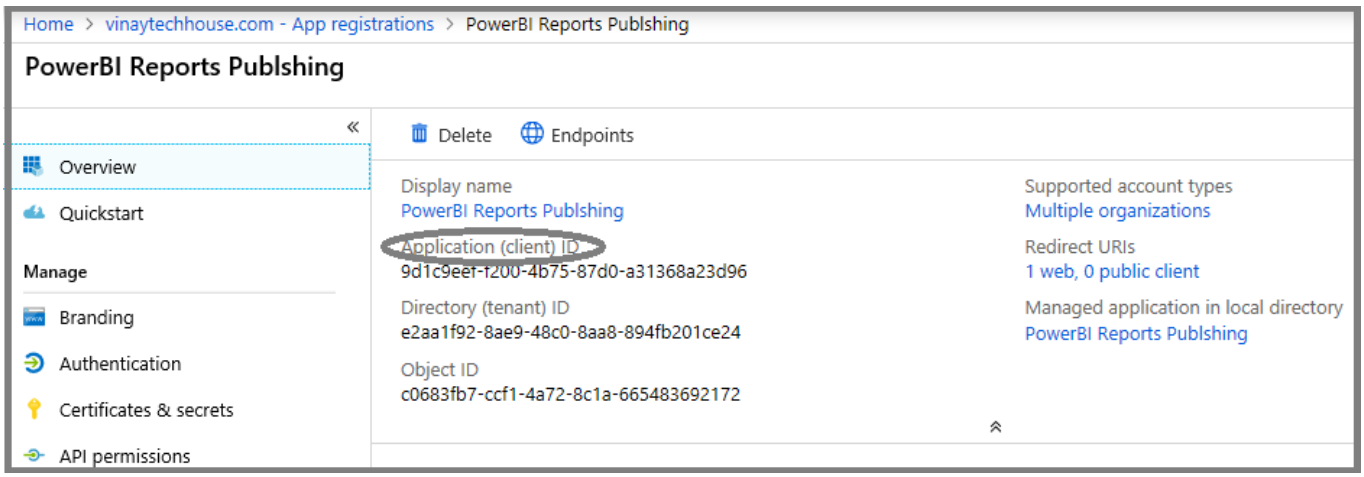




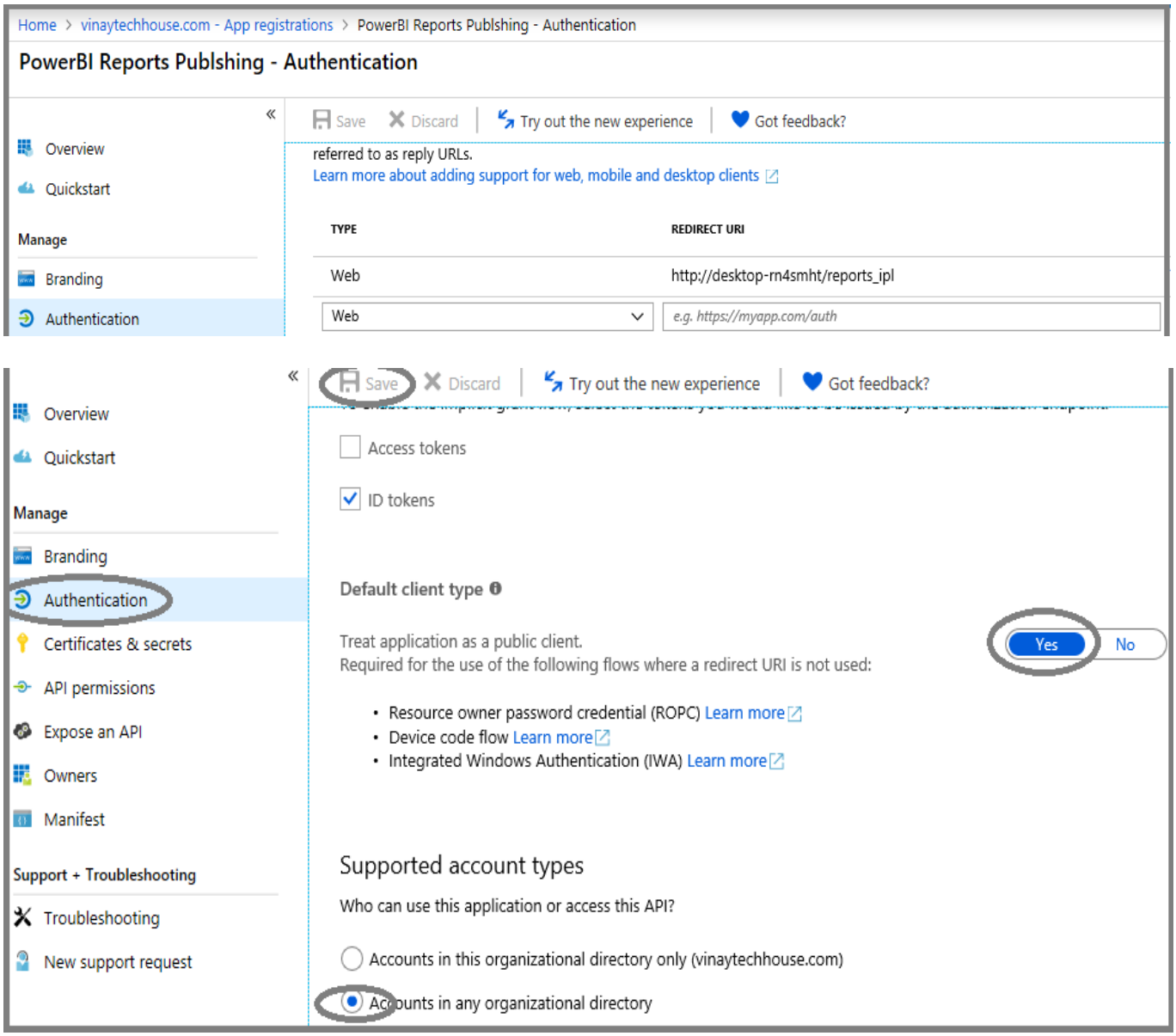
Click Register

Go to Azure Portal→Connect with your account

Click App Registrations→Click on Power BI Reports Publishing



Click App Registrations→ Authentication



Open the folder→ go to project file(.cs)→ Open with Visual Studio 2017→

Change the ClientID, Hardcode username and password, workspace, import type.

Comment the manual enter credential statements.

Comment Gatewayfunction calling in Main method.

Run it.

It will open a command prompt and then takes secure code and then publish the report under import type.

Go to the Power Bi Cloud and observe the report with report type.