



Planning a Power BI Enterprise Deployment

Whitepaper

Summary: This is a technical whitepaper which outlines considerations and best practices for an enterprise level Power BI deployment.

Writers: Melissa Coates ([Coates Data Strategies](#)), Chris Webb (Microsoft)

Technical Reviewer: Meagan Longoria ([DCAC](#))

Microsoft Reviewers: Lukasz Pawlowski, Kim Manis, David Iseminger, Alon Baram, Nimrod Shalit, Miguel Martinez, Tessa Hurr, Anton Fritz, Chris Finlan, Ranin Salameh, Kate Follis, Will Thompson, Miguel Llopsis, Ben Sack, Christian Wade, Colin Popell, Arthi Ramasubramanian Iyer, Aaron Meyers, Adi Regev, Yaron Canari, Aviv Ezrachi, Maya Shenhav.

Version: 3.1

Published: May 2020

Contents

Section 1. Introduction.....	12
Purpose of this Whitepaper.....	12
Scope of this Whitepaper.....	12
Roadmap and Future Features	13
High Level Overview of Power BI.....	14
Definitions and Terminology	15
Section 2. Power BI Usage Scenarios.....	16
Personal BI.....	17
Small Team Collaboration	18
Large Team Collaboration and Distribution.....	21
Enterprise Content Distribution	22
Centralized Datasets for Corporate Reporting.....	23
Prototyping Activities and Sharing	25
Embedding for your Organization.....	26
Embedding for Your Customers (External Embedding).....	27
Workflow Automation + Updating Data from Within Power BI.....	29
Self-Service Data Preparation and Machine Learning.....	31
Near Real-Time Reporting	32
On-Premises Reporting Portal.....	33
Section 3. Power BI Architectural Choices	34
The Power BI Service	34
What is the Power BI Service?.....	34
Why Use the Power BI Service?	35
Power BI Service Modern Release Cycle	37
Power BI Premium.....	37
What is Power BI Premium?.....	37
Power BI Premium Capacities.....	38
Premium Tiers and SKUs.....	39
Deciding Whether to Use Power BI Premium	41
Power BI Dataflows	43
Why Use Power BI Dataflows?	43

What are Power BI Dataflows?.....	44
Two Types of Data Lake Storage with Power BI Dataflows.....	46
Power BI Dataflows and the Common Data Model.....	48
Additional Power BI Dataflows Functionality with Power BI Premium	49
Power BI Report Server.....	50
What is Power BI Report Server?.....	50
Why Use Power BI Report Server?.....	50
Power BI Report Server Release Cycle	51
Comparing Power BI Report Server Functionality with the Power BI Service	51
Embedding Power BI Content in Applications and Websites.....	51
Embedding for Productivity.....	52
Embedding For Your Organization	53
Embedding For Your Customers (Power BI Embedded)	54
Publish to Web.....	55
Section 4. Power BI Licensing and User Management.....	56
Power BI Licensing Summary	57
Power BI User Licenses	59
Power BI Free.....	59
Power BI Pro.....	59
Organizational Purchasing of Power BI User Licenses	60
Self-Service Purchasing of Power BI User Licenses.....	61
Ways to Assign Power BI User Licenses	63
Group-Based Licensing Assignment.....	64
Workflow-Based Licensing Assignment.....	64
Customizing the Power BI Signup Process	65
Managing Power BI User Trials and License Assignments.....	66
Automating the Conversion of Trial Accounts to Pro User Licenses.....	66
Handling Auto-Created Office 365 Tenants from Self-Service Signups	67
Blocking Users from Signing Up for Power BI	67
Other Power BI License Types.....	67
Power BI Premium Capacity Licensing.....	67
Power BI Embedded Capacity Licensing	68
Power BI Report Server Licensing	69

Section 5. Power BI Source Data Considerations.....	71
Identifying Various Types of Data Sources.....	71
Data Source Considerations.....	72
Cleansing, Filtering, Transforming and Integrating Data	75
Centralizing Data Preparation Logic with Dataflows.....	77
Section 6. Power BI Dataset Storage Options	78
Dataset Storage Modes.....	78
Imported Data.....	78
Live Connection to a Power BI Dataset or to Analysis Services.....	79
DirectQuery Mode.....	80
Composite Models	82
Aggregations.....	82
Near Real-Time Data	83
Choosing a Dataset Storage Mode.....	85
Is Analysis Services Already in Use?.....	85
How Much Data is Needed to Produce the Reports?	87
How Frequently Does Data Need to be Refreshed?.....	88
How Complex is the Dataset? Are any Complex Calculations Necessary?	90
How Many Concurrent Users Need to be Supported?.....	91
Are there Legal or Regulatory Reasons for Storing Data in a Specific Location?.....	92
What Technical Skills are Present In-House?.....	92
What Budgetary Constraints Need to be Considered?	92
Summary of Dataset Storage Recommendations	93
Section 7. Power BI Data Refresh and Data Gateway	95
Introduction to the On-Premises Data Gateway	96
Gateway Terminology	97
Standard Mode.....	98
Personal Mode.....	99
Summary of Features Per Mode.....	101
When a Data Gateway is Required.....	101
Overview of Process for Setting Up the On-Premises Data Gateway.....	102
Managing Who Can Install a Gateway.....	103
Considerations for Using the Data Gateway.....	103

Data Gateway Installation File.....	103
Using a Gateway Cluster	104
Gateway Naming Conventions	104
Identities and Credentials Used by the Data Gateway	106
Associating Datasets with a Gateway.....	108
User Interactions with the Data Gateway	108
Personal Gateway Warnings	109
Data Gateway Credentials.....	109
Sharing of Data Gateway with Other Applications.....	110
Using One or More Data Gateways	111
Performance Impact on the Server Running the On-Premises Data Gateway	111
Where to Install the On-Premises Data Gateway	112
Firewall Ports.....	113
Communication Protocol.....	113
Other Items To Be Installed on the Gateway Server.....	113
Monitoring and Managing Data Refreshes	114
Data Refresh Success.....	114
Paused Data Refreshes	114
Deactivated Data Refreshes.....	115
Data Refresh Limitations Based on License Type	115
Monitoring and Managing Gateways.....	115
Managing Data Gateways.....	115
Keeping the Data Gateway Updated	115
Process for Gateway Updates.....	116
Data Gateway Permissions	117
Monitoring Gateway Health.....	117
Data Gateway Configuration Files	120
Data Gateway Server Uptime and Health.....	120
Data Gateway Disaster Recovery	120
Types of Data, File, and Cache Refresh in Power BI.....	121
Dataset Refresh.....	121
Dataflow Refresh.....	121
Package Refresh (File Level).....	122

Cached Tile Refresh.....	122
Cached Visual Container Refresh.....	122
Section 8. Power BI Dataset and Report Development Considerations.....	123
Choosing the Right Tool.....	123
Power BI Desktop.....	124
Power BI Service	124
Excel	125
Power BI Report Builder	127
Best Practices for Dataset and Report Design in Power BI Desktop	127
Managing Dataset Authorship.....	127
Version Control for Power BI Content.....	128
Separating Datasets from Reports	130
Using Row-Level Security for Securing Data.....	132
Using Templates to Speed Up and Standardize Report Development.....	132
Using Power BI Data Source Files	133
Minimizing the Amount of Data Loaded	133
Reducing Dataset Size.....	134
Naming of Tables, Columns, and Measures	135
Minimizing the Amount of Data Displayed on a Page	136
Focusing on Quality of Content	137
Focusing on Information Design Principles	137
Verifying Accessibility on Reports	138
Using Certified Custom Visuals.....	138
Using Custom Data Connectors	139
Creating Dashboards to Unify Key Metrics from Various Reports.....	140
Developing for the Power BI Service and Power BI Report Server.....	140
Section 9. Power BI Collaboration, Sharing and Distribution.....	141
The Role of Power BI Desktop	141
Collaboration, Sharing, and Distribution in the Power BI Service.....	142
Workspaces in the Power BI Service.....	142
Content Distribution in the Power BI Service	149
Allowing Users to Copy and Customize Existing Reports	157
Shared and Certified Datasets and Dataflows	158

Advantages and Disadvantages of Shared Datasets	159
Workflow for Using Shared Datasets	160
Lineage View.....	161
Dataset Discovery Experience for Certified and Promoted Datasets	162
Permissions Required to Use a Shared Dataset	163
Certification Permissions.....	163
Shared and Certified Dataflows	164
Deployments to Development, Test, and Production	164
Option 1: Use App as Production	164
Option 2: Use of Separate Workspaces.....	166
Option 3: Deployment Pipelines + Separate Workspaces	167
Suggestions For Getting the Most From Deployment Pipelines	169
Collaboration, Sharing and Distribution with Power BI Report Server.....	172
Consuming Content Published to the Power BI Service.....	172
Viewing Content in a Web Browser.....	172
Power BI Mobile Apps.....	172
Power BI Universal App for Windows 10 Devices.....	172
Analyze in Excel	173
Embedded Reports in Microsoft Teams.....	174
Embedded Reports in SharePoint Online and SharePoint On-Premises.....	175
No-Code Embedding with iFrames.....	175
Distributing Reports Publicly on the Web.....	176
Sharing and Distribution with External Users Using Azure B2B	177
Subscriptions and Alerting.....	178
Subscriptions	178
Alerts.....	180
Export-To-File API.....	180
Integration with Power Automate	181
Section 10. Power BI Administration.....	181
Initial Power BI Tenant Setup.....	181
Creation of a Power BI Tenant	181
Geographic Data Region.....	182
Administrator Roles Related to Power BI	183

Tenant Setting Recommendations	184
Help and Support Settings.....	186
Workspace Settings	187
Information Protection Settings.....	188
Export and Sharing Settings	189
Content Pack and App Settings.....	192
Integration Settings	193
Power BI Visuals Settings	193
R Visuals Settings.....	194
Audit and Usage Settings	194
Dashboard Settings.....	195
Developer Settings	195
Dataflows Settings.....	196
Template App Settings	196
Q&A Settings.....	197
Managing Custom Help Menus.....	197
Internal Support Pages	197
Licensing Requests	198
Governing the Use of Power BI Visuals and Organizational Visuals	198
Managing Use of Power BI Visuals in the Power BI Service.....	200
Managing Use of AppSource Visuals and Visuals from a File in Power BI Desktop.....	201
Managing Use of Organizational Visuals.....	201
Managing Embed Codes for Publicly Available Reports	203
Managing Workspaces.....	204
Managing Custom Branding	205
Managing Protection Metrics	205
Managing Featured Content.....	206
Managing Dataflow Settings.....	207
Monitoring Usage of Power BI	207
Power BI Service Usage	208
Power BI Report Server Usage	215
Automating Power BI Management.....	216
Power BI Management PowerShell Cmdlets	216

Power BI Data Gateway PowerShell Cmdlets	218
Power BI REST APIs.....	218
Using Power Automate.....	220
PowerShell Cmdlets for Power BI Report Server.....	220
Microsoft 365, Azure Active Directory, and Microsoft Graph REST APIs	220
Managing Power BI Premium	221
Creating Capacities in the Power BI Service	221
Assigning Workspaces to Power BI Premium Capacity	222
Assigning Capacity Admins and Capacity Assignment Permissions	223
Configuring Workloads.....	223
Power BI Premium Capacity Metrics App	224
Power BI Premium Capacity Overload Alerts	225
Deploying Installers and Applications	225
Power BI Desktop.....	226
Power BI Report Builder	228
Power BI Desktop Optimized for Report Server.....	228
Power BI Mobile Apps.....	228
Power BI App for Windows 10 Devices	229
Analyze In Excel Provider	229
Data Gateway	230
Other Potential Installers.....	230
Managing Power BI Mobile Apps with Intune.....	230
Monitoring Service Health	231
Power BI Support Site	231
Azure Status	231
Microsoft 365 Admin Center	232
Firewall Settings/Whitelisted URLs	233
High Availability and Disaster Recovery	233
Section 11. Power BI Security and Data Protection.....	233
Integration with Azure Active Directory.....	234
Managing Access to the Power BI Service	234
Managing Access to Content in the Power BI Service.....	235
Workspace Security.....	235

App Security.....	236
Sharing of Dashboards and Reports.....	237
Dataset Permissions.....	237
Subscription Permissions	238
Location of Original Data Files.....	238
Data Protection and Privacy	238
Unified Sensitivity Labels	238
Data Protection Risk Detection.....	243
Data Privacy Levels in M Queries.....	243
Dashboard Data Classification.....	244
Compliance Certifications.....	244
Securing Data at Rest.....	244
Dataset Row-Level Security.....	245
Securing Data at the Data Source Level with SSO.....	246
Data Storage for Imported Datasets (< 10 GB)	246
Data Storage for Large Imported Datasets (> 10 GB).....	246
Storage for Metadata and Visual Cache.....	246
Storage of Credentials to Connect to Source Data.....	246
Storing Data in a Specific Geographic Region.....	246
Securing Data in Transit.....	247
Securing the Connection Between Data in Azure and Power BI.....	247
Securing the On-Premises Data Gateway.....	247
Data Transmitted Outside of the Power BI Service.....	247
ExpressRoute	248
Protecting Data on Mobile Apps.....	248
Azure Active Directory	248
Intune	248
Custom Mobile Device Management (MDM)	248
Securing Power BI Report Server.....	248
Managing Encryption Keys	249
Azure Premium Encryption Key.....	249
On-Premises Data Gateway Recovery Key	249
Encryption Keys for Gateway Credentials.....	249

Encryption Key for Power BI Report Server.....	250
Power BI Report Server Product Key	250
Power BI in the National Clouds.....	250
Section 12. Power BI Deprecated Items.....	251
Power BI Publisher for Excel	251
Classic Workspaces in the Power BI Service.....	251
Old Filter Experience	251
Windows 7 and .NET 4.5 Support for Power BI Desktop.....	251
Ribbon in Power BI Desktop.....	251
Windows 10 Mobile.....	252
TLS 1.0 and 1.1 Support.....	252
Power BI APIs .NET SDK v2	252
Get Data Experience in the Power BI Service	252
Organizational Content Packs	252
MSI Installer for Power BI Desktop	252
Custom Visuals API Version 1.1 and Older	253
Certain Behaviors of Data Analysis eXpressions (DAX)	253
Certain Color Classes within Report Theme JSON Files.....	253
Azure Workspace Collections	253
Power BI CLI.....	253
Section 13. Support, Learning, and Third-Party Tools.....	253
Power BI Release Plan (Roadmap)	254
Support and Help	254
Learning More About Power BI.....	256
Community and Third-Party Tools.....	257

Section 1. Introduction

Purpose of this Whitepaper

Deploying Power BI in a large enterprise is a complex task, and one that requires a lot of thought and planning. The purpose of this whitepaper is to help you make your Power BI deployment a success: it covers key considerations, the decisions which will be necessary throughout the process, and potential issues you may encounter. Best practices and suggestions are offered when possible.

The target audience for this whitepaper are technology professionals. Some knowledge of Power BI and general business intelligence concepts will be assumed.

Scope of this Whitepaper

Unless specifically noted, all features mentioned in this whitepaper are generally available (GA) as of April 2020.

The following topics are out of scope or limited in scope for this whitepaper:

- Features which are in **public preview** (not reached general availability yet) are limited to awareness only, and not discussed in depth. An important example of this is the XMLA endpoint, which will be a very important aspect of enterprise-level Power BI deployments for many customers.
- Discussion of the **V1 workspace experience** (which has been replaced by the new workspace experience and is the focus of this whitepaper).
- Detailed explanation of **Power BI fundamentals**, such how to create reports and dashboards or write DAX calculations.
- **ISV deployment scenarios**, which are handled differently from enterprise deployment scenarios. Limited information about the Power BI Embedded service in Azure is included in this whitepaper for clarity.
- **Performance tuning** of reports, or of DAX or M expressions.
- **SQL Server Reporting Services**, beyond direct comparisons to Power BI Report Server.
- Deploying and managing **Power BI Report Server** in-depth.
- Other **Power Platform tools and features** beyond Power BI-specific features.
- **Third party solutions** which integrate with Power BI.
- The custom development process involved with **creation of custom visuals**.
- Distributing content to **external guest users with B2B**, as it is covered in a [separate whitepaper](#), and **multi-tenant scenarios**.
- Deploying and **managing Power BI Premium capacity** in-depth, as it is covered in a [separate whitepaper](#).
- **GDPR compliance** in-depth, as it is covered in a [separate GDPR whitepaper](#).
- **Power BI security** in-depth, as it is covered in a [separate whitepaper](#).
- Features and functionality for the **national clouds** (US Govt, China, Germany), or **government**. The guidance in this whitepaper is focused on the commercial cloud.



Make it a practice to stay current

There is an abundance of information about Power BI online, due to a vibrant user and partner community. However, the frequent pace of change represents a challenge for accuracy. Make it a practice to stay current, and verify any information found online since it can quickly become out of date. Indeed, some information in this whitepaper may be out of date by the time you read it: this [What's New](#) page contains a reference list of major changes to features and functionality since this whitepaper was published.

Roadmap and Future Features

This whitepaper refers to future features, to communicate meaningful changes to be aware of that are not be available at the time of this writing.

This whitepaper purposely makes no estimation of release dates in conjunction with future features. Some of these features may be released very quickly after this whitepaper becomes available, and some may take longer to be released.



Follow the Power BI release plan

The latest Power BI roadmap information can be found in the [release plan](#). The roadmap is released twice annually in accordance with the two release waves per year:

- April through September, and
- October through March

At the time of this writing, Power BI is continuing toward its vision of enabling customers to standardize on a single platform for self-service analytics and enterprise BI, which involves the following investment areas:

- ✓ AI-infused data exploration experiences for self-service analytics
- ✓ Modern enterprise reporting, enterprise semantic models, and administration
- ✓ BI woven into the fabric of the organization, through deep integrations with Azure, Power Platform and Office

The Power BI release plan is published by the Business Applications Group, which includes the entire Power Platform as well as Dynamics 365. The release plan is an excellent resource to review regularly so your organization can be prepared for new changes.

High Level Overview of Power BI

Power BI is collection of tools and services which facilitates the acquisition of data, data modeling, visualizations and reports, as well as the distribution of analytical solutions.

Power BI is used by many thousands of companies around the world.

Microsoft has been consistently ranked a leader on [Gartner's Magic Quadrant for Analytics and Business Intelligence Platforms](#).

As indicated in the [2019 annual report](#), Power BI hosts more than 25 million models and processes 12 million queries per hour.

Figure 1. Magic Quadrant for Analytics and Business Intelligence Platforms



Figure 1 Gartner magic quadrant as of January 2020

Definitions and Terminology

The most prominent components of the Power BI suite of tools are defined as follows:

Power BI	The collection of software, services, applications, and connectors which facilitate the creation of reports and dashboards and the visual analysis of information.
Power BI Desktop	<p>Power BI Desktop is the authoring tool of choice for connecting, transforming, and modeling data, as well as for creating interactive reports.</p> <p>Power BI Desktop saves files in the .pbix format. Power BI Desktop can produce template files in the .pbix format.</p>
Power BI Service	<p>The Power BI service is a cloud-based service which supports collaboration, sharing, enterprise distribution, creation of dashboards, and additional functionality such as alerts. It can serve as a central location to consolidate distribution of datasets for content creators and reports, dashboards, and apps to consumers.</p> <p>Additional ways it may be referred to include:</p> <ul style="list-style-type: none">• PowerBI.com• Power BI cloud service• Power BI web portal• Power BI web service• Power BI site
Power BI Workspace	<p>A Power BI workspace is a collaboration area within the Power BI service. A workspace is a distinct area, separate from My Workspace, dedicated to a specific team, subject area, or project.</p> <p>Reports are published to a workspace so that colleagues can view the content, collaborate, and contribute as appropriate.</p>
Power BI App	A Power BI app is a set of packaged content in the Power BI service for the purpose of distributing related reports and dashboards to a larger base of consuming users. One Power BI app may be published for each workspace.
Power BI Mobile Apps	Power BI mobile apps are native applications for iOS and Android for viewing content on a mobile device.
Power BI Windows Application	The Power BI Windows application is available from the Microsoft Store. It is an alternative for consuming content which has been published to the Power BI service, and is particularly useful for rotating reports on display monitors.

On-Premises Data Gateway	The On-Premises Data Gateway is an agent installed within the corporate network to allow secure access to organizational data stored on-premises or within a corporate network. There are two gateway modes: standard mode and personal mode.
Power BI Premium	Power BI Premium is an offering that provides dedicated resources for reports, dashboards, and datasets within the Power BI service. This provides organizations more predictable performance, larger storage volumes, larger dataset sizes, higher refresh rates, and incremental data refresh. Premium also enables widespread distribution of content without requiring a Pro license for each read-only user.
Power BI Report Server	<p>Power BI Report Server is an alternative to the Power BI service for deploying Power BI reports within an on-premises data center, as opposed to the cloud-based Power BI service. Power BI reports are deployed and delivered in an on-premises portal alongside Reporting Services paginated reports, Excel reports, and Mobile reports.</p> <p>The intention of Power BI Report server is not feature parity with the Power BI service – its intention is to be a straightforward report portal.</p>
Power BI Embedded	Power BI Embedded is a product intended for software vendors and developers that allows them to embed reports and dashboards into applications - without the time and expense of building and maintaining visual analytics features themselves. Power BI Embedded is the companion capacity offering to Power BI Premium and is an Azure resource that runs on an hourly meter.
Power Platform	A collection of services including Power BI, Power Apps, Power Automate, AI Builder, and Power Virtual Agents.

Section 2. Power BI Usage Scenarios

The Power BI ecosystem is diverse and can be implemented in different ways. This section presents common usage scenarios to illustrate different ways Power BI may be deployed. After becoming familiar with the usage scenarios, the remainder of the whitepaper provides technical information and considerations to aid with a successful implementation.

In each of the individual usage scenarios presented in this section, the most prevalent components of Power BI are identified based on how Power BI is intended to be used. The objective here is *not* to call out every possible option for each usage scenario – rather, each diagram illustrates the primary features to investigate for each scenario. Many of the scenarios build upon each other, in which case the key aspects are not repeated for every scenario.

It is highly likely you will “mix and match” the ideas presented in this section to formulate a Power BI strategy that fits your situation. It’s also very possible that multiple Power BI strategies will be used by different departments or functional business units within an organization.

Personal BI

With personal self-service BI scenarios, content is created for personal use, with no sharing or collaboration involved. In personal BI scenarios, the content author generally has a lot of freedom and flexibility. Many Power BI users begin using the product for personal data analysis, then evolve to sharing their solutions with colleagues – at which time additional data and governance policies may become applicable.

The following is a high-level overview of the primary Power BI components to support personal self-service BI without any sharing or collaboration:

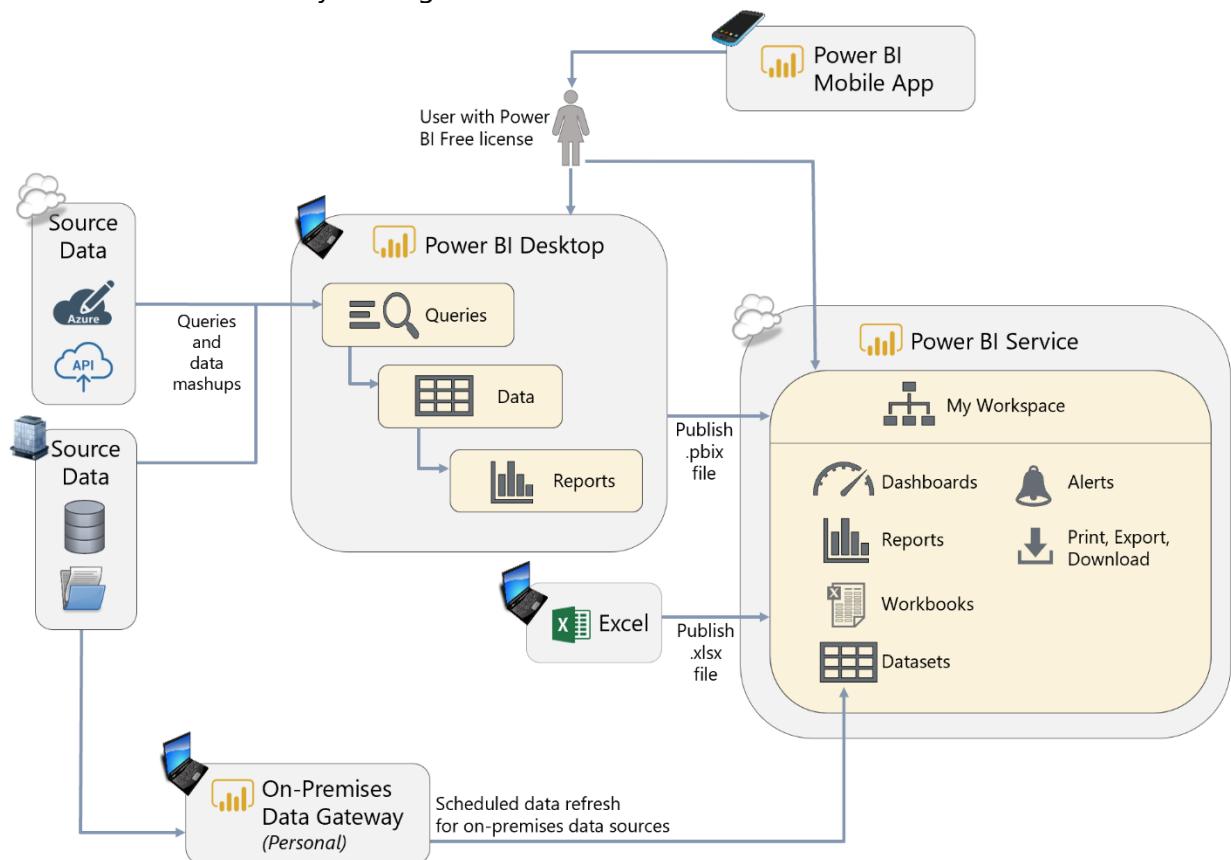


Figure 2 - Diagram of personal BI components in the Power BI service

Power BI Desktop. Power BI Desktop is the authoring tool for queries, datasets, and reports. The .pbix file is then published to My Workspace in the Power BI service for personal use.

Power BI Service with Power BI Free License. For personal use (i.e., no sharing or collaboration), the Power BI service can be used with a Power BI Free license. When using a free license, all activities within the Power BI service are constrained to within My Workspace. Features such as Power BI service Live Connections, e-mail subscriptions, and Analyze in Excel are not available with a free license. Although no sharing or collaboration is occurring within the Power BI service in a personal BI scenario, the Power BI service is still very useful for functionality such as scheduled data refresh, dashboards, data alerts, as well as the ability to use the Power BI mobile app. As will be covered later in this whitepaper, we recommend that use of My Workspace be extremely limited.

On-Premises Data Gateway in Personal Mode. For refreshing datasets imported to the Power BI service, a data gateway is required for accessing data sources which reside within the corporate network or cloud sources within a virtual network or firewall. The data gateway in personal mode supports data refresh but does not support Live Connection or DirectQuery operations. When in personal mode, the data gateway is typically installed on an individual user's machine, and therefore is best-suited to minimal personal usage scenarios only.

Power BI Mobile Application. Reporting 'on the go' via mobile apps is available for viewing reports and dashboards which have been published to the Power BI service. This is a very nice convenience for users.

Small Team Collaboration

When a Power BI user has created a solution of value, it is time to collaborate with colleagues. This is sometimes thought of as a "bottom-up" approach wherein the subject matter experts in the functional business units drive the analytical initiatives for their team.

For small team scenarios with a group of users who work closely together and interact regularly, collaboration and distribution could be handled directly within a workspace as depicted in the following diagram:

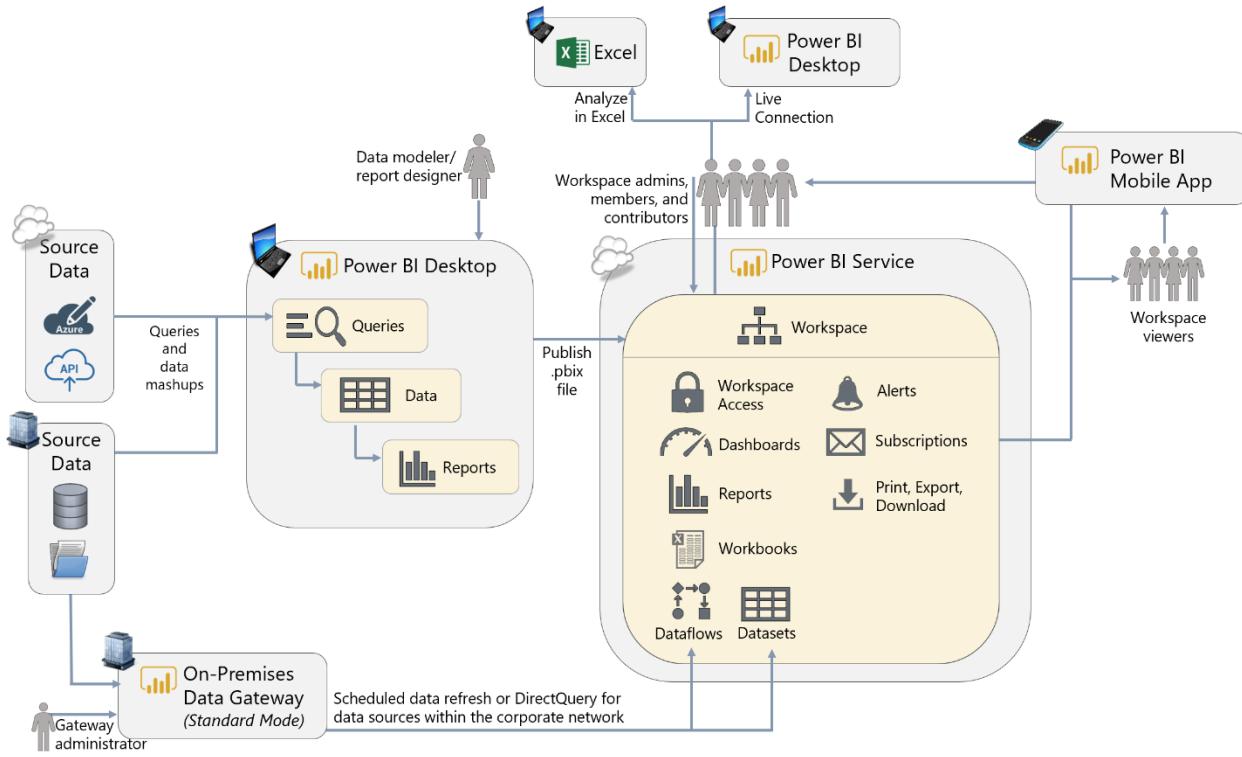


Figure 3 - Diagram of small team collaboration in the Power BI service

Power BI Desktop. Power BI Desktop is the authoring tool for queries, datasets, and reports. Storage for the original Power BI Desktop .pbix file in a common location such as OneDrive for Business is recommended because it is easily accessible by colleagues and has built-in versioning history capabilities. The common location also facilitates shared development responsibilities: for instance, if one person works on the dataset and another works on the reports.

Power BI Service with Power BI Pro License. For collaboration and distribution purposes, the Power BI service allows the most flexibility and functionality. In addition to facilitating collaboration and distribution, the Power BI service has additional features such as dashboards, scheduled refresh, data alerts, e-mail subscriptions, and other advantages. All content creators and consumers require a Pro license unless the workspace has been assigned to Premium capacity (in which case users with a Power BI Free license may view content which resides in Premium).

Workspace. A Power BI workspace serves as a container in the Power BI service for storing related datasets, reports, dashboards, workbooks, and dataflows. The workspace is considered the collaboration area (along with Power BI Desktop itself when the development work is divided among multiple people). In small team scenarios, it can be most practical to use the workspace for both authoring and collaboration, as well as the distribution and delivery of reports and dashboards. An app could optionally be used to deliver read-only content to users

(as shown in the next scenario for large teams), but is not required for small teams which don't want the additional deployment and security management of an app.

Power BI Service Live Connection and Analyze in Excel. These two features are very important additions to a team collaboration scenario. These capabilities permit a live connection, from Power BI Desktop and Excel, respectively, to a dataset which has already been published to the Power BI service. Usage of these capabilities is recommended (particularly over exporting to Excel) because:

- ✓ It reuses an existing dataset which greatly reduces the number of redundant datasets, and
- ✓ Reduces the risk of datasets and calculations being out of sync, and
- ✓ Allows the user to use all slice, dice, and pivoting capabilities within Excel while remaining pointed to the original data which resides in the Power BI service

On-Premises Data Gateway in Standard Mode. For refreshing datasets in the Power BI service, a data gateway is required for accessing data sources which reside within the corporate network or cloud sources within a virtual network or firewall. The use of a centralized data gateway in standard (enterprise) mode is strongly recommended over gateways in personal mode for team/group scenarios. In standard mode, the data gateway supports Live Connection and DirectQuery operations in addition to scheduled data refresh operations.

Power BI Mobile Application. Reporting 'on the go' via mobile apps is available for the reports and dashboards which reside in the Power BI service. Teams which use the mobile app, in addition to the Power BI service, for consuming content tend to have a better experience with Power BI.

Power BI Microsoft Teams integration (not depicted above). Using a modern collaboration tool like Microsoft Teams can engage users more in data-driven decision-making. This is ideal for small teams where faster discussions about data can lead to faster decision-making. It can also be used by different constituencies within a team to have different discussions using channels or chats to quickly loop in the required people for a targeted discussion about a specific report.

Large Team Collaboration and Distribution

When teams grow larger, it becomes impractical to use a workspace effectively for all report distribution (as was shown in the previous small team collaboration scenario). **The best way to handle larger team-based scenarios is via the usage of a workspace for collaboration, and a separate app for distributing the content to a larger group of people:**

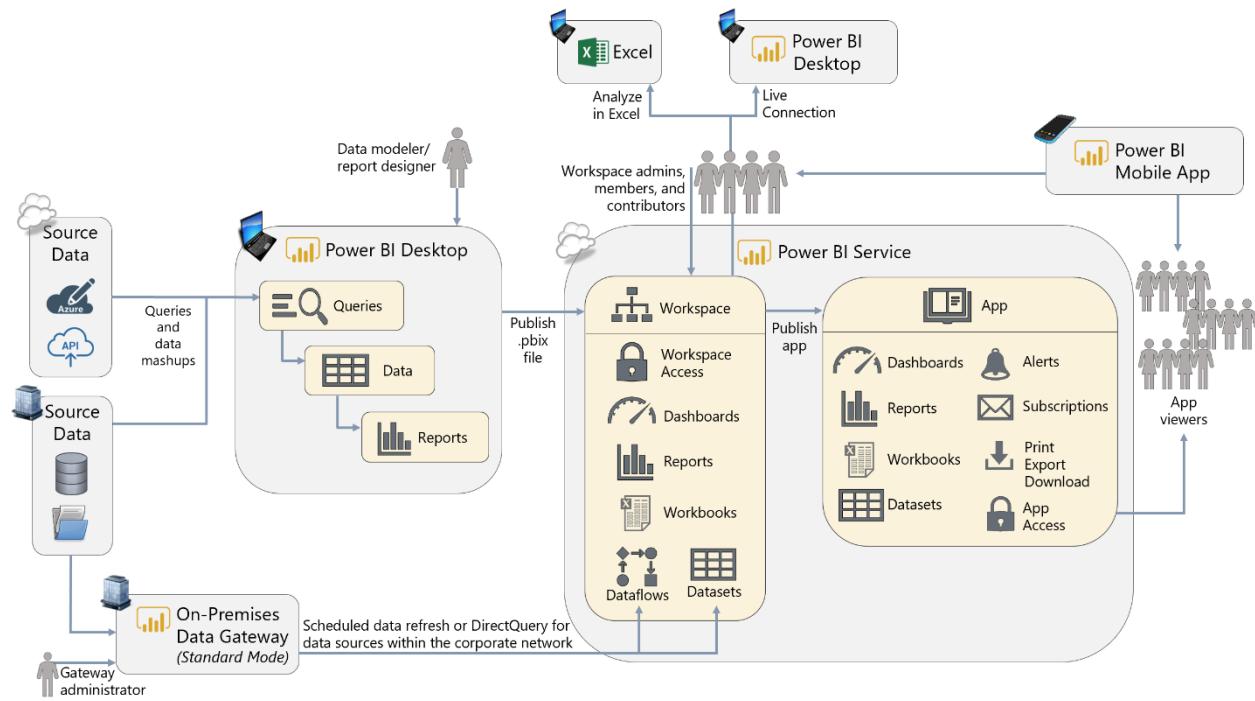


Figure 4 - Diagram of large team collaboration and distribution in the Power BI service

Power BI Desktop. Power BI Desktop remains the authoring tool for queries, datasets, and reports. As a team grows with multiple collaborators, processes should be developed to address the challenges of collaborating on the creation of a .pbix file. One of the best techniques for this is to decouple dataset from reports, which is discussed later in this whitepaper. Storing the original Power BI Desktop .pbix file in a common location such as OneDrive for Business which has built-in versioning history continues to be strongly recommended.

Power BI Service with Power BI Pro License. For collaboration and distribution purposes, the Power BI service has the most flexibility and functionality. All content creators and consumers require a Pro license unless the workspace has been assigned to Premium capacity, in which case Power BI Free users may view content from the app (or the workspace, if viewer permissions exist).

Workspace + App. A Power BI workspace serves as a container in the Power BI service for storing related datasets, reports, dashboards, workbooks, and dataflows. The workspace (plus Power BI Desktop) is considered the collaboration area, whereas the app is responsible for delivering content to users, as shown in the diagram above. For larger team situations, it is a best practice to limit access to the workspace to only the persons(s) handling authoring, development, and quality assurance activities – this is typically a small number of people. Most, if not all, read-only users then obtain access to view the content via a Power BI app rather than directly via the workspace (which was shown previously in the small team scenario). An app becomes particularly useful in larger team situations where the user base is more diverse, and when you want to control explicitly when the content is redistributed to users (i.e., via updating of the app).

Enterprise Content Distribution

Very large enterprise BI implementations often employ a “top-down” approach because it is common for the Power BI content to be largely created and maintained by a centralized IT / BI team.

Enterprise-level reporting scenarios commonly have a significantly larger number of read-only users who consume content, compared to a much smaller number of authors who create content to publish for others. A common way to cost-effectively support organizational/enterprise level reporting is via the use of Power BI Premium:

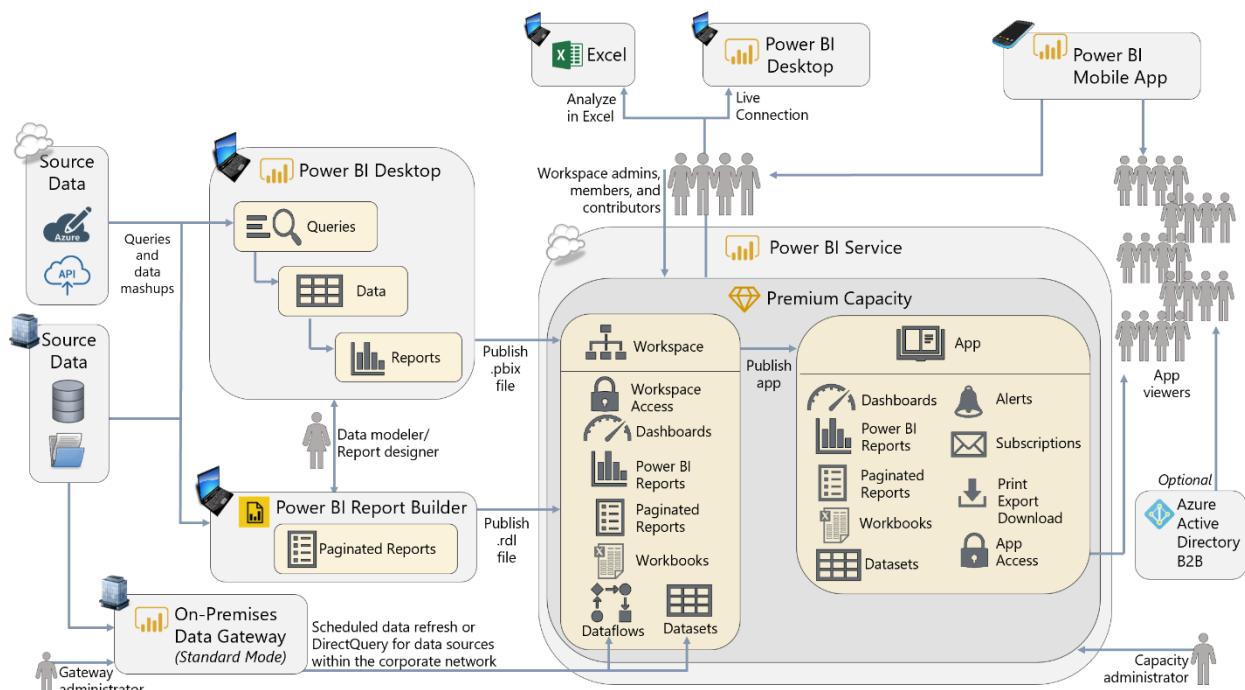


Figure 5 - Diagram of enterprise content distribution in the Power BI service

Power BI Desktop. Power BI Desktop remains the authoring tool for queries, datasets, and reports. For organizational-level reporting scenarios, centralization of datasets becomes even more important (and will be highlighted in the Centralized Datasets for Corporate Reporting scenario next).

Premium Capacity in the Power BI Service. Some larger implementations may justify the use of dedicated Premium capacity in the Power BI service, in which a specific amount of computing resources is purchased for use by a single customer. When there are many view-only consumers of reports and dashboards, the use of Free licenses afforded by Power BI Premium becomes a cost-effective alternative to purchasing Pro licenses for all Power BI users. Larger implementations may also find advantages from the larger storage limits, larger dataset sizes, higher refresh rates, and other Premium features.

Azure Active Directory B2B (Business-to-Business) or B2C (Business-to-Consumer). *(Optional.)* If any of your Power BI users are external to the organization, such as partners, vendors, and customers, they can be [invited to your AAD \(Azure Active Directory\) tenant as guest users](#). Guest users will only be able to view content via a browser, as the Power BI mobile applications are not available to guest users.

Centralized Datasets for Corporate Reporting

There are several options for implementing centralized datasets. This scenario illustrates one such option, in which a Power BI shared dataset is reused among many different Power BI reports. This approach facilitates delegating, or decentralizing, the report development activities to functional business users via usage of centralized source data.

This scenario has many characteristics in common with the Enterprise Content Distribution scenario discussed previously. The intent of the previous Enterprise scenario was to focus on the distribution of a Power BI app to many users via a Premium workspace, whereas this scenario focuses on how Power BI can reuse existing a centralized data model. The content can then be distributed broadly as described in the previous Enterprise scenario.

There are many advantages to separating datasets and reports. One advantage of separating .pbix files is the improved ability for dataset authors to work independently from report creators. Dataset authors may very well be from centralized IT / BI or may be a subject matter expert in a business unit. There are typically far more report creators, who may publish content to separate workspaces. The .pbix files containing reports are using Power BI purely as a report development tool.

There are several ways to accomplish delivery of centralized data with Power BI: Use of a shared dataset in Power BI, as shown in the scenario. Use of Analysis Services is also a valid approach. Sometimes use of DirectQuery to a data warehouse or reporting database may also be a viable option.

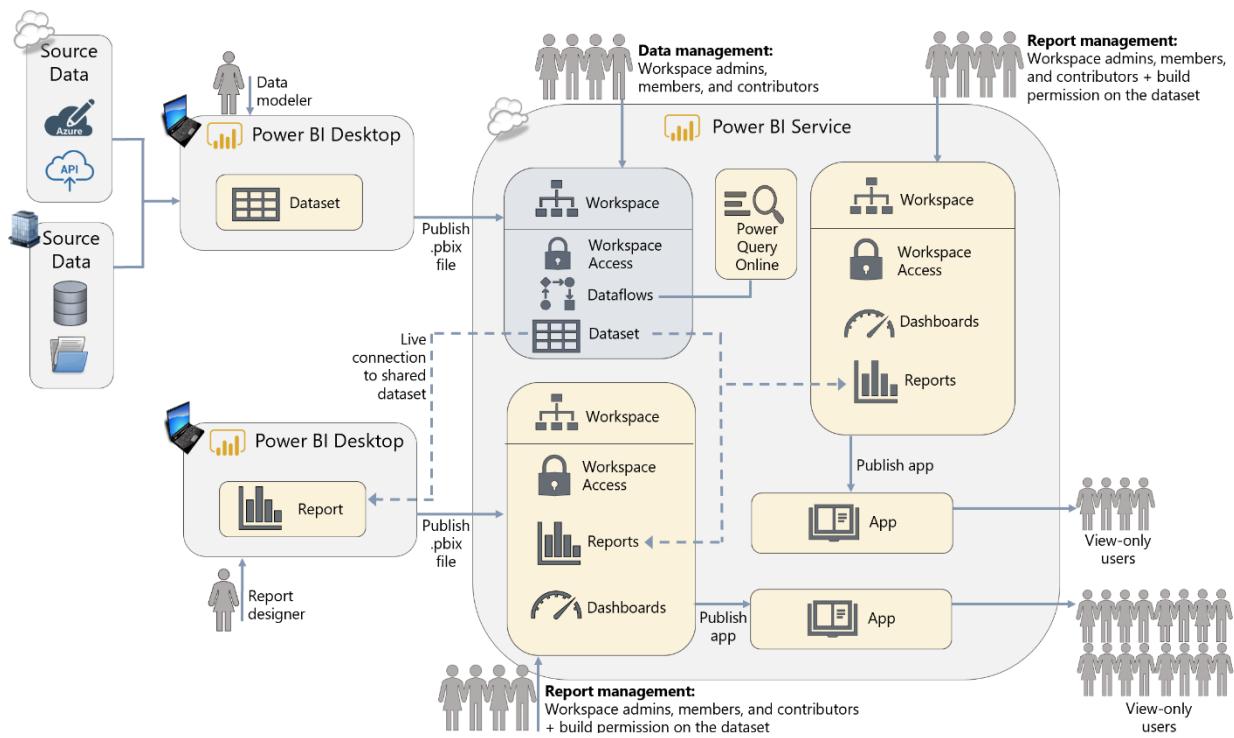


Figure 6 - Diagram of using centralized datasets for corporate reporting in the Power BI service

Power BI Desktop. In this scenario, Power BI Desktop is used by a data modeler to create and publish the dataset. In this case, the dataset and reports are completely decoupled. The authoring experience becomes significantly simpler because only reports are present in the .pbix file. **Queries, calculations, and the dataset are largely omitted (apart from report-level measures)** because they are all handled in the shared dataset which is being accessed via a live connection.

Workspace Permissions. The separate workspaces shown in this scenario facilitate the ability to separate out who has data management permissions (i.e., the ability to publish and update datasets), versus who can publish and update report content. Typically, a much larger number of people have report management permissions.

Shared Dataset. Having a centralized, user-friendly, shared dataset to serve as a semantic layer for reporting helps organizations work towards achieving “one version of the truth” and reduces duplicative copies of datasets. Using Analysis Services as a shared dataset is a valid option as well.

Build Permission. To allow access to the shared dataset, a “build” permission must be assigned to the dataset. This allows approved report authors to create new content using the dataset. This is a separate permission than workspace permissions and separate from the dataset read permission.

Prototyping Activities and Sharing

Prototyping techniques are very useful for confirming requirements for visuals and calculations with end users before significant development time is incurred.

Prototyping solutions may be temporary, short-lived solutions or they may ultimately evolve into a solution which is fully validated and released. For simplicity, this scenario illustrates the usage of sharing functionality for collaboration purposes (as opposed to workspaces and apps covered in the previous scenarios).

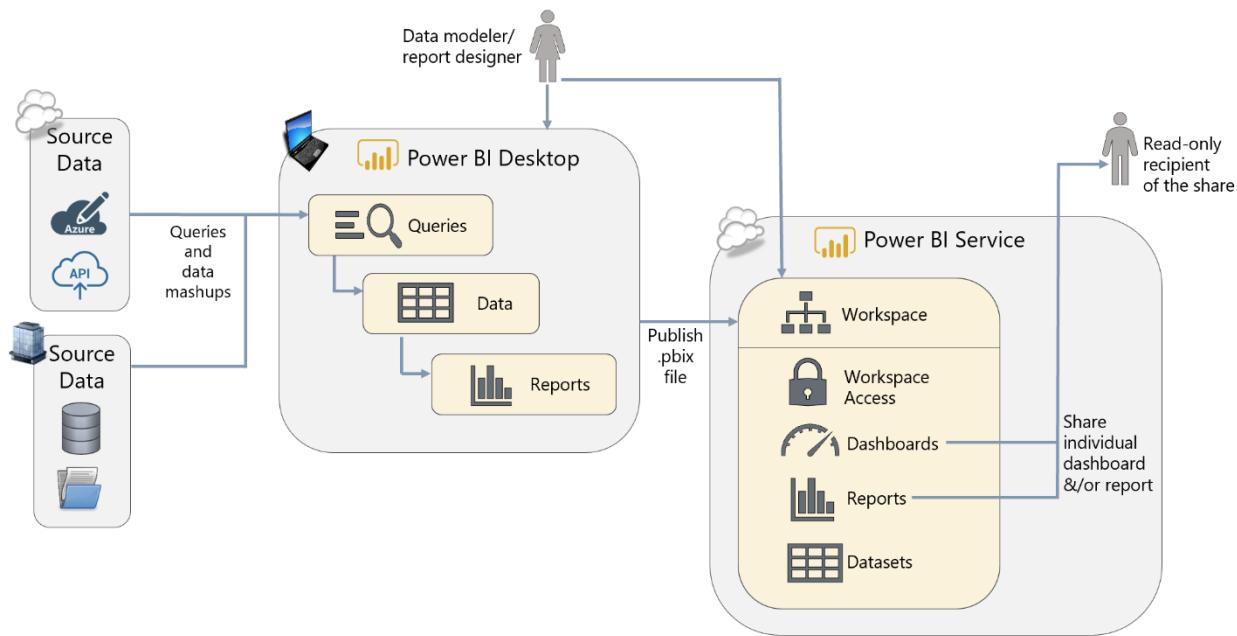


Figure 7 - Diagram of prototyping activities and sharing functionality in the Power BI service

Power BI Desktop. The authoring tool can be used during interactive prototyping sessions conducted with business users and subject matter experts. Prototyping sessions are very useful for validating calculation rules, how the dataset should be designed, and how the reports should display data.

Power BI Service. Optionally, the Power BI service can be useful when you need to share preliminary results for feedback and decision-making purposes. It is recommended that you segregate a prototyping solution into its own workspace, so it is clearly segregated from non-production content.

Workspace. A workspace is appropriate in this scenario since we are dealing with a small team-based collaboration scenario. With a workspace in place, once the solution has been finalized it can quickly be promoted to production, or operationalized, by publishing to the appropriate workspace, then creating an app for delivery of content to read-only users.

Sharing Reports & Dashboards. The above diagram depicts sharing directly to a recipient (without usage of an app). Sharing is appropriate for collaboration scenarios when colleagues work together in an ad hoc way, and typically works best when limited to a small number of colleagues who primarily need to review content. Another valid alternative, not depicted, is to specify the colleague as a member in the workspace (as was shown in the Small Team Collaboration scenario). Adding colleagues to a workspace works best when those colleagues need access to many pieces of content in the workspace and/or have reason to see all the content of the workspace rather than just one item. We recommend that the use of direct sharing be minimal.

Embedding for your Organization

The [Power BI REST APIs](#) can be used for embedding dashboards and reports into custom applications, SharePoint Online, Microsoft Teams, or even in a third party application such as Salesforce. Making analytical data easier for business users to access, by integrating visuals in the tools and applications they use every day, allows data-driven decision-making to become more prevalent and improves workflow efficiency for functional users.

As discussed in [Section 3](#), there are two main techniques for delivering embedded analytics with Power BI: Embedding for your Organization and Embedding for your Customers. This first technique describes the usage of Power BI Premium (P or EM SKUs) for internal/organizational users:

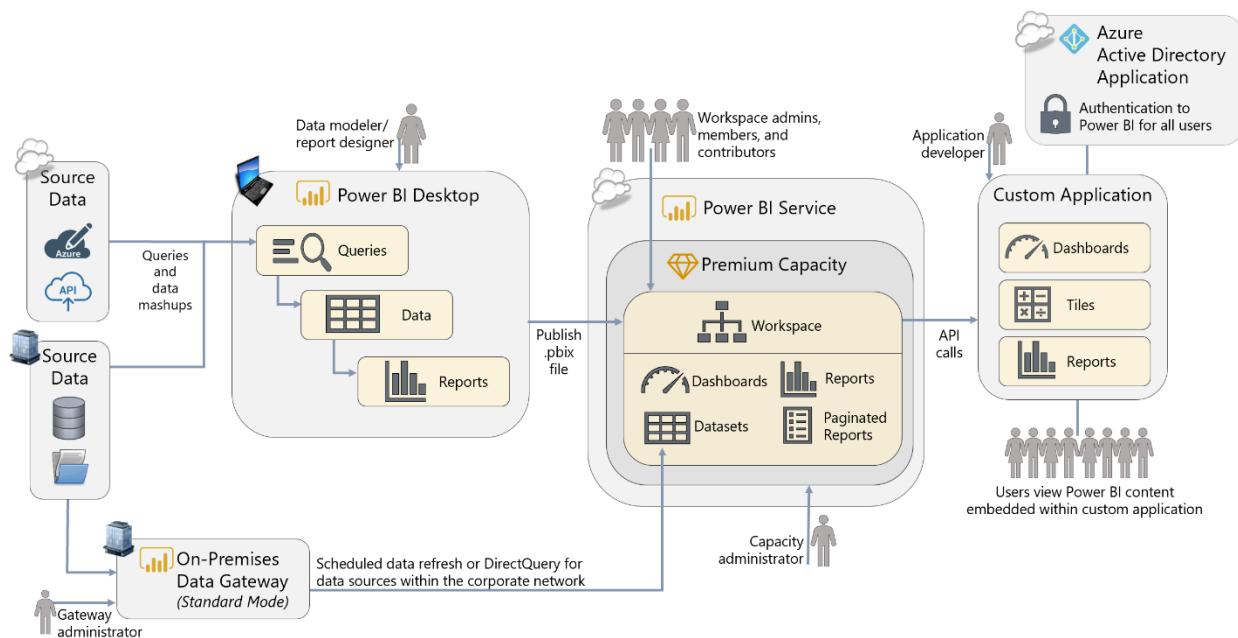


Figure 8 - Diagram of embedding for your organization with the Power BI REST APIs

In smaller organizations where all users have Pro licenses, or in organizations covered by Office 365's E5 license, organizational embedding works for all users who have Pro licenses. Or, as depicted in the next section.

Power BI Desktop. The authoring experience is the same in Power BI Desktop, even if you intend to consume the reports within a custom application. This is beneficial because a business user who is the subject matter expert can change content and publish to the Power BI service with no changes to their normal practices. If row-level security is required, it is defined in Power BI Desktop using the same technique as when the reports are delivered in the Power BI service.

Workspace. The .pbix file needs to be published to the Power BI service so the APIs can render report items embedded in the custom application. The reports, dashboards, and datasets should be published to a specific Power BI workspace which has been designated for the embedded analytics project. Note that publishing to an app, however, is optional for this scenario - the workspace suffices since users will view content directly in the application.

Authentication and Authorization. When embedding content in an organizational scenario utilizing Power BI Premium (as opposed to utilization of Power BI Embedded described in the next scenario), every user is authenticated to Power BI to view the data embedded in the custom application. Therefore, authentication to Azure Active Directory (AAD) is based on the user's individual AAD account. This allows the embedded content to leverage the same row-level-security configured for your analysts and report consumers in Power BI.

Power BI Premium. Since Power BI is aware of every user in this scenario (see the next scenario for the alternative), all users will need a Pro license -or- the content needs to reside in Premium capacity-backed workspace so that Power BI Free users may view the content as well. For this reason, Power BI Premium is often discussed in conjunction with embedding scenarios.

Embedding for Your Customers (External Embedding)

The [Power BI REST APIs](#) can be used for embedding Power BI content into custom applications which are used externally by customers and partners.

As discussed in [Section 3](#), there are two main techniques for delivering embedded analytics with Power BI: Embedding for your Organization and Embedding for your Customers. This second technique, most used with independent software vendors (ISVs), describes the usage of the A SKUs in Power BI Embedded for rendering content for external users:

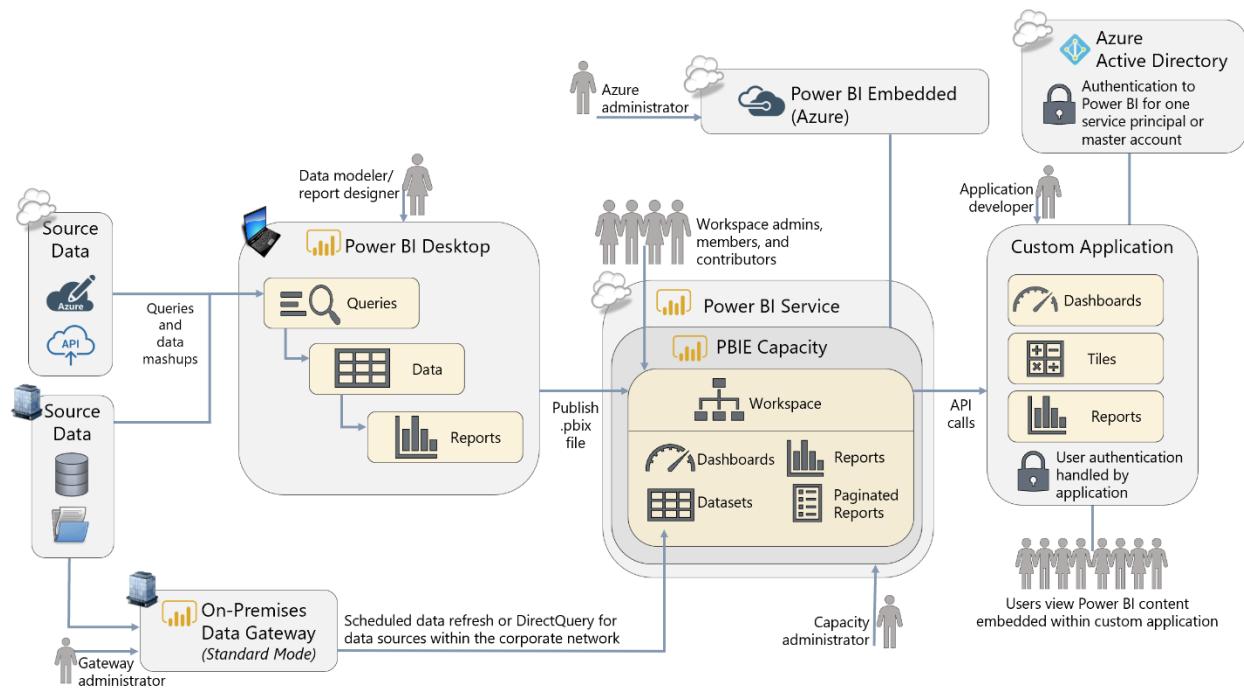


Figure 9 - Diagram of embedding for your customers with the Power BI REST APIs

Power BI Desktop. The authoring experience is the same in Power BI Desktop, even if you intend to consume the reports within a custom application. This is beneficial because a business user who is the subject matter expert can change content and publish to the Power BI service with no changes to their normal practices. If [row-level security](#) is required, it can be defined as well.

Workspace. The .pbix file needs to be published to the Power BI service so the APIs can render report items embedded in the custom application. The reports, dashboards, and datasets should be published to a specific Power BI workspace which has been designated for the embedded analytics project. Note that publishing to an app, however, is optional for this scenario - the workspace suffices since users will view content directly in the application.

Authentication and Authorization. When embedding content via use of Power BI Embedded, the application user is authenticated and authorized by the application. The application end user is not required to have a Power BI license. The custom application makes the connection to Power BI, which it does through Azure Active Directory based on either a service principal or a master account. This is a significant difference than the previous scenario.

Power BI Embedded. A subscription in Azure to Power BI Embedded provides capacity resources for rendering Power BI content which is embedded in a custom application. Specifically, this scenario refers to purchasing an A-series SKU from Power BI Embedded. The Power BI Embedded service can be paused when not in use (during which no costs are incurred), which can be useful for dev/test scenarios. Power BI Embedded can also be scaled up or down

to handle the peaks and valleys of varying workload levels (whereas Power BI Premium is not intended for frequent scaling).

Workflow Automation + Updating Data from Within Power BI

The Power Platform family of tools allows for many interesting use cases which involve integration between these business applications. There are many possible scenarios, and the integration is continually improving.

In the following scenario, envision that we are checking for data quality issues or missing data. Power Automate is used to send a customized e-mail alert which directs a user back to Power BI to update the source data:

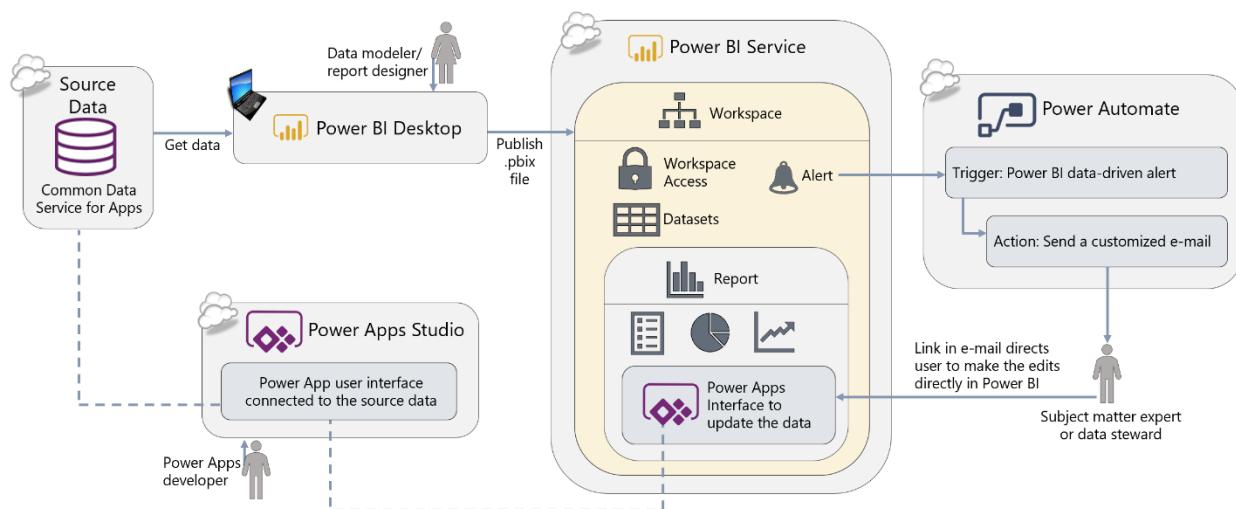


Figure 10 - Diagram of workflow automation and updating data from within the Power BI service

Power BI Desktop. The Get Data experience within Power BI Desktop is the same as usual. Creation of queries, calculations, datasets and reports all work as expected.

Alerts in the Power BI Service. Alerts in the Power BI service can be used to identify when an issue exists. This could be many things, such as missing data, invalid data, presence of a critical situation, or exceeding a numeric threshold. Alerts can be set up on dashboard tiles which display a singular value such as gauges, KPIs, and cards. Alerts are fired when data is updated in the Power BI service.

Power Automate. Power Automate is a Power Platform service for workflow automation between services. Used in conjunction with alerts in Power BI, Power Automate can be used for many types of workflow automation, such as:

- Send a customized e-mail alert or text message
- Create task in tools such as GitHub, Trello, or a SharePoint list
- Post a message on forums such as Teams or Slack
- Request an approval
- Trigger another workflow

In the above diagram, Power Automate is being used to send a detailed, data-driven message to the subject matter expert once the Power BI alert has triggered the Flow in Power Automate to kick off. There are many, many interesting use cases that can be done with Power BI and Power Automate.

Power Apps. Power Apps is a Power Platform service for creating small line of business applications and user interfaces in a low-code way. Power Apps can be used for many things, such as:

- Task and issue tracking
- Budget input
- Cost estimations
- Surveys

Within the e-mail message shown in the above diagram, a URL is provided which directs the recipient back to the Power BI service to make the data updates – this is for user convenience since the user is frequently in the Power BI service anyway. The user interface being exposed in Power BI is backed by Power Apps, so the data updates could certainly be made directly in Power Apps if preferred. When updates are done via the interface in Power Apps, the source data is immediately updated; the updates will be reflected in Power BI the next time a data refresh operation is executed – or DirectQuery could also be used. There are many interesting use cases for integrating Power Apps into Power BI, as well as the inverse where Power BI is integrated into a Power App.

Self-Service Data Preparation and Machine Learning

As discussed in [Section 3](#), the objective of dataflows is to centralize data cleansing activities. Power BI dataset authors can then obtain data from the dataflow rather than from the original data source.

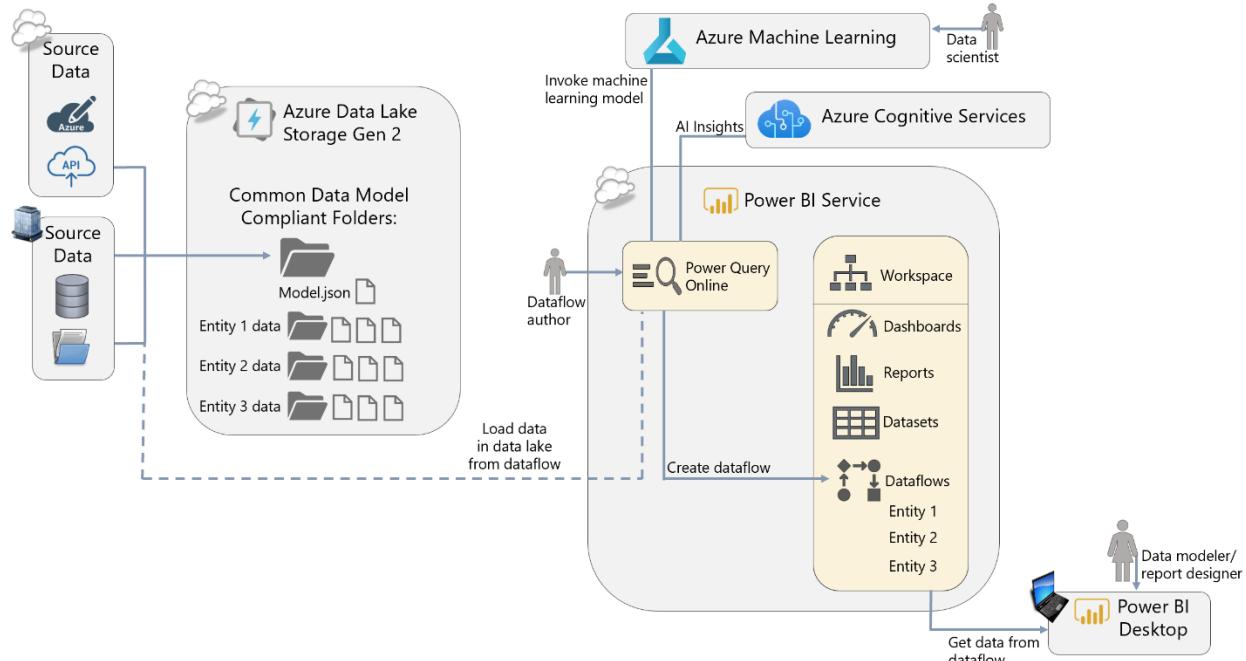


Figure 11 - Diagram of self-service data preparation and machine learning in the Power BI service

Power BI Desktop. To consume data output from dataflows, use the standard 'Get Data' experience within Power BI Desktop. Creation of queries, calculations, datasets, and reports all work as expected in order to create interactive reports. This is because the dataflow is just a data source for a Power BI dataset.

Dataflows. Data becomes inherently more valuable once it has been integrated with related data. Accordingly, dataflows are intended to make data integration from various sources easier. Dataflows are a workspace artifact like datasets, reports, and dashboards. A key advantage of dataflows is that the data integration work is done only once, and then reusable among many Power BI datasets. Entities can be linked, to avoid loading the data multiple times.

Azure Data Lake Storage Gen 2. The output from a dataflow resides in Common Data Model-compliant folders in Azure Data Lake Storage Gen 2. A folder is considered 'compliant' due to the presence of a metadata file which contains schema and metadata information associated with the data file(s). The default state is usage of a built-in storage account provided to the customer. Alternatively, a data lake account within an existing Azure subscription can be used if preferred. Most large organizations will prefer using their existing Azure subscription to be able to directly access the data in the data lake.

Azure Machine Learning. Functionality for [integrating with Azure Machine Learning models](#) is integrated with Power BI dataflows. A data scientist can publish the model to Azure Machine Learning, then Power BI dataflow authors with permission to the model can invoke it for the purpose of enriching the dataflow output.

Azure Cognitive Services. [Integration with Azure Cognitive Services](#) models are available for Power BI dataflows which reside in a Premium capacity-backed workspace. Power BI dataflow authors may use built-in algorithms for enriching the output of the dataflow.

Power BI Premium (optional). Usage of dataflows in conjunction with Power BI Premium makes additional capabilities available such as linked entities, computed entities, Azure Cognitive Services, among other features discussed in [Section 3](#). It is possible to get started using dataflows without Premium and evolve into using Premium as the solution grows.

Near Real-Time Reporting

Access to near real-time data allows the business to react faster when time-sensitive data is involved.

Use of Automatic Page Refresh, in conjunction with DirectQuery, is the easiest way to implement near real-time reporting. Further details are covered in [Section 6](#).

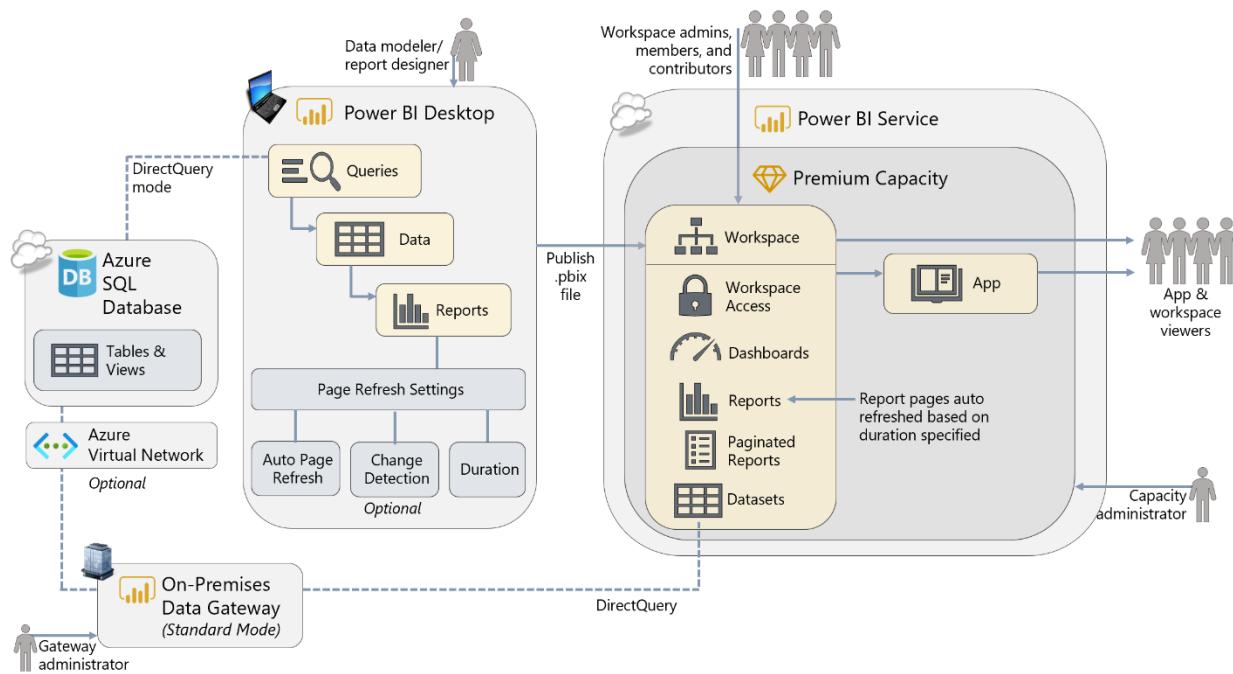


Figure 12 - Diagram of near real-time reporting in Power BI Desktop and the Power BI service

Power BI Desktop. Power BI Desktop remains the authoring tool for queries, datasets, and reports. Each report page has specific properties for configuring automatic page refresh, including the duration. The dataset must be in DirectQuery mode in order to configure automatic page refresh.

Change Detection (optional). As further discussed in [Section 6](#), the use of change detection (available with Premium) is a way to reduce the amount of data continually being sent which is helpful when data arrives on an inconsistent pace. Change detection uses a specific measure to determine if a change occurred; if affirmative, then all queries are sent to refresh the page.

Power BI Premium (optional). The minimum refresh interval in shared capacity is 30 minutes; the use of Premium capacity allows the minimum level to be configured by the capacity administrator. The use of change detection is also available with use of Premium capacity.

On-Premises Reporting Portal

An additional benefit of Power BI Premium (also available with a SQL Server Enterprise license + Software Assurance) is the ability to install and publish Power BI content to a Power BI Report Server which resides within the corporate network or an Azure virtual machine.

Power BI Report Server is useful for customers who wish to have an alternative to the Power BI service for deployment:

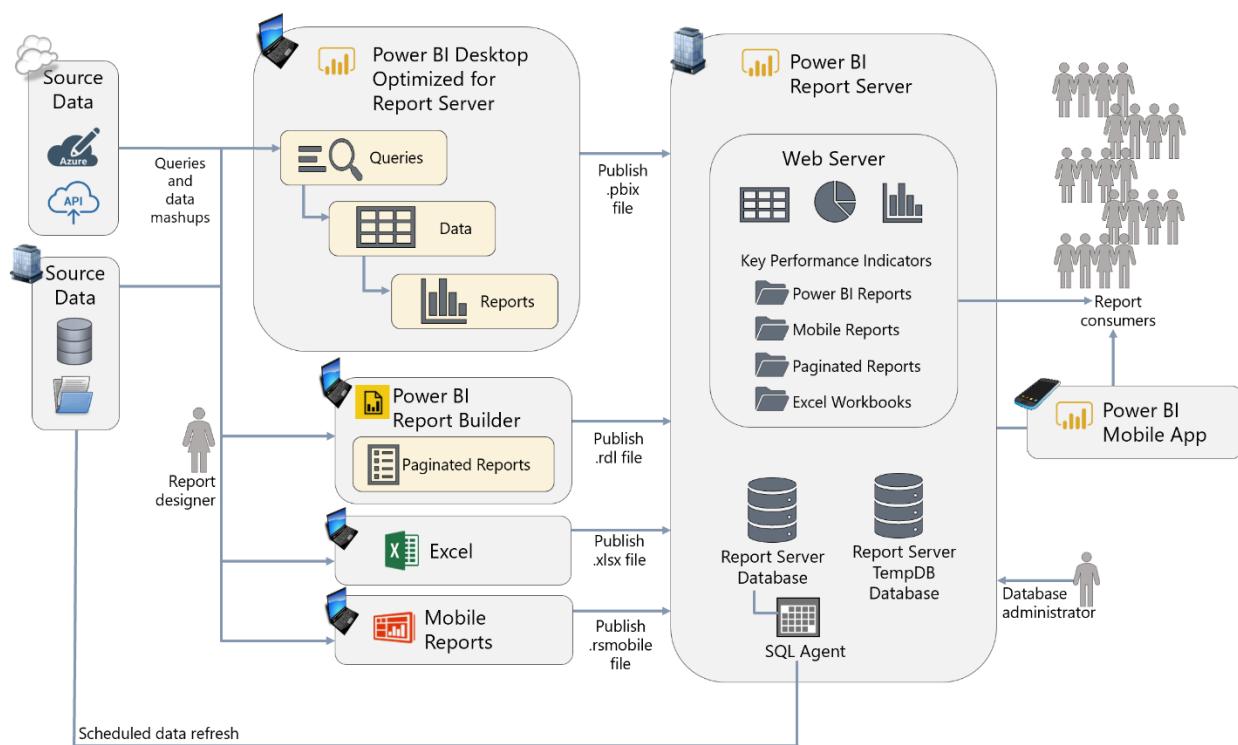


Figure 13 - Diagram of Power BI Report Server

Power BI Desktop Optimized for Report Server. The authoring experience remains the same for report authors. However, Power BI Desktop authors must use a version of Power BI Desktop which is designated as being compatible with the Power BI Report Server release cycle. Updates are released approximately 3 times per year.

Power BI Report Server. The Power BI Report Server offers a repository and user portal for delivering Power BI reports alongside paginated reports, Excel reports, and mobile reports. See [Section 3](#) for additional discussion about Power BI Report Server.

Section 3. Power BI Architectural Choices

This section discusses some of the most important decisions you must make when planning a Power BI enterprise deployment:

- The decision whether to use the Power BI service, to use Power BI Report Server on-premises, or to use both
- The decision whether to use Power BI Premium
- **The decision whether to use Power BI dataflows**
- The decision whether to use one of the options for embedding Power BI content into a custom application or website

Other architectural decisions are discussed in other areas of this whitepaper. See [Section 6](#) for dataset storage options (such as import or DirectQuery). See [Section 7](#) for data gateway information.

The Power BI Service

What is the Power BI Service?

The Power BI service is a cloud service where Power BI users can:

- Discover and access data, reports, dashboards and other business intelligence-related content which has been shared with them
- Publish data, reports, dashboards and other business intelligence-related content that they have created
- Connect to on-premises and cloud data sources seamlessly, with scheduled refresh
- Share and distribute this content with authorized users, both inside and outside of the organization

The Power BI service appears to users as a web-based portal plus connected mobile applications, but it is much more than that. Among other things, it can facilitate:

- Distributing reports and dashboards to large numbers of users
- Storing and managing data used by reports
- Enabling access to analytics through mobile devices using rich device-specific applications
- Managing access to external data sources
- Applying row-level security both to reports and the underlying data
- Generating alerts
- Setting up subscriptions to inform users that new content is ready to be viewed by email
- Connecting to published data from other tools, such as Excel
- Performing centralized administration and governance activities

Why Use the Power BI Service?

Analysts may be accustomed to storing the reports they create (for example in Excel) on their PC and then distributing the files by email. Or, they may store the files on a network file share or a SharePoint folder for other people to access the files.

Solutions created in Power BI Power BI Desktop (.pbix) could technically be distributed in the same way - but there are many reasons why these methods should be abandoned in favor of the centralized approach that the Power BI service offers, including:

- A lack of centralized visibility over data access, usage, and the ability to build a single source of truth which can fuel a data culture.
- The storage of reports in a variety of locations makes it difficult for users to discover all the content that is available to them. Although it is recommended that organizations have a policy for where Power BI .pbix files should be located, it becomes difficult to enforce security and compliance policies for a myriad of file locations.
- Different versions of the same report may exist in different places, making it difficult for users to know which one is correct.
- Distributing reports as files makes it very likely that others will alter calculations or even the data model structure.
- Updating and distributing reports can be a time-consuming manual process, one that often breaks down when the person responsible is ill, on vacation, or leaves the organization.
- When a .pbix file is opened within Power BI Desktop, much more than just a report is visible to the user: they can also see raw data, calculation definitions, and other technical details of the implementation. For executives and consumers, this can be confusing and require extra training to deal with. This experience can be a detriment to adoption of Power BI solutions.
- Conveniences such as scheduled data refresh, alerts, and subscriptions cannot be implemented with Power BI Desktop alone.
- Row-level security cannot be implemented when delivering reports via Power BI Desktop alone.

- Power BI Desktop is a Windows application, so for Mac users viewing content in the Power BI service is ideal.

The Power BI service addresses all the above concerns, making it easy for users to discover the information they need to do their job. The Power BI service is considered a place where users can collaborate, share, and distribute content in a more reliable manner with additional security options.

Most consumers (who don't author datasets or reports) of Power BI content use the Power BI service exclusively. The following are reasons why you may want to consider using the Power BI service in your Power BI deployment:

Fast Setup and Little Maintenance

Once you have decided to use Power BI, the Power BI service can be up and running for your organization very quickly – usually within a few minutes. The Power BI service includes data storage and connectivity to services such as Dynamics and Salesforce – this allows business users to quickly start using certain data sources for making decisions without waiting for IT to provision data sources or integrate data. As a cloud service, Microsoft handles all maintenance, updates, and patching activities. It is also natively integrated with Azure Active Directory, so users and groups are unified.

Subscription Pricing

User-based licensing options for the Power BI service are based on a monthly subscription. Even when an organization chooses to incorporate additional paid capabilities, such as Power BI Premium, the up-front costs are significantly less than a server-based approach which requires purchasing and provisioning server hardware and software. As usage of Power BI grows, you can purchase additional licenses and services to support the expansion.

Content Distribution and Collaboration

The Power BI service is available in the cloud, allowing your users to access content from anywhere through device-specific mobile apps, a web browser, or customized portals. This allows your users to use data effectively wherever they are located. Access to content can be constrained to users within the organization or it may also include access for authorized external partners, customers, and vendors which you allow to be a guest user.

Governance, Administration, and Usage Monitoring

When the Power BI service is used as the destination for publishing user reports and dashboards, it gives administrators a single, centralized place to enforce governance and support users. This also permits auditing of user activities, and understanding how solutions are (or are not) being used which is crucial to a successful deployment. Additionally, Power BI administrators have a variety of options available to govern use of the environment.

Access to the Full Functionality of Power BI

The Power BI service provides access to the full functionality of Power BI and includes several features (such as dashboards, dataflows, apps, alerts, subscriptions, AI visuals, and Quick Insights). The best and most comprehensive user experience includes the use of the Power BI service.

Power BI Service Modern Release Cycle

Updates to the Power BI service are released frequently, usually on a weekly basis. New features and improvements are announced on the official [Power BI blog](#) and are described in more detail in the documentation. The goal for the frequent release cycles is to continually enable greater productivity.

 Some preview features in Power BI Desktop may not be supported in the Power BI service immediately.

Power BI Premium

What is Power BI Premium?

Power BI Premium is an offering for organizations with a greater number of read-only users, and/or larger data volumes, and/or more demanding data requirements. As such, Power BI Premium is aimed squarely at the enterprise market.

Power BI Premium provides the following high-level benefits:

- More **cost-effective licensing options when there are large numbers of read-only users** who only need to view (rather than create their own) content. These view-only users may be internal or external to the organization. Premium permits consumers to view content with just a Power BI Free license, provided that the content resides in a workspace added to Premium capacity. This Free user benefit translates to embedded content as well.
- Use of **paginated reports** to meet requirements for highly formatted, pixel-perfect reports. Paginated reports may be published alongside Power BI and Excel reports in a workspace that has been added to Premium capacity.
- Additional dataset reusability via use of **XMLA endpoint** for accessing a dataset published to the Power BI service from other reporting tools.
- **Higher limits** for individual dataset size, and the overall amount of data that can be stored in Power BI.
- Access to functionality that supports working with **larger data volumes**.
- **Isolated, dedicated hardware**, operating in single tenancy mode rather than multi-tenancy. All P-series SKUs are isolated, as well as EM3 and above.
- Ability to define the geographic area for data storage, to meet **data residency requirements**.
- Additional **control** over Power BI operations, for example frequency of data refresh.

- More **consistent query performance**, especially when there are many users viewing reports and dashboards.
 - Additional features for **Power BI dataflows**, such as linked entities, computed entities, enhanced compute engine, among others.
 - Use of **AI capabilities** associated with Azure Machine Learning and Azure Cognitive Services.
 - Ability for customer to manage their own **encryption key**, to meet security or compliance requirements.
 - The additional option to use **Power BI Report Server** for publishing certain content on-premises and/or using a hybrid approach to content delivery.
-



Performance Expectations for Power BI Premium

Scale, control, consistency, and cost-effectiveness are the main objectives for Premium. Although performance may be more predictable with Premium, improved performance is not typically a *primary* objective. In fact, performance may be reduced if the capacity is under-provisioned (with not enough resources) or over-assigned (with too many workspaces allocated to the capacity, thus it cannot support the simultaneous workloads). Monitoring the health of Premium capacity is very important to avoid performance issues as well as ensuring that you are not paying for more than you need to.

To understand these benefits, the remainder of this section is devoted to the technical details of how Premium works while the licensing benefits are explored in detail in [Section 4](#). Throughout the remainder of this whitepaper, there will be references to how Premium enhances various aspects of Power BI. A brief discussion of managing Power BI capacity can also be found in [Section 10](#) of this whitepaper.

Power BI Premium Capacities

Power BI workspaces are stored in capacities. When a Power BI tenant is created, initially it has one capacity available: Power BI shared capacity. The resources supporting shared capacity are allocated to all Power BI customers (multi-tenancy mode). Accordingly, Microsoft has implemented a series of limits and throttles to ensure that all customers receive the same high quality of service. Examples of these limits include maximum size of a dataset, the maximum amount of data that can be stored in a workspace, and the maximum number of dataset refreshes that can take place in a day.

When you purchase Power BI Premium, you purchase one or more Power BI Premium capacity nodes. A Power BI Premium capacity node can be thought of as something like a virtual machine: it is a [specified amount of computing resources](#) within the Power BI service that can only be used by your organization. Since these resources are no longer shared among multiple customers, the limits and throttles that exist in Power BI shared capacity can be relaxed or even removed altogether.

After a capacity node has been purchased, the resources it provides must be configured as one or more Premium capacities within the Power BI service. These Premium capacities may then be used to host workspaces. When a workspace is moved from shared capacity and assigned to a Premium capacity, it can take advantage of additional features as well as relaxed limits.

The following is a depiction of how workspaces can be allocated to multiple Premium capacities, as well as shared capacity, within the Power BI service:

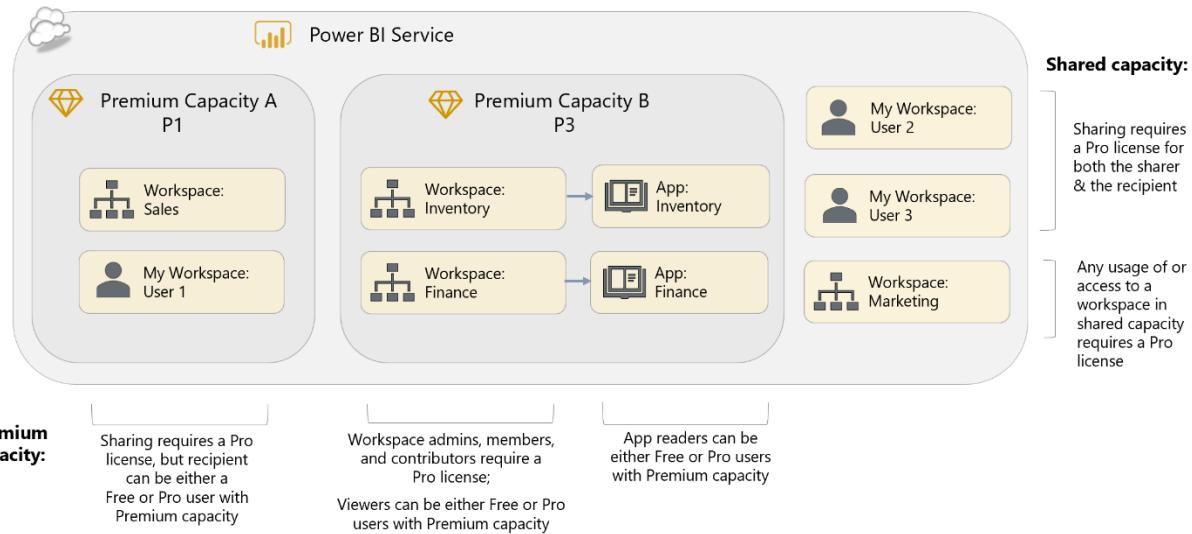


Figure 14 - Depiction of allocating workspaces to multiple Premium capacities

In the above diagram:

- Capacity A contains an workspace called Sales which has a demanding workload.
- Capacity A also supports a "My Workspace" area for User 1, which is a private workspace where User 1 can create and manage personal reports or dashboards.
- Capacity B supports an Inventory app and Finance app.
- My Workspace for User 2 + User 3, as well as the Marketing workspace, all remain in shared capacity (i.e., the standard multi-tenant service).

Premium Tiers and SKUs

Power BI Premium is available in two tiers for organizational use, each of which are divided up into several SKUs. Each of the two tiers is intended for different use cases. They are:

1. **EM-series**, which is intended for use with organizational embedding scenarios, discussed later in this section.
2. **P-series**, which supports the full array of features and functionality mentioned earlier in this section.

Note that A-series SKUs, from Power BI Embedded, are discussed later in this section. A-series SKUs may be used instead of P-series SKUs in some situations.

The Power BI Premium EM-series SKUs have a [subset of the functionality available](#) in the P-series SKUs and are less powerful in terms of the resources they provide.

The differences between the two sets of Power BI Premium SKUs may be summarized as follows:

	Power BI Premium EM-Series	Power BI Premium P-Series
Share embedded content to organizational users with Power BI Free licenses via Microsoft Teams, SharePoint Online, and custom applications	✓	✓
Share content broadly to organizational users with Power BI Free licenses in the Power BI service portal	No access to the Power BI service	✓
Additional features such as paginated reports, XMLA endpoint, full dataflows features, etc.	No access to the Power BI service	✓
Use of Power BI Report Server (optional)	No access to Power BI Report Server	✓

Within each tier, the SKUs are differentiated by the number of virtual cores (known as "v-cores"), the amount of storage each one provides, additional limits including the number of DirectQuery/live connection queries per minute, and the number of page renders per hour (above which throttling and slow-downs occur), and their cost. The v-cores themselves are divided into [two types](#):

- **Back-end v-cores** which handle query processing, dataset refresh, natural language processing, and other data-related activities like report rendering. Cache management (i.e., active datasets loaded to memory) is also handled on the back-end v-cores.
- **Front-end v-cores** which are responsible for the web service and user experience. Tasks such as document management, access rights management, APIs, uploads, and downloads are also handled by the front-end v-cores.

A given SKU might have a different number of front-end and back-end v-cores. For a complete list of all of the SKUs within each tier see the [official documentation](#). The online [Power BI Premium Calculator](#) is available as one resource for beginning to estimate the Premium SKU level needed to support your usage patterns and the associated costs. We recommend that you discuss your requirements with a Microsoft representative.

Key considerations for choosing a capacity level include:

- Various workloads to be run on the capacity including datasets, dataflows, paginated reports, and artificial intelligence
- Number of concurrent users

- Query volume generated by users interacting with reports and dashboards
- Query complexity (particularly complex DAX calculations in measures)
- Number of concurrent data refresh schedules
- Competing, concurrent query and data refresh operations
- Data size of refreshed datasets, including efficiency of filtering and query folding operations
- Use of import datasets (vs. DirectQuery or live connections, both of which delegate much of the query workload to the source system)
- Use of streaming datasets and the volumes of data they are expected to continually handle
- Number of datasets which are actively queried (dynamic memory management offloads idle datasets from memory)
- Frequency of potentially resource intensive operations

A larger Premium capacity node which has been purchased may be divided into multiple smaller capacities in the Power BI service. It is worth considering why you might consider buying one larger capacity node rather than several smaller capacity nodes:

- Larger capacities allow for **larger individual datasets** to be published. For example, the current maximum [size of a dataset](#) in a P1 capacity is 3 GB, whereas for a P2 capacity the limit increases to 6 GB per dataset.
- Multiple smaller capacities allow for **isolation of resources**. For example, if there is one group of users who run a lot of reports, their workspace(s) can reside on one capacity where it won't affect the performance of other users whose workspaces are on different capacities.

Deciding Whether to Use Power BI Premium

The benefits Power BI Premium offers are diverse, and it may be that you make the decision to purchase it because you only need one or two specific features. It is likely you will best be able to assess the need for Power BI Premium after you have used Power BI in production for several months.

Following are some questions to bear in mind while planning your Power BI deployment that will help you make your decision on whether to use Power BI Premium:

What is your budget?

Go to the Office 365 Admin Portal and/or use the [Power BI Premium Pricing Calculator](#) to familiarize yourself with the pricing for Power BI Premium. If your project does not have the necessary budget for Premium, you may need to prepare a business case to obtain the extra funding. Refer to the benefits mentioned at the beginning of this section to help with that business case.

How many users need to publish content to the Power BI service vs. how many users will only view content?

These numbers will be critical for calculating your licensing costs and, in turn, working out whether purchasing Power BI Premium is more cost-effective than buying Power BI Pro licenses alone. Please see [Section 4](#) to gain an understanding of the different licensing options. At the beginning of a Power BI deployment it will be difficult to estimate these numbers. The more content you publish, the more users you will gain. As Power BI expands throughout your organization, you may be surprised by the number of users who embrace it as well as the way it gets used. Continually monitor actual usage (see [Section 10](#) for more details) as your adoption grows, and periodically review your licensing so that you know whether you have reached the point where buying Premium is the right choice.

Do you expect your users to view content in the Power BI service or the Power BI mobile apps, or will it be enough to view Power BI content embedded in another portal or line of business application?

If you have a relatively small, stable number of Power BI reports that can be delivered exclusively as embedded content, for example in SharePoint Online or in a custom portal, you can use the less expensive Power BI Premium EM-series SKUs rather than the P-series SKUs. However, do not underestimate the level of features and flexibility the Power BI service adds to the overall Power BI experience.

Do you need flexibility to use all possible deployment options for internal and external users?

The use of P-series SKUs is the only choice which makes *all possible* deployment options available, including: Power BI service, embedding for your organization, embedding for your customers, and Power BI Report Server.

How much data to you intend to store within the Power BI service, and how often do you need to refresh that data?

The increased limits on the size of a dataset, on the overall amount of data that can be stored in Power BI, and on how often that data can be refreshed, are compelling reasons to purchase Power BI Premium. To reiterate: estimating data volumes at the beginning of a deployment can be very difficult, so monitoring data volume growth over time is necessary to determine when Premium becomes necessary. In addition, as discussed in [Section 6](#), using different data storage options such as Analysis Services can reduce the amount of data that needs to be stored in Power BI.

How many concurrent users will need to view content within the Power BI service?

As with data volumes, ongoing monitoring is necessary to be aware of how many concurrent users view content in Power BI at any one time, and how well Power BI can support their requests. Poor performance can lead to user dissatisfaction and discourage users from going to Power BI. Purchasing Power BI Premium may alleviate some of those concerns.

Do you have requirements for highly formatted reports?

Workspaces which are assigned to Premium capacity are permitted to use paginated reports (Reporting Services RDLs), in addition to Power BI and Excel report types. The ability to publish content from three different types of reporting tools offers significant flexibility for meeting report requirements while utilizing a single portal for delivery of that content.

Do you intend to use Power BI Report Server for hybrid cloud/on-premises content delivery?

The only two ways to use Power BI Report Server are (a) if you buy a Power BI Premium P-series capacity node, or (b) if you have SQL Server Enterprise Edition licenses with Software Assurance. Details on licensing for Power BI Report Server are discussed in [Section 4](#).

Are there specific compliance requirements?

Certain aspects of Power BI Premium, such as dedicated resources, geographic location of data, and customer encryption key management can aid in meeting compliance requirements.

Do you have multiple report tools in your organization, and wish to take advantage of the investment in Power BI datasets?

Use of the XMLA endpoint for connecting other reporting tools is one way to be able to use a highly curated Power BI dataset beyond Power BI alone.

Power BI Dataflows

Why Use Power BI Dataflows?

The primary scenario which is enabled by [Power BI dataflows](#) involves use of a **centralized data repository for self-service data preparation**.

With self-service BI implementations, common downsides include duplication of effort and inconsistency of data produced across various self-service solutions. Historically, these types of concerns have been reduced by introducing a corporate data warehouse and/or centralized semantic layer where all the data preparation has been handled by a BI team. As important as centralized systems like this are (and remain), they cannot always handle every user need. They can be slow to react to changing requirements, and they require highly skilled IT professionals.

Power BI dataflows are geared towards:

- Content authors who are comfortable with Power Query / Power BI Query Editor for data preparation, and who want to **do the work of data cleansing only once**, rather than in every individual .pbix file that contains a dataset. This is particularly helpful when transformations and cleansing rules have grown complex and more complicated to maintain.

- Content authors who want to **consume data that has already been pre-prepared in order to save time, reduce duplication of effort, and improve consistency.**
- Situations when you need to **reduce the load placed on source systems to extract data.** With dataflows, you extract data from the source into the dataflow – then all downstream datasets use data from the dataflow without repeatedly returning to the source.
- Situations when there is **not already a data warehouse or cleansed, consolidated reporting system already in place** as a source for datasets. Or, the data you need is not contained in the data warehouse and will need to be used across multiple datasets.
- Situations where **staged data** needs to be delivered to Power BI data modelers to finish preparation and delivery.



There are two types of dataflows: Power Platform dataflows, and Power BI dataflows. This section only addresses Power BI dataflows. Exploring Power Platform dataflows is beyond the scope of this whitepaper. We recommend that you take ample time to educate yourself on the similarities, differences, and level of integration between Power Platform dataflows and Power BI dataflows if you plan to use both.

Power BI dataflows work particularly well for commonly used data such as an organization's master list of products, customers, or suppliers.

What are Power BI Dataflows?

Power BI dataflows can be thought of as:

- A pipeline to extract, transform, and load data, plus
- The entities that the data is loaded into

Power BI dataflows are created using Power Query Online, which is a web-based version of the same Power Query engine that is used in Power BI Desktop. After it's been created, the dataflow is another object in a workspace alongside datasets, dashboards, reports, and workbooks.

When a Power BI dataflow is refreshed, behind the scenes it loads its data into files located in a data lake.

This data lake is in an Azure resource called Azure Data Lake Storage Gen 2, which is discussed later in this section.

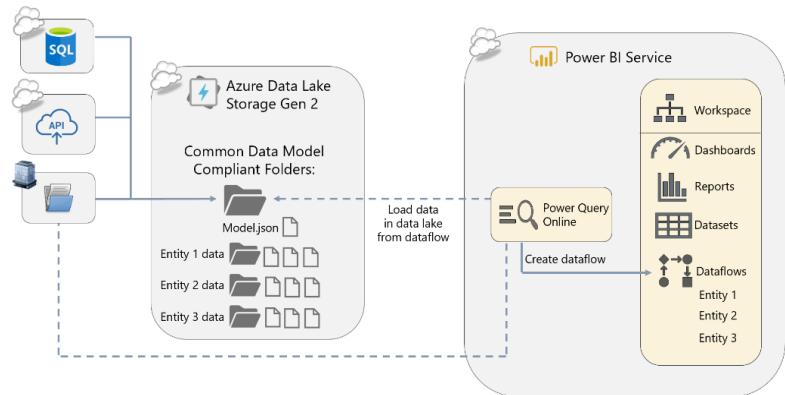


Figure 15 - Depiction of dataflows in the Power BI service with data stored in ADLS Gen 2

Power BI dataflows do not negate the need for datasets. Datasets are still created normally including relationships, measures, and row-level security. To use an existing dataflow, the Get Data experience in Power BI Desktop is used to access a dataflow:

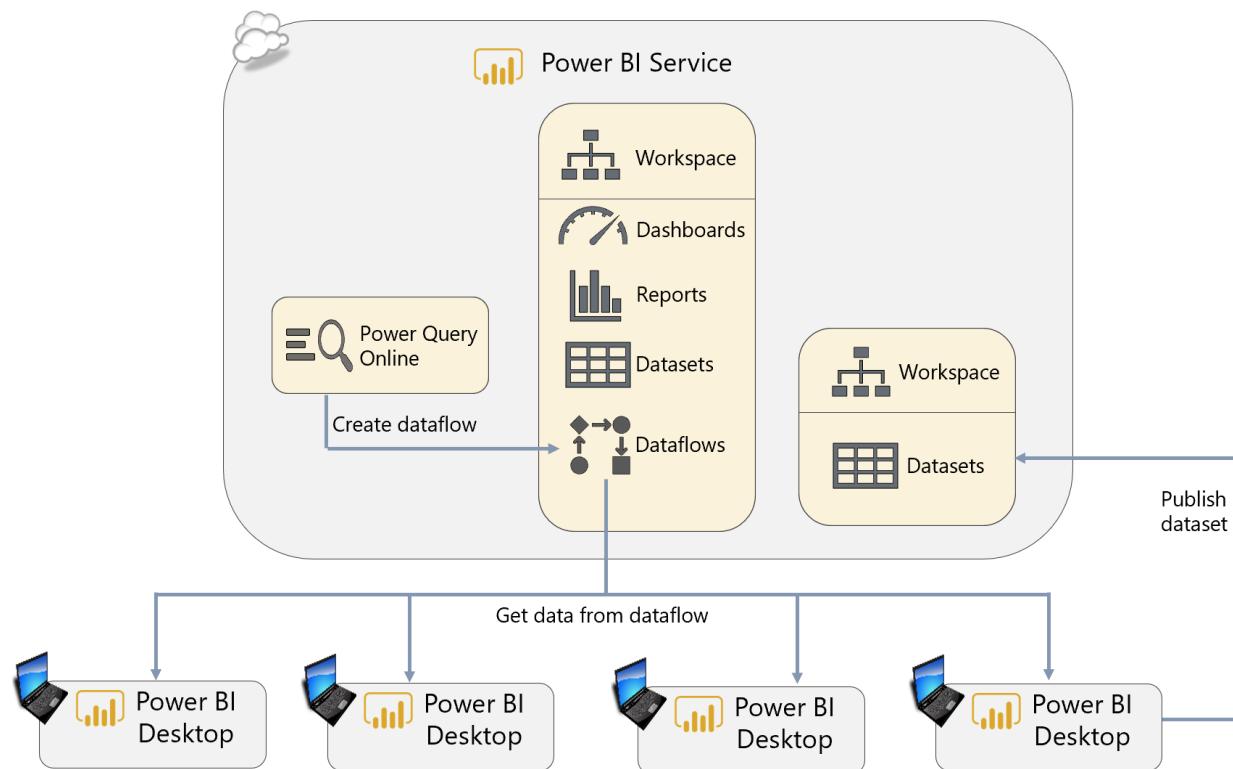


Figure 16 - Depiction of reusing one dataflow across numerous datasets

To gain maximum value from the use of Power BI dataflows, as much data preparation should occur in the dataflow as possible (or in the source for the dataflow, when appropriate). This improves consistency in the downstream datasets and reduces work for the dataset authors. Typical work includes:

- Friendly, readable column names using business terminology
- Creation of calculated columns when necessary
- Data types defined correctly
- Unnecessary columns are removed

Downstream datasets are refreshed using the dataflow as the source. Dataflow refresh and dataset refresh are separate operations.

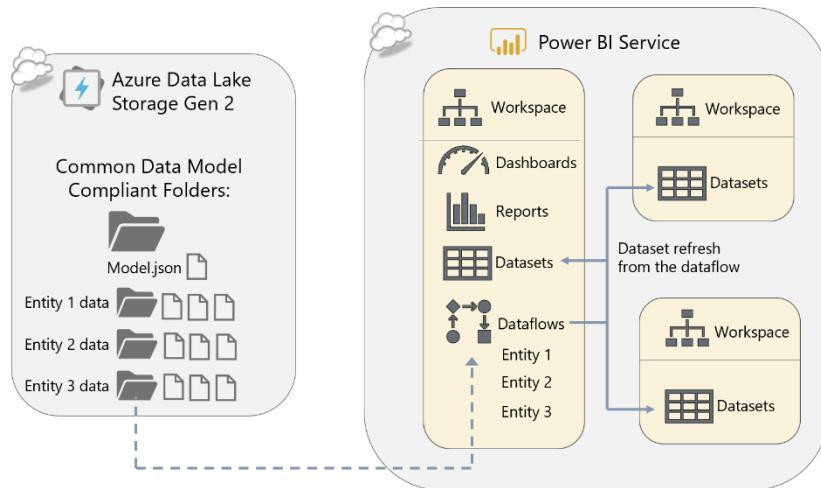


Figure 17 - Depiction of separate data refresh operations for dataflows and datasets

At the time of this writing, [DirectQuery support for Power BI dataflows](#) is in preview (not depicted above).

Two Types of Data Lake Storage with Power BI Dataflows

Power BI dataflows are always stored in Common Data Model (CDM) folders in Azure Data Lake Storage Gen 2 (ADLS Gen 2). By default, dataflows are stored in Power BI-provided data lake storage. Optionally, customers may choose to associate their own ADLS Gen 2 storage account, in which case Power BI will instead store their dataflows CDM folders in that data lake storage.

This happens behind the scenes regardless; for companies who have an existing ADLS Gen 2 account in their Azure tenant, they can [associate that data lake account](#) with their Power BI tenant.

When should we use the built-in dataflows storage?

When using the built-in data lake which is available using the Power BI service, dataflows may only be accessed via the dataflow connector. This is recommended for customers who:

- Intend to only use the dataflow connector as the interface to the data.
- Don't have a need to directly view or connect to this data other than through the dataflow connector.

When should we use an organization-provided ADLS Gen 2 account for dataflows storage?

When using an organization-provided data lake account (also known as "bring your own data lake") there is more flexibility. This is recommended for customers who:

- Would like to access the dataflow's CDM folder directly in the data lake storage. Use cases for this include:
 - Accessing the data from Azure Data Services such as Azure Data Factory, Azure Databricks, Azure Machine Learning, among others.
 - Accessing the data through custom tools.

- Are using Power Platform dataflows which output to analytical entities in ADLS Gen 2, which means you already have an existing data lake account you can associate with the Power BI service. (Note, however, that the Power BI dataflows will store its data in a "powerbi" filesystem whereas Power Platform dataflows stores its data in a separate "power-platform-dataflows" filesystem. Although the store data in different filesystems, they both do store data in CDM-compliant folders and files.)
-



Two ways to approach Power BI dataflows with "bring your own data lake"

Once you've associated an Azure Data Lake Storage Gen 2 account to your Power BI tenant, you can choose on a dataflow-by-dataflow basis:

- **If Power BI fully manages the dataflow** (in Power Query Online). Other tools can access the data as well.
 - **If Power BI is a consumer of the dataflow.** The data is managed elsewhere, by another tool and process.
-

As the integration with ADLS Gen 2 continues to mature, it is important to be aware of certain limitations, such as:

- Dataflows created in Power BI store their data in either Power BI-provided storage or the organization's ADLS Gen 2 account. Dataflows created in Power Apps, however, can either store data in either Common Data Service or the organization's ADLS Gen 2 account.
- Only one storage account provided by the organization can be associated with Power BI.
- Once an ADLS Gen 2 storage account is associated to Power BI, it cannot be changed. Therefore, it's not currently possible to use a separate location for testing or proof of concept.
- Power BI dataflows stored in the organization's storage account will always be stored in a subfolder with the name of the workspace in a specific filesystem (container) called "powerbi." For example, "Cleansed product data" dataflow was created in the "Marketing report" workspace so its path in the data lake storage would be "...\\powerbi\\marketing report\\cleansed product data." This level of data segregation and organization may not necessarily align with the organizational data lake design principles.
- Storing data in the explicit data lake needs to be enabled for every workspace. If not enabled for an individual workspace, the data resides in the built-in data lake instead.
- Configuring a dataflow to store its data in the organization's ADLS Gen 2 account must be enabled per individual workspace. The workspace setting will then apply to all dataflows in that workspace.

There are also some security considerations, which are noted in the [Section 11](#) of this whitepaper. Also, refer to the [best practices article on dataflows](#).

Power BI Dataflows and the Common Data Model

Dataflows store their data in Common Data Model (CDM) compliant folders. A CDM-compliant folder indicates:

- **Metadata:** The presence of a json file which contains metadata plus the schema associated with the accompanying data file(s).
- **Data:** Data files which use either a specified schema (documented in the metadata file). This could be a standard/pre-defined schema or a custom schema.

The Common Data Model is an [open source](#) initiative, part of the [Open Data Initiative](#) between Microsoft, Adobe, and SAP. The objective of the Common Data Model is semantic consistency and interoperability between tools. This interoperability is achieved by using the [Model.json file](#) for documenting the schema for the accompanying data files.

Common Data Model has its roots in Dynamics 365, thus most of the standard entities (such as customer and account) align with Dynamics. The standard entities are extensible, so customers may add additional columns in order to customize them. For non-Dynamics customers, it is highly likely that a lot of the columns will be inapplicable, so you will want to weigh the pros and cons of starting with a built-in schema versus creating a custom schema. If you intend to use CDM schemas across services and/or begin with standard reports, using standard (built-in) entities may be appealing.



Clarifying what the Common Data Model (CDM) is and what it isn't

You may have heard some people use Common Data Model (CDM) and Common Data Service (CDS) interchangeably. Misunderstandings of the scope of CDM, and how it is applied, can lead to a lot of confusion. To clarify definitions and expectations:

- There are two aspects to the CDM:
 - **CDM entity:** An entity (such as Customer or Account), with an agreed-upon schema.
 - **CDM folder:** A standard for storage of data + metadata which has been adopted by Power BI dataflows. CDM folders contain entities.
- A CDM folder is a folder in data lake storage that conforms to specific, well-defined, and standardized metadata structures and self-describing data. These folders facilitate metadata discovery and interoperability between data producers and data consumers.
- CDM folders can be created by any service that conforms to the standard. A CDM folder can contain one or more entities with a CDM entity schema.
- The CDM is not a database, and it is not a unified service.
- Technically, you can store CDM entity files (data + metadata) anywhere: a data lake, a hard drive, or any storage of your choice. Although Power BI dataflows require use of ADLS Gen 2 for data storage, technically CDM does not - because CDM is just a common schema for the purpose of saving time and improving integration across systems.

- Dataflow entities created in the Power Apps portal can also conform to CDM entity schemas. Dataflow entities in Power Apps can be stored in Common Data Service or in CDM folders in a customer-provided ADLS Gen 2 account. Whereas Power BI dataflows target analytical use cases, Power Platform dataflows typically target application development use cases + some level of analytics.
- The Common Data Service stores and manages data which is used by business applications. Data within Common Data Service is stored as entities. There are a base set of standard entities that cover typical scenarios, but customers can also create custom entities and populate them with data using Power Platform dataflows.

There is a specific [release plan for the Common Data Model](#) because it may be used not only with dataflows in the Power BI service, but also with Power Platform and Dynamics 365.

Additional Power BI Dataflows Functionality with Power BI Premium

You can get started using Power BI dataflows with a Pro license. Power BI Premium allows you to take full advantage of dataflows with [additional enterprise-level capabilities](#), such as:

Linked entities	A read-only reference to another entity, like a select statement. Enhances reusability factor by exposing an entity (such as a Date table) in different dataflows.
Computed entities	Another type of linked entity, conceptually like a select>insert statement. Relies on in-storage calculations to reload data to a new entity.
Enhanced compute engine	Use of a SQL-based cache, managed by the Power BI service, to improve performance. Data is available both in the data lake and in the database cache. Expensive operations such as filtering, sorting, grouping, and joins are performed by the SQL engine rather than the in-memory M mashup engine associated with Power Query Online.
Parallel execution of transformations	Rather than performing data ingestion and transformations in a serial fashion, allows execution in parallel.
Refresh frequency	Increases refresh limits to 48x/day vs. 8x/day in shared capacity.
Automated Machine Learning (AutoML)	Integration with Azure Machine Learning to enable business analysts to train, validate, and invoke machine learning in a Power BI dataflow. Note that the ability to access pre-published Azure ML models , which have already been trained and published, from Power BI dataflows does not require Premium.
Cognitive Services	Integration with Azure Cognitive Services to execute algorithms to further enrich data being prepared.

Since the dataflows concept becomes most useful at scale, Premium capacity allows dataflows to be used to its fullest.

Power BI Report Server

What is Power BI Report Server?

[Power BI Report Server](#) is an on-premises alternative for publishing, sharing, and consuming business intelligence content within your corporate network. Although it is built on top of SQL Server Reporting Services and has a superset of its functionality, Power BI Report Server is a distinct product from SQL Server Reporting Services and is licensed and installed separately.

Why Use Power BI Report Server?

The following are reasons why you may want to consider using Power BI Report Server:

On-Premises Only Solution or a Complement to the Power BI Service

The Power BI Report Server is useful for customers who want to have an additional choice for where to publish certain content. Some organizations require that certain sensitive or proprietary reports remain within the corporate network; in these situations, deploying to Power BI Report Server instead of the Power BI service is a helpful alternative.

- ✓ Pro: Allows customers to have flexibility and choice.
- ✗ Con: If both the Power BI service and Power BI Report Server are deployed, users will have to visit two different portals to access content.
- ✗ Con: If both the Power BI service and Power BI Report Server are deployed, users may be unclear on the feature differences between the two portals.

Power BI Report Distribution Flexibility for Highly Regulated Customers

For highly regulated customers who like using Power BI Desktop, but are not allowed to use the cloud-based Power BI service, Power BI Report Server offers an option for publishing and securing reports.

- ✓ Pro: Makes available an option for customers whose data must remain on-premises for regulatory, legal, or intellectual property reasons.
- ✓ Pro: More granular permissions are available in Power BI Report Server than the Power BI service.
- ✗ Con: If both the Power BI service and Power BI Report Server are deployed, there are two security models to maintain.

Build on Existing Investment in SQL Server Reporting Services

There are many thousands of customers who have Reporting Services implementations. Existing content can be migrated to a Power BI Report Server without any loss of functionality for existing Paginated and Mobile reports.

- ✓ Pro: Expand usage of existing system to also include Power BI reports.
- ✗ Con: Additional Power BI Pro licenses may be required in order to publish Power BI reports. Power BI Pro is required for anyone who publishes Power BI reports, whereas there is no such requirement for publishing RDL or Mobile reports. This requirement exists so that licensing requirements for publishing of Power BI content is consistent regardless of destination, in the cloud (Power BI service) or on-premises (Power BI Report Server).

Power BI Report Server Release Cycle

Power BI Report Server releases occur approximately 3 times per year which coincide with a specific Power BI Desktop release that is optimized for the Power BI Report Server. Support for Power BI reports in Power BI Report Server is backward-compatible but not forward-compatible.

Comparing Power BI Report Server Functionality with the Power BI Service

Although Power BI Report Server and the Power BI service are developed by the same team and share much of the same code base, there are substantial differences in the functionality of the two products. The intention for Power BI Report Server is to be a basic report portal for on-premises publishing and rendering.

Feature parity between Power BI Report Server and the Power BI service is not an objective. As a result, you should verify that critical features you intend to use are present in whichever delivery mechanism that you intend to deploy.

Embedding Power BI Content in Applications and Websites

There are several options available for embedding Power BI content into applications and websites:

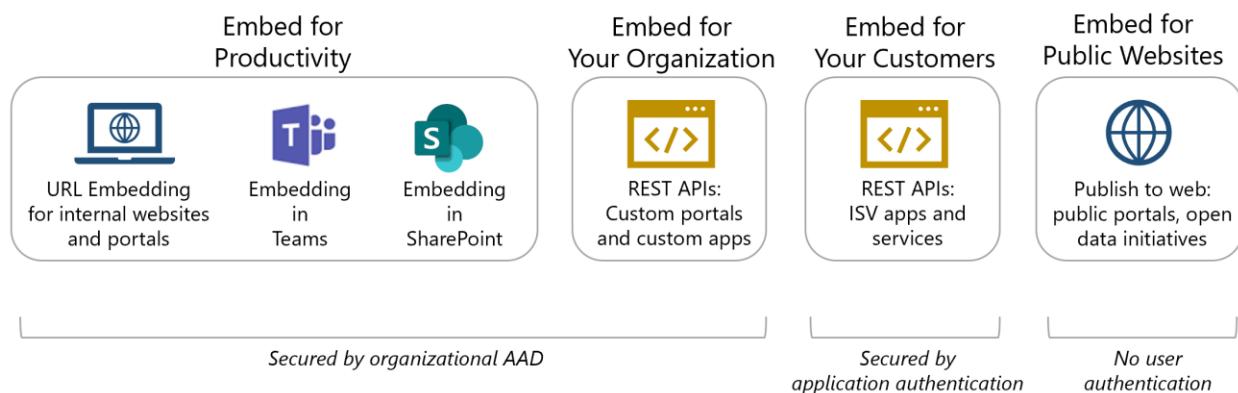


Figure 18 - Graphic of multiple embedding options for Power BI content

The remainder of this section introduces each of these options briefly. The options for embedding can be a little confusing, so we recommend that you work with your Microsoft representative to help determine the best fit.

Embedding for Productivity

URL Embedding for Internal Websites and Portals (No-Code Embed)

This simple URL embedding approach, commonly called no-code embed, is an alternative to the previous two options which require developer skills. An [embed code](#) may be generated for a report in the Power BI service, which can be used to display content to authorized users in an iframe within a portal or website.

Use no-code embedding URLs when:

- The intended location for consuming content is an application or website which can be customized.
- A no-code/low-code embedding solution is desired for a custom application or website which does not have native integration with Power BI.
- There is no need for use of the JavaScript APIs (i.e., the full stack of embedding capabilities is not available).
- Security for content viewing should be controlled by the Power BI service.
- Single sign-on is not a requirement; users being prompted to sign in is considered an acceptable trade-off for the simplicity of deploying this approach.
- All users have a Power BI license (either Pro or Free + use of Premium capacity via an EM or P series SKU).

Embedding in Teams

Power BI reports and paginated reports (which are published to the Power BI service) may be embedded as a [tab within a Teams channel](#). This simple embedding scenario allows members of a channel to discuss the contents of a report using threaded conversations.

Use Teams embedding when:

- A no-code solution is desired for displaying reports in Teams.
- There is no need for use of the JavaScript APIs (i.e., the full stack of embedding capabilities is not available).
- Security for content viewing should be controlled by the Power BI service.
- All users have a Power BI license (either Pro or Free + use of Premium capacity via an EM or P series SKU).
- There is no need for embedding dashboards (reports only are supported in Teams).

Embedding in SharePoint

Power BI reports can be embedded in a SharePoint Online modern page using the [SharePoint Online web part for Power BI](#).

Use SharePoint embedding when:

- A no-code solution is desired for displaying reports in SharePoint Online, using the web part.
- Mention single sign-on
- There is no need for use of the JavaScript APIs (i.e., the full stack of embedding capabilities is not available).
- Security for content viewing should be controlled by the Power BI service.
- All users have a Power BI license (either Pro or Free + use of Premium capacity via an EM or P series SKU).

Embedding For Your Organization

The “embedding for your organization” scenario targets the use of APIs for the purpose of extending the use of the Power BI service. This type of embedding scenario gives users the option of using the Power BI service for viewing content, or via a custom application. This alternative is sometimes called “user owns data” because users authenticate into Power BI.

 Although this option is called “Embedding for Your Organization” it is not limited to individuals in your organization. There are techniques to include external users (via B2B) in this scenario.

Use “Embedding for your organization” when:

- The intended location for consuming content is an application or website which can be customized.
- A developer with coding skills is available.
- The full flexibility of using the JavaScript and REST APIs is required.
- Security for content viewing should be controlled by the Power BI service.
- All users have a Power BI license (either Pro or Free + use of Premium capacity).
- Single sign-on is a requirement.

Embedding For Your Customers (Power BI Embedded)

The “embedding for your customers” scenario targets usage of Power BI Embedded, an Azure service primarily aimed at independent software vendors (ISVs) who intend to include Power BI content within their customer application. This alternative is sometimes called “app owns data” because users authenticate into the application rather than to Power BI.

 Although this scenario is called “Embedding for your Customers” it can also be used with internal users when the use case fits. It is named as such because users do not use their organizational credentials, thus it is primarily targeted to external users. Instead the application manages this aspect, so application users themselves do not need Power BI licenses to access the embedded content.

The Power BI Embedded service works in a very similar way to Power BI Premium in that you purchase capacity. Once workspaces have been assigned to that capacity, content can be consumed by embedding it in a custom-built application. The cost for Power BI Embedded is based on purchase of scalable capacity, rather than licensing for individual users.

Use “Embedding for your customers” when:

- The intended location for consuming content is an application or website which can be customized.
- A developer with coding skills is available.
- The full flexibility of using the JavaScript and REST APIs is required.
- Content consumers do not have a Power BI license, and all user authentication will be handled by the application.
- Power BI Embedded capacity has been purchased in Azure to achieve:
 - Flexibility to scale on-demand or on a schedule
 - Flexibility to pause compute services (and cost) on-demand or on a schedule
 - No billing commitment

Other than embedding scenarios, capacity can be purchased via the Power BI Embedded service in Azure for the purpose of:

- **Initial testing prior to the purchase of Power BI Premium.** Since the Power BI Embedded compute service can be paused when it is not being used, that makes it a great way to test out features which are specific to Premium (such as paginated reports, the XMLA endpoint, or additional dataflows functionality). Note that the use of Power BI Free users to view content is *not* available when capacity has been purchased via an A-series SKU from Power BI Embedded.
- **Use of certain Premium features, but not 24x7.** Since the Power BI Embedded compute service can be paused when it is not being used, that offers a very cost-effective way to use certain features which require dedicated capacity. One such scenario is sending paginated report subscriptions: an automated process could start the Power BI Embedded capacity, then paginated report subscriptions in the Power BI service could

be sent out, then an automated process stops the Power BI Embedded capacity (keep in mind, however, that paginated reports may not be viewed interactively while the capacity is paused, and subscription delivery will fail if capacity is paused during the scheduled time). Not every Premium feature is available via capacity purchased from Power BI Embedded, but most are.

Note that pricing is approximately 20% more per hour for an A4 SKU from Power BI Embedded, versus a P1 SKU from Power BI Premium. Therefore, if you intend to run the capacity 24x7 with little to no pausing (which is available with Power BI Embedded only), Power BI Premium is a more cost-effective option.



Clarifying the Use of Premium with Embedding Scenarios

As just discussed, Power BI supports the embedding of content in two main ways.

Organizational embedding is for users who have Power BI licenses (either Free or Pro) and who need to be signed in with their organizational account to view embedded content.

Whereas the **Power BI Embedded Azure service** allows users without Power BI licenses to view embedded content without needing to sign in with an organizational account; the application that contains the embedded content handles user authentication separately.

The REST APIs may be used by any Pro user. Power BI Premium is not required for organizational embedding, or to use the APIs. However, the ability to allow users with Power BI Free licenses to access content stored in Premium capacities, and the ability to handle larger numbers of concurrent users with more predictable performance, are particularly useful in organizational embedding scenarios. Therefore, Premium is often discussed in conjunction with embedding.

Publish to Web

It is possible to make a report publicly visible to anyone, inside or outside of the organization, with the [Publish To web](#) feature. Since this option is intentionally not secure, we recommend the tenant settings highly limit who is permitted to use this feature. The default setting does ensure that the Power BI administrator must allow use of Publish to web.

Use Publish to web when:

- The content is not confidential or proprietary and may be viewable by anyone on a public-facing website.
- You're building a public website, such as government website to inform citizens of public data, a journalist publishing data as part of an article, or a non-profit organization showing the impact being made.

Section 4. Power BI Licensing and User Management

All Power BI deployments have concerns around performance, security, governance, control, scalability, and monitoring. These are all things typically associated with an enterprise business intelligence rollout. The Power BI licensing model is based on options to simplify choices and meet the varying needs of individuals, small groups, medium companies, and large organizations.

Terminology related to licensing and users includes:

A single organization will typically have a single Power BI **tenant** (though some very large organizations may have multiple tenants). A Power BI tenant is associated with the Azure Active Directory (AAD) tenant.

In order to use the Power BI service, one or more **subscriptions** must be obtained for that tenant. Subscriptions (such as Power BI Free and Power BI Pro) have a start and end date, and a certain number of licenses are available within the subscription.

Licenses are, in turn, assigned to individual **users**. All users of the Power BI service require a Power BI license, either Free or Pro, to be assigned. (Note that capacity-based licenses are available as well from Power BI Premium or Power BI Embedded; however, this conversation focuses on user-based licensing.)

A license is not required to use Power BI Desktop. However, users will be routinely prompted to sign in, so having a license offers the best experience.



Although the Office 365 admin portal can be (optionally) used to create an AAD account, and it is where licenses are purchased, an Office 365 user license is NOT a prerequisite in order to use the Power BI service.

To use the Power BI service, the user requires an:

- a. AAD account, and
- b. Appropriate Power BI license
(Free or Pro)

Most Power BI users will have Office licenses as well, which allows the applications to work well together.

The following graphic is a high-level overview how a subscription has licenses, which are assigned to users:

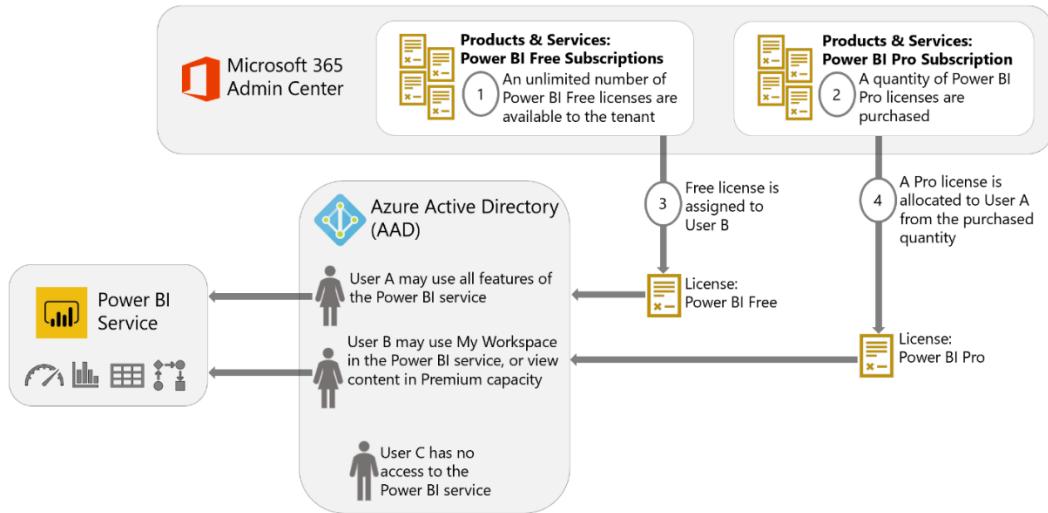


Figure 19 - Depiction of the flow of Power BI user license assignment

Power BI Licensing Summary

There are two types of user-based licensing options:

Power BI Free	Power BI Free is a user-based license which includes the ability to use the Power BI service for personal use. No collaboration or content distribution options are available with the free tier. Power BI Free licenses are not suited to an enterprise deployment, except when used in conjunction with a Power BI Premium license (discussed below).
Power BI Pro	Power BI Pro is a user-based license which includes all Power BI Free features, plus the ability to share and collaborate with colleagues in the Power BI service. The ability to use all features associated with workspaces are available to Pro users.

There are two types of capacity-based licensing options:

Power BI Premium	Power BI Premium is a capacity-based license. Power BI Premium includes many additional benefits and features (discussed in Section 3), as well as the capability to distribute read-only content to report consumers with Power BI Free licenses.
Power BI Embedded	Power BI Embedded is a capacity-based license, purchased as an Azure service, which provides the ability to render Power BI content in custom business applications. When to consider use of Power BI Embedded is covered in Section 3 .

Finally, there is one server-based option:

Power BI Report Server	Power BI Report Server is a reporting portal which supports the delivery of paginated reports alongside Power BI reports, Excel reports, and mobile reports. It is a basic reporting portal which offers an alternative to the Power BI service for publishing and consuming reports. When to consider use of Power BI Report Server is covered in Section 3 .
------------------------	--

The above licensing options can be mixed and matched within an organization, based on the workload for individual users and groups.

The following matrix is a high-level summary of the capabilities for each license type:

Capabilities available per license type:

License type:	Power BI Service	Embedding in a Custom Application	Power BI Report Server
Power BI Free	Use of the Power BI service for personal use (limited to My Workspace unless used along with Power BI Premium) for named users		
Power BI Pro	Use of the Power BI service (incl. collaboration, sharing, and distribution capabilities) for named users	Ability to distribute Power BI content embedded in a custom application to users who have a Pro license	
Power BI Premium	Capacity resources for isolated workloads + additional enterprise features + the ability to distribute read-only content in Power BI apps & workspaces to named users with a Power BI Free license	Capacity resources for distributing Power BI content embedded in a custom application to named users with a Pro or Free license (can also optionally be used for embedding for customers without a Power BI license)	Use of Power BI Report Server as an alternative for content distribution to any type of user, with no Power BI user authentication (requires at least one Pro license for publishing)
Power BI Embedded (Azure Service)	Capacity resources for testing capacity level needs or Premium features	Capacity resources for displaying Power BI content embedded in a custom application to for users without a Power BI license	
SQL Server Enterprise + Software Assurance			Use of Power BI Report Server as an alternative for content distribution to any type of user, with no Power BI user authentication (requires at least one Pro license for publishing)

The remainder of this section will clarify how to purchase licenses and define each license type in a little more detail.

Power BI User Licenses

Power BI Free

Power BI Free makes the Power BI service available for personal use. Power BI Free can be used one of two ways:

- A. **As a standalone implementation for personal use** of the Power BI service. No report distribution, sharing, or collaboration features are available as part of Power BI Free. For example, a Free user cannot access content in a workspace or an app with a Free license (except as noted in item B next). A user in the free tier will only be able to use 'My Workspace' in the Power BI service.
and/or
- B. **In conjunction with Power BI Premium.** When content is deployed to a workspace which has been assigned to Premium capacity, any number of users in the organization, may consume the content without a Pro license (with requisite permissions in place). Content may be viewed within a published Power BI app, within the workspace using the viewer role, or if the content is embedded in a custom application.

As the name suggests, Power BI Free has no subscription cost but does require a subscription with a named user license for system authentication purposes.

Power BI Pro

A Power BI Pro license is required for any form of sharing, collaboration, or content distribution. Put another way: both the person doing the sharing of content, as well as the person receiving the shared content, require a Pro license (except for Premium read-only users).



If a Pro user adds a Free user to a workspace, an app, or shares a report/dashboard with a Free user, there is no immediate validation of license, nor is an error displayed if the recipient does not have a Pro license. When the Free user attempts to view the content, the Free user will be prompted to upgrade to a Pro trial at that time:

Upgrade to Power BI Pro



This feature is only available to users with a Power BI Pro license. When you upgrade, you get access to collaborate with others and distribute content. Upgrade today or try it free for 60 days. [Learn more](#)

[Try Pro for free](#) [Upgrade account](#) [Cancel](#)

Figure 20 - Message displayed in the Power BI service to prompt user to begin a Power BI Pro trial

Note that the "Try Pro for free" button is not displayed if the custom link for licensing requests has been set in the tenant settings. This is discussed later in this section.

Organizational Purchasing of Power BI User Licenses

Power BI user licenses are purchased in the Microsoft 365 admin center. The purchase must be made by someone who is a member of the 'Global administrator' or the 'Billing administrator' role in Microsoft 365.

Within the [Microsoft 365 admin center](#), you may purchase Power BI user licenses one of these ways under the Billing > Purchase Services area:

Power BI Pro license

This is a standalone license which can be purchased for one user, or in bulk (for example, purchasing a quantity of 10 Pro licenses).



Figure 21 - Block displayed in M365 admin center to purchase Power BI Pro licenses

Office 365 or Microsoft 365 enterprise license

The E5 licenses represent the full suite of services, which includes Power BI Pro in the bundle of services.



Figure 22 - Blocks displayed in M365 admin center to purchase E5 licenses

Self-Service Purchasing of Power BI User Licenses

In larger and decentralized organizations, procurement of a Power BI Pro license is not always as easy to get done as it may seem. Microsoft is now allowing more flexibility for how individual licenses for the Power Platform suite of products may be purchased. The introduction of [self-service purchasing](#) aligns with how decentralized business units may elect to purchase any other software-as-a-service (SaaS) platform. Note that this process is only applicable to commercial cloud customers, and not the national clouds or government customers.

The following graphic is a high-level depiction of how self-service purchasing of a Pro license works:

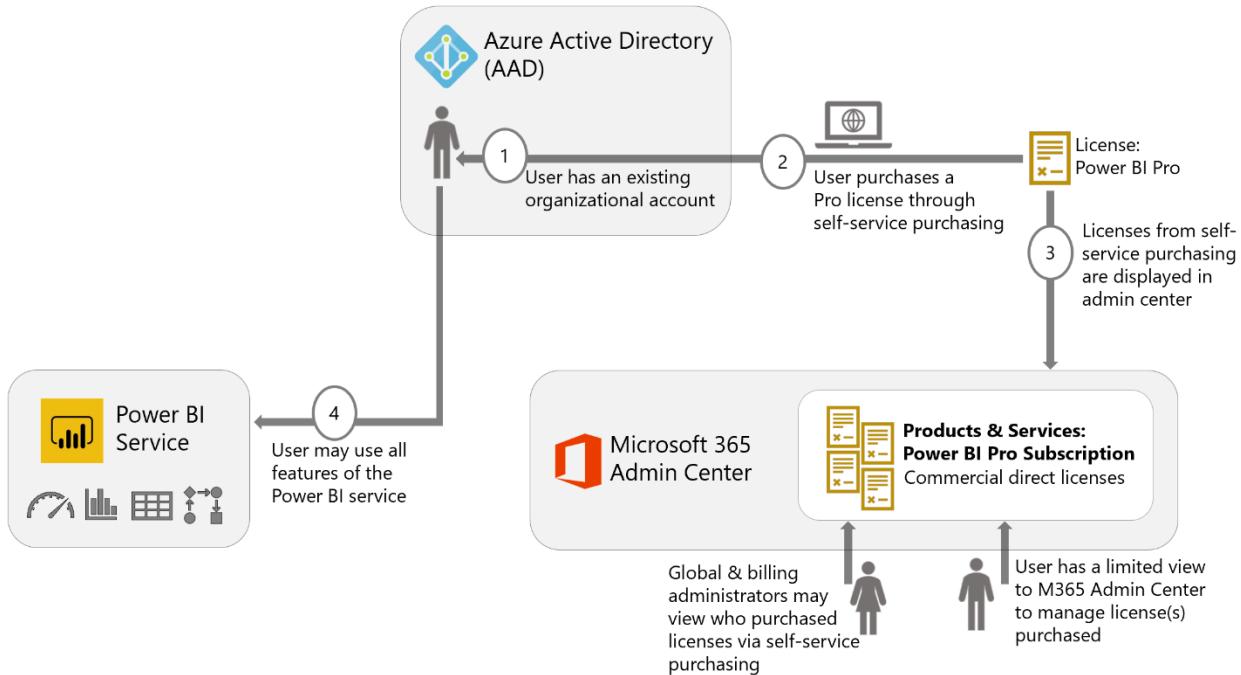


Figure 23 - Depiction of the flow for self-service purchasing of a Power BI Pro license

Users who acquire Power BI Pro licenses through self-service purchasing do so through "buy now" or "Get started with Power BI Pro" links displayed on the Power BI website and on documentation pages:

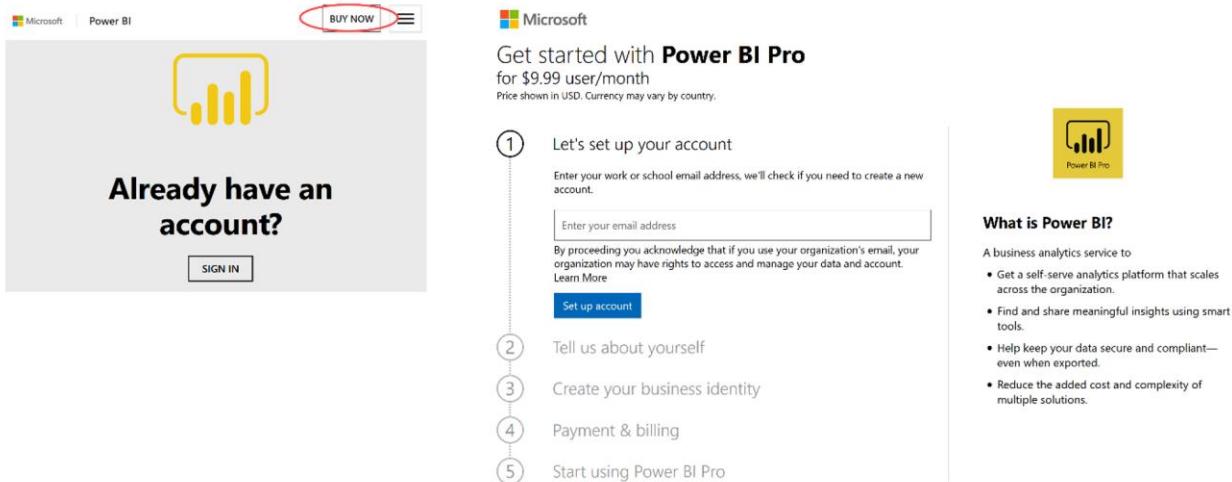


Figure 24 - User experience for beginning the self-service purchasing process for a Power BI Pro license

Once purchased, the user who obtained the Power BI license(s) via self-service purchasing has a limited view within the M365 Admin Center to add licenses, remove licenses, assign users, or cancel the self-service subscription.

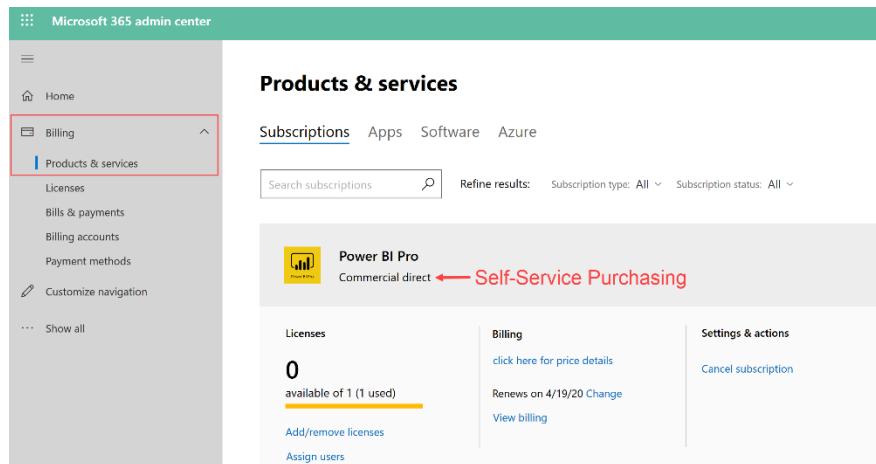


Figure 25 - Limited view in M365 Admin Center for person who obtained Power BI Pro license(s) via self-service purchasing

The global administrator also has full visibility in the M365 Admin Center to how many self-service purchasing licenses have been acquired, and by whom. The global administrator's view is the same as the previous screenshot, minus the links to make direct changes to licenses and users.

Self-service licenses for Power BI Pro are always a monthly commitment. Other than the credit card that is being charged for certain licenses, everything else functions the same for a license that is purchased. The same Power BI tenant is used to publish content; the same security model applies; all user auditing mechanisms are in place.

Disabling Self-Service Purchasing:

The ability for users to make self-service purchases with a credit card is enabled by default. There are some organizations who have existing procurement processes in place to meet compliance, regulatory, security, and governance needs. These organizations should consider disabling self-service purchasing to ensure that all Power BI Pro licenses are approved and managed according to defined processes. Disabling self-service purchasing may also be appealing for companies who have requirements for new Power BI Pro users, such as mandatory training or user acknowledgement of data protection policies.



Policy-Based Management

PowerShell can be used to set a policy to [disable self-service purchasing](#). The MSCommerce module will need to be installed. Disabling self-service purchasing needs to be done for each individual product in the Power Platform.

In addition to process-oriented concerns, the other main reason for disabling self-service purchasing relates to cost management. Self-service purchasing is always at retail pricing. For this reason, organizations may choose to disable self-service purchasing to ensure that all Power BI Pro licenses fall under the enterprise agreement in order to take advantage of negotiated/discounted licensing rates. Alternatively, a global administrator or user management administrator may routinely audit who has made self-service purchases and plan to terminate those licenses in order to re-assign licenses under the organizational agreement.

Additional FAQs about self-service purchasing of licenses can be found [here](#).

Ways to Assign Power BI User Licenses

Power BI Pro licenses can be [assigned to users](#):

- In the Microsoft 365 admin center
- Within the Azure portal in Azure Active Directory
- Via PowerShell

The assignment of licenses can be customized with the help of using a custom licensing requests page which is defined in the tenant settings.

In large organizations, handling requests for user licenses can involve a lot of administrative effort. Following are two ways to make that process more streamlined.

Group-Based Licensing Assignment

One option to improve efficiency is to use [group-based licensing](#) in which a license is automatically assigned to each user based on membership in an AAD group:

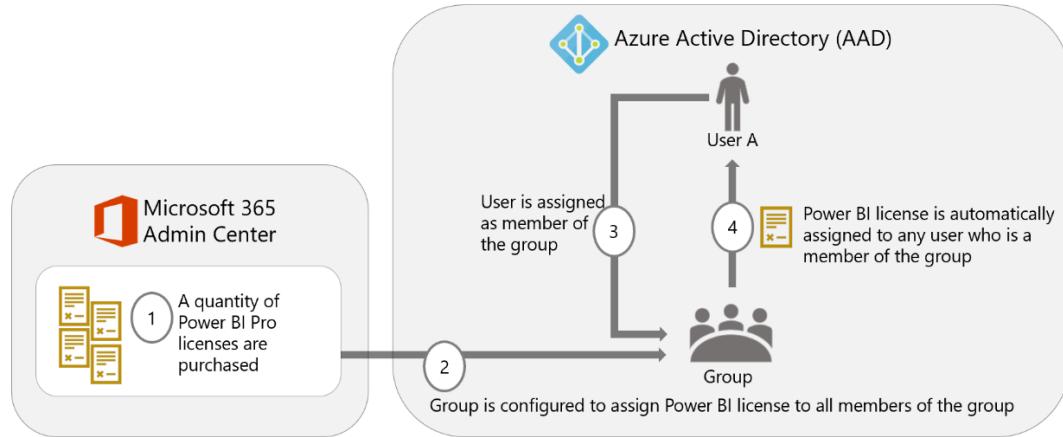


Figure 26 - Depiction of the workflow for using an AAD group for assignment of Power BI licenses

A group called "Power BI content authors" or "Power BI Pro users" works very well for this purpose of associating licenses and users in an efficient manner.

Workflow-Based Licensing Assignment

Group-based licensing can be particularly effective for assignment of Power BI Free licenses. For assigning Power BI Pro licenses which incur a subscription cost, additional workflow or approvals may be required. One way this could be accomplished is via the use of Power Platform tools:

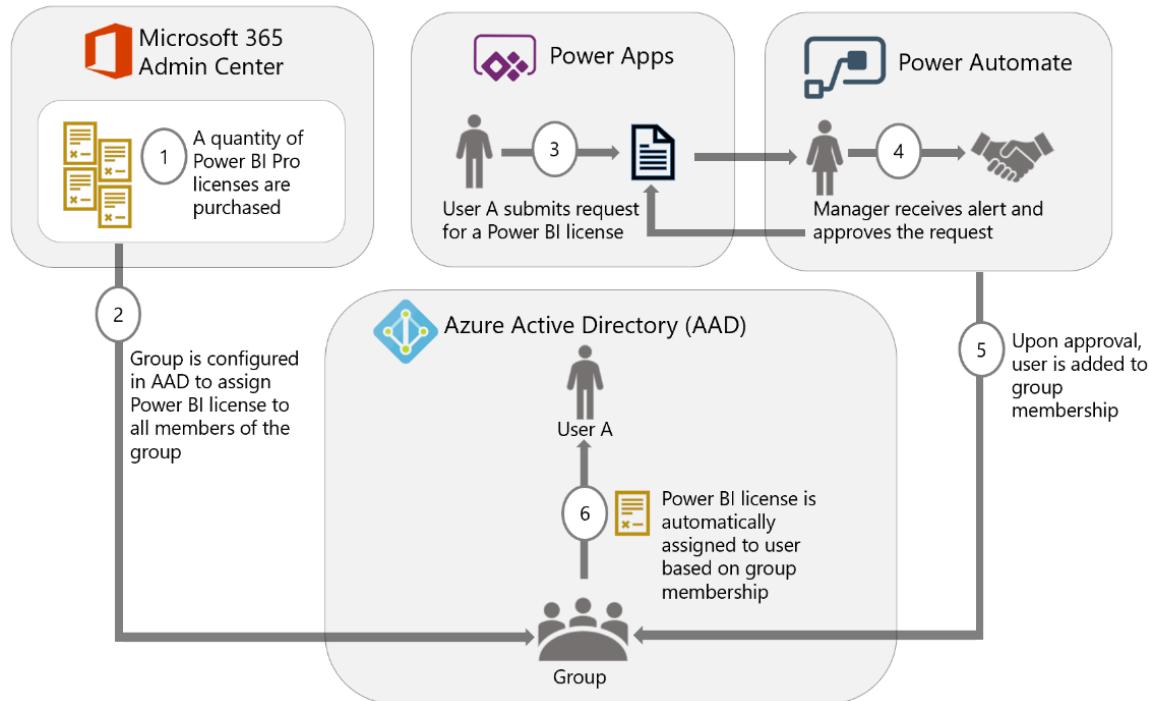


Figure 27 - Depiction of workflow for using an automated process to approve the purchase of a Power BI Pro license

Customizing the Power BI Signup Process

A custom licensing request page can guide users correctly through the license assignment process as defined by your organization. This can enable scenarios such as:

- Requiring acknowledgement of data privacy and data handling policies prior to being assigned a Power BI license.
- Directing users to training offerings as part of the Pro license assignment process.
- Integrating with an existing license management platform.
- Implementing an approval process for granting a Pro license.
- Determining how costs will be allocated, and/or how chargebacks will be handled.

Within the tenant settings in the Power BI admin portal, you may specify a custom URL for handling licensing requests:

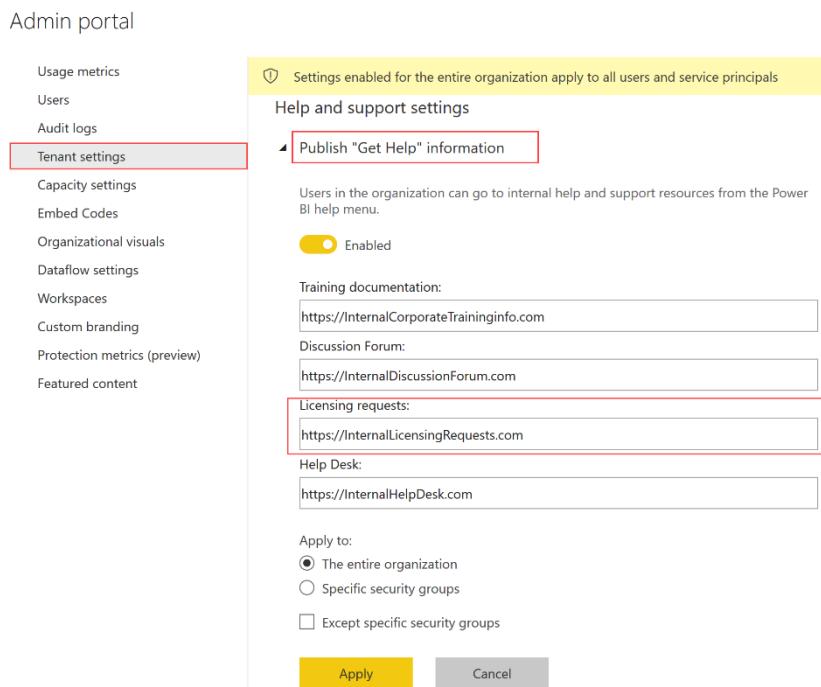


Figure 28 - Tenant setting in the Power BI admin portal for using a custom licensing requests URL

The custom URL is invoked when:

- A user with a Free license is presented with a message to upgrade. The "Upgrade account" button is redirected to the custom URL.

Upgrade to Power BI Pro

This feature is only available to users with a Power BI Pro license. Upgrade today to get access to collaborate with others and distribute content. [Learn more](#)

 **Upgrade account** Cancel

Figure 29 - Message displayed to prompt user to upgrade to a Pro license

or

- b. A user with a Free license visits their "Manage personal storage" page. The "Upgrade account" button is redirected to the custom URL.



Figure 30 - Message displayed to prompt user to upgrade to a Pro license

In addition to directing users to a custom page, users also no longer see the "Try Pro for free button" in the above two scenarios. This means the organizational process for requesting a Free license will need to be addressed, in addition to assignment of Pro licenses.

Managing Power BI User Trials and License Assignments

Above we discussed scenarios for a global administrator to assign a Pro license for a user. It is also very common for a user to get started with Power BI Pro using a 60-day free trial.

 When a Pro user distributes or shares content with a colleague, if the recipient does not have a Pro license (or if the content doesn't reside in a Premium capacity), then the recipient is prompted to start a Pro trial in order to view the content.

Automating the Conversion of Trial Accounts to Pro User Licenses

For a user to continue using Power BI Pro capabilities after the trial period has concluded, a valid user license is required. The objective is to avoid an interruption of service for the trial user (and colleagues as well – for instance, if the trial user has shared content and/or owns refresh schedules. Therefore, it is helpful if Power BI administrators are aware of who has signed up for a trial but does not yet have a license assigned. To avoid this being a time-intensive manual process on the part of the administrator, PowerShell can be used:

- Search the Power BI activity log for activities with a description of "OptInForProTrial." This can be done with the PowerShell cmdlet [Get-PowerBIActivityEvent](#).
- Verify who does and does not have Pro licenses with the [Get-AzureADUserLicenseDetail](#) cmdlet. The service plan ID to locate Power BI Pro licenses is "70d33638-9c74-4d01-bfd3-562de28bd4ba" as documented [here](#).
- Assign Pro license to users who signed up for a trial per the audit log. This can be done with the [Set-MsolUserLicense](#) cmdlet.

You may also wish to also obtain additional information about the user beyond just the e-mail address that appears in the activity log, which can be obtained with the [Get-AzureADUser](#) PowerShell cmdlet.

Handling Auto-Created Office 365 Tenants from Self-Service Signups

If there is not an Azure Active Directory (AAD) tenant in existence associated with an e-mail domain (ex: Contoso.com), then when the first user from Contoso.com signs up for a Power BI account (Free or Pro), an unmanaged AAD directory will be auto-created. This unmanaged tenant is also sometimes referred to as a shadow tenant. If a sanctioned organizational AAD tenant is later needed for this domain, the administrator can [take over management of the existing tenant](#).

Blocking Users from Signing Up for Power BI

Most organizations do permit Power BI Free licenses to be assigned automatically. However, certain organizations deem it necessary to prohibit use of the Power BI service due to data privacy or other concerns, and/or they need to control the assignment of Power BI Free user licenses very closely.

In this situation, you can use PowerShell to [disable automatic license distribution](#) (aka ad hoc subscriptions). If ad hoc subscriptions are disabled, the user will see a message that the IT department has turned off access to Power BI. This affects the Free subscription type only. Note that this process is only applicable to the commercial cloud, not the national clouds such as US Government.

 Disabling ad hoc subscriptions prevents users from being able to sign up for Power BI Free. Global admins & user management admins are still able to assign Free licenses in the M365 Admin Center to specific people.

Additional information about managing Power BI licensing in your organization can be found [here](#).

Other Power BI License Types

Power BI Premium Capacity Licensing

As discussed in [Section 3](#), Power BI Premium is a licensing option designed to address the varying workloads of larger enterprise deployments.

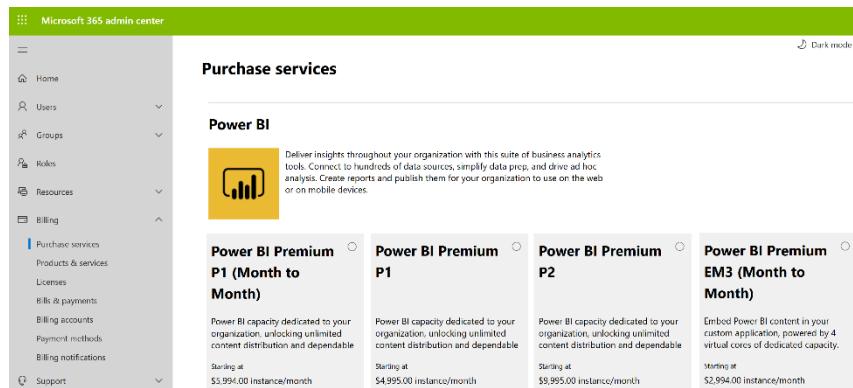
The Pro licensing model, discussed in the previous section, becomes difficult cost-wise for companies when there are a large number of report consumers (who only view reports), and a small number of report authors (who create content for others to consume). In this situation, purchasing Power BI Pro licenses for all authors and consumers can become very expensive. In this situation, purchasing Power BI Premium may be a cost-effective alternative. In addition to

other Premium-only features, all Power BI Free users may consume content (read-only) which is published to workspaces or an app hosted in Premium capacity.

The point where Premium becomes most cost-effective from a licensing perspective depends upon the number of users who only need to view content, whether content is accessed via custom applications only or via the Power BI service portal (or both), and the pricing arrangements your organization has with Microsoft. For more information on how to make this decision, see the discussion of Power BI Premium tiers and SKUs in [Section 3](#), including a list of considerations for selecting a Premium capacity level.

[Power BI Premium](#) P-series SKUs, and certain EM-series SKUs, are purchased via the Microsoft 365 admin center as shown above. Certain EM-series SKUs are available only through volume licensing, thus requiring you to contact your Microsoft representative to purchase.

Power BI Premium generally involves an annual commitment. There is a month-to-month option for the P1 SKU which has a higher price in exchange for a monthly rather than annual commitment.



Power BI Premium SKU	Description	Starting at
Power BI Premium P1 (Month to Month)	Power BI capacity dedicated to your organization, unlocking unlimited content distribution and dependable	\$5,995.00 instance/month
Power BI Premium P1	Power BI capacity dedicated to your organization, unlocking unlimited content distribution and dependable	\$4,995.00 instance/month
Power BI Premium P2	Power BI capacity dedicated to your organization, unlocking unlimited content distribution and dependable	\$8,995.00 instance/month
Power BI Premium EM3 (Month to Month)	Embed Power BI content in your custom application, powered by 4 virtual cores of dedicated capacity.	\$2,994.00 instance/month

Figure 31 - Blocks displayed in the M365 admin center to purchase Power BI Premium SKUs

[Multi-geo support](#) is a feature of Power BI Premium, wherein different Premium capacity nodes may be provisioned in the data region of your choosing independently of the region where the Power BI tenant is located. From a licensing perspective, a multi-geo implementation may impact cost if it involves purchasing more Premium capacities for the purpose of locating them in different geographic regions.

Power BI Embedded Capacity Licensing

As discussed in [Section 3](#), Power BI Embedded is an Azure service which provides the ability to embed Power BI content in custom business applications.

Power BI Embedded is licensed via A-series SKUs purchased in the [Azure portal](#) (as opposed to the EM and P-series SKUs from Microsoft 365 mentioned above which are associated with Power BI Premium). Billing for Power BI Embedded is computed down to the second, and billed on an hourly basis, with no required pre-commitments.

One very big difference between Power BI Premium and Power BI Embedded is that the Power BI Embedded service can be paused when it is not in use. While the service is paused, no user queries can be served and there are no billing charges. Pausing capacity is particularly helpful for dev/test scenarios, and when the capacity does not need to be running 24x7. The capacity level can be scaled up and down as needed based on demand in order to optimize cost and performance.



Using PBIE for Testing Capacity Levels

Because there is no billing commitment, and the service can be paused, the capacity purchased via Power BI Embedded can be used for testing different capacity levels to determine which level of Premium SKU to purchase. For instance, a Power BI Embedded A4 SKU is equivalent to a Power BI Premium P1 SKU.

When selecting a location, try to select the same data center as other Azure services if possible, in order to improve performance and reduce data egress charges. You may also want to consider co-locating the Power BI Embedded service in the same region as your custom application.

Power BI Report Server Licensing

As discussed in [Section 3](#), Power BI Report Server is a reporting portal which offers an alternative to the Power BI service for publishing and consuming reports.

Customers may acquire a license to use Power BI Report Server in one of two ways:

1. **Power BI Premium subscription.** With the purchase of a Power BI Premium P-series capacity node, Power BI Report Server may be installed on an on-premises server with the same number of cores as the capacity node's v-cores. With this method, customers can implement a hybrid approach wherein content may be published to either the cloud-based Power BI service, and/or to Power BI Report Server. Note that this does require a P-series SKU; the other capacity-based SKUs (EM series and A series) do not offer this benefit.
2. **SQL Server Enterprise Edition Core license with active Software Assurance (SA).** Customers who have licensed SQL Server Enterprise with Software Assurance may elect to install or migrate to Power BI Report Server (PBIRS). **Azure Hybrid Benefit (AHB)**, a benefit of Software Assurance, allows eligible customers the flexibility to install Power BI Report Server to on-premises infrastructure, or to an Azure virtual machine. This allows the customer to install Power BI Report Server on a server with the maximum licensed number of cores (or virtual cores for an Azure VM), with a minimum of four core licenses per operating system environment.

We recommend consulting your volume license agreement, and speaking with your Microsoft representative, for specific details about licensing.

The following is a summary of the available licensing combinations permitted for Power BI Report Server:

Method of purchase:	May deploy to:		
	On-Premises Data Center	Azure Virtual Machine (IaaS)	Azure SQL Database vCore Tiers (PaaS)
Power BI Premium (P-series)	✓	Not available	Not available
SQL Server Enterprise + Software Assurance (with Azure Hybrid Benefit)	✓	✓	Not available
Azure Pay as You Go	Not available	Not available	Not available
SQL Server Developer License	Development/ non-production use	Development/ non-production use	Not available

The use of Power BI Report Server does not require a separate SQL Server license unless you intend to use other components such as the SQL Server database engine or SQL Server Integration Services. A limited-use rights database engine comes with Power BI Report Server for hosting the Report Server catalog and temporary databases.

At least one Power BI Pro user license is required for publishing content to Power BI Report Server.



Do you need hybrid flexibility?

When purchased via Power BI Premium, customers have the most flexibility: using either the Power BI service, or Power BI Report Server, is permitted for paginated reports. However, when purchased from SQL Server Enterprise licensing, only Power BI Report Server capabilities are available.

Section 5. Power BI Source Data Considerations

All the data presented in reports and dashboards originates somewhere. The nature of the data sources can have a significant impact on a Power BI project. In this section, we look at how different types of data sources can affect the amount of time needed to develop reports, the reliability of the system, and report refresh duration and reliability.

Identifying Various Types of Data Sources

It is likely that, when you speak to the various people in the organization who create reports, you'll be surprised at the number and diversity of the data sources currently being used. You will likely encounter what is known as "shadow IT" as part of this exercise: systems and data sources that are not under the control of the official IT department and, in many cases, not even known to IT. These can range from Excel spreadsheets and Access databases, to accountancy systems running on a local PC, or cloud-based services.



One mistake often made by Power BI projects run by IT departments is to ignore these "shadow IT" systems and data sources, or to try to force their users to use officially sanctioned data sources instead. Doing so increases the risk of alienating Power BI content authors and potentially even risking project failure. We recommend planning for the following:

1. Determine which data sources will be available in a centralized way vs. departmental or personal data sources.
2. Encourage self-service authors to publish all their content to the Power BI service. This allows Power BI administrators to gain visibility into what data sources are being used, which is a positive aspect from a governance perspective.
3. Establish a process for gateway connections to on-premises sources. This allows data owners and gateway administrators to have a clear understanding of how data is accessed and distributed.

One of the advantages of Power BI as a self-service BI platform is that it makes it easy for users to build reports from many, many types of data sources. Since users are already likely to be well aware of the problems associated with their current reporting methods (such as the amount of time it takes them to prepare reports, and the frequency with which these reports break or show incorrect figures), focus instead on how Power BI can solve some of these issues. Once the user base has begun using Power BI (in particular, the Power BI service), the Power BI administrator will have greater visibility about how reports are used and by whom, and it will be much easier to work with users to migrate their reports to use sanctioned and/or centralized sources of data when appropriate.

The inventory of data sources created as a result of this exercise is useful not only when planning a Power BI deployment; it will also be useful when that project goes into production. Having an enterprise-wide catalog of all of the data sources used for reporting and analysis and

making that list available to everyone using Power BI will help people discover, gain access to, and understand data sources they may not have known existed. In turn, this increases the value of the reports created by subject matter experts. Consider using a data catalog for inventorying data assets in your organization.

Data Source Considerations

While Power BI can connect to many different types of data sources, some types of data sources should be preferred over others. Even if Power BI can connect to all the data sources in your inventory it is important to assess how well Power BI works with these data sources. Some things to consider include:

Provide data to self-service authors from a centralized data warehouse	To minimize the amount of work that self-service authors need to take on, we recommend investing in a centralized data warehouse whenever possible. This may be a traditional data warehouse, or it may be exposed to users as a semantic model from Analysis Services or a Power BI shared dataset. It may even be dataflows in Power BI. Regardless of the technical implementation, the objective is to improve data consistency and accuracy for commonly used organizational data. Then, self-service authors can focus their efforts on reporting, visuals, and data management only for custom data sources. When self-service authors use a centralized data warehouse, teach them to use Live Connection or DirectQuery whenever possible so as not to duplicate the data.
Minimize use of Excel, csv, and text files as sources when practical	<p>Excel workbooks, csv files, and other text files - where there is the chance that the format can change regularly as a result of human editing or mistakes - often cause the most problems for Power BI. Excel workbooks are frequently used as a data source for Power BI because of the flexibility and the ease with which users can edit the data in them.</p> <p>However, this flexibility also means that there is a greater chance that users will add new columns to tables, rename columns, delete columns, or enter data in the wrong place (for example, entering text in a column that should contain numbers). When unplanned changes happen, it is highly probable that Power BI datasets will fail to refresh or may return incorrect results. The extra time spent dealing with these issues, and the damage they can do to the credibility of the project, should be considered.</p> <p>Using tables or named ranges in Excel to identify different datasets within a single workbook can reduce the frequency with which these issues occur, but these issues can never be completely avoided when working with files. To mitigate these types of issues, be sure to have user documentation readily available related to what types of updates may be made to the file.</p>

<p>Store source files in a central location, accessible by all co-authors of the Power BI solution</p>	<p>If Excel workbooks, csv files, or other text files are used as data sources, it is strongly recommended they be stored in a place where they can be accessed easily by all co-authors. This could be OneDrive for Business, a SharePoint folder, or a network file share. Locations such as Azure Data Lake Storage or Azure Blob Storage are also acceptable.</p> <p>Specifically, avoid referring to a path such as c:\Users\User1\FileName.txt because that path will not be valid for User2, nor will the Power BI service be able to reach that path. Using centralized locations is a very important consideration for data access and refresh operations.</p>
<p>Be aware of API connectivity issues and limits</p>	<p>Many software-as-a-service (SaaS) solutions provide access to data via APIs. In many cases Power BI can easily connect to these APIs to retrieve data. However, it cannot do so in all cases: for example, not all types of authentication are supported by Power BI.</p> <p>Even when Power BI can connect to an API, it may involve writing code in the M language, something which many business users and experienced IT professionals may struggle with. Developing a Power BI custom data connector, and then distributing to your end users, can remove the need to write M code when connecting to an API and make importing data much easier.</p> <p>There may also be limits on how often the APIs can be called and how much data can be retrieved from the service. Downloading large amounts of data from APIs may potentially be slow due to throttling imposed by the service provider. The API owner typically documents limits on their website.</p>
<p>Know how to support SaaS solutions from AppSource and expect further data integration requests</p>	<p>Datasets, reports, and dashboards from several popular SaaS solutions, such as Salesforce and Google Analytics, can be accessed using AppSource. These pre-built solutions make it easy to get started quickly with analyzing data. However, these services use APIs behind the scenes and may suffer from the same restrictions on the frequency and volume of data discussed in the previous point: many SaaS providers place a cap on row limits and/or charge an additional fee for extracting more data. Also, the datasets which are created as part of the SaaS solutions may or may not be easily modified; this can make it difficult to customize the reports and dashboards you need.</p> <p>Expect that users who begin to get value from the SaaS solutions will often request the data to be integrated with other corporate data. This is a very positive development, as it informs the Corporate BI/IT team what users find useful - but may add some new requests to the backlog. We recommend integration of data from independent SaaS be included in a centralized data warehouse. This will avoid duplicate data extractions, enable better historical analysis, and integrate the data with other relevant sources which makes the data fundamentally more valuable.</p>

<p>Minimize the query load on source systems</p>	<p>Querying data from source systems may, in some cases, cause problems for the data source itself. For example, if users frequently connect to line-of-business (LOB) databases and retrieve large amounts of data, or run complex queries (particularly with DirectQuery), this can result in the LOB application becoming slow or unresponsive because it is experiencing contention or locking conflicts.</p> <p>Because it can be challenging for systems to satisfy both transactional and analytical reporting needs, Power BI users may be prevented from accessing certain transactional/OLTP databases at certain times (such as during peak working hours), or even altogether, by the system administrator.</p> <p>Some ways to minimize the load on source systems include:</p> <ol style="list-style-type: none"> 1. Use incremental refresh in Power BI for the dataset(s). 2. Use a Power BI dataflow, which will extract the data from the source on a schedule. There can then be many downstream datasets which import the data from the dataflow – without any additional burden on the source system. Dataflows are discussed further in Section 3. 3. Reduce the dataset size by only extracting the amount of data needed, such as the last two years. This can be handled with a parameter to avoid manual changes later.
<p>Expect data refresh operations to take some time</p>	<p>If Power BI reports are deployed to the Power BI service and the data sources they use are on-premises, then bear in mind that moving large amounts of data into Power BI over an internet connection may take some time. This is one of several reasons to minimize the number of Power BI datasets which import data (that is, use existing shared datasets whenever possible rather than creating new datasets).</p> <p>The On-Premises Data Gateway, discussed in detail in Section 7, will need to be installed within the corporate network to allow Power BI to connect to on-premises data sources or cloud data sources which are within a corporate network or behind a firewall. The standard gateway mode is strongly recommended over personal mode for use in enterprise scenarios.</p>
<p>Test data refresh in the Power BI service regularly during development</p>	<p>At the time of writing, there are limitations within the data gateway that prevent certain types of complex M queries (used within the Query Editor to load data into a Power BI dataset) from running successfully, even if these M queries succeed when the dataset is refreshed within Power BI Desktop. Scenarios where this happens include using M functions to make certain types of call to web services.</p> <p>If your solution uses complex M code in their queries, it is strongly recommended to test whether data refresh works once the dataset has been published to the Power BI service. A successful data refresh test in the Power BI service should be verified on a regular basis during development.</p>

Use relational database sources when practical	<p>Relational databases, such as SQL Server, are one of the most suitable data sources for Power BI. Typically, a well-tuned database can handle large data volumes and deliver data very quickly. Power BI generates SQL to request just the data that it needs when a dataset is refreshed (a process called “query folding”) which can make data refresh much faster because actions such as filtering, grouping, and sorting are performed by the underlying database engine. You can use Query Diagnostics to determine if query folding is occurring.</p> <p>In addition, relational databases have many other features that can be used to avoid the kind of problems described above with using files as data sources. Even the fact that the columns in a table can only contain one data type eliminates one of the most common problems associated with using Excel as a data source.</p>
Make accessing data as easy as possible	<p>Power BI can access a wide range of data sources, but not all data sources are as easy to connect to and work with as others. For example, connecting to some web services requires M programming skills that many users do not possess, and the data returned by many web services often requires further transformation before it can be used. The use of – and the creation of, if necessary – Power BI custom data connectors can make it much easier for authors in your company to connect to a wider range of data sources. They usually return data in a Power BI-friendly format too, meaning that less transformation work needs to be done by the dataset author.</p> <p>To aid your self-service dataset authors, consider distributing connection information for commonly used data sources through Power BI data source files (.pbids file format).</p>
Invest time to optimize the data model	<p>There are numerous tactics for optimizing the data model. The optimization guide for Power BI is a good place to start.</p>
Spend time deciding on the best storage mode	<p>As discussed in Section 6, the storage mode is a critical decision for each .pbix file. There are numerous considerations around whether to use import mode, DirectQuery, Live Connection, a composite model, and/or aggregations. This is one of the biggest areas of education/training for self-service authors.</p>

Cleansing, Filtering, Transforming and Integrating Data

It is inevitable that the data used by reports will need some extra work on the part of the dataset author to transform the data into a format suitable for Power BI. Examples of the kind of changes needed include:

- Integrating data from multiple, inconsistent data sources
- Denormalizing a dataset (i.e., “flattening the data” to reduce the number of tables) to make the dataset more suitable for reporting
- Filtering out unnecessary columns or rows
- Handling dirty data, incorrect data, or errors



Make good decisions about where the data cleansing work is performed

In some cases, the amount of effort required for data cleansing might be trivial. In other cases, it might represent as much as 80% or 90% of the effort involved to ultimately deliver a report. If this is the case, consider where and how this work is done. The greater the effort and complexity within the Query Editor, the greater justification for the use of dataflows to centralize this logic in one place.

Power BI has some very powerful functionality for cleansing, filtering, transforming and integrating data in its Query Editor. Therefore, this is not a question of whether Power BI is able to do the job. Instead, ask:

- Do dataset authors want to spend so much time and effort preparing data?
- Will cleansing, transformation, and integration of data be done consistently across all datasets?
- Should cleansing and transformation be handled closer to the source so that it's reusable for more purposes?

Creating a centralized data repository to store data that has already been cleansed is one way of reducing the amount of data preparation work that end users need to do in Power BI. If that repository is a relational database, such as SQL Server or Azure SQL, then it can also solve some of the problems described earlier in this section to do with loading data from Excel files, web services, and line of business systems. Analysis Services can also be used for this purpose.



This is work that is best undertaken by someone with strong data management skills. This is a good example of where the IT department can effectively collaborate with the rest of the business in a self-service BI project.

A centralized repository or reporting database is often referred to as data warehouse, although this has become an overloaded term in that there are competing methodologies describing how a data warehouse should be designed – so be sure to determine how the term is used within your organization. Whatever name you use, there are obvious benefits in creating this kind of centralized reporting repository, however simple or sophisticated it is. When building a data warehouse, avoid challenges others have faced by focusing on:

- Fast time to value
- Accepting a smaller initial release that grows over time
- Responding to user demand and changes in requirements

Centralizing Data Preparation Logic with Dataflows

Power BI dataflows, as discussed in [Section 3](#), make it much easier to create a simple, centralized data repository like the one described above, albeit one that is not as sophisticated as a full-fledged data warehouse.

[Dataflows](#) are available for all users with a Power BI Pro license, with additional features available for Power BI Premium customers.

There are a few ways that dataflows improve interacting with source systems:

1. If you need to **query the source system as few times as possible**. With dataflows, you extract data from the source into the dataflow – then all downstream datasets use data from the dataflow without returning to the original source system. This can significantly reduce contention on busy sources.
2. If you want to **reduce the amount of data preparation work that is being duplicated across many datasets**. The work of creating a dataflow is done in Power Query Online, rather than in an individual .pbix file. This not only makes the output reusable; it also provides one place to maintain complex data preparation logic.
3. If you need a **single storage location for integration of data from multiple sources**.

Keep in mind that usage of dataflows does not diminish the need for a dataset. A dataflow removes the data preparation work from the Query Editor inside of the .pbix file. A dataset is still required for forming relationships, creating measures, defining row-level security, and so forth. Querying datasets is faster as well, due to the in-memory data structure.

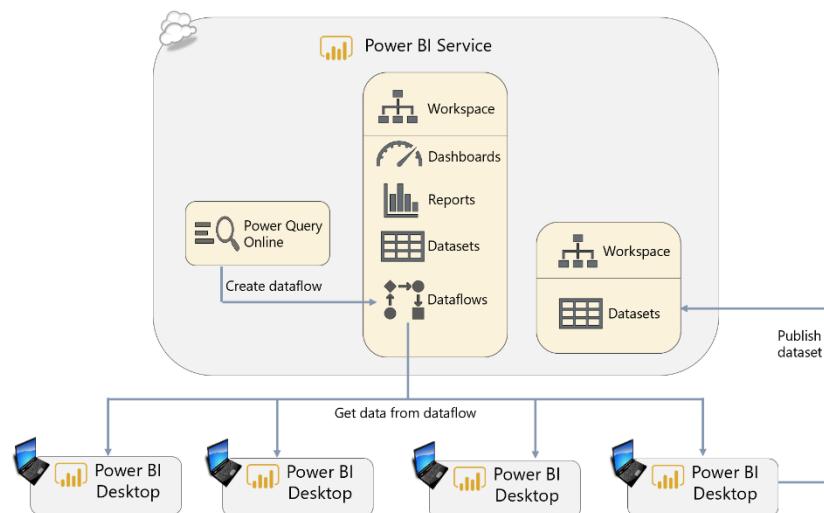


Figure 32 - Depiction of using one dataflow as input for many datasets

When you use a dataflow as your source in Power BI Desktop, the standard Query Editor experience is still available. This offers flexibility to perform additional data cleansing and data integration steps if necessary.

As of this writing, dataflows are supported for import mode only. A future feature to support [DirectQuery operations](#) is on the release plan. Also, not all Power Query connectors are yet available to dataflows; feature parity for Power Query Online is still evolving at the time of this writing.

See [Section 3](#) of this whitepaper for further discussion about the use of dataflows.

Section 6. Power BI Dataset Storage Options

Once you have identified the data sources you intend to use in your Power BI reports, the next major decision is how Power BI should store the data it needs for the goal of producing visuals and reports. In this section, we will discuss the available options and the factors to consider when making this decision.

Dataset Storage Modes

Excluding the storage of dataflows (discussed in the previous section), Power BI takes advantage of several mechanisms for storing and connecting to data:

- Imported data
- Live connection to a Power BI dataset or to Analysis Services
- DirectQuery
- Composite models
- Aggregations
- Near real-time data

Each of the above storage modes have different advantages and disadvantages, and you may choose to use different storage modes for different PBIX solutions. The following is a short summary of the technical details for each storage mode.

Imported Data

The most common way of working with data in Power BI is to connect to your data sources and then import a copy of that data into a dataset stored within Power BI. Imported data is also sometimes referred to as being cached into memory within the dataset.

In this scenario, Power BI uses its high-performance, in-memory columnar database engine to store the data. When reports and dashboards are viewed, Power BI queries its copy of the data stored within the dataset rather than going back to query the original data source(s) to serve user queries.

For imported datasets, a data refresh is required to update the data. The data refresh can be scheduled in the Power BI service, and may be executed in full or incrementally. [Incremental refresh](#) is particularly helpful for reducing the time to load larger datasets and reducing the query load on source systems.



The widest range of Power BI functionality is available with imported datasets.

Below is a depiction of an imported dataset in Power BI:

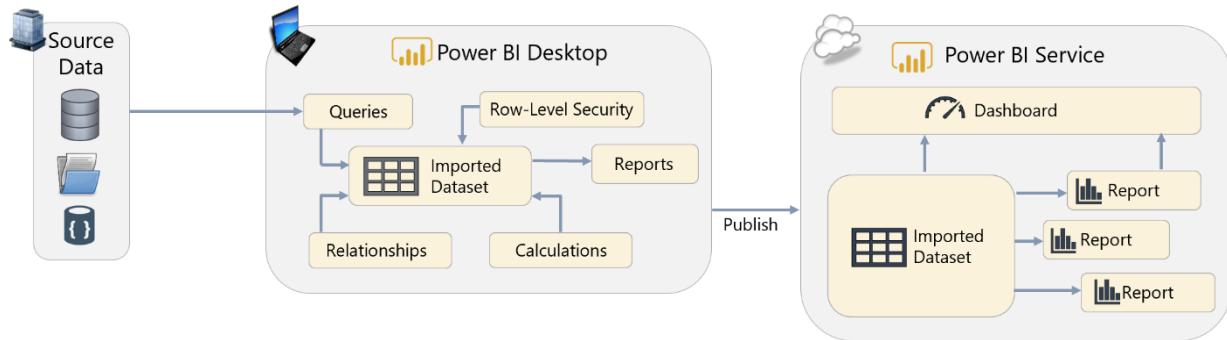


Figure 33 - Depiction of an imported dataset

Refer to [Section 7](#) for more details on when a data gateway is required for data refresh, plus when and where credentials are stored.

Live Connection to a Power BI Dataset or to Analysis Services

The database engine that Power BI uses to store an imported dataset is, essentially, the same engine that runs inside an Analysis Services Tabular model. As a result, Power BI can use SQL Server Analysis Services, or Azure Analysis Services, as a data repository instead of using an imported Power BI dataset. In this situation, DAX queries are issued via a live connection back to the Power BI dataset, or Analysis Services, whenever a report is rendered.

Any version and edition of SQL Server Analysis Services Tabular, and any tier of Azure Analysis Services, can be used in live connection mode.

SQL Server Analysis Services Multidimensional 2012 and 2014 is supported as well, provided the database is SQL Server 2012 SP1 CU4 or greater, and is either Enterprise or Business Intelligence Edition. With Analysis Services 2016 and newer in Multidimensional mode, Standard Edition may also be used.

Below is a depiction of using a Power BI shared dataset in live connection mode:

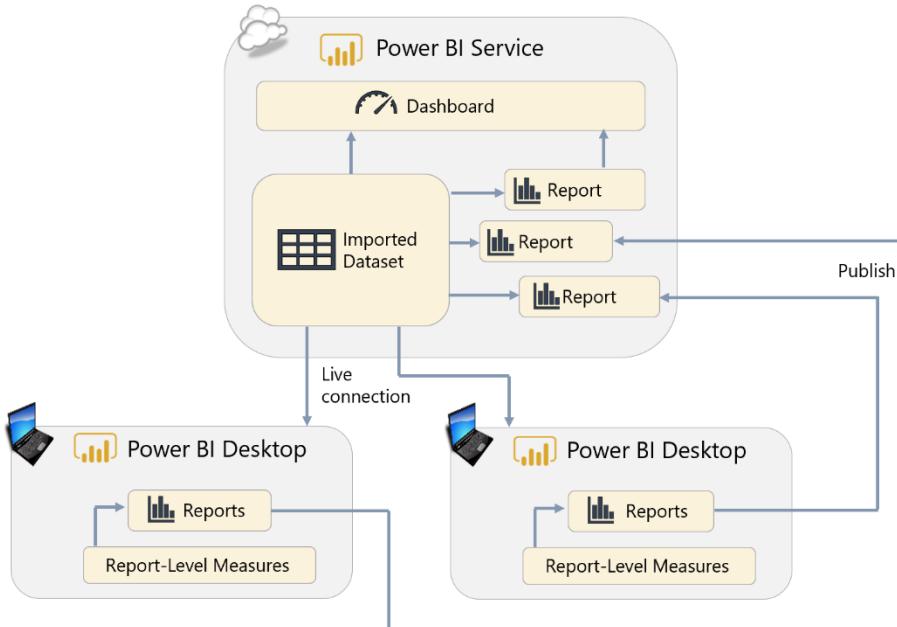


Figure 34 - Depiction of using a Power BI shared dataset in live connection mode

The advantage of using shared datasets in this way is discussed in [Section 9](#) of this whitepaper. Certified and promoted dataset endorsements are often used in conjunction with shared datasets.

Currently, a live connection requires that all the data that a report creator will need be contained in that dataset. Although the creator can create [report-level measures](#), it's not currently possible to mash up multiple data sources in live connection mode (essentially, this would be utilizing a composite model with some tables in live connection + some tables in import mode which is not supported at this time). This capability, however, is on the [roadmap](#).

DirectQuery Mode

Another alternative is to use a Power BI dataset running in DirectQuery mode. In this scenario, Power BI creates an internal dataset, but this dataset does not store any data. Instead, when your Power BI reports and dashboards query the dataset, the dataset in turn passes the queries to the original data source.

Only certain types of data sources, such as relational databases, can be used in DirectQuery mode. Certain custom data connectors also support DirectQuery mode as well. See [here](#) for a full list of data sources that can be used in DirectQuery mode.

When a new data source is in beta or preview, it is not yet supported for production use. Although it may work in Power BI Desktop, functionality in the Power BI service may be delayed to a later date or may not be fully supported.

The following diagram shows a Power BI dataset which retrieves data from an Azure SQL Database using DirectQuery mode:

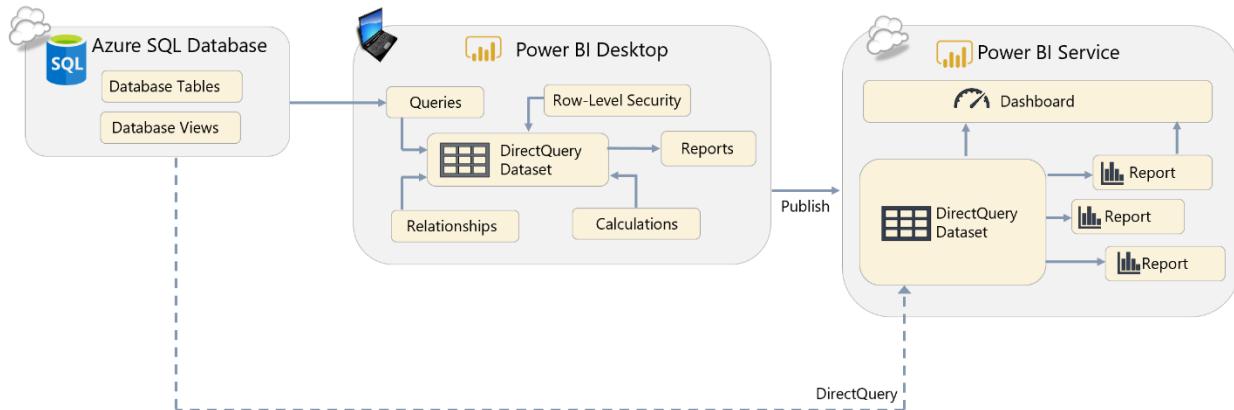


Figure 35 - Depiction of a DirectQuery dataset

Some, but not all, DirectQuery data sources also support [single sign-on \(SSO\)](#) via the data gateway using Kerberos. SSO allows Power BI to pass the identity of the user viewing the report back to the data source so that security defined in the data source can be applied correctly. At the time of this writing, [SSO in Power BI](#) is supported for SQL Server, SAP HANA, SAP BW, Oracle, Teradata, Spark, and Impala. There is also an enhanced connector for connectivity to [Snowflake](#).

A future capability is planned for enhanced DirectQuery [connectivity with Azure Synapse Analytics](#). For larger data sizes, materialized views will be created in Synapse in order to improve query performance and support a larger number of queries (i.e., concurrency). (*Alternatively, [large datasets](#) are also supported in import mode.*)



Difference between DirectQuery and Live Connection

Note that DirectQuery and Live Connections to Analysis Services both involve sending a query to an external data source. However, the key difference is that with Live Connection, Power BI uses the model metadata from Analysis Services and sends native DAX or MDX queries to retrieve data.

Conversely, in DirectQuery mode, there is a dataset created inside Power BI - it just doesn't store any data. With DirectQuery, since the underlying data source does not natively support DAX, Power BI maintains a tabular model that reflects the data source's schema. The connector translates native DAX queries into the query language of the data source, such as T-SQL for SQL Server.

Composite Models

[Composite models](#) are a feature which allow individual tables in a Power BI dataset to be defined as:

- a. An **imported** table, or
- b. A **DirectQuery** table, or
- c. **Dual mode**, in which either import or DirectQuery is used depending on the context of the query submitted

Composite models, also referred to as mixed mode, allow Power BI content creators to gain the benefits of DirectQuery mode only for the tables in a dataset that actually need them, and use import mode for the other tables where that is a most appropriate choice. Setting the [storage mode per table](#) allows a dataset developer to make the best possible choices for balancing **performance** (usually best with import mode) with **data latency** needs (most up-to-date with DirectQuery mode) and **data size** considerations. There are also implications for composite models with respect to security and relationships to be aware of.



Support for composite models with Power BI and Analysis Services (via Live Connection) is a future feature on the [release plan](#).

A few common uses for composite models include:

- Create a custom grouping in a Power BI which does not exist in the source in an import table, while continuing to use the standard taxonomy from the source in a DirectQuery table.
- Use data from a spreadsheet in an import table, such as territory manager names, alongside other data in DirectQuery mode, such as sales data from an operational data store.
- Obtain data from two different DirectQuery sources, such as sales data from an operational data store plus sales targets from a departmental database.

Current limitations with respect to use of composite models can be found [here](#). There are also some limitations for supported sources when [incremental refresh](#) is used alongside composite models.

Aggregations

[Aggregations](#) are a feature which allow for summarized data to be created for the purpose of improving query performance over large-scale datasets.

The use of aggregations allows scenarios where, for example, a large table is stored in DirectQuery mode but aggregated data from that table is stored in import mode. If a user's query can be answered from the aggregated table, Power BI will use it automatically and the query will return results significantly faster. If the query cannot use the aggregated table, the more granular table will be used instead.

Aggregations are stored as another table in the Power BI dataset. Most commonly, aggregations are stored as imported, in-memory, tables. However, an externally defined aggregation table (in DirectQuery mode) may be used as well, which is helpful in certain circumstances. For instance, if you want to take advantage of clustered columnstore indexes in Azure SQL Database or SQL Server. Aggregation tables are not visible in the logical data model which is exposed to end users – only to the dataset developer/owner.

Near Real-Time Data

When near real-time data is requested, it's important to establish what is really meant by real-time. Generally, real-time can mean one of two things:

- a. It could refer to lowering the latency of viewing new data on reports. This type request could be achieved with use of DirectQuery, or by scheduling frequent data refresh, depending on the level of delay that is acceptable to satisfy the requirements.
- b. Alternatively, it could refer to viewing streaming data displayed on reports as the data arrives. This may be able to be achieved with Automatic Page Refresh or Real-Time Streaming Datasets, depending on requirements.

In both cases above, near real-time data typically comes with an expectation that consumers or users of that information are expected to act quickly. Sometimes this may mean the same day; other times it may mean within a few minutes.

Automatic Page Refresh

A newer alternative available in Power BI is [Automatic Page Refresh](#), which allows you to configure how frequently a report page is refreshed. This allows you to achieve near real-time reporting for monitoring of events, sensors, and other time-sensitive data.

Automatic Page Refresh sends queries to the data source at a pre-defined interval. It does this via:

- a. A dataset in DirectQuery mode, or
- b. Individual DirectQuery tables in a composite model

Power BI Desktop has no restriction for setting the refresh interval. Technically, it can be set as frequently as every second, though in practicality the data source may not be able to support this level of constant data writes and reads. The Power BI service, however, does implement some [restrictions](#) in order to ensure the health of the service. In shared capacity, the minimum allowable interval is 30 minutes. Whereas on dedicated Premium capacity, the minimum threshold is configurable by the capacity administrator.

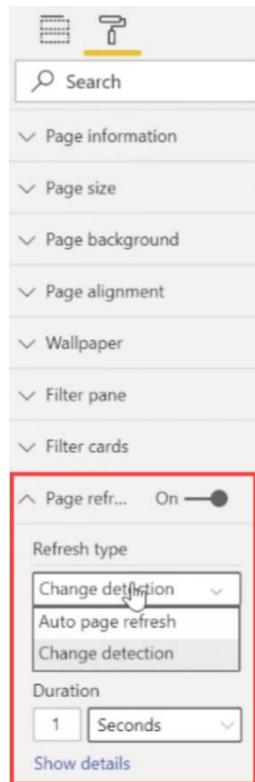


Figure 36 - Page refresh options available in Power BI Desktop

There are two ways to configure the page refresh type:

1. Auto page refresh
2. Change detection

With auto page refresh, all queries related to the report page are sent to the source system at the interval specified. For example, if the report page requires 25 queries to refresh all visuals, all 25 queries will be sent every time the page is refreshed. When data has not changed since the last interval, this results in inefficient use of system resources.

Change detection is a way to reduce the amount of data transmitted. It requires use of a "change detection measure" such as count of rows or maximum date (sometimes called a canary query or a polling query). At the interval specified, one query is initially sent to determine if a change occurred. If the result is affirmative, then all queries are then submitted to perform a complete page refresh.

Use of change detection is recommended when data is arriving on an inconsistent pace. This results in more efficient use of system resources, both for the source system as well as in Power BI.

The ability to use change detection is a Premium feature and must also be enabled in the Power BI service. Note that even though change detection is configured in Power BI Desktop, it is only invoked in the Power BI service when it is running on Premium capacity.

Real-Time Streaming Datasets

There is another alternative in the Power BI service for handling [real-time streaming datasets](#). Real-time datasets are more of a 'push' mode for handling truly streaming data. Currently, real-time streaming datasets have some key limitations related to visuals support, as well as the [number of rows which can be processed](#).

 We recommend you first investigate use of Automatic Page Refresh for displaying near real-time data. Automatic Page Refresh is currently the best option if you can rely on a 'pull' solution using DirectQuery, and it offers the best experience for report authoring.

Choosing a Dataset Storage Mode

The remainder of this section focuses on which storage option to use in which scenario and is structured as a series of questions to ask about your project, your report, and your requirements. In many cases it may not be easy to choose the right storage mode for a report. If there is uncertainty, a technical proof of concept involving creating the same report using different storage modes may be beneficial.

Is Analysis Services Already in Use?

If you already have an investment in Analysis Services (SQL Server or Azure), and you have built cubes or Tabular models which contain the data that you wish to use in your Power BI reports, there is a strong argument to be made in favor of using live connections to those cubes or Tabular models in Power BI. There are two primary reasons why:

- ✓ It will greatly reduce the amount of development work needed for your Power BI reports because the data modeling investment has been made in Analysis Services.
- ✓ Data structure, naming, calculated values, and derived columns shown in your Power BI reports will match the data and calculated values displayed in other tools (such as Excel or paginated reports) that also use the same Analysis Services databases as their data source.

In this scenario, Power BI becomes an extra data visualization tool in your existing corporate BI deployment.

However, there could be several reasons why you might choose not to use the existing Analysis Services cubes or Tabular models with live connections:

- ✗ Analysis Services cubes and Tabular models are usually designed and maintained by a centralized BI department and not by the typical Power BI user. Power BI users who want to have a significant level of control over the way that data is modelled may not wish to route requests for changes through the BI team or IT department. A centralized source such as Analysis Services is one option for use with corporate BI scenarios (*in addition to Power BI datasets + other features such as the XMLA endpoint*). However, utilizing a centralized source may result in a slower pace to introduce changes as well as limitations to what customizations can be made to satisfy individual team needs. When connecting to an Analysis Services Tabular model (though not an Analysis Services Multidimensional cube), users can define their own report-level DAX measures, but no other changes to the way the data is modeled are possible. In some situations, it may be appealing to create your own imported dataset in Power BI. However, there is always risk of introducing discrepancies and differences, so we recommend carefully considering the trade-offs.
- ✗ If you are using a version of Analysis Services Multidimensional that is older than SQL Server 2012 SP1 CU4, then live connections with Power BI are unavailable. If you cannot

upgrade SQL Server, perhaps for cost or other reasons, then other options should be explored.

- ✖ Live connections perform the best with Analysis Services 2016+ Tabular instances. With older instances of Analysis Services Tabular, and especially with Analysis Services Multidimensional instances, you may find that query performance is not as good as using an imported Power BI dataset. We recommend doing a technical proof of concept to test performance early in the project.
- ✖ If you are connecting to Analysis Services through the data gateway (for on-premises AS, or AAS within a virtual network), the latency of querying data could potentially cause unsatisfactory performance. You may want to consider investigating options such as Azure ExpressRoute or Azure Private Link to minimize the impact of latency.
- ✖ With SQL Server Analysis Services Multidimensional instances, certain cube features are not fully supported, or at all, with live connections from Power BI. Examples include translations and calculated members on parent/child hierarchies. A full list can be found [here](#). In addition, users who are members of roles that use cell security will not be able to connect to a cube at all from Power BI.
- ✖ Similarly, when using live connections to a SQL Server Analysis Services Multidimensional instance, certain Power BI features such as Q&A are not available.



Power BI Premium is the next generation of Analysis Services

Power BI Premium is continuing to acquire more and more of the features of Azure Analysis Services to enable supporting both IT-owned and self-service workloads. Indeed, Power BI Premium can be thought of as the next generation of Azure Analysis Services. This future direction should be considered when planning your organization's usage of Azure Analysis Services and Power BI Premium in the long term.

Power BI Premium supports the Analysis Services protocol (XMLA), so the same clients and management applications as Azure Analysis Services and SQL Server Analysis Services can connect. This includes a rich set of third-party data visualization tools and management applications.

There are several capabilities in Azure Analysis Services not yet available in Power BI Premium that might encourage you to choose Azure Analysis Services. These need to be carefully considered with respect to the advantages of Power BI when deciding:

- Backup/restore
 - Object level security
 - Scale-out read replicas alias server names
 - Asynchronous refresh (via REST APIs)
-

How Much Data is Needed to Produce the Reports?

Size limits for individual Power BI datasets stored in import mode differ depending on if you are using shared or Premium capacity. Note: there is no longer a per-user storage limit, as was documented in the past.

Limits in shared capacity:

- An individual Power BI dataset may not exceed 1 GB.
- An individual Excel workbook, containing an Excel data model, may not exceed 250 MB.
- There is a per-workspace storage limit of 10 GB. The exception to this is Power BI Free users are limited to a maximum of 1 GB storage in their My Workspace area.
- There is an aggregate quota of 10 GB * the number of Pro user licenses.

If the amount of (compressed) data will exceed those limits, or will grow to exceed those limits in time, alternatives include:

- Use Power BI Premium to store some or all imported datasets. Premium allows significantly **more overall storage capacity** and allows for larger individual datasets depending on the SKU. More details on dataset size that each Premium SKU support can be found [here](#).
- Power BI Premium enables support for **large models**. At the time of this writing, [large model support](#) does not change the limit for initial upload size: that remains at 10GB max size (depending on the size of your Premium capacity), but the dataset may continue to increase in size beyond that initial 10GB limit via incremental refresh in the Power BI service. Note that when large model support is enabled for a specific dataset, that data is now stored in Azure Premium Files.
- Consider using an Analysis Services Tabular model for data storage and use **live connections to Analysis Services from Power BI**. The amount of data that can be stored in Analysis Services is constrained by the specifications of the server instance.
- Use a **DirectQuery connection back to a relational database**, such as SQL Server or a cloud-based service such as Azure SQL Database, for some or all the tables in the dataset. The pros and cons of using DirectQuery are discussed later in this section. DirectQuery may be the only feasible option for extremely large data volumes.
- Consider using a **composite model** for balancing the needs of DirectQuery and imported data, on a table-by-table basis.
- Consider using **aggregations** for improving query performance for large-scale datasets.



Cost considerations

The above options may result in additional costs, and for some small and medium-sized organizations these costs may be unacceptable. If this is the case, you may need to look at your requirements again and try to reduce the volume of data needed for your reports, for example, reducing the number of years or months of data that you import. Additional techniques are discussed in [Section 8](#) of this whitepaper.

Another performance aspect to consider relates to imported datasets which are used infrequently. Power BI may [evict unused datasets from memory](#) to make resources available for more frequently queried datasets. If evicted, the next time a user views a report, that dataset will need to be loaded back into memory, making report performance slower for that user.

How Frequently Does Data Need to be Refreshed?

The data in the sources used by your Power BI reports will change at different rates. On one hand, the data in a corporate data warehouse may only be updated once a day, whereas sensors connected to machines in a factory may return thousands of readings every minute. As a result, you need be aware how often the data in your Power BI reports needs to be updated to enable timely decision-making.

For streaming data, which arrives at the highest velocity and needs to be analyzed in near real-time, use of Automatic Page Refresh is recommended.



Most datasets do not need to be near real-time and, in fact, can introduce confusion when the data changes too often. However, it is still important to determine how often the data does need to be refreshed because this can be an important factor in choosing between the remaining data storage options.

With respect to data refresh operations, if you are using a Power BI Pro license without Power BI Premium, imported datasets can be refreshed up to eight (8) times per day. Imported datasets stored in Premium capacity may be refreshed up to 48 times a day, which equates to every ½ hour. While datasets stored in Premium can be scheduled more reliably, there are still limitations on the number of scheduled refreshes that can take place in parallel. The limits are related to the number of v-cores available on the Premium SKU that you are using (detailed [here](#)), and this can also influence when scheduled refreshes actually take place and how long the refresh takes.



Data refresh start time

The Power BI service starts scheduled refreshes at the time you specified on a best effort basis. The target is to initiate the refresh within 15 minutes of the scheduled time. However, a delay may occur if there is significant load on the capacity used to execute the refresh.

[Incremental refresh](#) is very important aspect of dataset refresh performance on larger datasets, because it means that only new data, and changed data, needs to be loaded from the data source. When configuring incremental refresh, the dataset author can specify:

- A date range to store data for (for example, to retain data for the last two years in a dataset).
- A date range where data may have changed and where data is loaded from the data source.
- Optionally, a column in the data source that indicates precisely which data has changed.



Incremental refresh availability

When it was initially introduced, the incremental refresh feature was only available for datasets residing in Premium capacity. As the feature has evolved and matured, it has become available for all Pro users as well.

Since you have full administrative control over an Analysis Services instance, you have very granular control over how and when data refresh takes place. Incremental refresh is also currently available in Analysis Services if you use partitioning (which is available for tables in Analysis Services Tabular and measure groups in Analysis Services Multidimensional on premises, but in both cases only in (a) Enterprise or BI Edition with older versions of SQL Server and (b) Standard tier of Azure Analysis Services). Configuring and optimizing data refresh in Analysis Services can become a complex task that Analysis Services developers spend a lot of time working on, but it is also well-documented and well-understood.

If your data resides in a data source which is supported by DirectQuery mode, such as SQL Server, then the fact that Power BI queries this data when a user interacts with a report means there is no need to schedule data refresh: when the data changes in your data source the reports will reflect that change almost immediately. If your team is more comfortable with relational database technology versus Analysis Services, you may consider loading your data into a relational database such as SQL Server that is supported by Direct Query. When using the composite model feature, only the tables where data changes frequently will need to be stored in DirectQuery mode which makes DirectQuery a more compelling option. We recommend doing a technical proof of concept to validate functionality and performance expectations.



How often should data be refreshed?

If you ask your end users how often they would like the data for their reports to be refreshed, they will inevitably say “as often as possible.” If you take them at their word this can make your life as a developer much more difficult than it needs to be. If you do some research into the actual business requirements, you might find that less frequent data refresh is perfectly acceptable. A few questions you might ask include:

- How often will report consumers view the data?
- How quickly are report consumers be expected to act on the data?
- Does refreshed data have to be available first thing in the morning?
- Are there end-of-day processes that would make a data refresh at the close of business helpful?
- Are there international users with different timing needs?

How Complex is the Dataset? Are any Complex Calculations Necessary?

A dataset is composed of a series of tables linked together by relationships, DAX (Data Analysis eXpressions) calculations, and other metadata. The complexity of the dataset, and the number and complexity of any measures used in it, will have a significant impact on the performance and usability of reports.

Tuning the DAX used in measures plus optimizing the modeling of imported datasets can make a significant difference in performance. If you are working with live connections to Analysis Services or DirectQuery you may also be able to tune the performance of those data sources, but there are some general remarks that can be made about complexity and the different options for storing data:

- As complexity of a dataset and calculations increase (including the use of complex row-level security), performance of a DirectQuery dataset will usually degrade faster than the other types of datasets. This is because it becomes increasingly difficult for Power BI to generate efficient SQL to retrieve the data needed by a report. By default, Power BI prevents you from using many DAX functions in your calculations when you are in DirectQuery mode because it cannot generate efficient SQL for them. This behavior can be changed by going to the Options dialog: on the DirectQuery pane check the “Allow unrestricted measures in DirectQuery mode” box to enable the use of more DAX functions inside measures and calculated columns. Be aware that doing this greatly increases the risk of poor report performance.
- If you are using a live connection to Analysis Services 2016 Tabular or newer, the version of the database engine used to store the data in Analysis Services is more or less equivalent to the one used to store an imported dataset in Power BI. However, for older versions of Analysis Services Tabular and for Analysis Services Multidimensional, Power BI generates less efficient DAX queries which will perform worse because these versions do not support the latest additions to the DAX language (referred to as SuperDAX). This

performance difference will be more noticeable the more complex the dataset becomes. Older versions of Analysis Services may not be able to evaluate certain types of calculations as efficiently as the version of the database engine inside Power BI, which does support SuperDAX.

- A live connection to an on-premises Analysis Services database may perform better than an imported dataset depending on the hardware the Analysis Services instance is running on, how the instance has been configured, and how well-optimized the Analysis Services database is. Important factors include network connectivity, the speed of the CPUs, the amount of memory on the server, and, in the case of Analysis Services Multidimensional, disk performance.

How Many Concurrent Users Need to be Supported?

The number of concurrent users who will be using reports and dashboards will also have a significant impact on performance. Power BI imported datasets that are stored in shared capacity (i.e. not in Power BI Premium capacity) are subject to throttling by Microsoft so that the shared hardware is governed well across customers. Therefore, reports and dashboards that use imported datasets stored in shared capacity will be unable to support an extremely large number of concurrent users. Power BI Premium capacity can support a much larger number of concurrent users because it runs on dedicated hardware not shared by other customers, and therefore has fewer restrictions on the usage of that hardware.

To handle many concurrent users, you will need to consider which operations need to be scaled out. Power BI Premium allows for scale-out for both front-end and back-end operations:

- Front-end operations include everything relating to the user experience: the web service, dashboard and document management, access rights management, APIs, uploads, and downloads.
- Back-end operations include query processing, cache management, running R servers, data refresh, natural language processing, real-time feeds, and server-side rendering of reports and images.

Using a live connection to Analysis Services provides scale-out for only a subset of the back-end operations that Power BI Premium allows: only those operations that are concerned with query processing and data refresh. As a result, Power BI Premium is essential for scenarios where you need to support many concurrent users. You may choose to use Power BI Premium in combination with live connections to Analysis Services in some cases to get the best results.

The question of how much capacity to purchase in Power BI Premium, or how to scale your Analysis Services server, is much harder to answer. This will depend on several factors.

Are there Legal or Regulatory Reasons for Storing Data in a Specific Location?

In some industries or countries there may be legal restrictions on what data you can store in the cloud, forcing you to use on-premises data storage options instead. There may also be legal restrictions on which database technologies can be used to store data: for example there may be a corporate security mandate to store certain data in Always Encrypted columns in SQL Server, resulting in the use of DirectQuery datasets in Power BI. If you think that these restrictions might apply to your project, seek legal advice.

A Power BI Premium capacity may be configured to store its data in a specific geographic area (the list of supported geographies may be found [here](#)). This affects the geographic data storage (data at rest) location for only the individual workspaces which are assigned to that specific Premium capacity.

More details on how and where Power BI stores data when different storage modes are used is available in the [Power BI security whitepaper](#). Compliance information and list of certifications for Power BI are available in the [Microsoft Trust Center](#).

What Technical Skills are Present In-House?

An Analysis Services instance, whether in the cloud or on-premises, will need someone to configure it and to monitor it. The overhead of managing Azure Analysis Services will be somewhat less than with an on-premises instance of SQL Server Analysis Services, but it will still require oversight by an administrator. Administering Power BI Premium, in contrast, requires less time and technical skill (though to be clear: capacity administrators do still have to monitor memory levels, eviction, and other health aspects).

The flip side of the previous point is that Analysis Services is more flexible and configurable while Power BI Premium is less flexible but much easier to manage. If you already use Analysis Services in-house and have staff on your team who have experience administering Analysis Services, then it is unlikely that using it with Power BI will present any extra concerns as you have people who can take advantage of the extra opportunities it offers. However, if you do not have existing Analysis Services experience in-house then it may be appealing to choose Power BI Premium in conjunction with imported datasets.

What Budgetary Constraints Need to be Considered?

No project has an unlimited budget, and you will need to understand what options are available for the funds which are available. Equally, you will want to avoid over-specifying and paying for capacity and features that the user base does not need or will not use.

If you are considering using Power BI Premium or Azure Analysis Services, the recommendation is to start using the least expensive tier of either Power BI Premium or Azure Analysis Services, and then run a series of load tests on your reports to determine how well they will perform with

your expected number of users. If you find that performance of the reports under load is unacceptable, then upgrade to the next tier and repeat the process until you achieve acceptable performance, or you reach the limits of your budget. You should also pay close attention to usage once you have gone into production and be prepared to upgrade in the future as more and more reports are deployed to Power BI and reports are used more frequently.

Summary of Dataset Storage Recommendations

The recommendations for options discussed within this section may be summarized as follows:

Consider using **imported datasets** in the Power BI service if:

- ✓ You want the best possible performance, and
- ✓ You want to take advantage of the full feature set of Power BI, and
- ✓ You want the data to change on a predictable cadence (the refresh schedule)

Consider using **live connections to imported datasets** stored in the Power BI service if:

- ✓ You want to reuse an existing shared dataset across numerous reports in order to decouple the dataset from the report, or
- ✓ You want different people, with different skills, to create reports from the dataset(s), and
- ✓ All the data need for report creation resides in the imported dataset, and
- ✓ Your end users do not need to customize the datasets (beyond report-level measures), and
- ✓ You want centralized access management and governance of enterprise data

Consider using **composite models** if:

- ✓ Performance of your reports in DirectQuery mode is a concern, but you still need the benefits of DirectQuery mode, and
- ✓ You only need the benefits of DirectQuery for a subset of tables in your dataset

Consider using **aggregations** if:

- ✓ You are using import mode or DirectQuery mode, and
- ✓ You want to improve query performance for larger tables, or tables with high cardinality attributes

Consider using **live connections to Analysis Services** if:

- ✓ You already have all the data you need in a fully developed Analysis Services model, and
- ✓ Row-level security is required but managing RLS within a lot of individual, various, Power BI datasets would introduce too much management overhead and/or risk of discrepancies, and
- ✓ Your end users do not need to customize the datasets or work with data that is not currently stored in Analysis Services, and
- ✓ You have the resources in-house to support an instance of Analysis Services, and
- ✓ Your end users already use, or wish to use, Analysis Services with other tools such as Reporting Services or Excel, and

- ✓ You require a greater degree of control over processing, monitoring, fault tolerance and other architectural features

Consider **Analysis Services instead of Power BI for dataset development** if:

- ✓ You have a strong preference for developing in Visual Studio vs. Power BI Desktop, or
- ✓ You have a requirement for object-level security, or
- ✓ You want to be able to scale up, down, or pause compute in Azure Analysis Services

Consider using **Power BI Premium with imported datasets** if:

- ✓ Your mix of report developers and read-only consumers means that it is more cost-effective to use Power BI Premium versus Analysis Services, or
- ✓ You need support for large datasets, or
- ✓ You have legal or compliance requirements for isolated hardware, data residency, or encryption key management, or
- ✓ You want the flexibility to use additional features such as paginated reports, full dataflows functionality, or AI capabilities, or
- ✓ You need to create many small or temporary datasets that can be managed outside the IT department in a decentralized manner

Consider using **Power BI Premium**, regardless of whether you are using an imported dataset stored in Premium capacity or you are storing your data in Analysis Services, if:

- ✓ You have many concurrent users and find that the user experience for operations like publishing reports, using the API, and changing settings in the web interface is unacceptably slow with just Pro licenses, or
- ✓ Your mix of report developers and read-only consumers means that it is more cost-effective to use Power BI Premium versus just Power BI Pro licenses, and
- ✓ You want to access existing datasets from other reporting and management tools using the XMLA endpoint

Consider using **Azure Analysis Services over on-premises SQL Server Analysis Services** if:

- ✓ Some or all your data sources already reside in the cloud, and
- ✓ Testing shows that the latency involved with using data from an on-premises instance of Analysis Services through the data gateway in a Power BI report has a noticeable effect on report performance, and
- ✓ You have no legal requirements to store data on-premises, and
- ✓ You do not already have an on-premises instance of Analysis Services (or you want to migrate an existing instance to Azure), and
- ✓ You want to be able to scale compute up and down, or scale out, on demand, or
- ✓ You want to be able to pause compute to reduce cost when users are not utilizing the AAS model, assuming there is not a 24x7 uptime requirement

Consider using **DirectQuery to a relational data source** if:

- ✓ Your data needs to be refreshed more quickly than it can feasibly be loaded into a Power BI imported dataset or an Analysis Services database, or
- ✓ Regulatory concerns mean that your data cannot be stored in a Power BI imported dataset or an Analysis Services database, or
- ✓ Your data volumes are so large that the data you need cannot be loaded into a Power BI imported dataset or an Analysis Services database, and
- ✓ Your dataset is simple, consisting of relatively few tables and relationships, and
- ✓ You are using very few DAX measures or calculated columns in your reports, and
- ✓ You are using a data source such as SQL Server that supports DirectQuery mode

Consider using **automatic page refresh** if:

- ✓ You need data to be displayed as soon as possible after the data is generated

Consider using **real-time streaming datasets** if:

- ✓ You need data to be displayed as soon as possible after the data is generated, and
- ✓ You are using a service such as Azure Stream Analytics or a custom application that can push data through the API to Power BI

Section 7. Power BI Data Refresh and Data Gateway

Data stored in the Power BI service must be refreshed for reports and dashboards to stay relevant. If you are importing data into a Power BI dataset, it is important to ensure that the Power BI service can connect back to the original data source(s) for scheduled refresh to take place. Similarly, if you are using a live connection to Analysis Services or DirectQuery mode, then Power BI must be able to connect to the data source to run queries when reports are rendered.

This section is a brief introduction to the data gateway. The online documentation is divided into two areas:

- [Data gateway content specific to Power BI](#) (*note power-bi in the URL*)
- [Data gateway content applicable to all services](#) (*note data-integration in the URL*)

Since many enterprise data sources reside within the corporate network, and the Power BI service is a cloud service, a data gateway must be used to allow Power BI to connect to those data sources within the corporate network.



Corporate network

Throughout this section, you'll see references to data which resides inside the corporate network. This includes all the following:

- On-premises data sources
- Cloud data sources which reside in an IaaS (infrastructure-as-a-service) virtual machine
- Cloud data which resides within a virtual network (VNet)

Introduction to the On-Premises Data Gateway

The On-Premises Data Gateway is an agent, installed within your corporate network, which securely handles data connectivity and data transfer between the data sources residing within the corporate network and cloud services.

A data gateway is required for the following Power BI activities:

Refreshing imported datasets	Refreshing data which is stored in an in-memory dataset from a source within the corporate network
Refreshing dataflows	Refreshing data which is stored in Azure Data Lake Storage Gen 2 from a source within the corporate network
Using DirectQuery	Passing queries to an original data source, such as SQL Server
Using Live Connection for Analysis Services	Passing queries to an Analysis Services semantic model

See [Section 6](#) of this whitepaper for a thorough discussion of the distinctions between imported data, DirectQuery, and Live Connections for datasets.

In addition to handling data communications, the gateway handles encryption/decryption of credentials for on-premises data sources, as well as compression of data transmissions.

Power BI content published to a Power BI Report Server does not require implementation of a data gateway. Therefore, this section applies exclusively to the Power BI service.



This section refers to other services such as Logic Apps, Power Automate, and Power Apps. Although these services are beginning to introduce some interesting integration points with Power BI, they are not directly related nor required for Power BI.

Gateway Terminology

The On-Premises Data Gateway is frequently also referred to as the **data gateway**.

The On-Premises Data Gateway can be installed in one of two modes:

1. **Standard mode** (formerly known as enterprise mode)
2. **Personal mode**

Any references to On-Premises Data Gateway refer to standard mode. In this whitepaper, we will be focusing on the functionality of standard mode, unless personal mode is specifically noted.

A gateway **cluster** represents one, or more than one, server which appears in the Power BI service as one logical gateway cluster, and acts as a single unit. Even if a gateway cluster only includes one server, Power BI still uses the term cluster.

The gateway in standard mode supports a variety of services beyond just Power BI:

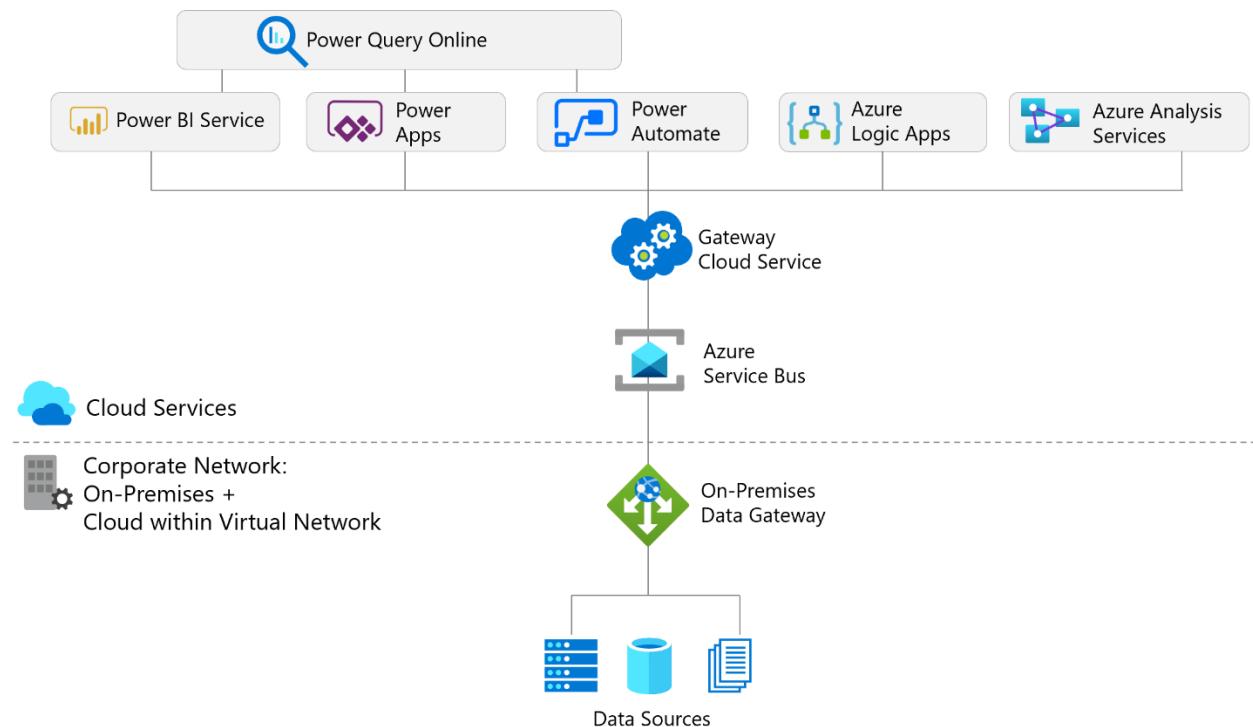


Figure 37 - Depiction of services supported by the data gateway in standard mode

Following is a summary of gateway support per service:

Connectivity for Services	On-Premises Data Gateway (Enterprise Mode)	On-Premises Data Gateway (Personal Mode)
Power BI Service	✓	✓
Power Apps	✓	
Power Automate	✓	
Azure Logic Apps	✓	
Azure Analysis Services	✓	

When sharing a gateway with the other above-mentioned services, each service needs to reside in the same geographic data region in order to be able to use a single gateway.

Both the standard and personal gateway can co-exist on the same machine. However, it is highly recommended that the Power BI gateways are not installed on the same machine where other gateways reside (i.e., for Azure Data Factory or Azure Machine Learning).

Standard Mode

The data gateway, in standard mode, is intended for enterprise-wide deployments. It may be installed on a Windows 64-bit platform.

The following diagram depicts a scheduled data refresh for an imported dataset, which is executed via a 2-node gateway cluster:

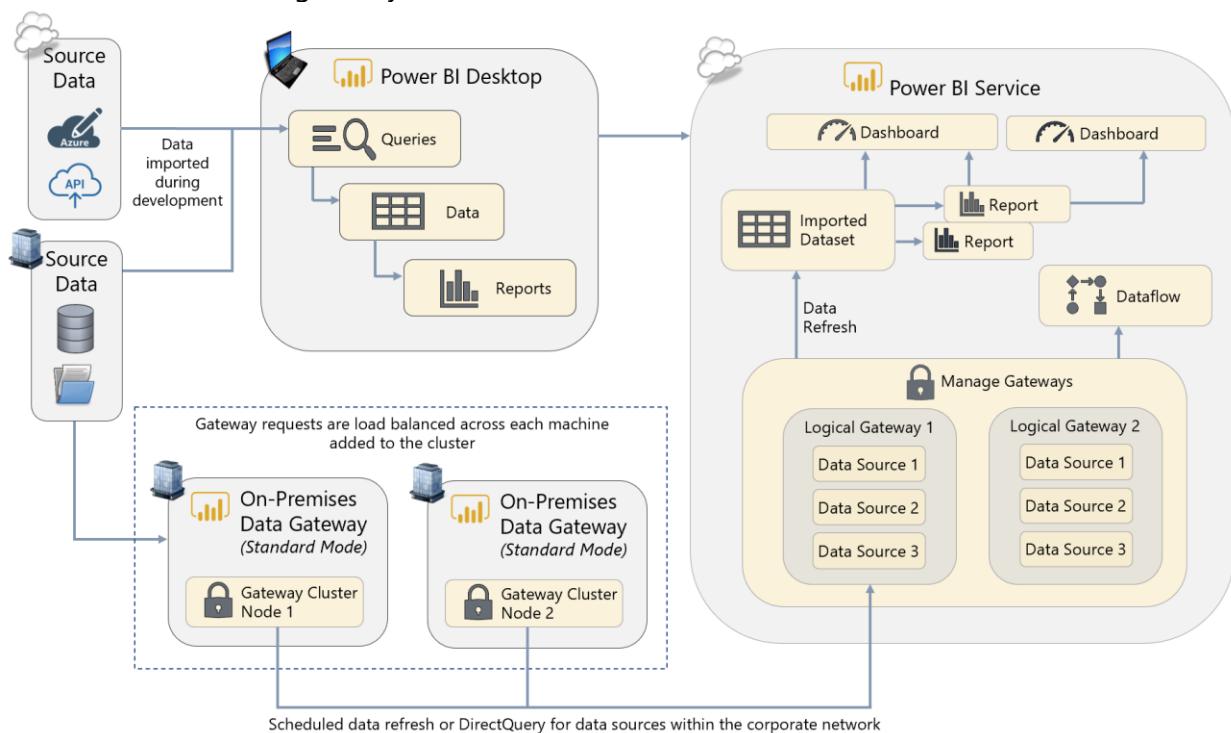


Figure 38 - Depiction of scheduled data refresh with a 2-node gateway cluster in standard mode

Advantages of the standard mode include:

- ✓ The gateway and data source configurations are **reusable for multiple datasets and dataflows, across numerous users**. If a source database is migrated to a new server, the configuration can be changed once in the Power BI service (whereas every individual user who runs personal mode will need to make this change).
- ✓ The gateway is **intended to run on a server**, rather than a user machine, which makes it more reliable for uptime and performance.
- ✓ Standard mode can run in a cluster of multiple machines, allowing both **load balancing** and **high availability**.
- ✓ Standard mode supports **DirectQuery and SSAS live connections** (whereas personal mode only supports dataset refresh).
- ✓ **Dataflow refreshes** are supported on the standard mode (whereas personal mode does not support dataflows).
- ✓ Standard mode supports **paginated report queries** (whereas personal mode does not).
- ✓ The standard mode is **reusable among other applications**, such as Azure Analysis Services, Logic Apps, Power Apps, and Power Automate (whereas personal mode supports Power BI only).
- ✓ **Privacy levels** per data source can be configured for the data gateway in standard mode.

Personal Mode

The data gateway, in personal mode, involves installing the gateway on an individual user's machine for the purpose of scheduled dataset refresh. The personal mode gateway may be installed on a 64-bit Windows PC.

The following diagram depicts a scheduled data refresh for an imported dataset which is executed via a data gateway running in personal mode:

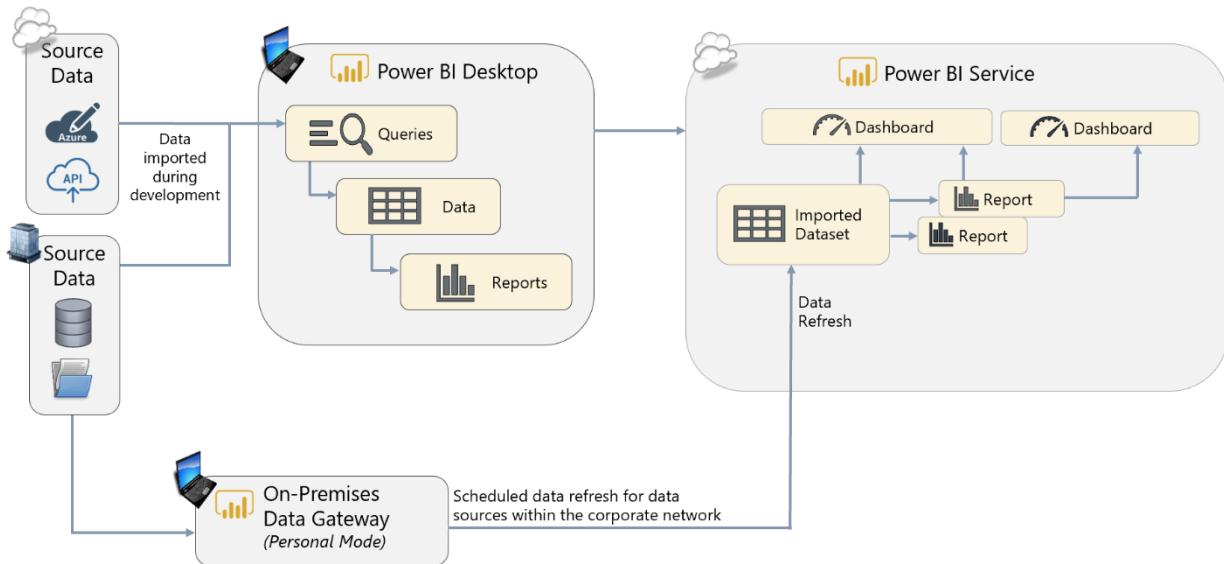


Figure 39 - Depiction of scheduled data refresh with a data gateway in personal mode

Advantages of personal mode include:

- ✓ It is **fast and easy** to set up, making it useful for prototyping and personal projects.
- ✓ It **offers freedom** for business users to get a data refresh accomplished if they do not have permissions to use or configure a data gateway in standard mode, or if they do not know which administrator to contact.
- ✓ New features may arrive for the personal mode gateway first.
- ✓ When using R or Python scripts for data access, data refresh must use the personal gateway. We recommend you use known connectors for enterprise use cases instead whenever possible.
- ✓ Personal mode is capable of ignoring data privacy levels for the purpose of improving data refresh times (though it should be used with caution for sensitive data). See [Section 11](#) for more details on data privacy levels.

 **A personal gateway is associated with one user**
 The data gateway in personal mode may be installed once per Power BI licensed user. Therefore, the same user is not permitted to install the personal mode gateway once on their laptop and again on a server.

Summary of Features Per Mode

Feature	Personal Mode	Standard Mode
Scheduled data refresh for imported datasets	✓	✓
Scheduled data refresh for Excel files (Power Pivot / Excel data model)	✓	✓
Scheduled refresh for dataflows		✓
Paginated report queries		✓
DirectQuery queries		✓
Analysis Services live connection queries		✓
Usable by multiple services: Azure Analysis Services, Power Apps, Power Automate, Azure Logic Apps		✓
Clustering for high availability and disaster recovery		✓
Logging of activity and health		✓
Intended for use with enterprise-wide scenarios		✓
Intended for limited, personal use by one individual	✓	

To reiterate, personal mode is not intended for enterprise deployment scenarios. The remainder of this section focuses primarily on the enterprise-level features for the data gateway in standard mode.

When a Data Gateway is Required

A data gateway is not required when working within Power BI Desktop. After a Power BI file has been deployed to the Power BI service, a gateway is required to access **data sources within the corporate network** for dataset refresh, dataflow refresh, DirectQuery, or SSAS Live Connections.

A data gateway is not necessary to access cloud-based Platform-as-a-Service (PaaS) services (such as Azure SQL Database or Azure Analysis Services), or Software-as-a-Service (SaaS) services (such as Salesforce or Google Analytics), with the following **exceptions**:

1. Data sources which reside in an IaaS (Infrastructure-as-a-Service) virtual machine.
2. Data sources which have been added to a virtual network or within a firewall.
3. If the Web.Page() function is used in an M query.
4. When a single M query combines data from cloud and on-premises data sources.

Overview of Process for Setting Up the On-Premises Data Gateway

Setting up the data gateway in standard mode involves the following process:

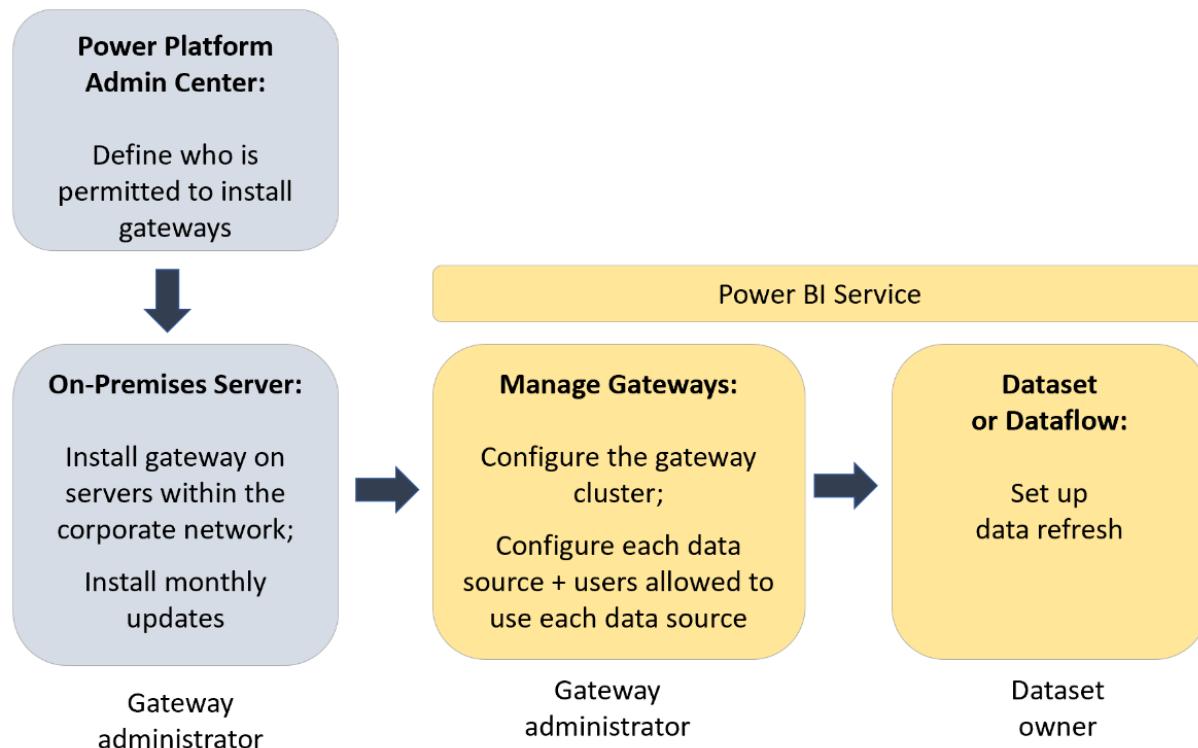


Figure 40 - High level overview of process for setting up a data gateway

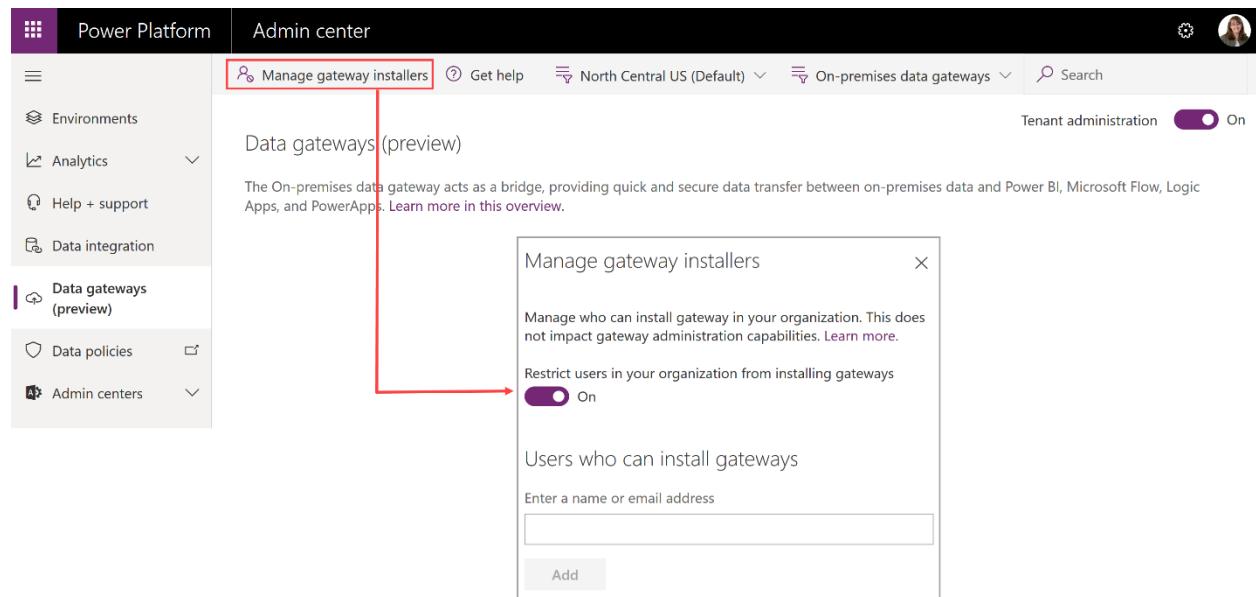


Tip for working with a combination of on-premises and cloud data sources

Combining on-premises and cloud data can be accomplished easily when handled via separate M queries. When combining data in a single query, the On-Premises Data Gateway requires the "Allow user's cloud data sources to refresh through this gateway cluster" option to be explicitly enabled on the gateway.

Managing Who Can Install a Gateway

It is possible to restrict who may install gateways. This is handled in the [Power Platform Admin Center](#).



The screenshot shows the Power Platform Admin Center interface. On the left, there is a navigation sidebar with various options like Environments, Analytics, Help + support, Data integration, Data gateways (preview), Data policies, and Admin centers. The 'Data gateways (preview)' option is selected. In the main content area, there is a section titled 'Data gateways (preview)' with a sub-section 'Manage gateway installers'. A red box highlights the 'Manage gateway installers' link. A red arrow points from this link to a larger, detailed view of the 'Manage gateway installers' dialog box. The dialog box has a title 'Manage gateway installers' and a sub-instruction 'Manage who can install gateway in your organization. This does not impact gateway administration capabilities. [Learn more](#)'. It contains a toggle switch labeled 'Restrict users in your organization from installing gateways' which is set to 'On'. Below this is a section titled 'Users who can install gateways' with a text input field 'Enter a name or email address' and a 'Add' button.

Figure 41 - The Manage Gateway Installers dialog box in the Power Platform Admin Center

Considerations for Using the Data Gateway

Data Gateway Installation File

There are two ways to obtain the installation file:

1 Power BI

The download link within the Power BI service navigates to installer links for each mode.

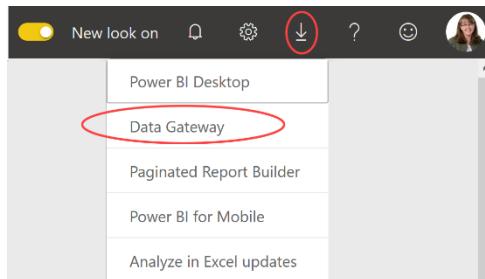


Figure 42 - Download menu in the Power BI service

2 Microsoft Downloads

The data gateway [available from Microsoft Downloads](#) is an installer that installs only the standard mode (with no option for the personal mode).

The same installer is used for initial installations and updates. Details about installing the gateway, including minimum requirements and prerequisites, may be found [here](#).

Using a Gateway Cluster

When the gateway cluster includes more than one server, two things are achieved:

High Availability	Eliminates having a single point of failure.
Load Balancing	Automatically distributes the workload across all gateway servers in the cluster.

We recommend that the production gateway cluster be configured with at least two servers to achieve high availability and load balancing. Ideally, another dev/test cluster is also configured as closely as possible to the production cluster, with the following exceptions:

- Dev/test gateways may have fewer overall nodes in the cluster.
- Dev/test gateways may have less resources (CPU and RAM) assigned to the servers. This is likely if a production gateway server has been scaled up over time to enhance workload performance.

Gateway Naming Conventions

The following are suggestions for naming the gateway and the data sources.

Gateway Cluster Name

The gateway cluster name is a logical name. It is helpful to have some variation of 'Data Gateway' or 'GW' in the name. This identification is useful with respect to logging, auditing, and troubleshooting on the server, particularly if the data gateway resides on a multi-purpose server. You may also choose to include a specific purpose, if this gateway is dedicated to supporting certain types of operations.

We recommend indicating if this gateway cluster represents dev, test, or production.

It's a best practice to have at least one gateway separate from production, for testing of monthly updates.

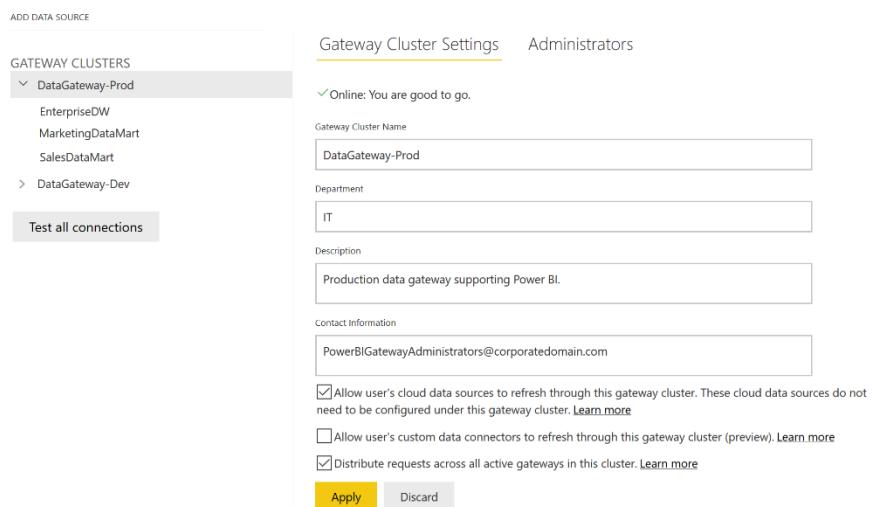


Figure 43 - Gateway Cluster Settings in the Power BI service

For the servers which support this gateway cluster, we recommend server names which align closely with the cluster name, such as:

DataGW-Prod-Node1	DataGW-Dev-Node1
DataGW-Prod-Node2	DataGW-Dev-Node2
DataGW-Prod-Node3	

When setting a data refresh schedule, the Power BI dataset/dataflow owner does see the gateway name, department, description, and contact information. This presents an opportunity to add helpful information for the user when applicable (for instance, if the dataset/dataflow owner should select a particular gateway for certain circumstances).

Data Source Name

The data source name should equate as closely as possible to the source itself, to avoid any ambiguity. This is usually the name of the database.

If applicable, other clarifications (such as Dev/Test/Prod) can be included in the data source name, if non-production environments are set up on a single gateway rather than multiple gateways.

The logical data source name may be modified after the fact. However, the type, server, and database may not be changed after it is initially configured.

The screenshot shows the 'ADD DATA SOURCE' dialog in the Power BI service. On the left, under 'GATEWAY CLUSTERS', 'DataGateway-Prod' is expanded, and 'New data source' is selected. On the right, the 'Data Source Settings' tab is active, showing the following fields:

- Data Source Name:** EnterpriseDW
- Data Source Type:** SQL Server
- Server:** BISQLVMT
- Database:** EnterpriseDW
- Authentication Method:** Windows
- Username:** BI_Svc_PowerBI_ReadOnly
- Password:** (redacted)
- Advanced settings:**
 - Use SSO via Kerberos for DirectQuery queries
 - Use SSO via Kerberos for DirectQuery And Import queries
- Privacy Level setting for this data source:** Organizational

At the bottom are 'Add' and 'Discard' buttons.

Figure 44 - Gateway Data Source Settings in the Power BI service

Identities and Credentials Used by the Data Gateway

There are several types of identities associated with the data gateway:

Gateway owner	<p>The gateway owner is specified when the gateway is initially installed on the corporate server. The e-mail account input during installation will be marked as the gateway owner and will be specified as the contact in the Power BI service.</p> <p>It is recommended this be a service account rather than an account associated with an individual user.</p>
Gateway service account	<p>When the data gateway is installed, a PBlgwService local account is automatically configured on the local machine. The gateway service then runs in the context of this account and is thus granted "Log On As A Service" permissions during installation.</p>

Name	Description	Status	Startup Type	Log On As
On-premises data gateway service	The on-premises data gateway service	Running	Automatic	NT SERVICE\PBIEgwService

Figure 45 - The windows gateway service

Some organizations may want to change this service account to run instead as a domain service account:

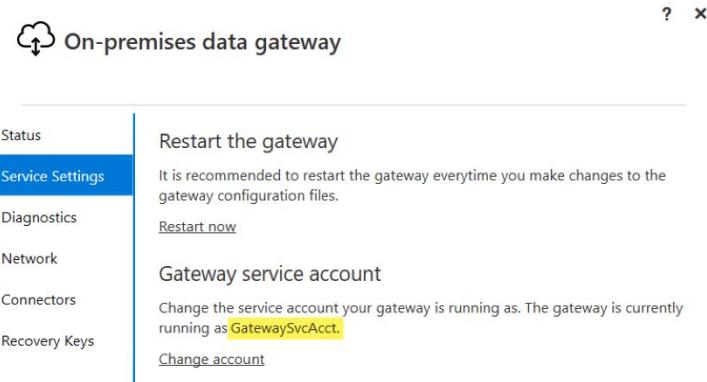


Figure 46 - Data gateway settings

When changing the [service account](#), be sure to also update:

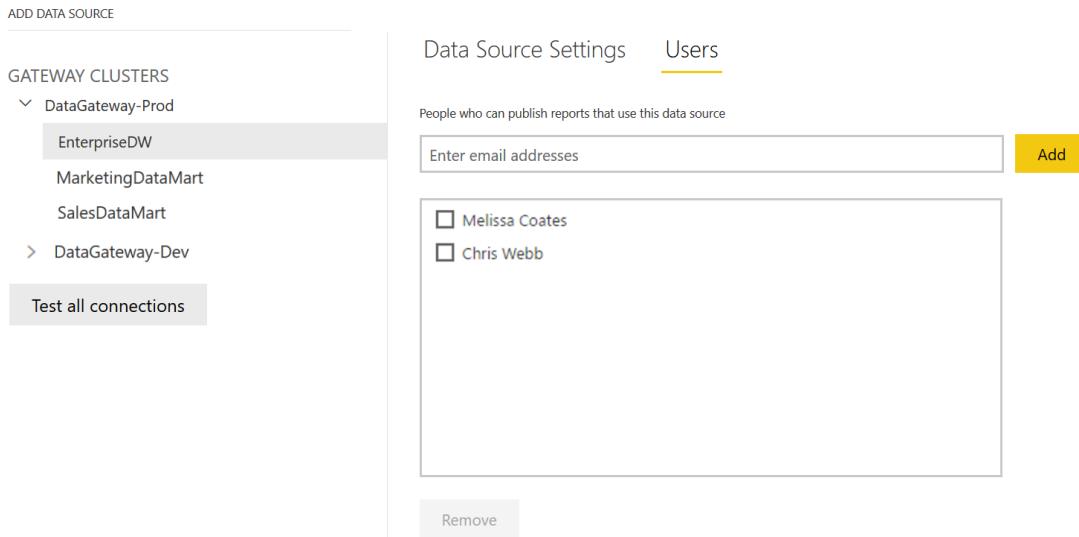
- Full control permissions for the installation location (default is located at: C:\Program Files\On-premises data gateway), as well as the location where logs are retained (if modified from the default path).
- Local security policy. User rights assignment requires (a) impersonate a client after authentication, and (b) log on as a service.

Data source credentials in the Power BI service	<p>Within the data source configuration in the Power BI service, an authentication method, username, and password need to be specified for access to the data source. These credentials are encrypted (using asymmetric encryption) on the data gateway, which also handles decrypting those credentials when the data source is accessed.</p> <p>Whenever possible, a domain-based service account should be used for data source credentials rather than an individual user account. User accounts become problematic when password changes occur or upon terminations and transfers. The service account should have a password that does not expire (or is managed). Using a service account will also be helpful for logging and auditing purposes.</p> <p>This account typically requires only read permissions on the data source. One exception is when specifying an Analysis Services data source: Analysis Services requires the account to have administrator privileges on the instance. With Analysis Services, the account for the user interacting with the report is sent via the EffectiveUserName connection string property in the form of an e-mail address. The domain in this e-mail address must match what is in the Azure Active Directory associated with Power BI, or match a user principal name mapping.</p>
Administrators for the gateway cluster	<p>Administrators of the gateway cluster are permitted to change configuration, and to add new data sources. It is strongly recommended there always be more than one administrator specified, in the event of employee terminations, transfers, or vacations.</p>
Users for the data source in the Power BI service	<p>The users specified for each data source are those who are permitted to:</p> <ul style="list-style-type: none"> • Schedule data refresh in the Power BI service, or • Publish a dataset which uses DirectQuery, or • Publish a dataset which uses Analysis Services live connection <p>Consumers of the published content do not need to be listed in the Users pane. Only dataset and dataflow owners need to be specified as Users for relevant data sources.</p>

Associating Datasets with a Gateway

For a gateway to show up as an option when the user sets up data refresh, there are 3 criteria:

First, the user who is attempting to set up data refresh must be listed on the Users page of the data source within the gateway. This omission is a common source of confusion for users trying to set up data refresh.



The screenshot shows the 'Data Source Settings' page for a data source in the Power BI service. The 'Users' tab is selected. On the left, under 'GATEWAY CLUSTERS', 'DataGateway-Prod' is expanded, showing 'EnterpriseDW', 'MarketingDataMart', 'SalesDataMart', and 'DataGateway-Dev'. A 'Test all connections' button is at the bottom of this list. The main area shows 'People who can publish reports that use this data source' with an input field containing 'Enter email addresses' and a list of users: 'Melissa Coates' and 'Chris Webb'. An 'Add' button is to the right of the input field, and a 'Remove' button is at the bottom of the user list.

Figure 47 - Gateway users assigned to a data source in the Power BI service

Second, the server name and database name used in Power BI Desktop need to match what is configured in the Power BI service. These names are not case-sensitive. For usability and readability, it is recommended that names be used instead of IP addresses. If SSL is configured for a source server, the server name for the gateway is required to be the fully-qualified domain name.

Third, each of the data sources referenced by a dataset need to be set up in the gateway (except cloud data sources that don't need a gateway). This is because a data refresh operation can communicate through only one gateway.

User Interactions with the Data Gateway

The only awareness of a gateway that a typical business user (non-administrator) should have is when setting up data refresh. This impacts dataset and dataflow authors, but not report consumers.

Ideally, data sources for commonly used data sources in the organization are pre-configured so that users working in DirectQuery or Live Connection mode have no issues. Suggestions to make this process as smooth as possible for the user include:

- ✓ Set up a data source only once. This is most common and is recommended.
- or
- ✓ If the same data source is purposely set up in multiple gateways, configure the users for each data source to be distinct groups of users. This will ensure each user sees only one data source and will prevent the user from needing to guess which gateway to select.

Personal Gateway Warnings

If a user has set up a personal mode gateway for themselves, that is the first option listed for the dataset in the Power BI service. If the user does not have a personal gateway set up, it displays as a warning even though the mapping to the standard mode gateway is correct:

Settings for Call Center Data

This dataset has been configured by [REDACTED]

Last refresh succeeded: Wed Feb 05 2020 12:53:15 GMT-0500 (Eastern Standard Time)
[Refresh history](#)

► Gateway connection

To use a data gateway, make sure the computer is online and the data source is added in [Manage Gateways](#). If you're using an On-premises data gateway (standard mode), please select the corresponding data sources and then click apply.

Use a data gateway

On

Gateway	Department	Contact information	Status	Actions
<input checked="" type="radio"/> DataGateway-Prod	IT	PowerBIGatewayAdm...	Running	

Data sources included in this dataset:

<input checked="" type="radio"/> SqlServer("server": "f...", "database": "...")	Maps to:	<input type="button" value="EnterpriseDW"/>
---	----------	---

⚠ You have no personal gateways installed. [Learn more](#)

[Install now](#)

This is a user training issue. In this situation, the data refresh will succeed so the warning message can be ignored by the dataset owner. The goal is to educate dataset owners to use the standard data gateway when possible.

Figure 48 - Warning related to the personal gateway displayed when scheduling a data refresh

Data Gateway Credentials

For DirectQuery, [single sign-on](#) may be configured so that user permissions in the source system are honored.

For import datasets, the credentials are inherited from the data source as configured in the data gateway. This is extremely important point because it may impact permissions on data retrieved from source systems.

As an alternative, [the dataset owner's identity may be used for import queries](#) if the data source is configured to allow it:

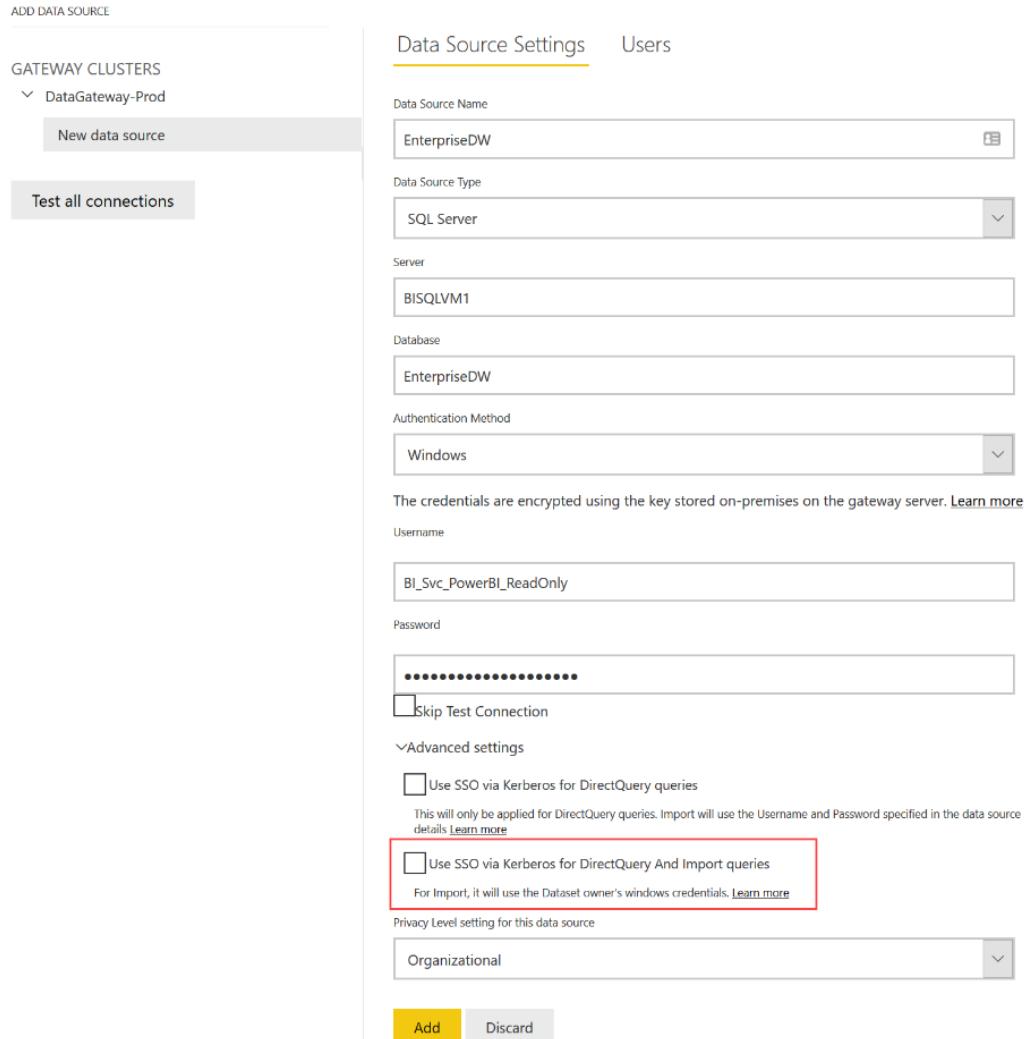


Figure 49 - Gateway data source settings to configure single sign-on

The above setting is very helpful for re-using one single data source for a wide number of people/teams who would retrieve different data based on their underlying permissions. This avoids having to set up the same data source multiple teams, with different credentials, and different user populations assigned.

Sharing of Data Gateway with Other Applications

As mentioned earlier, the On-Premises Data Gateway can be reused among Power BI and Azure Analysis Services, as well as Logic Apps, Power Apps, and Power Automate. This applies at the gateway (infrastructure) level only; the data sources for each application are independently configured. If one of the services begins to place more demand upon the gateway server, it may become necessary to introduce more than one gateway to support all operations.

Using One or More Data Gateways

One data gateway can communicate with multiple data sources. Depending on the volume of activity, a single gateway may be adequate.

However, for an enterprise deployment, there are a few concerns with using a single gateway:

- This represents a single point of failure should there be a hardware failure.
- If the gateway server needs maintenance, it may be offline for a period.
- If there are numerous types of data sources, and/or a lot of DirectQuery activity, and/or multiple services are involved, the load may become too much for a single gateway machine to handle effectively.
- There is no place for testing the monthly gateway update installation.

We recommend that a production gateway cluster include two servers as minimum. As mentioned earlier in this section, the separate servers appear as a single logical gateway to Power BI. This supports the need for high availability: all requests for data that result from the refresh of imported data sources, or come from DirectQuery or live connections, are routed to the primary gateway in the cluster if it is online; if it is not online these requests will be routed to another instance in the cluster.



If a Power BI Desktop file contains multiple data sources in order to perform a mashup, each of those data sources needs to be configured in the same gateway for the scheduled refresh to succeed. Put another way, a scheduled refresh can only refer to one gateway. For this reason, a data source may end up being configured in multiple gateways. This scenario should be approached carefully so as not to be confusing for users.

Before the introduction of gateway clusters, older guidance discussed separating gateways which serve data refresh operations for import models vs. DirectQuery models. However, with the introduction of gateway clusters, that is no longer a recommendation. Scaling up (more CPU and/or RAM) and scaling out (adding another machine to the cluster) is the approach for handling the multiple workloads.

Performance Impact on the Server Running the On-Premises Data Gateway

Before discussing where to install the data gateway, it is important to understand the potential performance impact on a gateway server. The impact can vary greatly.

There are three main performance-related considerations:

CPU	DirectQuery and Live Connection activity are more CPU-heavy workloads.
Memory	Import mode data refreshes are highly memory-dependent workloads. When operations cannot be pushed down to the source system, then Power BI may need to perform filtering, matching, or other operations itself to prepare the dataset. For imported datasets, memory is extremely important.
Network Bandwidth	Network bandwidth is extremely important for passing volumes of data and/or handling frequent queries.

[Query folding](#) can have an impact on the gateway server. For simple queries which extract small amounts of data into imported datasets, the impact is typically minimal on the gateway server. When operations such as predicates, calculations, aliasing, joins, and type conversions can be pushed down to the source system using query folding, the gateway avoids incurring the load of performing those operations.

 The data gateway handles more than just data connectivity. Work is performed on the gateway machine, which is why the server(s) may need to be scaled up or out to accommodate the workload.

The data privacy settings per source can also have a very significant impact on performance and can even prevent refresh from working completely. It is possible to stop the gateway from checking data privacy levels completely (applicable to personal mode only), but this option should be used with extreme caution.

Where to Install the On-Premises Data Gateway

The following are some considerations for where to install the data gateway.

- Use of a machine dedicated to gateway operations vs. a multipurpose machine (such as a server with SQL Server already installed). If an existing server is currently underused, using an existing machine is often a good place to start. However, using a dedicated server for the data gateway is a best practice.
- The machine should always remain powered on, with near full-time uptime.
- The gateway server should reside as close to the data sources as practical to minimize network latency.
- An Azure virtual machine can be a good option to install a gateway if on-premises equipment is not available. In this situation, the VM should reside in the same region as

the Power BI tenant unless there are other considerations such as proximity to the data or minimizing data egress charges.

- The gateway must reside in the same domain as the data source(s), or in another domain which is trusted.
- The gateway server requires strong network connectivity (preferably not wireless), high bandwidth, and consistent internet access. ExpressRoute or Azure Private Link is highly recommended. The [Azure Speed Test](#) site can be used to check network latency between a client machine and an Azure region.
- A potential starting point for a gateway server is 8 CPU cores and 16GB RAM, as noted in the [Gateway Best Practices](#). Plan to scale up on CPU and/or RAM if bottlenecks present themselves, and/or scale out to multiple gateways if the volume of traffic becomes heavier than one data gateway can handle.

Firewall Ports

The On-Premises Data Gateway communicates with the Azure Service Bus via TCP on [outbound ports 443 by default](#). It can also use 5671, 5672, or 9350 – 9354. With TCP, the On-Premises Data Gateway communicates with the Azure Service Bus with a combination of IP addresses and domain names.

Communication Protocol

As of mid-2019, the default for new gateway installations changed to be [HTTPS](#). This was done to improve customer experience and reduce connectivity failures for new installations.

There are two options for setting the gateway communication protocol:

- **HTTPS** (the default). When enabled, an additional layer of SSL security is added on top of the transport security. This should have only a nominal performance impact.
- **Direct TCP**. With this method, the gateway uses TLS (Transport Layer Security) to communicate with the Power BI Service. Data is encrypted in transit by the transport protocol which is [TLS 1.2](#).

Other Items To Be Installed on the Gateway Server

Depending on the data source, other components may need to be installed, such as:

- Custom drivers (ex: Oracle or SAP HANA)
- Custom connectors

Monitoring and Managing Data Refreshes

Data Refresh Success

A scheduled [data refresh job will time out](#) if not completed within 2 hours for datasets in standard capacity, and 5 hours for datasets in Premium capacity. As discussed in [Section 5](#), it is very important to extract the smallest amount of data from the source that will meet your objectives. Using incremental data refresh is another excellent strategy to avoid longer data refresh as data volumes grow over time.

For critical datasets which a lot of people around the organization depend upon, we recommend that a support group be part of the notification list should a failure occur:

Settings for Call Center Data

This dataset has been configured by: [redacted]

Last refresh succeeded: Wed Feb 05 2020 12:53:15 GMT-0500 (Eastern Standard Time)
[Refresh history](#)

- ▶ Gateway connection
- ▶ Data source credentials
- ▶ Sensitivity label (preview)
- ▶ Parameters
- ◀ Scheduled refresh

Keep your data up to date

On

Refresh frequency:

Time zone: (UTC-05:00) Eastern Time (US and Canada)

Time: [Add another time](#)

Send refresh failure notifications to the dataset owner

Email these users when the refresh fails

Power BI Support

Figure 50 - Scheduled refresh settings for a dataset in the Power BI service

Paused Data Refreshes

Scheduled refresh will be [paused](#) by the system if no users use any dashboard or report built on the dataset in 60 days. If any user revisits a dashboard or report built on the dataset, the scheduled refresh will resume.

Deactivated Data Refreshes

After 4 consecutive failed refreshes, or an unrecoverable error (such as invalid credentials), scheduled refresh will be [deactivated](#) by the Power BI service. When a data refresh schedule has been paused or deactivated, an e-mail notification is sent to the owner of the refresh schedule.

Data Refresh Limitations Based on License Type

The expectations for data refresh differ based on the type of Power BI license, as follows:

Feature/Capability	Power BI Free	Power BI Pro	Power BI Premium
Scheduled data refresh with gateway in personal mode	Up to 8x/day My Workspace	Standard mode is recommended	
Scheduled data refresh with gateway in standard mode		Up to 8x/day	Up to 48x/day
DirectQuery and SSAS live connection in standard mode		✓	✓

Monitoring and Managing Gateways

Managing Data Gateways

There are three areas encompassing management of a data gateway:

- The gateway server itself
- The [Manage Gateways](#) area in the Power BI service
- The [Power Platform Admin Center](#)

Keeping the Data Gateway Updated

Data gateway updates are not auto-installed. We recommend staying as current as possible with the latest data gateway version to ensure the mashup engine in the gateway is aligned with the latest mashup engine used by Power BI Desktop and Power Query Online. The goal is to avoid unpredictable behavior.

Data gateway updates are released as often as once per month. Some updates may introduce new dependencies (for example on the .NET framework), so be sure to check whether this is the case before starting the update process.

As [announced](#) in March 2020, Microsoft only supports the last 6 releases of the data gateway. We recommend trying to stay within the last 2 releases if possible.

It's very important to monitor the Power BI blog, as that is where new gateway prerequisites and changes to default configuration are announced.

As a standard practice, we recommend backing up each of the gateway configuration files before beginning the upgrade process.



There is not a separate installer for gateway updates vs. initial installation.

If it detects that you are installing a **newer version** of the gateway: the existing settings are retained (though we still recommend backing up all configuration files before beginning).

If it detects that you are installing the **same version or lower**: the installer would throw an error.

Process for Gateway Updates

As discussed previously, we recommend having a separate dev/test gateway to test updates prior to implementing them on the production gateway. Once the updated gateway has been installed on the dev/test gateway, several repeatable tests should be done such as: a successful data refresh operation and a successful DirectQuery report execution, each using the most important data sources and/or each distinct type of data source.

When all the requisite tests have been performed, then the production gateway may be updated. It is very important to use the same installer file for the production gateway update that was used for the dev/test gateway update (i.e., use the same file rather than downloading it from the web again).

We also recommend that you disable a gateway node before performing an update so the gateway cluster does not try sending requests to that gateway member until you re-enable it after the update successfully completes.

When performing the updates, the steps are repeated for each node in a gateway cluster. To avoid any downtime, it is very important that the update for node A be completed and validated prior to beginning the update for node B in the gateway cluster.

Sometimes a new prerequisite is required, such as a new version of .NET. Depending on IT policies, it is possible that a server infrastructure team will handle certain types of updates, and a BI team (gateway administrators) is permitted to manage the remainder of the process. Other companies allow the BI team to fully manage the dev/test gateway, with production gateway updates handled by the IT server infrastructure team with a change ticket. When there is a division of responsibilities such as this, then performing initial unit tests (as mentioned a few moments ago) becomes even more important.

Data Gateway Permissions

There are two roles related to gateways: **Administrators** and **Users**.

Administrators are defined for each logical gateway cluster. An administrator may configure the gateway cluster settings, configure data sources, and add users. Because the gateway administrator configures data sources – including credentials to read data from the source – that makes the gateway administrator a high privilege role.

Users are defined for each individual data source. Users are the Power BI users who are authorized to use a data source which has been registered to a data gateway by an administrator. This can be for DirectQuery, Live Connection, or performing data refresh operations.

 Dataset and dataflow owners who publish and refresh data *do not* need to be added as gateway administrators, unless the expectation is for them to also administer the gateway for the organization. Dataset and dataflow owners can use a gateway for their specific data source if:

1. The dataset/dataflow owner has been assigned to the data source.
 2. The connection defined in Power BI Desktop exactly matches how the connection is configured for the data source registered to the gateway. A common example of a mismatch is to use “*servername*” in one and the fully qualified “*servername.company.com*” in another.
 3. The gateway cluster where the data source has been registered is active and healthy.
 4. Any custom driver(s) required for communicating with the data sources are installed on the gateway server.
-

Monitoring Gateway Health

There are various techniques for monitoring discussed in the documentation for [troubleshooting gateways](#). Commonly used techniques include the following:

Available in the Power BI service:

Data Refresh History	<p>History may be reviewed on an individual dataset basis in the user interface, or via the REST API.</p> <p>Refresh history</p> <p>Scheduled OneDrive</p> <table border="1"><thead><tr><th>Details</th><th>Type</th><th>Start</th><th>End</th><th>Status</th><th>Message</th></tr></thead><tbody><tr><td>Scheduled</td><td></td><td>5/28/2017, 5:01:06 PM</td><td>5/28/2017, 5:14:12 PM</td><td>Completed</td><td></td></tr><tr><td>Scheduled</td><td></td><td>5/21/2017, 5:01:06 PM</td><td>5/21/2017, 5:13:04 PM</td><td>Completed</td><td></td></tr></tbody></table> <p><i>Figure 51 - Refresh history for a dataset displayed in the Power BI service</i></p>	Details	Type	Start	End	Status	Message	Scheduled		5/28/2017, 5:01:06 PM	5/28/2017, 5:14:12 PM	Completed		Scheduled		5/21/2017, 5:01:06 PM	5/21/2017, 5:13:04 PM	Completed	
Details	Type	Start	End	Status	Message														
Scheduled		5/28/2017, 5:01:06 PM	5/28/2017, 5:14:12 PM	Completed															
Scheduled		5/21/2017, 5:01:06 PM	5/21/2017, 5:13:04 PM	Completed															

Available on the Data Gateway Server:

Gateway Performance Monitoring Logging

The best option for managing gateway health is to enable the gateway performance monitoring logs as discussed [here](#). When enabled via a configuration file, several log files are written to the local server with details about query executions and system counters. For health monitoring, using these logs is significantly easier than getting the data from Windows Performance Monitor (PerfMon).

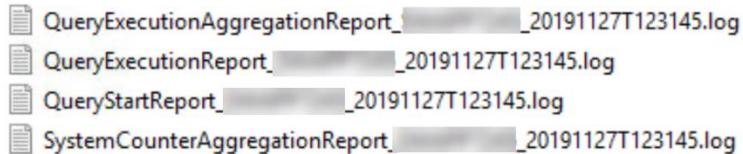


Figure 52 - Gateway performance monitoring log files saved to local drive

To enable, the "QueryExecutionReportOn" and "SystemCounterReportOn" should be set to true in the `Microsoft.PowerBI.DataMovement.Pipeline.GatewayCore.dll.config` file. Consider also modifying the aggregation time associated with each of these; 5 minutes is the default which may generate more data than is desired. You may also consider modifying the output path for the logs from the default to a more easily accessible location. This is handled in the "ReportFilePath" property.

We recommend creating a process to compile the performance monitoring logs for each gateway cluster machine into a central location so they can be analyzed together for reporting purposes.

Event Log Viewer

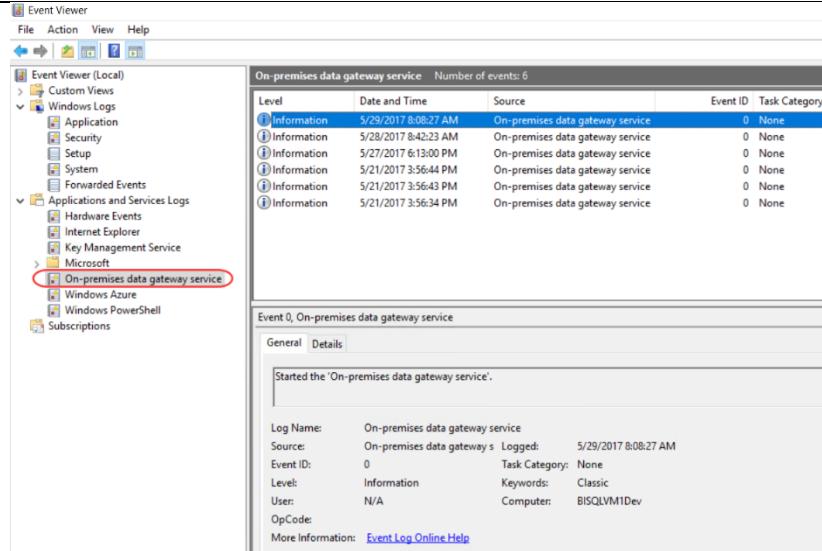
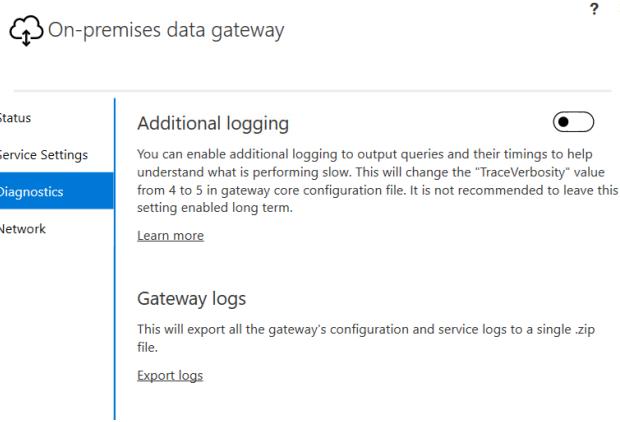
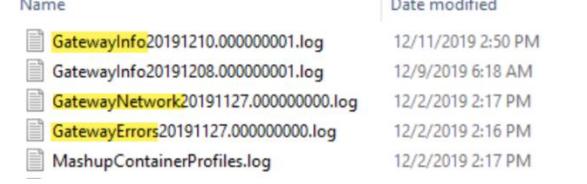
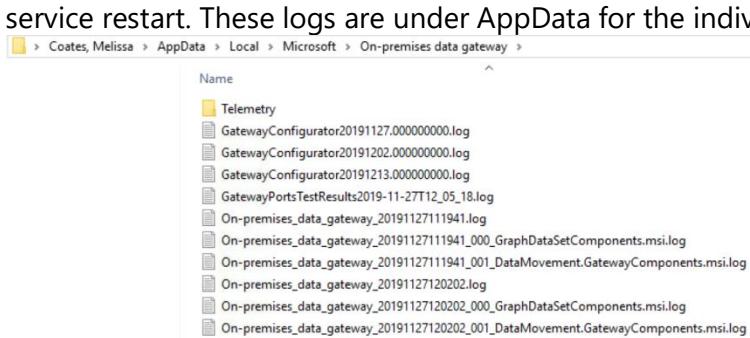


Figure 53 - Event log viewer in Windows for the data gateway service

<p>Verbose Logging and Log Exports</p>	<p>Additional verbose logging may be enabled temporarily to actively troubleshoot issues:</p>  <p>Figure 54 - Settings on the gateway server to enable additional logging</p>
<p>Gateway System Logs</p>	<p>Unless the installation path was modified, gateway logs are available under: C:\Users\PBIEgwService\AppData\Local\Microsoft\on-premises data gateway\Gateway*.log.</p> <p>There are several types of system log files. In particular, note the presence of the GatewayErrors log:</p>  <p>Figure 55 - Gateway system log files saved to local drive</p> <p>The config file Microsoft.PowerBI.EnterpriseGateway.exe.config specifies the number of logs to retain. The default is 20.</p>
<p>Gateway User Logs</p>	<p>The other type of system logs relate to the individual user (gateway administrator) who performed activities such as an installation, upgrade, or service restart. These logs are under AppData for the individual user:</p>  <p>Figure 56 - Gateway system log files saved to local drive</p>

Be sure to check these logs on a regular basis for problems. Users may not always report issues when they encounter them, so we recommend active monitoring for health and performance.

Data Gateway Configuration Files

There are several configuration files which are very important to customizing how the gateway server runs in your environment. The configuration files are co-located with the executable (dll) files, so be very cautious when working in this folder.

Unless the installation path was modified, the configuration files are located at C:\Program Files\On-Premises data gateway.

Most of the configuration files are for internal use only. The main file that customers will use to customize behavior is: Microsoft.PowerBI.DataMovement.Pipeline.GatewayCore.dll.config. Here, various customizations may be made such as:

- Updates related to single sign-on (FullDomainResolutionEnabled)
- Additional performance monitoring for queries (QueryExecutionReportOn)
- Additional performance monitoring for health counters (SystemCounterReportOn)
- Various database connection settings

The gateway service needs to be restarted every time a configuration file has been modified.

Data Gateway Server Uptime and Health

When possible, we recommend registering the gateway servers in the standard IT monitoring platform. This allows for real-time alerting if there are issues with the production gateway cluster.

There are also PowerShell cmdlets for checking the [status of a gateway](#) and the [status of a data source](#). See [Section 10](#) for additional discussion about using PowerShell for monitoring purposes.

In addition to the PowerShell cmdlets, there are also APIs for checking the [status of a gateway](#) and for checking the [status of a data source](#).

If there are issues on a regular basis, you may want to consider scaling up (more resources) or scaling out (more servers in the cluster).

Data Gateway Disaster Recovery

A recovery key is assigned (i.e., not auto-generated) by the administrator at the time the data gateway is installed. The recovery key is required if the gateway is to be relocated to another machine, or if the gateway is to be restored.

The recovery key is also required when joining a new server to an existing gateway cluster.

The recovery key should be retained where other system administrators can locate it if necessary. A corporate password vault is very suitable for this purpose.

Types of Data, File, and Cache Refresh in Power BI

There are several different scenarios where data stored in Power BI is refreshed:

Dataset Refresh

Description	A dataset refresh sends queries to the underlying data source to perform a full or incremental refresh of data and metadata for an imported dataset. For DirectQuery datasets and Live Connections, only metadata is refreshed.
How Refresh Is Handled	<p>There are several options for refreshing a dataset.</p> <p>Within the Power BI service:</p> <ul style="list-style-type: none">• Scheduled refresh• Refresh now button <p>Outside of the Power BI service:</p> <ul style="list-style-type: none">• REST API• Power Automate• Manually re-uploading a .pbix file

Dataflow Refresh

Description	A dataflow refresh sends queries to the underlying data source to perform a full or incremental refresh of data stored for the dataflow.
How Refresh Is Handled	<p>There are several options for refreshing a dataflow.</p> <p>Within the Power BI service:</p> <ul style="list-style-type: none">• Scheduled refresh• Refresh now button <p>Outside of the Power BI service:</p> <ul style="list-style-type: none">• REST API

Package Refresh (File Level)

Description	A package refresh synchronizes a .pbix or .xlsx file in the Power BI service from OneDrive or SharePoint Online. It does not run a data refresh from the underlying source data—it only copies the dataset out of the file and into the Power BI service. It also reflects any changes made to the reports in the .pbix file.
How Refresh Is Handled	File-level refreshes are handled automatically by the Power BI service, approximately every 1 hour.

Cached Tile Refresh

Description	Dashboard tiles display cached data, to ensure performance is optimal. The cached tile refresh updates the cached data displayed by dashboard tiles. This is useful after a data refresh has occurred to bring the dashboard tiles up to date.
How Refresh Is Handled	<p>There are several options for refreshing cached tiles.</p> <p>Automatically by the Power BI service:</p> <ul style="list-style-type: none">• After data refresh for imported datasets• Approximately every 1 hour (or as specified in the dataset properties) for DirectQuery and live connections datasets <p>Manually:</p> <ul style="list-style-type: none">• Refresh option from the menu

Cached Visual Container Refresh

Description	The visual container refresh updates the charts, graphs, and tables which have been cached in reports for performance reasons. This is useful after a data refresh has occurred to bring the visuals up to date.
How Refresh Is Handled	<p>There are a couple of options for refreshing cached tiles.</p> <p>Automatically by the Power BI service:</p> <ul style="list-style-type: none">• The visual containers are usually refreshed automatically once a data refresh operation completes <p>Manually:</p> <ul style="list-style-type: none">• The Refresh menu item at the top of a report

With respect to cached tiles and visuals, DirectQuery and live connections datasets will send queries back to the underlying source to refresh the cache. This can result in a lot of queries depending on the number of dashboards with pinned visuals.

 **Tip:** The Refresh menu item when viewing a report executes a cached visual container refresh as described above, which does *not* trigger a refresh of the underlying dataset.

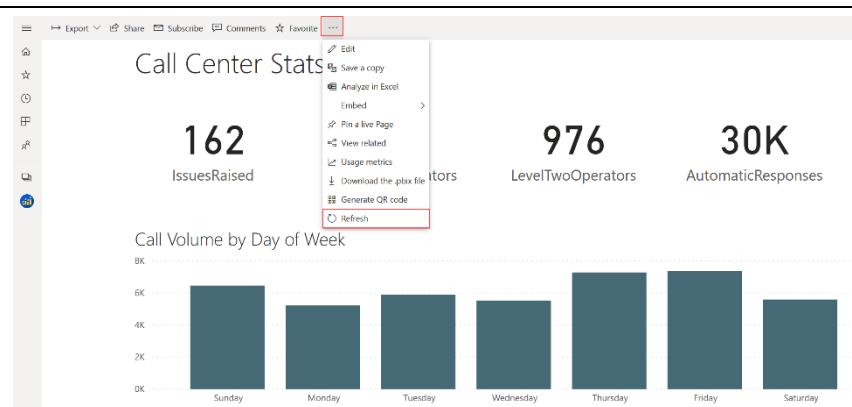


Figure 57 - Menu option in the Power BI service for refreshing the visual cache

This refresh option is typically most useful for DirectQuery or live connection datasets. It can also be useful if you manually refreshed an imported dataset just a few moments earlier.

Section 8. Power BI Dataset and Report Development Considerations

This section focuses on content creation. There are many related considerations regarding use of shared datasets, dataset permissions, reusability of data, and distributing content through apps discussed in other sections of this whitepaper.

Choosing the Right Tool

Power BI supports dataset and report development in multiple ways:

- Power BI Desktop on a PC
- Power BI service in a browser
- Excel on a PC
- Power BI Report Builder on a PC

In addition to this list, there are two additional options which are excluded from the scope of this section. The first concerns editing reports in an embedded scenario, as this would typically occur in the Power BI service. Also, the [new XMLA read/write feature](#) will introduce flexibility with what tool is used to create content.

To choose the right tool for the job, it is helpful to be aware of the strengths and weaknesses of each option.

Power BI Desktop

Power BI Desktop is the default choice for report development and recommended in most scenarios. It allows you import data from many data sources, combine data from multiple data sources, clean and transform that data, create a dataset, add DAX calculations, and build reports all in one application. Factors to be aware of include:

- Reports can be created in Power BI Desktop, whereas Power BI dashboards can only be created in the Power BI service (because one dashboard can contain pinned visuals from numerous reports).
- Power BI Desktop is available as a Windows desktop application. It does not run natively on other platforms such as Mac OS (though third-party virtualization platforms, such as Parallels, can enable this scenario).
- Power BI Desktop is typically updated every month. This is good news for users who wish to receive new features as quickly as possible. However, rolling out these updates for a large number of desktops requires planning. This is discussed in more detail in [Section 10](#).
- The .pbix files created by Power BI Desktop can be saved to a common location, such as One Drive or SharePoint, with versioning enabled. This is considered a best practice.

Power BI Service

The web-based Power BI service offers a subset of the functionality of Power BI Desktop for report development, in that it allows you to build reports and save them within a workspace. You have access to most report building capabilities in the Power BI service; however, it does not give you access to data modeling capabilities. It does not currently allow you to import, clean or transform data, nor create nor edit datasets (with the exception of dataflows). Measures cannot be created in the Power BI service, nor can you import images to a report.

The Power BI service is mainly used for its sharing and distribution capabilities:

- Dashboards can be created in the Power BI service, as well as reports.
- Apps can be created to package up a set of related reports and dashboards, for the purpose of distributing official content to a broad audience.
- Since it is a web-based application, it is available to all users regardless of what desktop operating system they use (including Mac). A list of supported browsers can be found [here](#).
- As a web-based application, the Power BI service needs no specific maintenance or management by the customer to "upgrade" to the latest version. Though there are some opt-in features, as a customer you cannot delay the introduction or control the timing of receiving new features and updates.
- There is no version history for reports created in the Power BI service currently.



We recommend that report creators use Power BI Desktop as the primary report creation tool rather than the Power BI service. Original files should be stored in a common location (such as the team One Drive) which is secured for authorized personnel. A process should exist, that all members of the team are familiar with, for how content is published to minimize the risk that someone will overwrite someone else's work. This all becomes easier to do when using Power BI Desktop.

Excel

Newer versions of Excel contain several features that are very closely related to Power BI: originally available as separately downloadable add-ins, Power Pivot became a native feature of Excel 2013 as the Excel Data Model, and Power Query became a native feature of Excel 2016 as the "Get & Transform" feature. The Excel Data Model uses the same database engine as Power BI for storing imported datasets; "Get & Transform" offers almost identical functionality to the Power BI Query Editor for loading data.

Using Excel in combination with these features to build reports provides a useful alternative to Power BI Desktop in certain circumstances. However, Excel does have several limitations compared to Power BI Desktop:

- ✗ The maximum [size of an Excel workbook](#) (which contains an Excel Data Model, otherwise known as Power Pivot) that can be published to Power BI is 1GB. Conversely, the size maximum for a Power BI Desktop file that can be uploaded to the Power BI service is 10GB (with Premium capacity, that is – without Premium, the same 1GB size limit applies). Outside of the Power BI service, the file size limit only depends on the individual user's machine specifications.
- ✗ Excel reports that use a live connection to a dataset in the Power BI service, via Analyze in Excel, may not be refreshed or edited after being published to the Power BI service. At this time, an Excel Data Model is required for an .xlsx file to be refreshed in the Power BI service. Excel in SharePoint Online has the same restriction.
- ✗ In some organizations, users have the 32-bit version of Excel installed rather than the 64-bit version. The realistic maximum size of dataset in the 32-bit version of Excel is around 1-2GB after compression. The 64-bit version of Excel can support much larger data volumes and generally performs much better than the 32-bit version.
- ✗ Some features that are present in Power BI Desktop are not present in the Excel Data Model or are introduced on a significantly slower cadence. Therefore, complex modeling requirements cannot always be handled as easily in Excel.
- ✗ Some data sources that are available in Power BI Desktop are not available in Excel.
- ✗ When opened in the Power BI mobile apps, Excel reports that use data from the Excel Data Model appear as static reports (they are fully interactive when viewed in a browser).
- ✗ The Power BI Publisher for Excel add-in is no longer supported with the new workspace experience and should not be used going forward.
- ✗ The Power View add-In for Excel, a very basic data visualization tool, is no longer being developed by Microsoft and is no longer visible in the Excel 2016 ribbon by default. The

Power View add-in should not be used for report development. Rather, Excel-based report development should primarily focus on pivot tables, pivot charts, and [cube functions](#).

- ✗ Not all Excel functionality is supported when an Excel workbook is published to Power BI and viewed in a browser. For example, the 3D Maps functionality in Excel 2016 (previously known as Power Map in Excel 2013) is not available in a published workbook.
- ✗ Custom and certified connectors are not yet available for Excel.

That said, there are some cases where Excel does offer a better experience for report design than Power BI Desktop:

- ✓ Excel has the full power of Excel formulas for creating calculations, in addition to what you can do in DAX, and a vast number of users are familiar with writing Excel formulas.
- ✓ Excel has a very mature list of charts and graphs for visualizing data, with slightly different options available versus Power BI Desktop. Like Power BI Desktop, Excel also supports the [use of Power BI custom visuals](#) in a workbook.
- ✓ Excel PivotTables are preferred by many users for ad hoc data exploration over the Power BI matrix visualization.
- ✓ Excel [cube functions](#) allow for fine control over where values appear in a report layout, and have no equivalent in Power BI Desktop. Cube functions are extremely useful for creating financial-style reports, for example.



Using Power BI and Excel Together

We recommend that you use a combination of Power BI Desktop and Excel based on each use case. Power BI Desktop enables fast creation of interactive reports over any data that is in a data model. Authors can control the interactions and analysis that consumers can perform. The data stays up to date with a scheduled refresh once it's published to the Power BI service.

Excel enables detailed ad hoc analysis and what-if scenarios. Users can manipulate data in sheets, use formulas to combine data, or shape data to fit into custom processes.

Some organizations have existing processes built on Excel that can be **modernized by connecting the Excel file to refreshable data in Power BI**. Ensuring Excel files are connected to a dataset in the Power BI service, rather than using a data export, is highly recommended. This saves users from repeating tedious steps, prevents data from becoming stale, and ensures data security is applied when users refresh the Excel data.

Power BI Report Builder

Power BI Report Builder is a desktop tool for authoring paginated reports, which are most useful for highly formatted reports such as financial statements. If you have authored Reporting Services reports in the past, it is the same banded report creation experience.

Factors to be aware of:

- Approach paginated reports with a different mindset. That is, a paginated report is always focused on the creation of one individual report. There is a dataset within an .rdl file, but it is scoped to that one report (i.e., it is not seen as a separate object in the Power BI service, nor is there is the concept of a shared dataset).
- Publishing of paginated reports to the Power BI service is available for workspaces which have been assigned to Premium capacity. This involves at least a P1 SKU or, if using capacity from the Power BI Embedded service in Azure, at least an A4 SKU to ensure adequate resources.
- For situations when data is accessed through a database stored procedure, such as a SQL Server stored procedure, that is possible with paginated reports.
- Development of paginated reports is generally considered to be a higher learning curve than Power BI Desktop. However, the benefit is fine-grained control over layout and placement, as well as improved printing and exporting capabilities with paginated reports.
- Report consumers do become accustomed to the built-in interactivity of Power BI reports, which is not available with paginated reports.
- There are some [feature differences](#) and unsupported items when a paginated report is intended to be published to the Power BI service (as opposed to Power BI Report Server).

See the [paginated reports FAQ](#) and [paginated report design tips](#) for additional information.

Best Practices for Dataset and Report Design in Power BI Desktop

It is beyond the scope of this whitepaper to go into detail on best practices for dataset and report design in Power BI Desktop. Entire books have been written about creating and optimizing DAX calculations, for example. However, it is possible to offer some simple tips and tricks that will make your reports as responsive and easy to manage as possible. You may also want to refer to [this page](#) for additional suggestions.

Managing Dataset Authorship

Monitoring of who builds datasets, where these datasets are published, and how many datasets are published, are essential if you are going to avoid thorny issues later in your project. If you allow unrestricted dataset authorship, you will find that duplicate datasets are created. In turn, this means that:

- Report creators are uncertain which dataset to use as a source for their reports.
- Maintenance of datasets becomes time-consuming and difficult, for example when data sources change, or calculations need to be altered.

- You reach the limits for how much data a user can publish using a Power BI Pro subscription, or even the available memory of your Power BI Premium capacity, more quickly.
- Scheduled refresh for large numbers of reports at the same time can put excess load on data sources, making both the data sources and the report refresh slow.
- Different data modeling decisions, variations in how calculations are defined, and even inconsistent column and measure names, mean that users have difficulty comparing the data displayed in different reports and are unsure which reports to trust.

The list above does not equate to a recommendation to centralize all report authorship in the IT department. Often that diminishes many of the beneficial aspects of self-service BI. However, it is important to ensure that:

- ✓ All dataset and report authors are properly trained.
- ✓ Authors follow your organization's documented best practices and standards when building datasets and reports.
- ✓ Duplicate datasets and reports are avoided whenever possible, and authors understand why this is crucial.
- ✓ Datasets and reports are thoroughly tested before they are published.
- ✓ Datasets and reports always have an owner identified who is responsible for maintenance, questions, and data quality issues.

 Many considerations such as these relate more to the 'process' and 'people' side of the equation, rather than strictly a 'technology' issue.

Organizations should encourage authors to use shared datasets (which are discussed in [Section 9](#)) that are certified or promoted. This minimizes dataset sprawl. It also helps report creators select the best dataset for their analysis.

We recommend training content authors to start by considering how datasets will be reused over time, and how not to overshare their datasets. Organizations should *not* block dataset authors altogether from creating new datasets since this is essential to self-service and removing IT as a bottleneck for users. Rather, Power BI administrators should monitor dataset creation through audit logs and usage to be aware of, review, and govern datasets that are becoming critical to the organization.

Version Control for Power BI Content

A significant amount of work goes into designing and validating content. Therefore, all original .pbix files should be stored in a safe location. The best way of doing this is through some form of source control or version control system. Although Power BI does not have any native features for version control or for interfacing directly with external version control systems (such as a Git repository), most version control systems allow you to store files of any type in a repository. This includes various Power BI file types, such as:

Power BI files	.pbix
Power BI template	.pbbit
Power BI data source	.pbids
Power BI custom visuals	.pbviz
Power BI theme files	.json
Paginated reports	.rdl
Excel workbooks	.xlsx



Use Power BI Desktop for most report authoring

Source control and versioning (discussed next) for Power BI Desktop files becomes particularly challenging when report changes are done directly in the Power BI service. To minimize the risk of overwriting/losing changes, we recommend implementing a standard practice for editing reports and publishing them from Power BI Desktop.

Source Control Systems

Source control, such as Git, is a common requirement for IT and centralized BI teams.

Consider using [Power BI template](#) files (.pbbit files) instead of .pbix files so that:

- The files you are storing in version control are as small as possible.
- Data is not stored in the source control system.

Power BI templates (.pbbit) only contain metadata. The actual data from the dataset is excluded, which this makes .pbbit files much smaller than .pbix files. Typically, template files are used for standardization of “look and feel” such as implementing organizational branding, so this usage of templates will need to be documented and communicated to the team.

Since .pbix and .pbbit files are basically treated like binary files, certain typical source control operations such as difference detection and merges are not available.

The ability to connect to Power BI datasets stored in Premium capacity, using the XMLA endpoint, allows tools that are currently used for Analysis Services development and deployment to be used for developing Power BI datasets as well. This capability is dependent on [read/write capability for the XMLA endpoint](#) (which is a Premium feature) offers another alternative for storing the definition of a dataset in source control.

[Deployment pipelines](#) are a new Premium capability which allow greater control for deploying content to development, test, and production workspaces in the Power BI service. At the time of this writing, deployment pipelines do not include source control or file versioning capabilities; they are focused entirely on the deployment process.

File Versioning Systems

Business users will usually be much more comfortable using a standard file sharing system like OneDrive or SharePoint. It's common for IT and BI teams to use this approach as well.

We strongly recommend storing .pbix files in a file sharing solution which saves historical copies of files. The number of versions retained is typically configurable. This allows access to the previous versions, which is particularly helpful if a change needs to be reverted. Recent advancements in OneDrive file size limits, and the efficiency of sync operations, make this a very seamless way to work.



Use consistent, unchanging file names

Some users like to implement their own file versioning with file names (ex: Inventory Stock Prices 2-23-2020). This technique of changing the file name does **not** work well in Power BI because data refresh schedules, pinned tiles, and related reports are dependent upon use of a consistent file name.

Separating Datasets from Reports

Creating a dataset requires a deep understanding of the data and how it should be modeled. Depending on the complexity level of data sources, calculations, data cleansing, and relationships, this requires some level of technical skill in addition to domain knowledge.

In many situations business users want to focus on report creation and may not have the skill, time, or desire to build a dataset for that report. If users create datasets associated with every report they build, this can result in the creation of many duplicate datasets which leads to the kind of issues discussed earlier in this section.



We recommend separating the development of the datasets used by reports from the reports themselves whenever possible.

Separation of datasets and reports can be achieved in a few ways:

1. Shared Power BI Dataset (a .pbix file with dataset only)

Create a dataset in a .pbix file without any analytical reports in that file.

Tip: instead of an empty report, you can use the report pages for dataset documentation, data quality testing, or a change log. This type of report would only be seen by dataset authors.

This .pbix file is published to the Power BI service which creates a new dataset. Report creators can create new reports in Power BI Desktop which connect to the published dataset. This creates

what is called a [Power BI service live connection](#), and behaves in a very similar way to creating a live connection to Analysis Services.

Alternatively, reports which use the published dataset can be created directly in the Power BI service using the web interface. However, as mentioned earlier in this section, we recommend using Power BI Desktop for more heavy-duty report development activities.

2. Use of Analysis Services

Another option is to use Azure Analysis Services or SQL Server Analysis Services as a semantic model for storage of data. The pros and cons of using a Live Connection to Analysis Services approach are discussed extensively in [Section 6](#) of this whitepaper.

3. Use of DirectQuery

Another option is to use DirectQuery. Conceptually, this is like the Live Connection options—however, the data is sourced from a relational database instead. The use of DirectQuery is covered in [Section 6](#) of this whitepaper.

The following graphic depicts of use of a shared dataset, as mentioned in option 1 above:

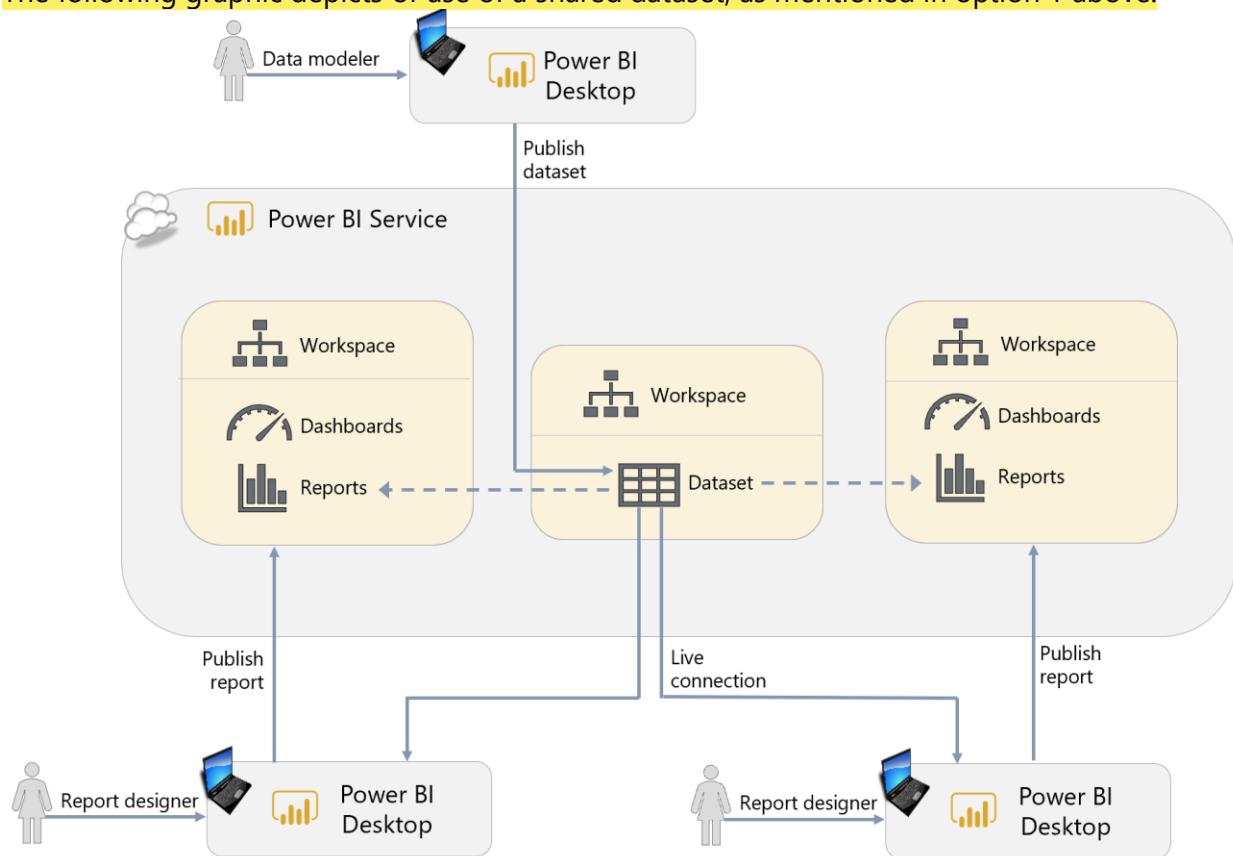


Figure 58 - Depiction of a shared dataset which is used by multiple separate reports

Shared datasets are intended for reusability across many reports and can be created by a centralized IT or BI team or perhaps by subject matter experts. When a dataset has significant value and has undergone validation, it can be marked as certified so that report creators know that dataset can be trusted. The diagram above shows the shared dataset in its own workspace; however, this technique can also be used effectively within one workspace. This makes it easier to add measures to a centralized dataset and prevent rework in numerous files.



We recommend you use both shared and certified datasets to the extent it is practical. Shared and certified datasets are discussed in more detail in [Section 9](#).

Using Row-Level Security for Securing Data

Use of [row-level security \(RLS\)](#) ensures the data in the dataset is secured based on each report consumer's job role and need-to-know. When RLS has been defined in a dataset, the same report can be distributed to multiple audiences which saves time and effort. With RLS defined on the underlying dataset, when a report is viewed by users in the USA, the report displays USA sales; when that same report is viewed by users in Germany it only shows sales for Germany.

Row-level security is defined in Power BI Desktop, and invoked in the Power BI service, for users who do **not** have edit permissions. This includes users with:

- Permission to view an app, or
- Viewer permissions in a workspace

Without RLS, if a user may access a dataset, they are permitted to see *all* the data in the dataset – not just the data shown in data visualizations.

We strongly recommend using RLS when appropriate rather than creating multiple versions of the same dataset or report.

Using Templates to Speed Up and Standardize Report Development

Building new reports from scratch can be time-consuming and enforcing standards on those reports can be extremely difficult. The use of Power BI templates (.pbix files) can speed up report development significantly and improve standardization across reports.

Instead of starting with an empty .pbix file, templates allow report creators to create a new file with:

- A corporate color scheme already in place (it is also possible to store a color scheme as a [Report Theme](#) which can be imported into Power BI Desktop).
- Corporate branding already applied to pages.
- Connections to commonly used data sources already created.
- [Parameterized queries](#) connected to a data source. This is particularly powerful because it allows your users to reuse queries that someone else has created yet customize the

way these queries behave. For example, you can have a parameterized query that applies a date filter to a large table so only rows that fall in each date range are imported. When a user opens a template containing the parameterized query, they are prompted to enter the date range for the data they wish to import.

- Commonly used DAX measures already created inside the dataset.
 - Sample visuals which employ data visualization and accessibility best practices.
-



Tips for delivering templates

Templates can be distributed multiple ways. One common way is within a documentation and resources area for Power BI content authors.

Another very useful way is to include template links directly within a report or app that you think users are likely to want to customize. This helps you manage the experience. It is also more secure because it prevents users from downloading data unnecessarily. Using this approach works well when decentralized authors are tasked with customizing a report, which was provided by a centralized BI team, to the needs of a local subsidiary, region, or department.

Using Power BI Data Source Files

To help report creators save time, connection information for commonly used data sources may be made available through [Power BI data source files](#) (.pbids file format). At this time, a .pbids file may contain only one data source connection.

A .pbids file is unlike a template: it does not contain table information. It merely contains connection string information to simplify connectivity for self-service dataset authors.

Minimizing the Amount of Data Loaded

When you are importing data into Power BI (rather than using DirectQuery or a live connection) it is extremely important to limit that data to only what is needed for the reports you are building. Two of the main ways to reduce the amount of data stored in a dataset include:

1. Removing columns from source tables in the Query Editor.
2. Filtering rows from the source tables down to just the subset of data which is required.
This can be done with hard-coded filters in the Query Editor or with query parameters.

Often the tables being accessed in the source will have a lot of data that seems as though it *might* be useful at some point in the future. Indeed, if you ask your users, they will probably tell you to import it just in case. However, this can have some serious negative implications with respect to performance.

First, be aware that importing rows and especially columns of unnecessary data will increase the size of the dataset. This is particularly true for columns with many unique values. There is an upper limit on the size of the dataset based on either the file size or capacity size. In many cases importing unnecessary data can contribute to a dataset approaching its size limits. Also, the performance of a report is partly related to the size of the dataset – and its memory consumption – so the larger the dataset the more likely the report will perform poorly.

Second, the more data – and again, especially the more columns of data – that is imported, the harder it will be for report creators and users to find the data that is most important. This usability issue makes report development slower and more frustrating, and it increases the chances of mistakes where the wrong column is used in a visual or a calculation. It may also be frustrating enough that report creators rely on old habits such as exporting data to Excel.

Third, too much data can lead to a fragile data refresh due to time-outs. This can lead to confusion and user dissatisfaction if user expectations for up-to-date data are not met.

It is easy to add columns back to a dataset, or change a filter to load more data, later. However, it is much harder to remove columns or add more restrictions to a filter after a report has been published.



Make a habit of minimizing dataset size

The best policy is to restrict the amount of data loaded as much as possible at an early stage. Add additional data, if it is necessary, rather than loading everything you possibly can initially.

Reducing Dataset Size

In addition to minimizing the amount of data loaded, there are other ways to reduce the size of a Power BI dataset. If you are using a data warehouse or similar data structure which is in a dimensional model (that is, a denormalized star schema with facts and dimensions), then a date dimension table is likely to be readily available. Power BI dataflows are another alternative for making a date table available to dataset authors.



Use a standard date table

We recommend working with a standard date table in your imported datasets. In this case, the date table should be marked as date. This disables the auto date/time feature of Power BI Desktop which automatically creates date tables in the background for [certain date columns in an imported dataset](#).

Additional techniques which can reduce dataset size include:

- Ensure a column is included in a dataset once, and only once (unless it's required for relationships).
- Reduce the number of distinct values in a column. One way to do this is to eliminate the time element from date/time columns if time is not relevant. Or, if time is required for analysis purposes, extract time into its own individual column separate from the date. This vastly reduces the number of possible combinations, thus reducing the dataset size and memory consumption.
- Avoid importing columns which contain values calculated from other columns in the same table.
- Use calculated measures rather than calculated columns when possible.

Naming of Tables, Columns, and Measures

For self-service report creators who build their own reports it is important that tables, columns, and measures have meaningful, human-readable names that are consistent with the terminology used throughout your organization. The objective for naming tables, columns, and measures is to avoid the misinterpretation of results. We recommend having naming conventions, including:

- **Clear, friendly names** which will display well on reports. Ex: "Product Name" looks far nicer on a report than ProdNm.
- **Clarity over brevity**. Ex: "Product Code" is far clearer for users than Product.
- **Prefixes for clarity**. Ex: if the dataset contains Product Name, Customer Name, and Sales Rep Name, they should all contain prefixes. This makes it clear which name is being used. Though the report creator can tell which name it is from the table it comes from, a report consumer does not have the same luxury if only "name" is displayed on the report. This also reduces the frequency of visual-level renaming.
- **Minimize use of acronyms and jargon**. If use of an acronym is required, a key or a tooltip can be of great help for others to understand the meaning.

A report creator always has the flexibility to update names which are displayed on a specific visual.



Expose a column once and only once in the dataset

If a column exists multiple times in a dataset (perhaps for relationships), it should be exposed for reporting only once. Allowing the same column to be visible in a model multiple times is confusing for anyone creating reports from that dataset, and it can lead to incorrect results if the wrong column is used in a visual.

The Description property for fields can be used as a data dictionary to help report authors understand the data model. This type of information can be invaluable.

Naming of tables, columns, and measures is also highly relevant if you have report consumers who want to build their own visualizations using Q&A, Power BI's natural language query feature. For Q&A to work well, however, you will need to add Synonyms to your model and [edit the linguistic schema](#) to improve Q&A's ability to interpret a user's requests.

Minimizing the Amount of Data Displayed on a Page

More is not necessarily better when it comes to report design – the more visuals you place on a page, and the more data displayed in these visuals, the slower the report will render. In addition to performance concerns, dense report pages also make it harder for report consumers to make sense of what is being shown. Reports which are slow and difficult to interpret are unpleasant to use and will discourage people from using solutions created with Power BI.

As the number of visuals on a page increases, so do the number of DAX queries requesting data from the dataset. Inevitably each query that is executed will increase the overall render time of the report.

The more data that these DAX queries request, the longer they are likely to take to execute, and the longer it will take for Power BI to display the results in a visual. In some cases, such as certain map visuals, there is a maximum number of data points that can be displayed in a visual (after which Power BI may begin using "[high density sampling](#)" techniques to ensure performance and interactivity of the visuals are maintained).



The performance of a report will depend significantly on storage of the dataset: imported, Analysis Services, or DirectQuery. This is discussed in more detail in [Section 6](#).

To maximize performance, aim to minimize both the amount of data displayed as well as the number of visuals displayed on a page.

Avoid "data dump"-style reports with tables containing hundreds of columns and thousands of rows, usually created when the user expects to export this data to Excel. If the user intends to export data to Excel, consider designing the report directly within Excel to provide the data required but with a live connection to the dataset.

Consider using slicers and filters in your reports to allow the user to select just the data that is needed to be displayed at any one time. We also recommend setting the slicer or filter at report publishing time, so the initial rendering of the report performs acceptably.

You may also consider using techniques such as [bookmarks](#) and [drillthrough pages](#) to reduce the amount of data displayed on a page. The drillthrough feature delays the rendering of detailed data until the user clicks on the link to drillthrough to an additional page (for instance, from a sales summary report to a sales detail report). In addition to improving performance, drillthrough can also simplify the higher-level reports thus improving the end user experience.

Focusing on Quality of Content

As more and more self-service report authors generate content for others to consume across the organization, that often exposes an opportunity to educate report authors on data literacy and data communication. A few suggestions include:

- Have a hyper-focus on usability of the report for end users.
- Have a specific purpose for each page, and for each chart on each page.
- Include clear interactions and instructions for report consumers.
- Have a process for verifying accuracy of data presented, and accuracy of measures with different filter selections.
- Use bookmarks to create 'help' information to guide users.
- Carefully review the data displayed to assess if it can be easily misinterpreted.
- Solicit feedback from consumers for both effective data presentation and data quality issues.

 One way to get feedback from report consumers is with a survey link available alongside the report or within an app. This allows report creators to get structured feedback on the quality of content, and understand how reports are, or are not, meeting user needs.

Focusing on Information Design Principles

The effective communication of information relies upon using sound information design principles. Learning about the Gestalt principles is highly recommended. A few information design tips include:

- Be intentional with selections for each visual. Focus on its ability to convey information, resisting the urge to use a variety of visual types for aesthetic appeal only.
- Use careful placement and a clear hierarchy for displaying information on the page.
- Use informative text so consumers can find subtle features like drill-through and/or find more information.
- Be consistent with placement of items which appear on multiple pages.
- Use a layout that focuses on telling the story you want to tell.
- Use color sparingly, and consistently, with a specific intention in mind.
- Use a color palette that follows visualization best practices. If you are required to follow a vibrant organizational color palette, investigate if you can mute the colors.
- Avoid clutter on the page to allow report consumers to focus on what is important.
- Address accessibility early in the design phase (discussed next).

We strongly recommend all report creators be educated in data visualization and information design practices. A few excellent resources include:

- [Storytelling With Data](#)
- [Alberto Cairo](#)
- [Stephen Few](#)

Verifying Accessibility on Reports

When designing reports, it is important to consider the accessibility needs of your users.



Designing for accessibility can be easy to overlook, but it should be a standard part of all report development activities.

Although there are some accessibility features [built in to Power BI](#), many aspects are in the hands of the report creator. For example:

- Using very **clear titles, labels, and descriptions** can be extremely helpful to aid comprehension by all report consumers, including those who are visually impaired.
- Adding **alternative text descriptions** (alt text) for visualizations in a report helps users who use screen readers understand what a visualization is showing. Alt text should be set for every object that conveys meaningful information.
- Choosing a **color theme** that has a high degree of contrast between backgrounds and text helps make reports easier to read, especially for users with vision impairment. There are many desktop apps and online sites to help with choosing colors with appropriate contrast ratios, including [Colour Contrast Analyser](#) and [WebAIM Contrast Checker](#). There are also helpful sites to simulate what colors look like to someone who experiences color deficiency, such as [Color Blindness Simulator](#) and [WhoCanUse](#).
- Using **visual indicators**, in addition to color, helps report consumers who are color-blind, and therefore unable to understand good and bad with color alone. Red and green can be particularly difficult to distinguish. This is important to keep in mind, especially when using conditional formatting.
- Setting the **tab order for visuals** assists visually impaired report viewers with how to go about viewing and processing the report in a logical order. Any images or objects which are purely decorative can be omitted to aid with this navigation.

For more tips on creating accessible reports see [this article](#), including the excellent [Report Accessibility Checklist](#).

Using Certified Custom Visuals

Custom visuals enable both third party developers, as well as the Microsoft Power BI team, to create many new visualization types for Power BI and enrich the available visualization stack. Some organizations also build their own custom visuals to meet their unique needs.

Independent and third party developers can list their custom visuals in [AppSource](#) to get wider public exposure and easier access to custom visuals they create.



Only use custom visuals from sources that you trust - preferably certified

While custom visuals provide rich ways for visualizing your data, they should be treated with caution. It is important to understand that custom visuals:

- Have access to the data in Power BI datasets, and
- Custom visuals are not blocked from sending data out of the Power BI service

Therefore, be certain to only use custom visuals from sources that you know and trust.

See [Section 10](#) for a discussion of governing the use of custom visuals and organizational visuals.

Custom visuals published to AppSource can optionally go through a validation process by Microsoft to become certified.

[Certification](#) reviews the quality and safety of a visual, but there is no guarantee that *all* code paths have been tested. Custom visuals downloaded from sources other than AppSource have not undergone any validation process by Microsoft and report authors should be very cautious when using them – make sure you trust the developer of the visual and test the visual before allowing it to be used widely in your organization. It is recommended to have a policy for using and testing all custom visuals in your organization.

As an alternative to allowing all custom visuals, or all certified custom visuals, Power BI administrators can create a repository of specific custom visuals that are approved for use within the organization.

Administrators have the following three capabilities for management of custom visuals which will affect the experience for report creators:

1. **Block all custom visuals.** This is done with a tenant setting in the Power BI service and a group policy setting.
2. **Allow certified visuals only.** This is done with a tenant setting in the Power BI service and a group policy setting.
3. **Deploy specific approved visuals** through the organizational visuals store. This is done in the admin center in the Power BI service.

See [Section 10](#) for additional information on managing custom visuals.

Using Custom Data Connectors

It is possible to extend the number of data sources that Power BI can connect to by creating custom data connectors. The same warnings that apply to using custom visuals (discussed in the preceding section) also apply to using custom data connectors created by third parties: they may not be properly tested or supported, and as a result you should have a policy for evaluating and testing the use of custom connectors.

Creating Dashboards to Unify Key Metrics from Various Reports

In Power BI, the word “dashboard” is a reference to a specific feature that exists within the Power BI service. Dashboards have a role to play in helping users focus on information that otherwise might be buried in reports.

Pinning visualizations to a dashboard allows report creators to provide an “at-a-glance” view of the most important metrics. A dashboard could be considered something like a start page or executive summary: if all looks well, the report consumer may not need to review the underlying reports, but if necessary the user can click on a visualization and navigate to the underlying report to see it in context.

From a consumer expectation perspective, be aware that dashboards are not designed to be interactive like reports. For instance, you cannot filter a visual from the dashboard page.

Other content, like videos, text, and web content may be easily added to dashboards to improve the user experience.

A dashboard can be created by pinning individual visualizations from multiple areas:

- Report visuals published to the Power BI service
- Q&A results in the Power BI service
- QuickInsights results in the Power BI service
- Visuals within reports published to Power BI Report Server
- Visuals within reports published to SQL Server Reporting Services
- Visuals within Excel workbooks published to the Power BI service



Usage of dashboards is how to provide a single view of data which spans multiple reports and/or datasets and/or tools.

Developing for the Power BI Service and Power BI Report Server

Different versions of Power BI Desktop exist for use with the Power BI service and Power BI Report Server. For users who publish content to **both** the Power BI service and Power BI Report Server, there are two ways of handling this:

1. Stay on the version of Power BI Desktop which is compatible for both the Power BI service and Power BI Report Server. New features are available to users approximately every 4 months in this scenario. This is the recommended approach because it's simplest for report authors to understand.
2. Run two versions of Power BI Desktop side by side. This is not typically recommended. Running two versions is only suggested for early adopters who are keen to obtain new features as quickly as possible and are willing to accept the additional complexity of managing two versions. If a .pbix file from the newer version is inadvertently published to Power BI Report Server, it may not be able to interpret the data model, refresh it, or render reports properly.

Section 9. Power BI Collaboration, Sharing and Distribution

Creating content (that is, datasets, reports, dashboards, workbooks, dataflows, apps) in Power BI is almost always a collaborative process, and content is usually intended to be used by more than just the person(s) who initially created it. Power BI offers a few different ways to collaborate, share and distribute content throughout an organization. It is important to understand which of these methods are appropriate for different scenarios.

The Role of Power BI Desktop

Reports built in Power BI Desktop are saved as .pbix files. It is very important to stress that .pbix files are **not** intended to be used as a method of sharing reports between multiple users in the way that Excel files often are, saved to a network file share or sent via email. There are several reasons for this:

- **Versioning** – There is always a risk that multiple copies of the same report will be created, which then means that changes and bug fixes may be made to one version and not another.
- **Stale data** – When data is imported into a .pbix file, the data in a dataset is refreshed manually in Power BI Desktop. This leads to a greater risk that the data in the report may be out of date.
- **Security** – Another effect of importing data into Power BI Desktop is that, since the .pbix file may contain data that is very sensitive, it must be kept in a secure location. Distributing .pbix files containing imported datasets may present a security risk and may not conform to your organization's security policies, especially as they change over time. Row-level security may be used with Power BI Desktop, when a live connection to the Power BI service to a dataset is used and for users who have view-only rather than edit permissions on the dataset. Within a Power BI Desktop file, using import mode the user can't edit. Otherwise, since row-level security is configured in Power BI Desktop, if the dataset is set to import data, the row-level security is not actually invoked.
- **File size** – Power BI Desktop files may become very large, frequently larger than the maximum allowed size for an email attachment and large enough to make them slow to open when stored on a network file share.
- **Accidental changes** – While Power BI Desktop does have the option to lock visualizations in a report to prevent them from being moved accidentally during design, this option is not preserved when the file is saved which means that users are likely to move visualizations around accidentally. There is no foolproof way to prevent users from making other changes to a report and saving them.
- **Printing** – Unlike the Power BI service or Power BI Report Server, Power BI Desktop does not have native support for printing reports at the time of this writing. An alternative does exist, however, to export to PDF and print from there.

- **Visibility and Governance** – If users form the habit of sharing through the Power BI service, IT administrators can use the audit logs to monitor sharing of data within the organization. Coupled with processes to intervene and educate users on best practices, this monitoring helps the organization safely and reliably use authoritative data to make decisions.

The role of Power BI Desktop focuses on authoring of content. All collaboration, sharing, and distribution of content in Power BI should be done via the Power BI service, Power BI Report Server, or via embedding content.



Multiple users may not edit a Power BI Desktop file concurrently.

Collaboration, Sharing, and Distribution in the Power BI Service

Workspaces in the Power BI Service

When a dataset author or report designer has finished developing and testing content created in Power BI Desktop, the .pbix file is published to a workspace in the Power BI service. There are two types of workspaces in the Power BI service:

My Workspace	Every Power BI user has a private area called "My Workspace" which is intended purely for personal use. My Workspace is auto-generated and may not be disabled.
Workspace	Workspaces are shared workspaces where multiple users can collaborate. Workspaces are typically created for a specific purpose and a specific audience.

Best Uses for My Workspace

Ideally, a user's My Workspace should only be used for:

- **Personal usage.** Content that will only ever be consumed by the individual user.
- **Temporary content.** Content that is only to be shared with a very small number of other users on a temporary basis.
- **Testing purposes.** The report designer may use My Workspace temporarily to see what a report looks like in a browser or on a mobile device before it is published to a workspace.

In all other cases, reports should be published to a workspace. Using My Workspace for anything other than the scenarios described above can be risky:

- If a user leaves an organization their My Workspace area may be deleted and you risk losing any content that has been published there.

- Content within My Workspace can only be managed and maintained by that one individual. If author goes on vacation or takes leave, you may be left scrambling to update content, fix issues, or refresh data.
 - Power BI apps are a key element of distributing content to the organization. You may not publish an app from My Workspace.
-



Discovering Over-Usage of My Workspace

The data in the Power BI activity log (audit log) should be used by system administrators to find situations where there is an abundance of sharing occurring from My Workspace. Look for content with large numbers of frequent viewers or viewers in senior leadership roles, since this indicates the content has become critical to the organization and should be moved to a workspace. This represents risk to the organization, so it's a situation that should be monitored. Content authors who are relying on sharing from My Workspace too heavily can be educated to instead use workspaces and apps for most situations.

Collaborating to Create Content in Workspaces

Power BI Desktop plus workspaces are where authors collaborate to create content. A key aspect of creating reports for an organization is that collaboration involves **a group of people taking an active part in the process of designing and testing content** rather than just being passive consumers of content. These people all know each other and are likely to work on the same team – with the term “team” defined in the loosest sense.

There are several ways that collaboration can happen:

- Multiple users can work together to build, test, and release a set of datasets, reports, dashboards, workbooks, and dataflows (collectively known as “content”).
- One user can take over the design of content that was started by someone else.
- One user may work on the design of an imported dataset, while other users to build reports from that dataset using Power BI live connections in Power BI Desktop.
- Users may need to perform data validations (either in the formal, IT development sense of the word, or perform an informal “sanity check”) for a report.
- Members of the same team, department, project, or even company will not just view content but discuss what it shows, act on the information shown, and suggest changes to existing content, and/or create new content as circumstances change.
- Stakeholders may need to review content during development. The team building the content may give view-only permissions to facilitate the review process without risking these users will change the content.

Importantly, workspaces can have multiple admins, so you can ensure there is redundancy in the management of the workspace. You can provide access to workspaces through security groups, distribution groups, or Office 365 groups. This enables easier access management as users enter or leave job roles and teams.



The importance of collaboration is another reason why workspaces are strongly favored for storage of content over My Workspace.

Determining the Scope of a Workspace

Different organizations will use workspaces in different ways depending on their requirements for collaboration. Though different departments may use workspaces slightly differently, it is important to have a clear and consistent policy on how workspaces should be used and what they contain.

The following are some ideas on how workspaces can be used:

Workspace Scope	Most Useful When	Usability
Per Department or Team Ex: Finance reports or Sales reports	Extreme simplicity in workspace design and permissions management in desired	Typically, this scope is too broad
Per Subject Area, Project, or Type of Analysis Ex: Quarter-end reports or Product launch reports	The subject area has a manageable number of reports -and- The permissions needed align well with the subject area	Usually the best choice, with a manageable number of reports and appropriate permissions boundaries 
Per Individual Report Ex: Sales summary report or Executive bonuses	Ultra-sensitive data exists which is purposefully segregated -or- A specific app is needed which drives the workspace design	Typically, this scope is too narrow
Shared Datasets	The datasets are critical to the organization -and- Multiple authors in multiple departments need to securely access this data	A best practice for larger datasets and using row-level security 

Per Department or Team [Row 1 in the table above]

Create one workspace for each department or team in your organization. All reports and dashboards needed by the people that work in that department will reside in the workspace. This technique is so broad that it's not recommended for teams with a large amount of content and/or large numbers of users.

- ✓ Pro: Makes it easy for users to find all the reports they need to access if they are all contained in a single workspace which serves as a report gallery.
- ✓ Pro: Controlling which users see which reports is easy if security is also defined along departmental lines.
- ✗ Con: May result in workspaces which contain many reports and dashboards, making it more difficult for users to locate specific items.
- ✗ Con: Since there is a 1:1 relationship between a workspace and a Power BI app, a broadly defined workspace will result in apps for users which have a lot of content to navigate. This can be mitigated with the use of menus in the app.
- ✗ Con: If users need to see reports and dashboards from departments they do not work in, then managing permissions per object will become more complex. There's the risk that sharing of individual objects will become overused to compensate.
- ✗ Con: If some content authors need permission to edit some content, but not all, that cannot be accomplished since edit permissions are defined at the workspace level.

Per Subject Area, Project, or Type of Analysis [Row 2 in the table above]

Create one workspace for each subject area, project, type of analysis, or business question that your organization needs to answer. All the associated content resides in that workspace.

Typically, this is thought of as being a manageable compromise between the other two options.

- ✓ Pro: Makes it easy to find reports and dashboards across departmental boundaries.
- ✓ Pro: A good compromise between workspaces that contain too much information and workspaces that contain too little.
- ✓ Pro: A good compromise with respect to managing permissions for content editors.
- ✗ Con: Depending on how broadly it's defined, can still potentially lead to many workspaces being created, making finding the right one difficult.

Per Individual Report [Row 3 in the table above]

Create one workspace for each report. This is not recommended except for very specific circumstances.

- ✓ Pro: Makes it easy to identify individual reports and dashboards, assuming that they are named well.
- ✓ Pro: Useful for scenarios such as ultra-sensitive content which is purposefully segregated.
- ✓ Pro: Permits fine-grained permissions related to editing of content (for instance, if a user is permitted to edit one report but not another).
- ✗ Con: If overused, will result in an exceedingly large number of workspaces being created, making finding the right one extremely difficult.

Shared Datasets [Row 4 in the table above]

Create workspaces for datasets (and possibly dataflows) for delivery of data to meet organizational/departmental needs.

- ✓ Pro: Enables the organization to establish certified and promoted datasets so authors can easily locate data they can trust.
- ✓ Pro: Centralizes access management to critical organizational data.
- ✓ Pro: Limits who can edit and manage datasets, ensuring no unintended changes occur to a critical dataset.
- ✓ Pro: Allows enforcement of row-level security, since most authors will work in different workspaces and will not have edit permission to the dataset.
- ✓ Pro: Streamlines management of Power BI Premium.
- ✗ Con: May require change management processes if report authors have requirements beyond what can be handled by composite models and report measures.

Separating Data Workspaces from Reporting Workspaces

Shared datasets, which are reused across various reports, can reside in separate workspaces from the reports if desired. This ability to use a shared dataset in a different workspace was introduced with the new workspace experience.

The following image depicts the use of a "data workspace" in the center, and two "reporting workspaces" on the left and right:

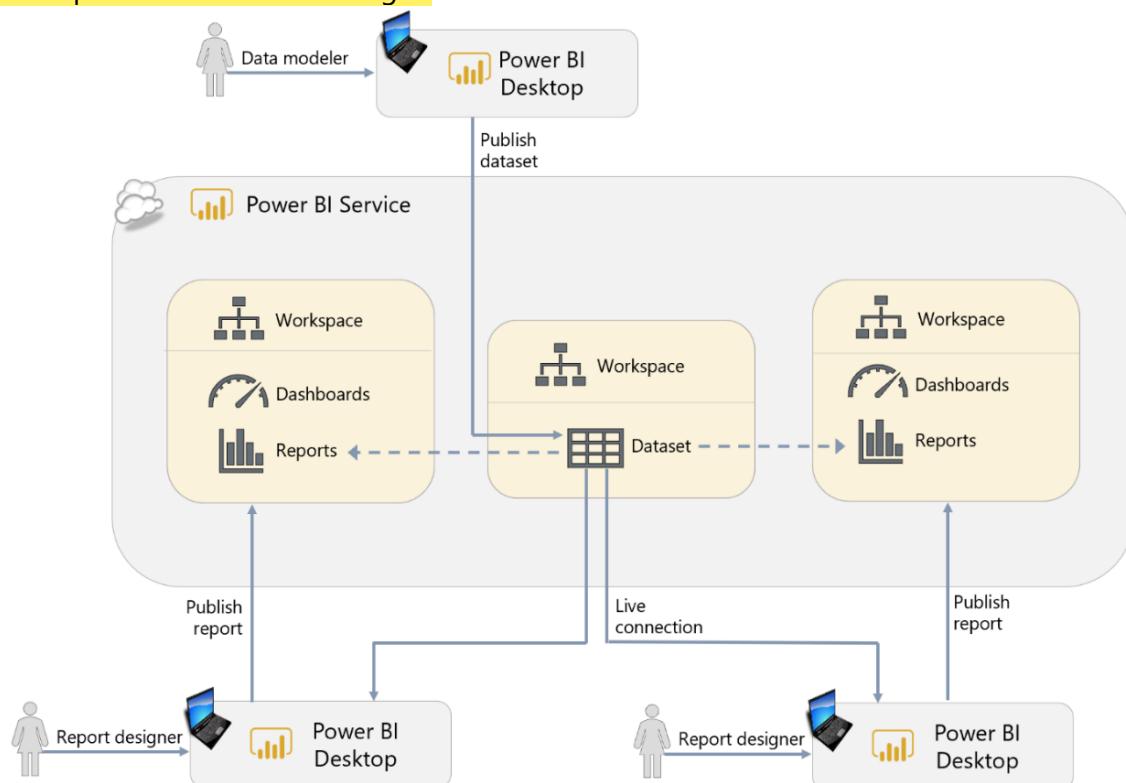


Figure 59 - Depiction of one shared dataset being used by multiple reports

Separating data workspaces from reporting workspaces has the following advantages:

- ✓ **Permissions per workspace can be defined separately.** This is very useful when there are many report creators and very few dataset developers.
- ✓ **Facilitates managed self-service BI and clarifies co-ownership scenarios.** When shared datasets are delivered from a centralized BI or IT team, while reports are published by functional business units, it becomes very helpful to segregate the datasets. This avoids the ambiguity of co-ownership scenarios because ownership and responsibility per workspace is more clearly defined.
- ✓ **Reduces risk that datasets are modified unintentionally.** The physical separation reduces the chances that the dataset is changed inadvertently. This additional layer of protection is particularly helpful for certified datasets, which are typically published by a centralized BI team and have additional expectations as to quality and trustworthiness.
- ✓ **Granular permissions management.** Workspace permissions are not a requirement to connect to the dataset. Rather, this can be handled through the dataset Read and Build permission. This offers flexibility to handle security very granularly if desired.

Naming a Workspace

When a workspace is created, various properties need to be specified which are more significant than they may first seem. One such property is the name of the workspace: it is important to choose a name that accurately reflects its contents.

It may be helpful to use a standard prefix to sort certain workspaces together, such as:

Workspace Name	Purpose
Financial Month-End	Reconciliation and month-end closing reports. Considered more of an informal, collaborative, workspace. An app is not necessary, as this content is used by a small group of people who work closely together. Most finance team members are likely to have permission to edit content in this workspace.
Financial Reporting	Final, presentation-level reports. Considered more of a more formal workspace with content distributed more broadly to the organization via an app. Publishing content here likely involves an approval process and is more tightly controlled.

Do keep in mind that longer workspace names do appear truncated in the user interface. Therefore, do choose wisely.

It is ok to rename a workspace if you need to refine the naming strategy. The GUID assigned (which you can see in the URL) does not change when the workspace is renamed.

When using separate workspaces for Dev/Test/Prod, those should typically be included as a suffix.

The New Workspace Experience

Power BI is in process of fully converting to the new workspace experience designed to simplify workspace creation, make managing workspace membership easier, and lay the foundation for other new capabilities.

The original workspace experience was tightly coupled with Office 365 groups, whereas the new workspace experience is not. Additional details about the differences can be found [here](#).

What to expect during the transition to 100% use of new workspaces:

If the workspace is created within the Power BI service:	<p>Workspaces created directly in the Power BI service use the new workspace experience, unless the creator specifically chooses to "revert to classic."</p> <p>A classic workspace is auto-generated in the Power BI service anytime a new Office 365 group is created.</p> <p>This is a known concern among customers and cannot currently be disabled. However, the workspace should <u>not</u> be deleted because it also deletes the Office 365 group the workspace is associated with.</p> <p>The workspace can safely be upgraded to the new workspace experience in order to use the newer features. After upgrading, the workspace is no longer tied to the Office 365 group. This means:</p> <ul style="list-style-type: none">• If you want to delete the workspace: you can delete the Power BI workspace without deleting or impacting the Office 365 group.• If you want to keep using the workspace: review the workspace settings for workspace roles, contact list, and the workspace OneDrive. These properties will still be associated with the Office 365 group after the upgrade. <p>The ability to hide or prevent creation of classic workspaces has been announced.</p>
If the workspace is auto-generated from an Office 365 service:	

You should start planning how to migrate your organization from classic workspaces to the new workspace experience. You will need to ensure management and governance processes are updated. In some cases, automation scripts to create or manage workspaces will need to be revised. The workspace roles have become more granular, which requires a review and often some changes. The Power BI Admin APIs and UI offer options for Power BI administrators to manage and govern the new workspace experience, something they would need to have collaborated with an Office 365 administrator for with classic workspaces.

Below is what you can expect to encounter in your Power BI deployment with respect to the new workspace experience.

Managing Workspace Role Memberships

The following includes recommendations when specifying settings and permissions for a workspace using the new workspace experience.

Visibility	<p>All workspaces are considered private. However, it is possible to allow broad access to the workspace by granting permissions to groups rather than individual users.</p> <p>The use of groups over individuals for access management whenever possible is strongly recommended.</p>
Workspace Roles	<p>There are four sets of workspace roles: admin, member, contributor and viewer. These workspace roles specify which users can edit content, and which users can only read content within an individual workspace.</p> <p>Note that row-level security defined for a dataset in the workspace only effects for users with viewer permissions (which could be obtained via the workspace, an app, or sharing).</p>
User and Group Membership	<p>Workspace role membership may be managed using the many different types of user groups that are already in use throughout an organization, including:</p> <ul style="list-style-type: none">• Security groups• Mail-enabled security groups• Distribution lists• Office 365 groups• Individual users <p>In the new workspace experience, the addition of a user to a workspace does not impact the underlying Office 365 Group membership.</p>

Content Distribution in the Power BI Service

Adding users to a workspace is one way to make the content in that workspace available to users throughout your organization. There are other options available as well. While workspaces are primarily intended as a place for **collaboration** within a team, reports and dashboards may also be **shared** with other users outside workspace; finally, content may be **distributed** by publishing apps from workspaces.

While we discussed “collaboration for authoring” content in Power BI above, it is useful to understand the meaning of the terms “sharing” and “distribution” in the context of Power BI as well:

Sharing	Sharing is typically thought of as an act where one user gives another user, or group of users, access to a specific piece of content. The sharing feature in the Power BI service is scoped to one dashboard or one report. Sharing most commonly takes place between individuals who know each other and work closely together.
Distribution	Distribution can be thought of as an act where content is delivered to others. This is often to a larger number of users, across multiple teams, who may not have explicitly requested the content but need it as part of their job. The recipients who consume distributed content may or may not know the original creator of the content. As such, distribution is usually considered a more formal mechanism within the organization.

Sharing refers to a feature in Power BI, and thus the term should be used carefully when communicating with others. Sometimes we are referring literally to the specific feature; other times it is used more generally.

Distribution via Apps

Content in a workspace may be distributed to a large number of users who are not members of that workspace by [publishing an app](#). An app is a read-only set of reports, dashboards, and workbooks which originate from the same workspace.

In a workspace, those with Administrator or Member permissions can publish and update an app (whereas Contributors and Viewers may not). For the purposes of this section, we will refer to this administrator or member with edit rights as the “app publisher.”

The process of creating and publishing an app is as follows:

1. A workspace app publisher chooses **which objects are to be included in the app** by utilizing the toggle for dashboards, reports and workbooks in the workspace. Datasets do not need to be specifically enabled in this way (they’re automatically added to the app based on the selected dashboards and reports). All content should be in a finalized and tested state.
2. On the **setup page**, the content publisher specifies:
 - o A **name for the app**. This is independent of the workspace name; we recommend they be very similar. However, they do not need to be identical.
 - o A **description for the app**. The description is required and should include relevant information about purpose and intended audience. This information appears very prominently on the apps page.
 - o A **support site**. This is optional, but highly recommended. The URL could be something like a SharePoint page which contains additional documentation. Or it

could be a link to a Microsoft Teams channel where app users can ask questions. Providing a support site helps you give better support experiences for larger audiences.

- A **logo for the app**. This is optional but recommended. If you are using logos for workspaces as an additional way to signify ownership or a functional business unit, this logo should be consistently used.
 - A **color for the app**. This is optional and includes a simple predefined set of colors to select. It defaults to the custom color selected in the Admin Portal. Note that this color is completely independent of the look & feel within reports.
3. On the **Navigation page**, the content publisher specifies:
- **Sections** of the navigation bar, order to organize the content in the workspace.
 - The **order** that reports and dashboards are presented within each section.
 - **Links** to additional information and resources.
 - **The names of items in the navigation** can be customized so they're easier for consumers to understand. This allows naming conventions in the workspace to support the needs of authors, but naming conventions in the app to focus more on the needs of content consumers.
 - Additionally, you can opt to **turn off the navigation** if you'd prefer to have a single item app or to show a list of content.
4. On the **Permissions page**, the content publisher specifies:
- If the app will be available to the **entire organization**. This option is enabled for app publishers who have been granted "Publish content packs and apps to the entire organization" in the tenant settings. Most content publishers are unable to select this, as that setting is typically reserved for centralized BI teams.
 - **If the app will be available to specific users and/or groups**. It is strongly recommended to use groups for this purpose so that an app does not need to be republished every time a new person needs permissions – this introduces additional unnecessary risk. All users with workspace permissions automatically see the app as well.
 - If the app users will be **assigned the 'build' permission to the dataset**. The [build permission](#) is a powerful setting which allows users to build their own reports and dashboards from the underlying dataset.
 - If the app users will be allowed to **make a copy of reports from the app**. This is another powerful setting which acts like 'save as' functionality. This feature is discussed further later in this section.
 - If the app users will be permitted to **re-share the app and the underlying dataset(s)**.
 - **If the app will be installed automatically**. This option is enabled for app publishers who have been granted "Push apps to end users" in the tenant settings. Most app publishers are unable to select this, as that setting is typically reserved for centralized BI teams. When an app is pushed to end users, it will automatically show up in their apps menu – if far too many items show up unexpectedly that leads to an unsatisfactory user experience. However, it is

extremely useful functionality in certain situations such as when an app is being published to a large group of people.

5. Finally, the app publisher **publishes the app**. When this happens, the contents of the workspace that were included in step 1 of this process are duplicated and packaged up in the app. Any new changes to the dashboards, reports, and workbooks in the source workspace do not affect the app until the app is updated. There is one notable exception, however, and that is datasets: changes to datasets always occur immediately when they're changed in the workspace.
-



App navigation links can significantly improve the user experience

Links in the App Navigation bar work particularly well for providing access to resources such as:

- KPI metric definitions
- Data definitions
- Location where data can be updated
- Related reports
- Training videos
- Feedback forms

Links can also be embedded in the content area, if the page linked to allows embedding. This can enable hybrid experiences, like giving access to on-premises systems. A link in the navigation bar is more prominently displayed for consumers of the app than the support site (which is defined on the Setup page).

There are two primary ways to view app content in the Power BI service:

1. **Using the apps page.**
 - If the app has been installed automatically on the user's behalf: the user will find the app present in the list of apps.
 - Otherwise, the user will need to install the app manually. They do so by clicking the Get Apps button. Under "My organization" the user will see any apps to which they have been assigned permissions. This is where a good app name and description are very useful, particularly when there are quite a few in the list.
2. **URL links** may be used as an alternative access method. The URLs can be communicated in email, documentation, specified in an enterprise portal, displayed on an internal web page, or even in another application. These types of links can even be included on a dashboard page, which can be another helpful way to make content available. After a user accesses the link, the app is shown in their apps page and on their home page.

After publishing an app, unique links are available for:

- The app itself
- Each dashboard published in the app
- Each report published in the app

The link is constructed to include GUIDs specific to each object (app, dashboard, or report). A URL to the app is constructed as follows:

<https://app.powerbi.com/Redirect?action=OpenApp&appId=AppGUID&ctid=TenantGUID>

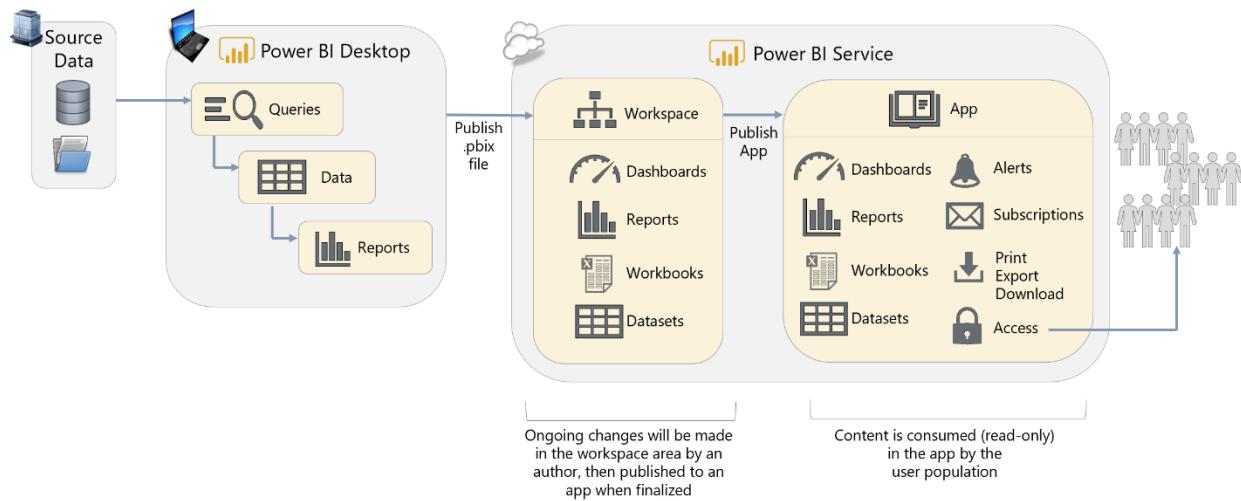


Figure 60 - Diagram of use of an app separate from the workspace for content distribution

The ability to push apps automatically to end users is disabled by default in the Power BI Admin Portal to avoid users being overwhelmed with new apps installed automatically for them by overzealous content publishers.

Admin portal

Usage metrics	
Users	
Audit logs	
Tenant settings	
Capacity settings	
Embed Codes	
Organizational visuals	
Dataflow settings	
Workspaces	
Custom branding	
Protection metrics (preview)	
Featured content	

🛡️ Settings enabled for the entire organization apply to all users and service principals

◀ **Push apps to end users** Enabled for a subset of the organization

Users can share apps directly with end users without requiring installation from AppSource.

Enabled

Apply to:

The entire organization

Specific security groups

Power BI Administrators × Power BI Content Admins - Corporate ×

Enter security groups

Except specific security groups

Apply Cancel

Figure 61 - The 'push apps to end users' tenant setting in the Power BI admin portal

This functionality should be enabled for content publishers who are familiar with how to use it judiciously. The ability to push apps does significantly increase the ability for consumers to easily find content, because it removes the necessity for them to go and locate it in AppSource.

Distribution of content using apps is, in many cases, a better choice than sharing, or adding large numbers of users as viewers of a workspace due to:

- ✓ **Navigation Menu.** The ability to define the sections, order, and links of the app navigation menu makes for a better report consumer experience than navigating the workspace directly.
- ✓ **Simpler Permissions Management.** An app allows for the distribution of multiple related reports, dashboards, and Excel workbooks rather than individual pieces of content (as is the case with the sharing feature), simplifying permissions management.
- ✓ **Specify Objects to be Included in app.** An app makes available only the objects which have been included in the app via the toggle. If there are additional objects in the workspace that report consumers do not need to see, they won't be seen by users who only possess app permissions. This is an advantage of apps having their own permissions management independently of the underlying workspace.
- ✓ **Control Timing of Updates.** Changes to content are only visible once the app is published or updated. This allows report and dashboard content to continue to be developed without affecting what is seen in the app until an update occurs (except for datasets—all changes to datasets always take place immediately).
- ✓ **Use of Power BI Free License.** If a workspace resides in Premium capacity (P SKU), users assigned via app permissions only need a Power BI Free license to view content. This is also true for the Viewer role for the workspace itself (for situations when a separate app is not desirable).

Distribution via Workspaces

One of the [roles available in the new workspace experience](#) is the viewer role. The previous section discussed the use of apps for distributing content – however, there are times that the additional layer of security and deployment is not desired. This is especially true for small teams who work closely together, and do not feel they need to use an app.



If a workspace resides in Premium capacity, workspace viewers only need a Power BI Free license to view content.

If row-level security is configured for a dataset, it does take effect for users who are assigned the viewer role for a workspace. Conversely, row-level security is ignored for users with edit permissions which includes the admin, member, and contributor roles.

To use the viewer role most effectively, it is useful to:

- Organize content well for each workspace so it is easily located by report consumers.
- Name reports and dashboards clearly and consistently so the right content can be easily identified. This becomes a bigger issue when there are more items published to a workspace.

- Consider separating development and test content from production so that in-progress work is not viewable by report consumers. Using a naming convention helps achieve this if you do not desire to manage separate workspaces.
- Be aware that all content published to the workspace is viewable by users with the Viewer role.

Sharing Dashboards and Reports

Both reports and dashboards may be [shared](#) – that is, you can provide a user or group read-only access to individual reports and dashboards with the Share button in the Power BI service. When a user shares a dashboard with other users, the supporting reports, workbooks, and datasets are viewable as well.



Sharing is considered an exception to security defined for Workspaces and Apps

Keep in mind that sharing for individual objects is done completely separately from workspace security and app security. This level of control and flexibility for defining read-only access to individual reports and dashboards (via sharing) is warranted sometimes. However, content authors should be educated to use the sharing functionality as the exception rather than the rule. Since sharing is defined per-object, it becomes more cumbersome to manage, and more difficult to audit security.

When sharing, the content author must provide a list of the individual users, distribution lists, or security groups to share the report or dashboard with. The individual users may be inside or outside your organization.

Sharing also includes the following options:

- Adding a message describing what has been shared.
- Allow recipients to share your report. This is known as the **re-share permission** and allows users with whom you have shared the content to share it with other people in turn. This is pre-selected by default, and in many cases should be de-selected.
- Allow users to build new content using the underlying datasets. This is known as the **build permission**, and it edits the permissions for the underlying dataset. The build permission allows report creators to use the underlying dataset for creating new reports. This is pre-selected by default. This allows users to use features like Analyze in Excel or export underlying data to do further ad hoc analysis based on the dataset to achieve their needs. It also enables users to discover and use the dataset to create a new report. Reusing data in this way is highly encouraged.
- Sending an email notification to individual users (though not groups).

Users inside your organization who are recipients of shared content will find the content in the "Shared with me" area of the Power BI service. External users, however, will not see shared

content in the same "Shared with me" area of their home tenant. External users will need to access the content via a URL such as:

<https://app.powerbi.com/sharedwithme?ctid=GUID&openReportSource=ReportInvitation>

The ctid parameter relates to the tenant where the original content resides. The ctid is included in sharing notifications as well. If it needs to be provided again to an external user, it can be found under Help > About Power BI:



Figure 62 - The ctid is available in the Tenant URL under the Help > About menu

Sharing of reports and dashboards is most appropriate for informal scenarios where content is to be shared with a small number of people. It is a good idea to educate users to investigate workspaces and apps, rather than sharing, when attempting to distribute content to large numbers of users or people outside your team. This is because:

- ✓ It becomes more difficult to determine which content has been shared with which users, since the permissions on each report and dashboard must be verified individually. In this way, sharing of individual items can be thought of somewhat as an exception to workspace security.
- ✓ By default, users that have a dashboard or report shared with them can re-share the dashboard and report with other users. This presents the risk of content being shared with users that the original owner did not intend to share it with. This can be prevented by unchecking the "Allow recipients to share your dashboard/report" option when sharing—but the sharer needs to remember this every time.
- ✓ Changes to reports and dashboards are viewable to others immediately, which may result in confusion for users if the content is a work-in-progress.
- ✓ When content has been shared from a user's My Workspace and that user leaves the organization, when the user's account becomes disabled the recipients will lose access to the content.



Recap of terms discussed in this section

Collaboration for Authors is associated to multi-person content creation and testing within Power BI Desktop and a workspace.

Sharing is usually referring to the specific feature for sharing individual reports and dashboards. However, it is frequently used more broadly as well.

Distribution is associated with the ability to publish content more broadly. This is frequently handled via apps.

Decision Tree for Using Apps, Workspace, or Sharing for Distribution

Sometimes it can be confusing for content authors to decide whether to distribute content via apps, workspace, or the sharing functionality. Following is a decision tree which should be used as a rule of thumb:

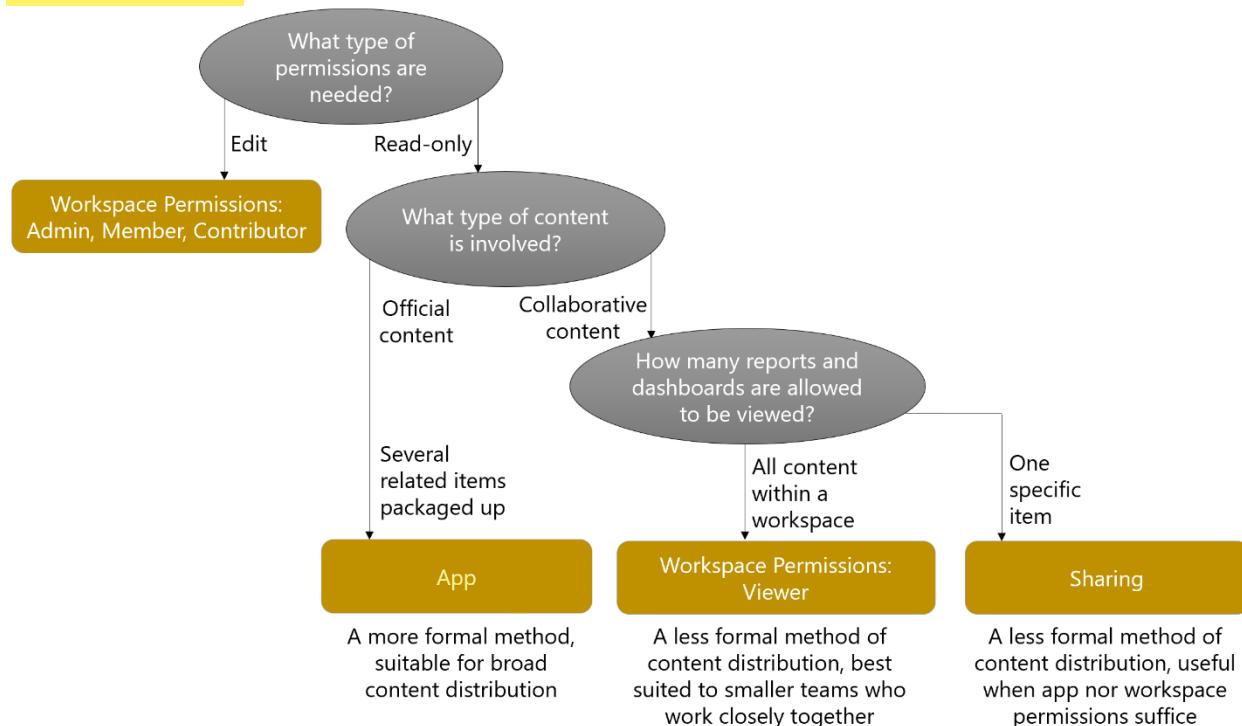


Figure 63 - Decision tree for using workspace, app, or sharing

Allowing Users to Copy and Customize Existing Reports

With a bit of planning, the Power BI service can be used effectively for corporate BI purposes, as well as self-service BI purposes wherein analysts create and publish their own reports. There are many ways to facilitate self-service BI (such as the use of certified datasets). One such way is to make purposeful choices related to how report creators will be able to create reports.

It can be a significant time-saver for report creators to be able to customize existing reports. This functionality is referred to as "[Copy a Report](#)" in the Power BI service.

The Copy a Report functionality does not duplicate a dataset. Rather, it creates a live connection back to the original dataset, while allowing the report author to assume control of the new report.

 Be very careful about allowing the Copy a Report functionality when report branding is in use. For instance, there may be a logo or label in a report footer which indicates who published the report, or the level of report validation. These types of trustworthiness indicators are no longer useful if self-service users are permitted to copy and edit reports with that type of branding.

Shared and Certified Datasets and Dataflows

[Shared and certified datasets](#) are two topics which are often discussed together because they are **highly critical to the success of a self-service BI implementation**.

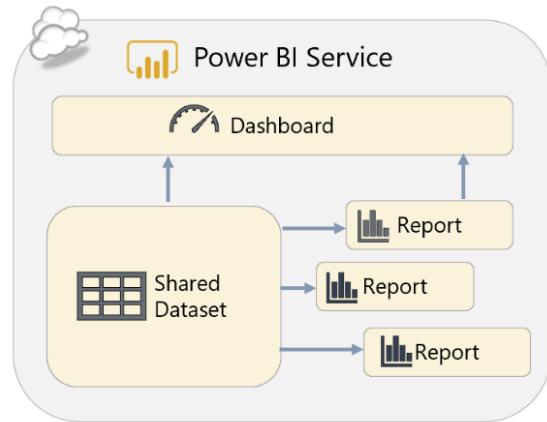


Figure 64 - Diagram showing one shared dataset being used by multiple reports

Power BI dataflows may be certified as well, as discussed in [this announcement](#). Although the discussion in this section focuses on datasets, most of the concepts apply to dataflows as well.

Following are a few definitions related to datasets specifically:

Shared Dataset	A dataset which is reused across many different reports.
Live Connection	The method for connecting to a shared dataset from Power BI Desktop.
Dataset Endorsement	A property assigned to a dataset. Possible values include Certified and Promoted . Dataset endorsements improve the ability for report creators to discover datasets.
Certified Dataset	A dataset endorsement which indicates the dataset has been certified by a subject matter expert. The intent is to convey that the dataset has been validated and is trustworthy. Only users who have been assigned permission in the tenant settings may certify a dataset. Certified datasets are typically reserved for use by centralized BI teams.
Promoted Dataset	A dataset endorsement which makes a dataset more discoverable. Any dataset author may set the dataset to be promoted, so its meaning does not convey as much rigor as a certified dataset. Promoted datasets are typically reserved for departmental use.

Advantages and Disadvantages of Shared Datasets

Use of shared datasets is highly encouraged because it decouples the dataset from report development.

There are many advantages to using shared datasets, including:

- ✓ Use of a single authoritative data source per subject area or topic
- ✓ Reduced level of data duplication and redundant datasets, leading to fewer compliance and/or security concerns
- ✓ Easier for report creators to know what dataset to use, and improved time efficiency
- ✓ Fewer data refresh operations, which in turn reduces the burden of data refresh queries sent to source systems
- ✓ Reduced level of maintenance needed across numerous datasets, and fewer issues with consistency across datasets
- ✓ Row-level security rules are maintained in fewer datasets, leading to fewer inconsistencies and/or security concerns
- ✓ Improved ability for multi-user teams to collaborate after decoupling the dataset from report creation

Depending on how your team works, there could be disadvantages to shared datasets which include:

- ✗ A lot of dependencies on a single shared dataset reduces its flexibility for change in the future
- ✗ Increased risk of introducing breaking changes, depending on what changes are introduced
- ✗ Responsibility/ownership of a centralized dataset may be with another team, leading to a situation where changes you propose are not approved
- ✗ The ability to introduce changes may be slower, particularly if a mission-critical shared dataset is required to go through formal change management
- ✗ The potential for introducing a performance bottleneck if a significant number of people unexpectedly build reports on a shared dataset which is not well-designed

Workflow for Using Shared Datasets

Following is the workflow for using a shared dataset in conjunction with a dataset endorsement:

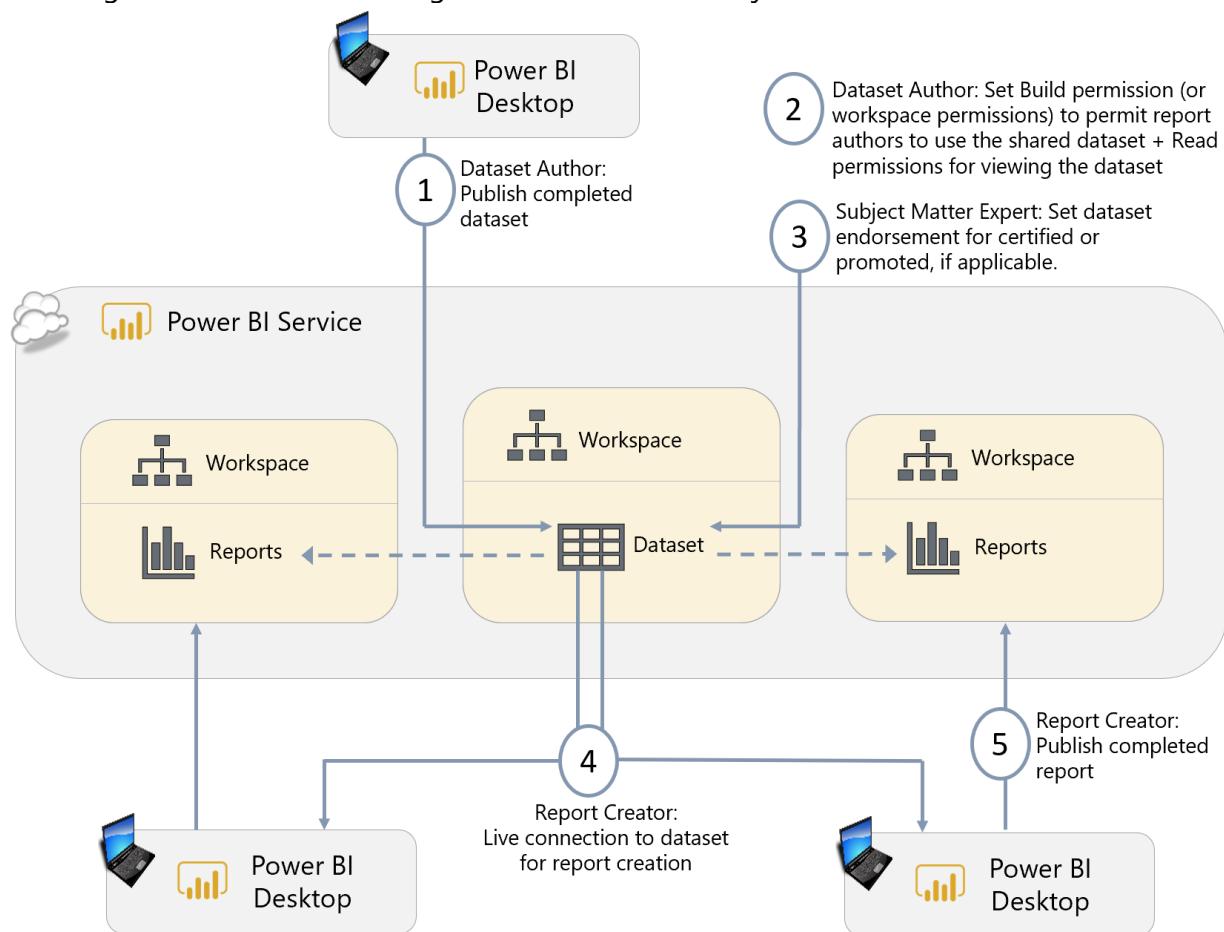


Figure 65 - Diagram of the workflow for using a shared dataset

1. **Publish completed dataset.** The dataset author completes the data model, relationships, and calculations in Power BI Desktop and publishes the dataset to the Power BI service. In this case, the dataset is typically the only thing in the PBIX file.
2. **Set build permission.** The dataset author sets the build permission for the dataset to allow report creators to connect to the dataset for their reports. The build permission is inherently read-only and applies to the individual dataset object. Alternatively, workspace permissions can be used.
3. **Set dataset endorsement (optional).** A subject matter expert performs the validations and sets the dataset endorsement to be certified or promoted, if applicable. For certified datasets, it is preferable that this person is separate from the original dataset author.
4. **Create reports from shared dataset.** Other report authors use the existing dataset, utilizing Live Connection, for creation of their reports.
5. **Publish reports which use shared dataset.** Report authors publish their completed reports to the Power BI service. The reports may reside in the same workspace, or in different workspaces, depending on the workspace strategy being employed.

Lineage View

There are two use cases addressed by the [lineage view](#):

Impact Analysis (Downstream Analysis)	Starting with the dataset: view the downstream lineage for a dataset across workspaces to do impact analysis for potential changes to a dataset.
Lineage (Upstream Analysis)	Starting with a report or dashboard: view the upstream external dataset and navigate to the source workspace to see the source lineage .

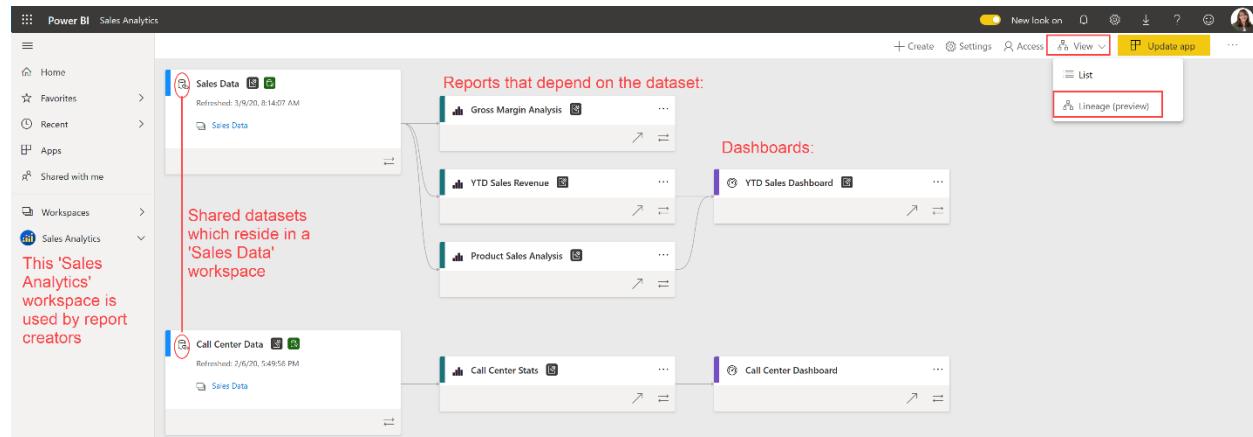
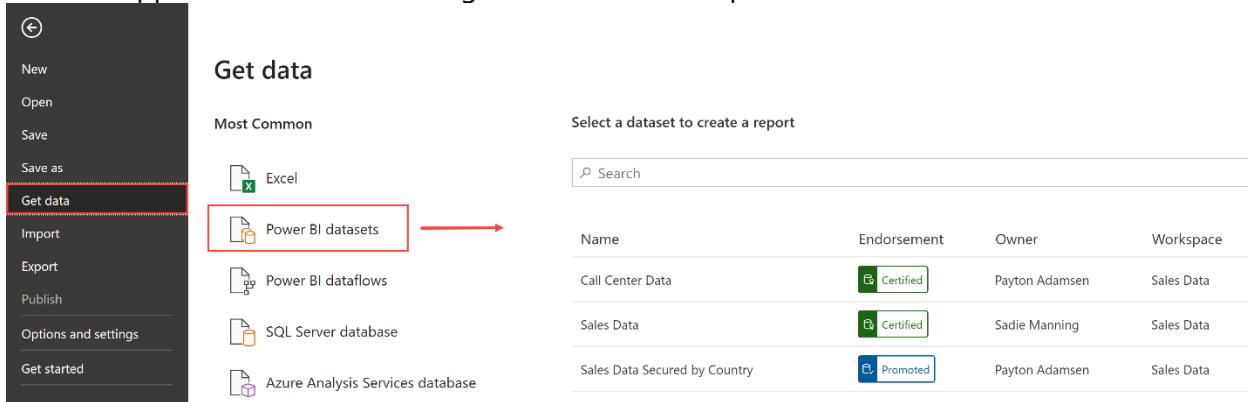


Figure 66 - Lineage view in the Power BI service

Dataset Discovery Experience for Certified and Promoted Datasets

When a dataset is certified or promoted, it becomes more discoverable because endorsed datasets appear first when searching in Power BI Desktop or the Power BI service:



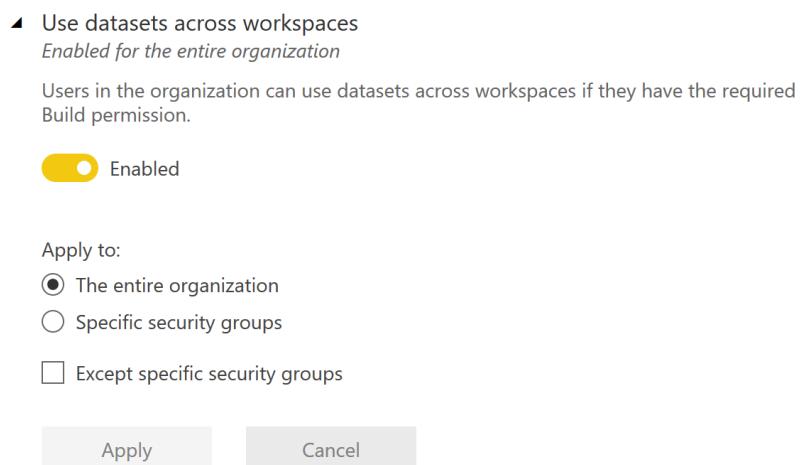
The screenshot shows the 'Get data' pane in Power BI Desktop. On the left, a sidebar lists options: New, Open, Save, Save as, **Get data** (which is selected and highlighted with a red box), Import, Export, Publish, Options and settings, and Get started. The main area is titled 'Get data' and shows 'Most Common' data sources: Excel, Power BI datasets (which is selected and highlighted with a red box), Power BI dataflows, SQL Server database, and Azure Analysis Services database. To the right, a table titled 'Select a dataset to create a report' lists datasets: Call Center Data (Certified), Sales Data (Certified), and Sales Data Secured by Country (Promoted). A red arrow points from the 'Power BI datasets' option in the sidebar to the 'Power BI datasets' option in the list.

Name	Endorsement	Owner	Workspace
Call Center Data	Certified	Payton Adamsen	Sales Data
Sales Data	Certified	Sadie Manning	Sales Data
Sales Data Secured by Country	Promoted	Payton Adamsen	Sales Data

Figure 67 - Get Data pane in Power BI Desktop which shows certified and promoted datasets first

For this capability to work, it requires the following tenant setting to be enabled:

The 'Use datasets across workspaces' setting impacts users in several ways. Refer to [this page](#) for the effect on report creators when this setting is disabled.



The screenshot shows the 'Use datasets across workspaces' setting in the Power BI admin portal. It includes a description, an enabled toggle switch, a 'Apply to:' section with radio buttons, and 'Apply' and 'Cancel' buttons.

◀ Use datasets across workspaces
Enabled for the entire organization

Users in the organization can use datasets across workspaces if they have the required Build permission.

Enabled

Apply to:

The entire organization

Specific security groups

Except specific security groups

Apply Cancel

Figure 68 - Tenant setting in the Power BI admin portal to use datasets across workspaces

Permissions Required to Use a Shared Dataset

The ability to connect to a shared dataset, for the purpose of creating a new report, requires the **Build permission** on the dataset. This may be granted several ways:

1. The report creator possesses **write permissions to the workspace where the dataset resides**. Any of the workspace level roles above viewer (contributor, member, or admin) work for this purpose. The Build permission (see #2) is implicitly granted to people with write permissions.
2. The report creator is assigned the **Build permission for the individual dataset**. This is helpful when granular control is needed per individual dataset. The Build permission can be granted a few ways:
 - By directly editing the dataset.
 - During a sharing operation (if the person doing the sharing has sufficient permissions).
 - During app publishing process (unless the dataset resides in a separate workspace, in which case the dataset will need to be managed separately).

Note that dataset **Read permission** is insufficient for the creation of new content using the dataset. The Read permission on the dataset allows access to data for viewing reports which use the dataset. Importantly, by default read permission gives access to all the data in the dataset, even if it is not directly used in a chart, map, or table in a report. Use row-level security (RLS) to limit what data a user can access within a dataset.

Certification Permissions

The tenant setting related to dataset certification specifies which security group(s) may certify content. Note that the certification ability may not be specified on an individual workspace basis.

We recommend the ability to certify content be **restricted to a very small number of people** to ensure that data trustworthiness for certified data is consistent and reliable.

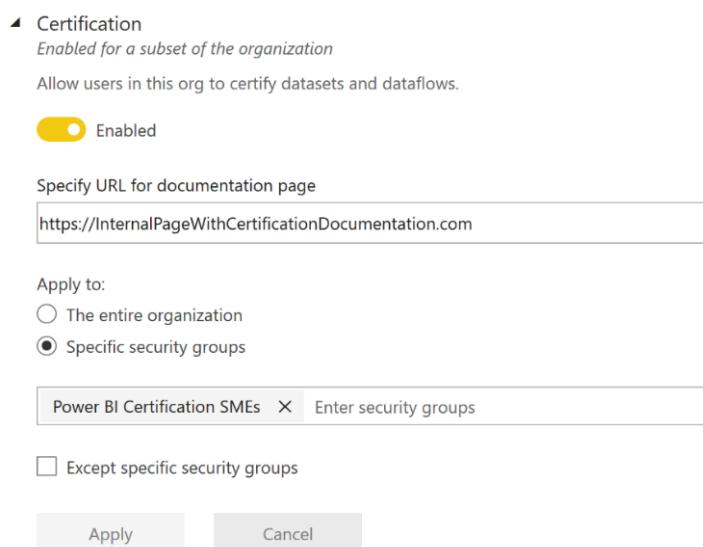


Figure 69 - Tenant setting in the Power BI admin portal for who may certify datasets and dataflows

Shared and Certified Dataflows

The ability to apply a certified or promoted [endorsement for a dataflow](#) is available as well. Most of the information shared in the previous section for shared and certified datasets also applies to dataflows.

Deployments to Development, Test, and Production

Enterprise IT standards usually require separate development, test, and production environments for business intelligence content which is considered mission critical. This typically involves a dedicated process for moving content from development to test, and from test to production, only when it is appropriate to do so. There are several ways to handle this in Power BI.

Option 1: Use App as Production

The first option involves utilizing Power BI Desktop + the workspace as the Development area. The act of publishing or updating an app represents moving content into Production.

- ✓ Pro: This option is lightweight and easiest to implement.
- ✓ Pro: It offers the ability to control exactly when the user base sees changes to reports and dashboards (with additional caveats as noted below).
- ✓ Pro: Allows a separate layer for defining access for viewers of the app vs. the workspace.
- ✗ Con: There are several challenges to using this approach. Three such very important considerations are discussed below.
- ✗ Con: A middle tier to handle testing separately is not available.
- ✗ Con: If there are different data source connections (ex: to query from, say, a test system instead of a production system), that cannot be easily handled in this option.

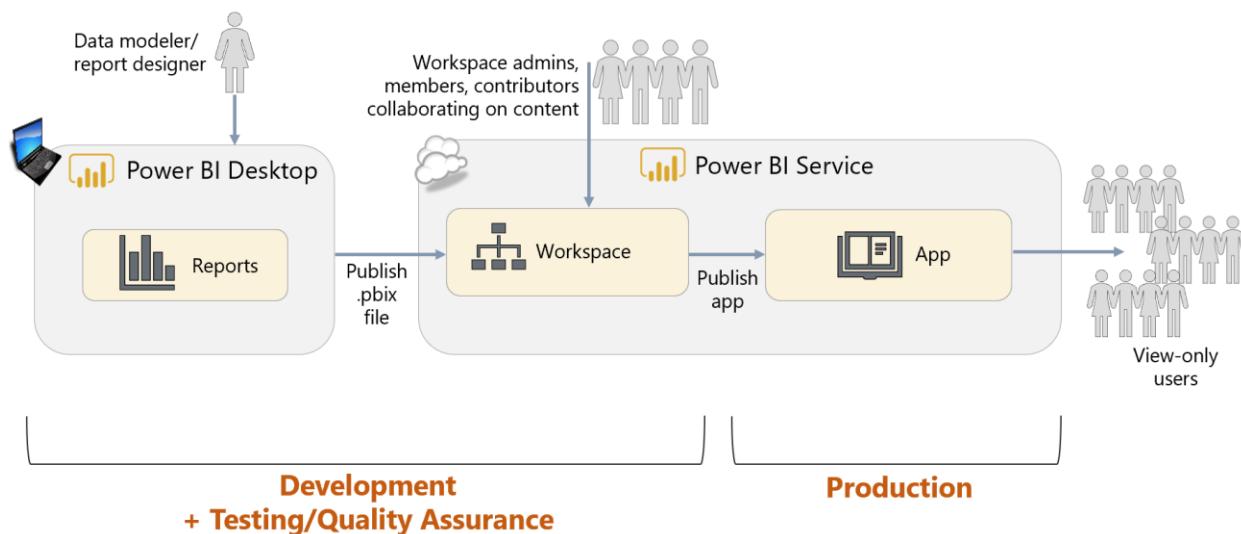


Figure 70 - Diagram of using an app as a lightweight 'production'

In the above graphic, the datasets could reside in a separate workspace. That is not depicted for simplicity. The collection of reports, dashboards, and workbooks do need to reside in the same workspace, as there is a 1:1 relationship between workspace and app.

There are three challenges to using this approach which are very important to understand before utilizing this method:

1. When republishing a .pbix file to the workspace, any **dataset changes always take effect in the app immediately**. The app publish process does precisely control when report and dashboard changes are visible; however, dataset changes in the underlying workspace are always immediately visible in the app too.

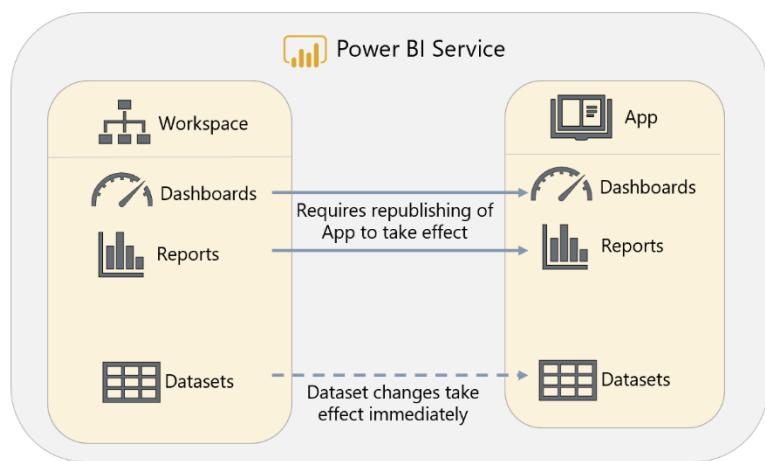


Figure 71 - Diagram showing that dataset changes always take effect immediately, without requiring app republishing

This includes data refreshes and changes to DAX calculation logic. It also includes things that can break visuals, such as relationship changes or edits to column names.

2. When republishing an app, **all items set as “Included in App” are always all republished**. This can prove challenging in a collaborative work environment when some content is ready to be published, while other content is not.
3. When republishing an app, **permissions and content are always published together**. If you need to redeploy the app merely to add a user permission, this can be a problem if report changes are underway in the workspace and not ready to be deployed. This can be mitigated with use of groups for app permissions rather than individual users.

For the above reasons, some teams find themselves ready to grow into using separate workspaces for Development, Test, and Production.

Option 2: Use of Separate Workspaces

The second option involves some additional complexity for lifecycle management. This option depicts the usage of separate workspaces for the separation of Development, Test, and Production content.

The **first variation of Option 2** involves publishing from Power BI Desktop to all three workspace environments.

- ✓ Pro: Since the different tiers are specifically enumerated, the number of environments needed can be easily customized to exactly what is needed. This could involve only two tiers if desired, or four tiers for instance.
- ✓ Pro: Solves the three main challenges experienced with option 1 above.
- ✓ Pro: There is additional flexibility to deliver content through an app, or through the workspace, depending on needs and preferences.
- ✓ Pro: This method can be used with shared capacity (whereas deployment pipelines discussed next in Option 3 requires the use of Premium capacity).
- ✗ Con: This option requires making sure that the .pbix file is in a state which can be deployed to Test or Production. Making additional changes to the .pbix file for Development could complicate keeping track of the state. This situation can be mitigated sometimes by downloading the file from the Power BI service (though it only applies to imported, not shared datasets; and not applicable for external users).
- ✗ Con: This option requires some processes/procedures, and efficient team communication, to execute seamlessly. Otherwise, it can be easy to lose track of the state of each workspace environment. IT teams are typically used to this type of rigor, whereas business-oriented teams are often less familiar with deployment processes.
- ✗ Con: Since this technique relies on building all applicable content in Power BI Desktop and publishing from there, usage of the Power BI service for content creation (i.e., web only reports) is discouraged.

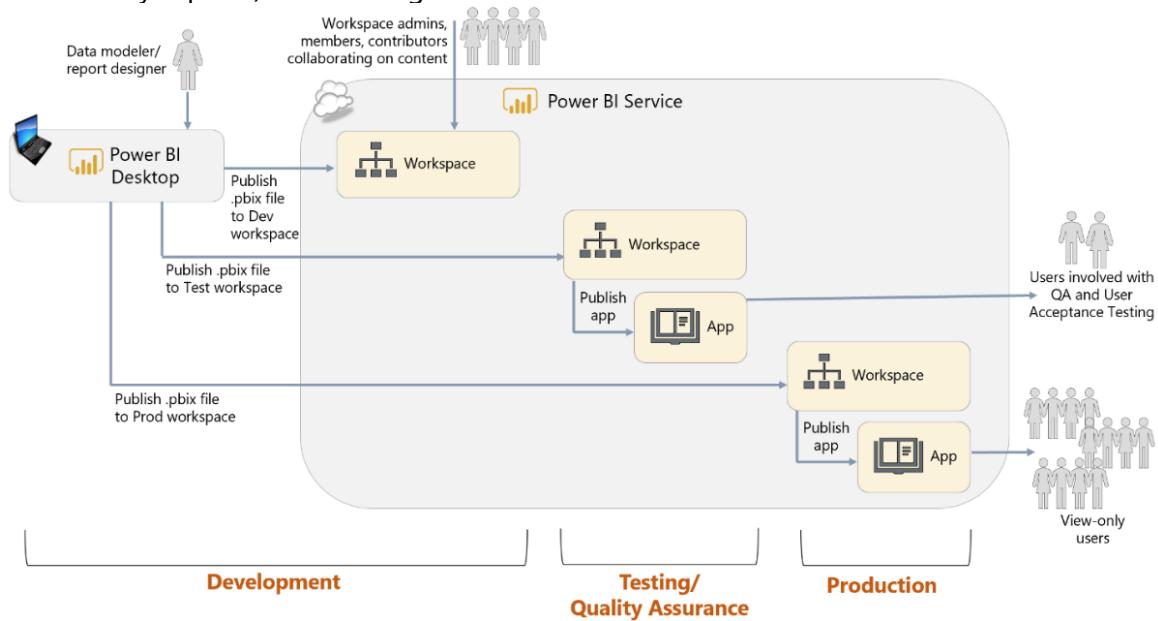


Figure 72 - Diagram showing use of separate workspaces for dev, test, and production with content manually published

The **second variation of Option 2** involves using the Power BI REST APIs to export content from a workspace and re-import to another. This technique is discussed in [this blog post](#), which also includes a sample PowerShell script. Additional considerations for this second variation include:

- ✓ Pro: This technique changes the Test and Production deployment techniques to a programmatic process. When Power BI content becomes mission critical, using more rigorous deployment techniques is a big advantage.
- ✓ Pro: This technique clones all the content from a workspace, so it can also deploy dashboards and web-only reports. Dataflows, however, are not yet included at the time of this writing.
- ✗ Con: Requires knowledge of using PowerShell to call the Power BI REST APIs.
- ✗ Con: There is not currently an API to publish an app. Therefore, it is not currently possible to completely automate the full end-to-end process.

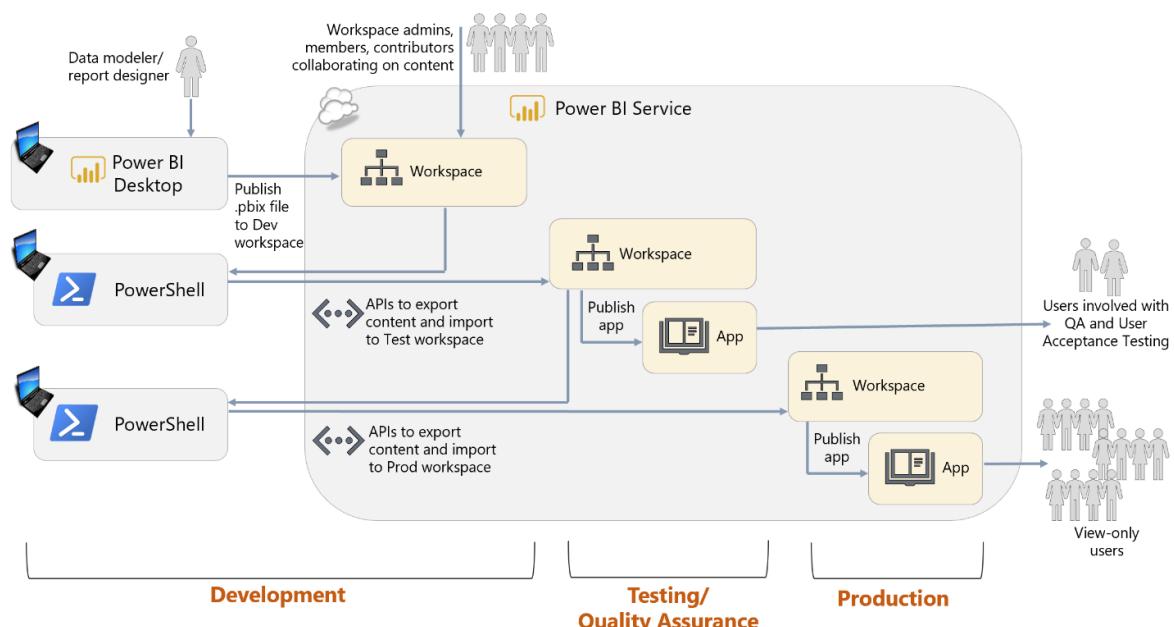


Figure 73 - Diagram showing use of separate workspaces for dev, test, and production with content published via REST APIs

Option 3: Deployment Pipelines + Separate Workspaces

The third option introduces a new feature in the Power BI service called [deployment pipelines](#), in conjunction with the use of separate workspaces which were introduced in Option 2.

A deployment pipeline can be thought of as a collection of workspaces, residing in Premium capacity, for the purpose of migrating content between them. Deployment pipelines are the future of content lifecycle management in Power BI. Any future features for content lifecycle management are expected to be built for supporting deployment pipelines.

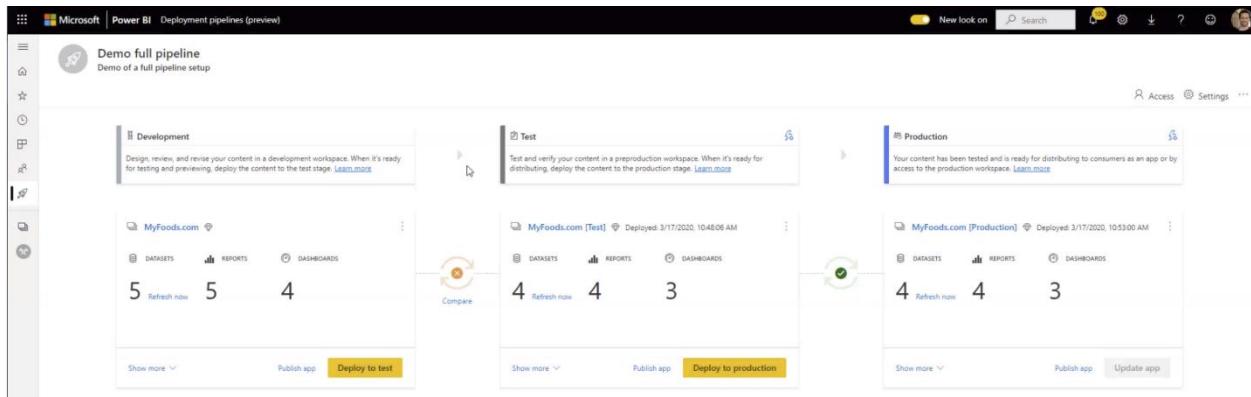


Figure 74 - Deployment pipeline in the Power BI service

A deployment pipeline consists of three stages: development, test, and production. Each workspace, and all its content, is assigned to a deployment stage, and content moves between the workspaces when a deployment occurs. Most commonly, an existing workspace is assigned to a development stage, then the test and production workspaces are auto-generated by the deployment pipeline (though there is flexibility to assign a workspace to any stage).

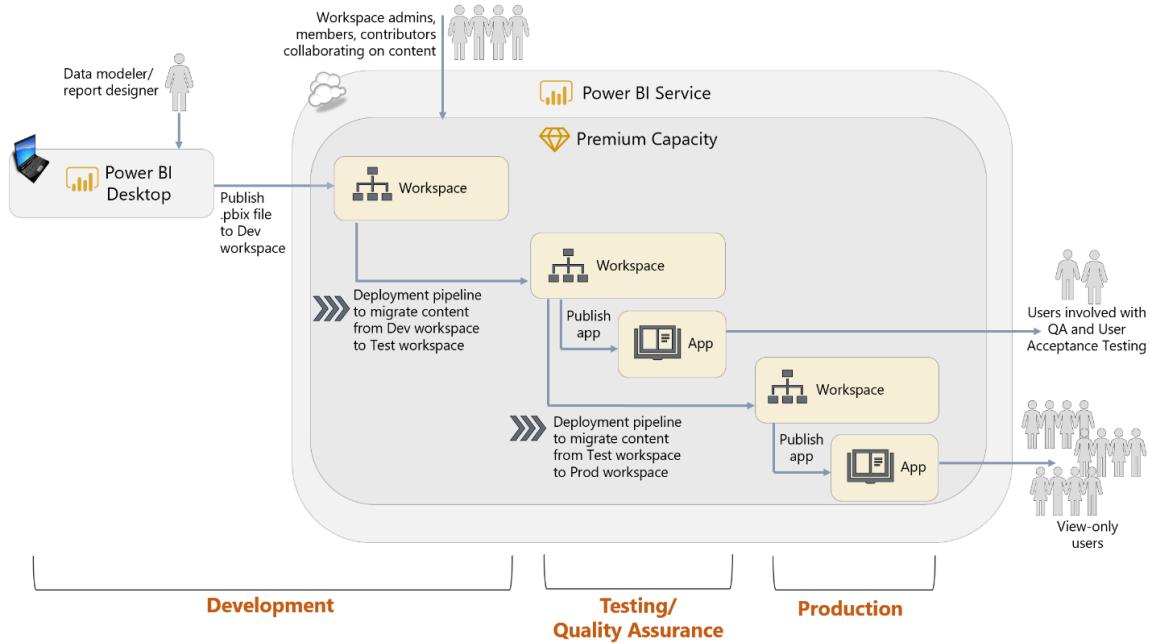


Figure 75 - Diagram showing use of separate workspaces for dev, test, and production with content published via deployment pipelines

Many of the considerations discussed in Option 2 above are also applicable with deployment pipelines. A few additional considerations for using deployment pipelines include:

- ✓ Pro: This option is driven by a user interface and easy to implement by users who may not have the skills or desire to use code-based deployments.
- ✓ Pro: Reduces the level of manual work involved to deploy content, which in turn reduces errors and inconsistencies.
- ✓ Pro: Once configured, a deployment pipeline is reusable many times. This allows for content updates to be promoted quickly once testing/validations have occurred.
- ✓ Pro: Data source rules and parameter rules are available for dynamically managing values which differ between development, test, and production.
- ✓ Pro: Because this approach relies on separate workspaces, full flexibility for applying different workspace security between dev, test, and prod is available. All existing workspace permissions will remain intact when content is deployed to the target environment.
- ✓ Pro: Pipeline deployment permissions are granted to workspace members and admins (provided they also have capacity assignment permissions if a new workspace is being provisioned during the initial deployment).
- ✓ Pro: Deployment pipeline menus will only be displayed for organizations that have at least one capacity assigned to the tenant (the capacity can be from either Power BI Premium or Power BI Embedded). This ensures that users will not be frustrated by viewing functionality which may not be available to them.
- ✗ Con: In order to prioritize ease of use, deployment pipelines are implemented with a user interface in the Power BI service. Corresponding APIs are not an option yet.
- ✗ Con: There is not currently an option to deploy an app as part of the deployment pipeline. App publishing remains a separate step once the pipeline deployment has completed.
- ✗ Con: Use of deployment pipelines requires dedicated capacity for all workspaces which have been assigned to the pipeline.

Suggestions For Getting the Most From Deployment Pipelines

Following are a few suggestions to consider as you begin to evaluate use of deployment pipelines in your organization. See also the [best practices article on deployment pipelines](#).

Plan for differences between dev/test/prod, including:

- The purpose and expectations for each workspace.
- Differences in user security per workspace (and app when appropriate).
- Different data sources used per workspace.
- Which workspace will be assigned to the deployment pipeline. Only one workspace may be assigned, and the other two workspaces are auto-generated by the pipeline. Most commonly the development workspace be assigned to the pipeline. A workspace may be assigned to only one deployment pipeline.

- Which values may need to be parameterized to accommodate differences between workspaces. The most common situation for parameterization is data source connections (ex: the test workspace sources data from a test database, whereas the prod workspace sources data from a Prod workspace).
- What approvals and validations are required in order to deploy to a production workspace. There is not currently a rollback capability once a deployment has occurred, so we recommend having a good process in place.

Plan for the **scope of workspaces** to not be overly broad. A best practice is to consider the entire contents of a workspace to be an analytical package which can be deployed together as a unit. While incremental/selective deployments are possible, it's ideal when a full deployment is a logical unit of content.

Be intentional with **workspace naming**, as discussed earlier in this section, and in conjunction with dev/test/prod suffixes.

Plan for **how to use the three stages**: development, test, and production. The number of stages cannot currently be customized, so if there is not a need for the 3rd slot, there are two main alternatives:

- Deploy to all three slots, but don't use the Test workspace for anything. In this case, Test could be considered as a staging area for Production. This is the suggested method.
- Only use the Test and Production slots, not the Development slot.

Plan for **how bug fixes will be handled**, in addition to scheduled/planned deployments. We recommend propagating all changes from development through test and production. However, if an immediate production fix is required, a process should exist to ensure it is also propagated to the other workspaces.

Consider **decoupling dataset development from report development** which has been discussed elsewhere in this whitepaper. Further, you may also consider **separating data workspaces from reporting workspaces**, which was discussed earlier as well.

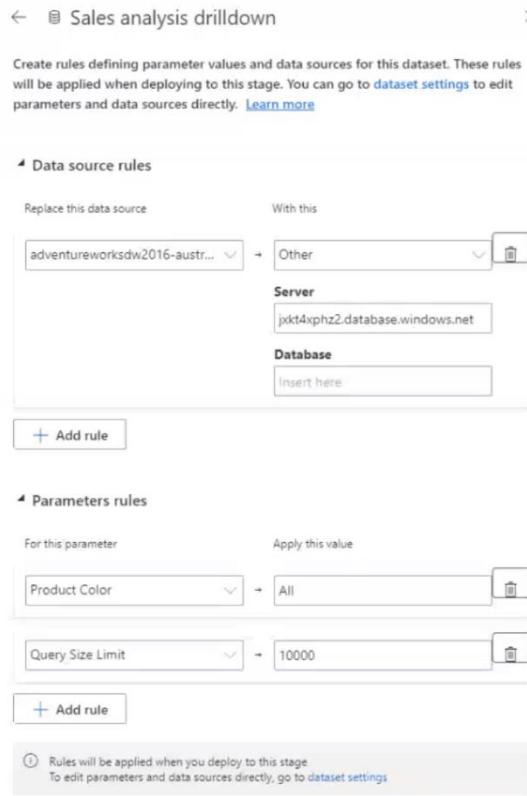
Plan needs for using **parameterization and data source rules**. There are two ways of achieving different values when deploying to different slots.

Plan for **capacity management**. Determine if you will use the same capacity for all workspaces in the pipeline, or if you will segregate the production workspaces into a separate capacity, and why.

If using apps, **consider using an app** associated with the test workspace as part of end-to-end testing prior to the production deployment.

Data source rules are created in the deployment settings of a stage in a pipeline. Data source rules can be used to manage the connection string for a data source (if the data source is supported by the [Update Datasources API](#)).

Parameters are created in the Query Editor in Power BI Desktop. Parameter rules may be created within the deployment pipeline for each stage. Parameter rules control different values when datasets are moved between deployment slots.



The screenshot shows the 'Sales analysis drilldown' deployment stage settings. Under 'Data source rules', a rule is defined for the data source 'adventureworksdw2016-austr...'. The 'With this' dropdown is set to 'Other', and the 'Server' field contains 'jxkt4xphz2.database.windows.net'. Under 'Parameters rules', two rules are defined: one for 'Product Color' with the value 'All', and another for 'Query Size Limit' with the value '10000'. A note at the bottom of the screen states: 'Rules will be applied when you deploy to this stage. To edit parameters and data sources directly, go to dataset settings'.

Figure 76 - Data source rules and parameterization for a deployment slot

Consider where **versioning or source control** will occur (as it is not a component of deployment pipelines). We recommend the continued use of OneDrive or SharePoint for source files (with versioning enabled), and that synchronization of files only occurs with the development workspace.

Consider what artifacts are managed through the **XMLA endpoint versus deployment pipelines**. For IT-managed datasets which reside in a Premium capacity, they may (optionally) be managed through the XMLA endpoint to take advantage of Analysis Services lifecycle capabilities and standard source control. Meanwhile, management of reports and dashboards may occur in a deployment pipeline.

Collaboration, Sharing and Distribution with Power BI Report Server

The options for collaboration, sharing, and distribution in Power BI Report Server are very different to those currently available in the Power BI service. Power BI Report Server is structured as a file system and all content published to it is either published to the root or to a folder; folders may contain subfolders. It is strongly recommended that you never publish to the root and that folders are created using the same methodology as described earlier in this section.

Access to content is controlled through Power BI Report Server's role-based security model, which is discussed in [Section 11](#).

Consuming Content Published to the Power BI Service

After a user has been given access to content, the next question is to decide through which medium to consume the content: via a web browser, through the Power BI mobile apps, through SharePoint Online and so on. There is such a wide variety of ways that end users can consume content that they may not be aware of all the options that are available, and as a result they may need guidance as to which options are appropriate for them.

Viewing Content in a Web Browser

The most common method for viewing content published to the Power BI service is via a web browser. Most users will use this option most of the time and it provides the richest set of options for interacting with content.

Power BI Mobile Apps

Many users prefer to view reports and dashboards on a mobile device such as a tablet or a phone, for example if they are at home, travelling for work, or just away from their desk. This is not just a matter of convenience: the [mobile apps for iPhone, iPad, Android phones and Android tablets](#) are designed for touchscreens and smaller screens, which makes it easier to interact with content.

To provide an optimal mobile experience, there is an alternative mobile layout when you are designing reports and dashboards in Power BI Desktop and the Power BI service.

Content published to Power BI Report Server may be displayed in the mobile apps as well.

Power BI Universal App for Windows 10 Devices

For desktops, tablets, or smartphones running Windows 10, the Power BI Universal App For Windows 10 can be an alternative to viewing reports and dashboards in a browser. The user interface closely mirrors what a user sees in a browser, and for less technical users it may be easier to navigate. Also, the touch-enabled interface can be a better experience than the browser.

Many organizations like to display reports on monitors located in office hallways, common areas, and conference rooms. This is particularly true for communicating progress towards goals, key performance indicators, or operational metrics. For this purpose, the [slideshow feature](#) available in the universal app is particularly helpful because it allows you to rotate between multiple report pages on a predefined frequency. The slideshow feature is part of [presentation mode](#).

Analyze in Excel

The [Analyze in Excel](#) feature of Power BI allows users to use Excel for the analysis of data stored in datasets which has been published to a workspace in the Power BI service.

This is particularly useful when significant effort has been put into shared and certified datasets.

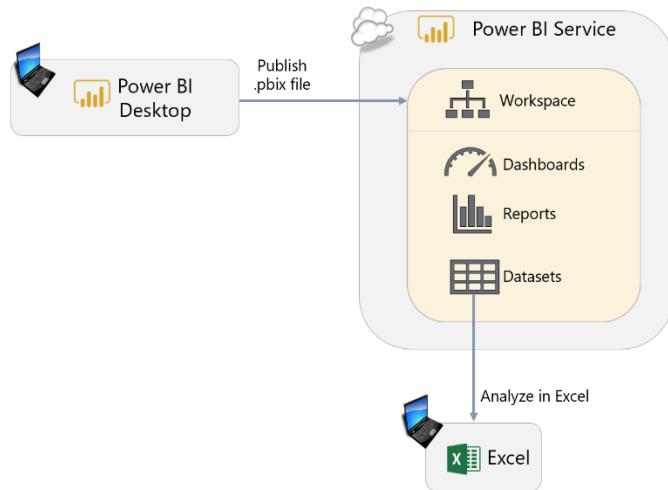


Figure 77 - Diagram of using Analyze in Excel with a dataset published to the Power BI service

Excel PivotTables are preferred by many users for exploring, slicing, and dicing data compared to functionality that is available in a Power BI report. Excel also has the advantage of familiarity: millions of Excel users know how to use PivotTables, and PivotTables connected to a Power BI dataset behave in a very similar way to regular Excel PivotTable. [Excel cube functions](#) can also be used to create complex and customized report layouts, for example financial reporting, that are not possible in Power BI Desktop.

Some users may have a strong preference for using Excel for their reporting and analysis, however many amazing Power BI demos they see. Rather than trying to fight against this attitude, promoting the use of Analyze in Excel allows users to keep using Excel but with data remaining in the centralized Power BI dataset. This gives you the advantages of using Power BI for creating datasets, the functionality of Excel for building reports, and visibility as users access the centralized datasets in Power BI.

Unfortunately, Analyze in Excel does suffer from a serious drawback. The live connection is done via a separate .ODC file which contains the connection string to the dataset in the Power BI service. Therefore, at the time of this writing, it is not possible to refresh an XLSX file in the Power BI service which uses Analyze in Excel (that is, a live connection to a Power BI dataset).

stored in the Power BI service). Although live connection reports created in Power BI Desktop may be republished to the Power BI service, this currently is not possible with live connection Excel files.

To use Analyze in Excel, the report creator must have [Build permission](#) for the dataset.

It's also important to be aware that a dataset must be prepared for use with Analyze in Excel. Whereas Power BI Desktop can handle implicit aggregations for numeric columns, Excel cannot. Therefore, specific measures need to be created for use with Excel PivotTables. Being unaware of this can lead to a suboptimal user experience for the report creator in Excel.

Live connections utilizing Analyze in Excel is not available for users of Power BI Report Server.

Embedded Reports in Microsoft Teams

[Microsoft Teams](#), a part of the Office 365 suite, is a collaboration application for teams of people working together. If Microsoft Teams is already widely used within your organization, then using Teams as a means for distributing Power BI content can promote greater engagement.

Power BI and
paginated reports can
be added as a tab
within a Teams channel

(but not dashboards).
This simple embedding scenario allows
members of a channel to discuss the contents
of a report using
threaded conversations.

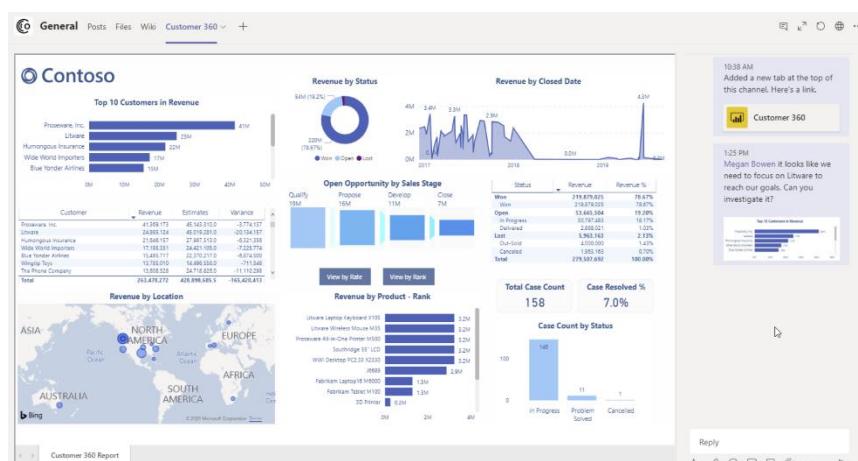


Figure 78 - Example report embedded in Microsoft Teams

Reports may be embedded in Teams from either a workspace or a published app. Utilizing the app version is recommended, as it ensures what will be viewable is the published app version rather than (potentially) an in-progress version in the workspace.

Content must first be published to the Power BI service; then it can be surfaced in Teams. Note that users do still need [permission to view the report in the Power BI service](#) as well as read permissions on the underlying dataset. Surfacing the content in Teams does not circumvent security, nor does it negate the need for a Pro license or use of Premium capacity.

Embedded Reports in SharePoint Online and SharePoint On-Premises

Power BI reports can be embedded in a SharePoint Online modern page using the [SharePoint Online web part for Power BI](#).

The URL for use with the web part in SharePoint can be obtained from the Power BI service.

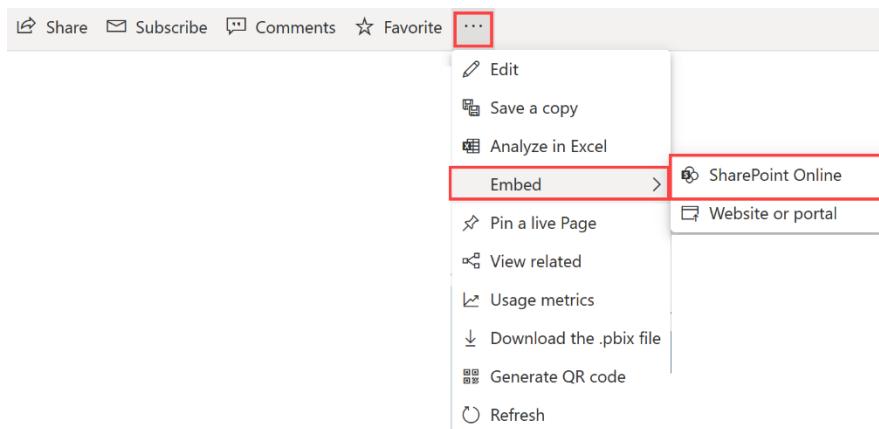


Figure 79 - Menu option for embedding a report or dashboard in SharePoint Online

No-Code Embedding with iFrames

The embed in website or portal option provides iFrame style embedding that can be used in portals or web pages. When viewing a page with the embedded report, users are prompted to sign-in to Power BI before accessing the content, ensuring all permissions and RLS are enforced before the user sees the content or data. While not as smooth an experience as you could build using APIs, this option requires little technical skill and therefore can be broadly adopted.

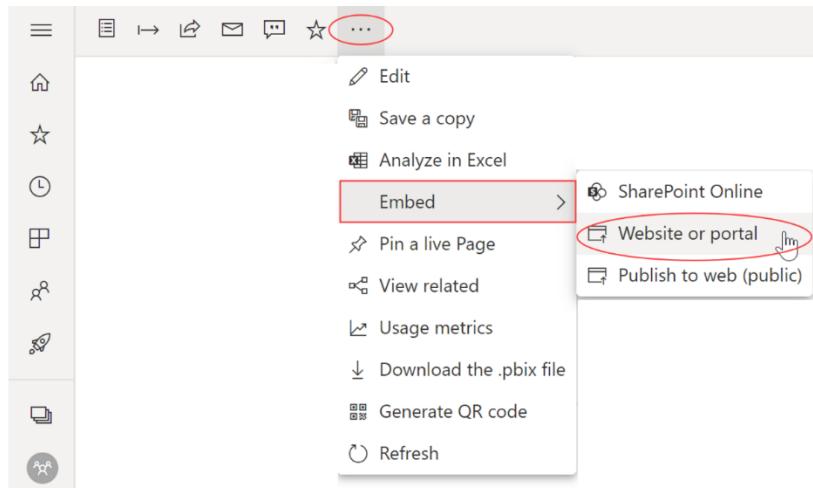


Figure 80 - Menu option for embedding a report or dashboard in an iFrame

This option is recommended for users with SharePoint on-premises since Power BI doesn't provide a web part. There are also third-party SharePoint add-ins available in the Office Store which use the API in the background and avoid the need for custom code.

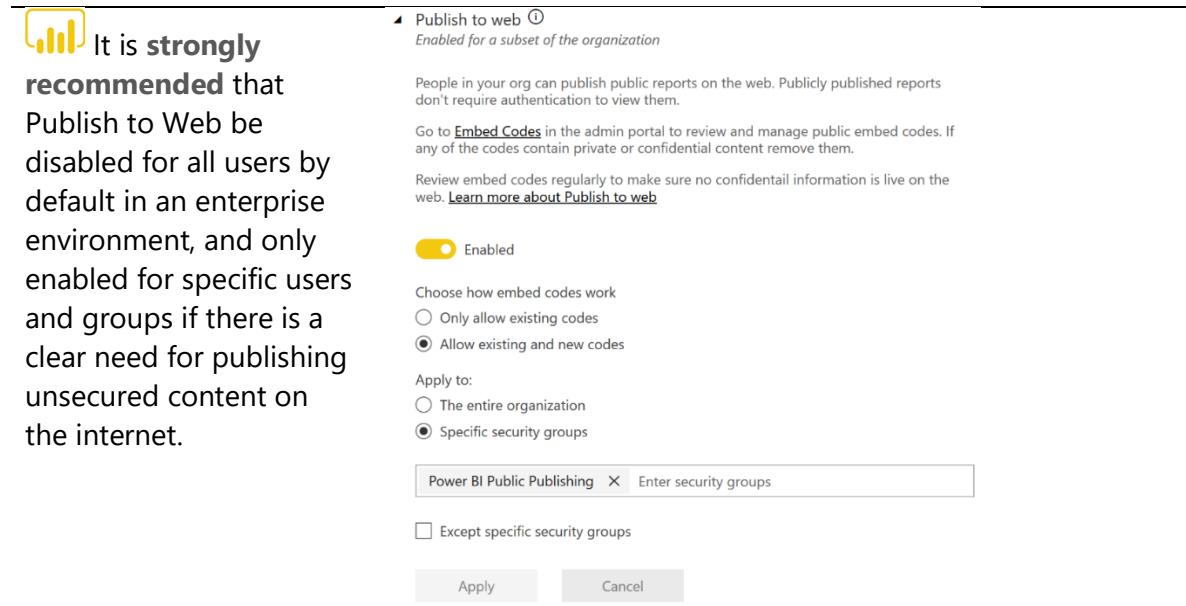
Distributing Reports Publicly on the Web

It is possible to make a report visible to anyone on the public Internet with the [Publish To Web](#) feature. In this scenario, once a report has been published to the Power BI service, when viewing the report in the browser, you can click on the Publish to Web option under the File menu. Once that has been done, a URL for the report and HTML code to embed the report in a public website will be generated. Anyone who knows the URL, whether they are inside or outside the organization, will be able to view the report.

This feature is useful if your organization needs to publish reports on a public-facing website. Typically, journalists, governments, public agencies, non-profits, corporate communications professionals, or Power BI enthusiasts will use the Publish to Web feature when sharing data intended for broad public consumption, including on social media.

WARNING: The Publish to Web feature presents **serious security risks**. It should **not be used for data that is confidential or proprietary** in any way. A common misconception is that viewers can only see the summarized, visual-level data. This is not always accurate – viewers can still use 'See Records' to view detail rows in some situations. An experienced user could also craft their own queries and retrieve anything from the underlying data model. Reports published in this way may be added to a public gallery of reports. As such, it is not a secure way of making content available within an organization, even if deployed to an intranet.

Be aware there are quite a few [limitations](#) associated with use of Publish to Web.



The screenshot shows the 'Power BI Public Publishing' settings. It includes a note that it is strongly recommended to disable Publish to Web by default for all users in an enterprise environment, and only enable it for specific users and groups if there is a clear need for publishing unsecured content on the internet. The 'Enabled' toggle is set to 'Enabled'. The 'Choose how embed codes work' section has 'Allow existing and new codes' selected. The 'Apply to' section has 'Specific security groups' selected. A text input field 'Enter security groups' is empty. At the bottom are 'Apply' and 'Cancel' buttons.

Figure 81 - Tenant setting for Publish to web

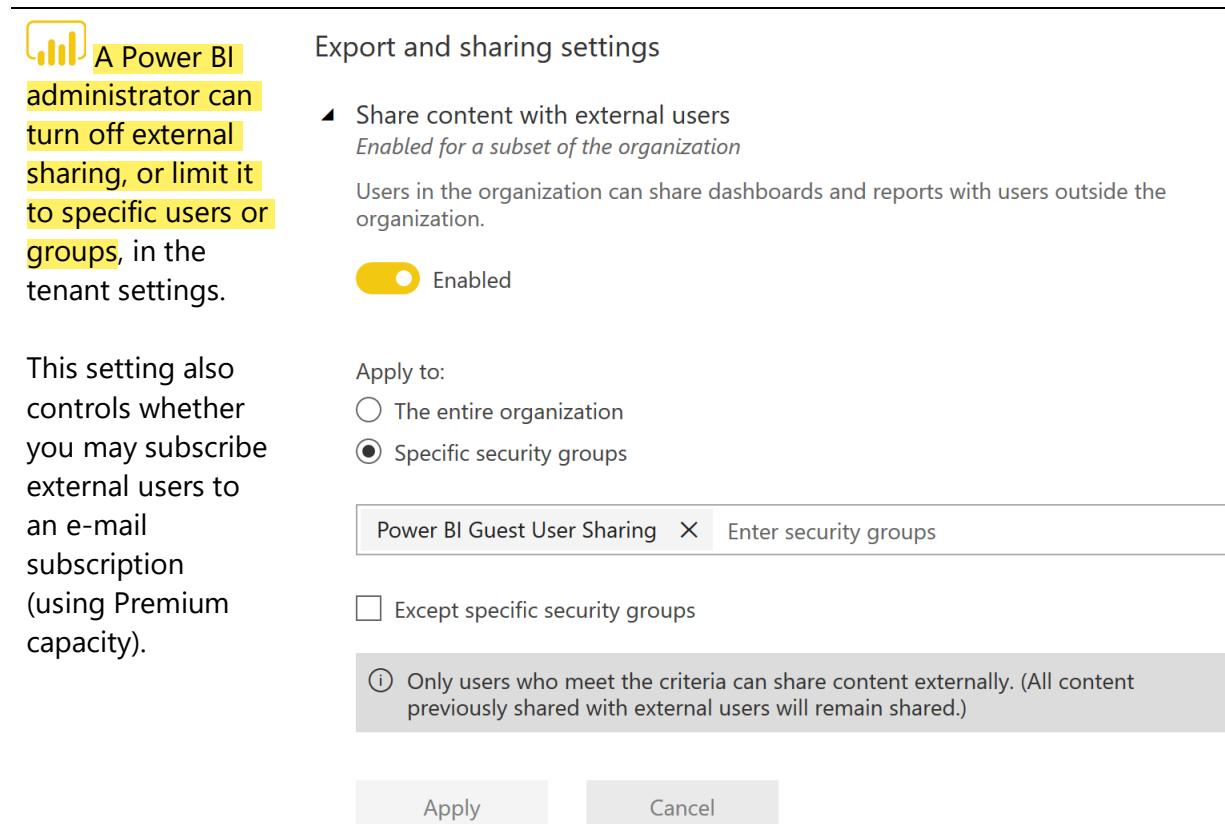
Sharing and Distribution with External Users Using Azure B2B

It is very common to work with external users, such as partners, vendors, customers, or suppliers. It is possible to work with users outside your organization through [Power BI's integration with Azure Active Directory B2B](#).

There are several aspects and decision points to making this work, including:

- ✓ Use of ad hoc vs. [planned guest user invitations](#)
- ✓ How the organizational relationships work across tenants
- ✓ Utilizing Premium vs. Pro licenses
- ✓ Guest user Pro licensing from home tenant vs. inviting tenant
- ✓ Allowing guest users to edit content vs. just view content
- ✓ Utilizing row-level security

All of these aspects are covered in the whitepaper, as well as guest user limitations and other important information: [Distribute Power BI Content to External Guest Users Using Azure Active Directory B2B](#).



The screenshot shows the 'Export and sharing settings' dialog. On the left, a sidebar lists steps for enabling guest sharing: 'A Power BI administrator can turn off external sharing, or limit it to specific users or groups, in the tenant settings.' Below this, a note states: 'This setting also controls whether you may subscribe external users to an e-mail subscription (using Premium capacity).'. The main area shows the 'Share content with external users' setting is 'Enabled for a subset of the organization'. It includes an 'Apply to:' section with 'Specific security groups' selected, a search bar for 'Power BI Guest User Sharing', and a note that only users meeting criteria can share content externally. At the bottom are 'Apply' and 'Cancel' buttons.

A Power BI administrator can turn off external sharing, or limit it to specific users or groups, in the tenant settings.

This setting also controls whether you may subscribe external users to an e-mail subscription (using Premium capacity).

Share content with external users

Enabled for a subset of the organization

Users in the organization can share dashboards and reports with users outside the organization.

Enabled

Apply to:

The entire organization

Specific security groups

Power BI Guest User Sharing Enter security groups

Except specific security groups

Only users who meet the criteria can share content externally. (All content previously shared with external users will remain shared.)

Apply Cancel

Figure 82 - Tenant setting for sharing content with external users

There is a separate tenant setting to specify which guest users are permitted to edit content. This is typically very minimal as compared to the number of guest users who are permitted to view content.

Allow external guest users to edit and manage content in the organization
Enabled for a subset of the organization

The specified guest users in the organization can edit and manage content in workspaces in the organization. They receive the ability to browse content and request access to content. [Learn more](#).

 Enabled

Apply to:

The entire organization
 Specific security groups

Power BI External Guest User Content Editors  Enter security groups

Except specific security groups

 Only guest users who meet the criteria can edit and manage content in the organization

Figure 83 - Tenant setting for external users who may edit content

The option to have guests work within your tenant can enable you to onboard contractors or partnerships with external entities more quickly. It can also have a licensing benefit since in the commercial clouds, external guest users can bring their own license to your tenant. This means that if the guest user is licensed in their own home tenant with Power BI Pro, when they access your tenant their Pro license is honored as well. One caution with this approach is it has some limitations: publishing from Power BI Desktop, connecting to Power BI datasets from tools like Excel or Power BI Desktop are not supported.

Subscriptions and Alerting

In addition to considering how users view Power BI content, it is important to consider how users will be brought to view that content. While many users will go to the Power BI service of their own accord, others may need to be prompted or reminded to view reports and dashboards daily or weekly. There may also be scenarios where users need to view a report when a particular event has taken place. This is where subscriptions and alerts become useful.

Subscriptions

For users who want to view the same report on a regular basis, [subscriptions](#) can be used to deliver email notifications. After a user subscribes to a report page, they will receive an email

containing a snapshot of a report page when the dataset used by the report has changed. Subscriptions may be created in the Power BI service for:

- Power BI report pages (scoped to one individual report page at a time)
- Power BI dashboards (scoped to the entire dashboard)
- Paginated reports (scoped to the entire report)

Subscribers can opt to receive the subscription hourly, daily, weekly, or monthly. A subscription email may also be generated after a data refresh (up to once daily).

There are quite a few considerations and limitations related to using subscriptions which are documented [here](#) for Power BI report pages and dashboards, and [here](#) for paginated reports.

There are some noteworthy feature differences between subscriptions for Power BI reports/dashboards vs. paginated reports. The top differences include:

	Power BI Report Page Subscriptions	Paginated Report Subscriptions
Requires Premium capacity	No (but Free users can receive subscriptions if using Premium)	Yes
Scope	Individual report page or dashboard	Entire report
Number of subscriptions per report	24 per report	No limit
Formats available	Image	Image (optional) + PDF, CSV, XLSX, PPTX, DOCX, XML
Ability to schedule after data refresh	Yes	No Frequently not necessary depending on data source
Also provide access to report (sharing operation)	Optional (but always gets re-checked every time you come back to the user interface)	Optional
Can subscribe others	Yes If RLS is <u>not</u> on the dataset	Yes If RLS is on the dataset, uses the subscriber's security context
Can subscribe to a live connection report	Yes For yourself only	Yes If RLS is on the dataset, uses the subscriber's security context

If the content resides in Premium capacity, additional subscription-related features become available:

- Ability to use group e-mail aliases for subscriptions, instead of subscribing users one individual at a time.
- Ability to subscribe users who only have a Power BI Free license.
- Ability to use paginated report subscriptions (paginated reports are a Premium capacity feature).
- Ability to subscribe [external users](#) outside of the organization.

In summary, subscriptions are best used when:

- ✓ A point-in-time snapshot of the report is desired.
- ✓ The user does not need or want an interactive experience.
- ✓ The user prefers a report to be delivered on a schedule.
- ✓ Data security is not at risk of being compromised by distributing report content via email.



Be cautious using subscriptions for reports with sensitive data

When an e-mail subscription is sent, an image of the report page is included in the e-mail. Be cautious with use of e-mail subscriptions when sensitive data could be included on the report page.

Alerts

Alerts can be an effective way of making users aware of important changes and events that require timely action on their part. [Power BI alerts](#) can be created for tiles on a dashboard that use card, gauge, or KPI visualizations. When the value displayed in the visualization changes, and reaches a given threshold, the alert will be fired.

Alerts can be configured to notify at most once an hour or once a day. When an alert is fired, the user receives a notification that is visible in the Power BI service as well as in the Power BI mobile apps. Optionally, an e-mail can be sent as well.

The alerting capabilities within the Power BI service are very basic. These capabilities can be extended with the use of Power Automate.

Export-To-File API

The [Export-To-File API](#) may be used in a number of ways:

- A button within an application to export a report
- A scheduled export of a report which sends an e-mail attachment
- A scheduled export of a report snapshot (bookmarks in a specific state)

Both the dataset and the report must reside in a workspace that is assigned to a Power BI Premium or Power BI Embedded capacity. Additional information is available <https://docs.microsoft.com/en-us/power-bi/developer/embedded/export-to>.

Integration with Power Automate

Power Automate is a cloud-based tool for creating automated workflows. A Flow in Power Automate can be triggered when a Power BI alert is fired, and Power Automate provides a wide range of integration options for actions that can happen after that. For example:

- Email alerts can be sent to multiple users with customized content
- Data can be written to a file or database
- An issue can be created in services such as Jira or GitHub

There is an array of possibilities for integration with Power Platform tools such as Power Automate and Power Apps.

Section 10. Power BI Administration

The Power BI documentation contains a lot of detailed information on [administering Power BI](#) and configuring security. This whitepaper will not be duplicating content which is available in the online administration documentation. Instead the focus here will be on providing an overview of key functionality available and, more importantly, providing recommendations for how this functionality should be used.

Power BI is part of the larger Microsoft cloud ecosystem in Azure. If your organization is already using services such as Microsoft 365 or Dynamics 365, then Power BI will be within the same tenant. This common tenant is built on Azure Active Directory.

Initial Power BI Tenant Setup

Creation of a Power BI Tenant

Ideally, planning a Power BI deployment should start before anyone within the organization has logged in to Power BI for the first time. This begins with finding whomever in the organization is responsible for administering Microsoft 365 and/or Azure Active Directory.

If there is not an Azure Active Directory (AAD) tenant in existence associated with an e-mail domain (ex: Contoso.com), then when the first user from Contoso.com signs up for a Power BI account (Free or Pro), an unmanaged AAD directory will be auto-created. See [Section 4](#) for additional discussion about this.

Geographic Data Region

When a Power BI tenant is created, it is created in an Azure data center. Choosing the correct Azure data center to host your Power BI tenant is important for two reasons:

- It may be important for legal reasons that corporate data is stored in a specific jurisdiction.
- The performance of reports and dashboards depends, in part, on users being in proximity to the Power BI tenant.

It is not possible to select a data center directly for commercial cloud tenants. Instead, Microsoft selects the data center closest to the country selected in the "Where's your company located" dropdown box when the tenant is first created (see this [article](#) for more details).

The location of the data center chosen is displayed in the 'About Power BI' dialog, accessible from the Help & Support dropdown menu (the question mark icon) in the top right-hand corner of the screen when logged into the Power BI service.

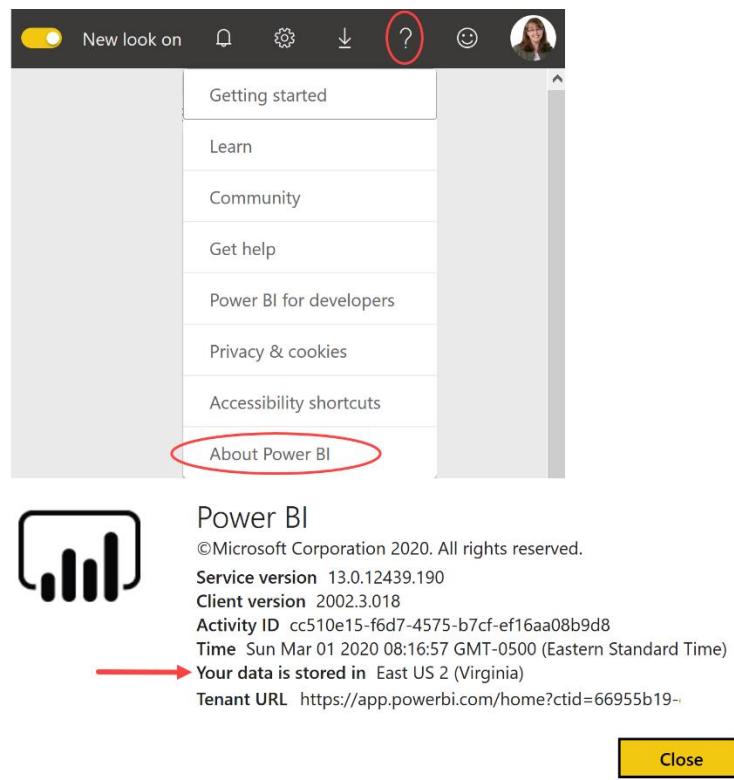


Figure 84 - About menu in the Power BI service



Moving a Power BI tenant to another Azure data region involves creating a support case. All content must be re-deployed by the customer after the brand-new tenant is provisioned by support in the desired region.

As discussed in [Section 3](#), specific data regions are supported with respect to Power BI Premium. Different Premium capacity nodes may be provisioned in the data region of your choice, independently of where home Power BI tenant is located. This enables organizations to:

- Meet data residency requirements for data at rest. Note that individual workspaces must be assigned to the correct Premium capacity to ensure this is occurring.
- Improve the ability to locate data near the user base.

Administrator Roles Related to Power BI

To reduce risk and follow the principle of least privilege, it is recommended to keep membership of the Power BI administrators to a **small number of people**. Multiple Power BI administrators are necessary to cover vacations, sickness, and employees leaving the organization. The standard recommendation is to have [2-4 global administrators at most](#).

Global administrators, via Microsoft 365 or Azure Active Directory, implicitly possess administrator rights in the Power BI service as well. A Microsoft 365 global administrator may assign individual users to the [Power BI service administrator role](#), which grants administrative rights over Power BI features only.

This section primarily refers to responsibilities related to Power BI administrators. There are several additional administrator roles which will be needed from time to time as well, including:

Type of Administrator	Scope	Permissions Available
Microsoft 365 Global Administrator	Microsoft 365	All administrative features for all Microsoft 365 services. Implicitly also a Power BI administrator, with the ability to add people to the Power BI administrator role. Manages unified data sensitivity labels.
Microsoft 365 Billing Administrator	Microsoft 365	Manages subscriptions and purchases services.
Microsoft 365 License Administrator	Microsoft 365	Manages user license assignments.
Power BI Administrator	Power BI service	Access to the admin portal in the Power BI service, including tenant-level settings. Able to manage all workspaces as well.
Reports Reader	Microsoft 365	View activity reports and usage data in Microsoft 365, including usage from Power BI.
Power BI Premium Capacity Administrator	One or more Premium capacity nodes in the Power BI service	Management of Power BI Premium capacity settings and workspaces assigned. View Premium usage measurements for CPU and memory.

Type of Administrator	Scope	Permissions Available
Power BI Workspace Administrator	A single workspace	Manage all aspects of a workspace, including the access. Includes the ability to delete the workspace
Power BI Embedded Capacity Administrator	Azure Power BI Embedded	Management of Power BI Embedded capacity settings.
Power BI Gateway Administrator	Gateway cluster	Provisions gateway cluster servers. Manages gateway settings. Registers data sources, including credentials, plus users assigned to each data source. Installs monthly gateway updates.
Desktop Support	All Power BI users	Manages process to install monthly Power BI Desktop updates.
System Administrator	Microsoft 365	Management of mobile device policies. Manage encryption keys.
Azure Active Directory Administrator	Azure Active Directory	Management of groups used for securing Power BI, and B2B guest access. Creation of service principals. Configures conditional access policies and other security features.

The Power BI service administrator role affects management capabilities across the entire Power BI tenant. The Power BI service administrator responsibilities cannot be delegated to individual groups, or functional areas, within a Power BI tenant. In some larger organizations, there is some friction between the centralized and decentralized Power BI development teams because the Power BI administrator role is applicable tenant-wide, and there is not a read-only administrator role. An example of this friction is when functionality like “push apps to end users” capability does not work and the user does not know why--a decentralized colleague may not be aware that there is a tenant-level setting to control it, nor is the colleague able to view or verify the current setting. We recommend publishing documentation about tenant settings to aid your decentralized user population with awareness settings and their impact on functionality.

Tenant Setting Recommendations

Global and Power BI service administrators have access to tenant settings in the [Power BI admin portal](#).

 Management of tenant settings is extremely important for proper governance and security of the Power BI environment. It is important to review the tenant settings when your organization first begins to use the Power BI service, and establish a periodic review process to ensure your settings align to your internal governance expectations.

Some tenant-level settings are enabled by default, while others are disabled by default.

Functionality for certain tenant settings may be specified for specific groups, whereas other settings are enabled or disabled across the entire organization. Consequently, tenant settings typically have three states:

- Disabled for the entire organization.
- Enabled for the entire organization.
- Enabled for a subset of the organization. Control for specific groups is available for most settings; however, there are a few settings which only apply to the entire organization. You can selectively allow or exclude users via the use of security groups.

Recommendations related to tenant settings include:

- The Power BI administrator role should be assigned to people who have knowledge of Power BI. It's not as well-suited to an administrator who does not have any familiarity with what users are trying to accomplish.
- To compensate for the lack of a read-only role for tenant settings, which introduces some challenges in larger decentralized organizations, we recommend **documenting the current settings**, and making the documentation available to the Power BI user population for reference. Documenting the tenant settings aids with monitoring of data governance policies as well.
- **Review settings on a regular basis**, including any new settings which have been released. Quarterly works well for this purpose.
- Use the **audit log and/or alerting** capabilities from Office 365 or Cloud App Security to be aware of when a tenant setting has changed, and by whom.
- For many organizations, Power BI is the central location where critical business data is aggregated. It is important to review data and sharing settings to ensure your organization understand how data is shared within and outside the organization.

Following are the specific tenant settings, including recommendations. All capabilities refer to the Power BI service unless otherwise noted.

Help and Support Settings

Property	Description
Publish "Get Help" information	<p>Custom URLs which can replace the built-in URLs on the help menu, including:</p> <ul style="list-style-type: none">• Training and documentation• Discussion forum• Licensing requests• Help desk <p>Use of the help URLs is discussed later in this section.</p> <p>Recommendation: Having exceptional internal support for the user population significantly impacts user experience and the adoption of Power BI in your organization. Use custom help menu options where possible, if you have made an investment in internal resources. Note that using this URL will also prevent users from signing up for a Pro trial. This is discussed further in Section 4.</p>
Receive email notifications for service outages or incidents	<p>Specifies which group which is informed via e-mail when an incident is occurring which impacts reliability. An e-mail is also sent when the incident has been resolved.</p> <p>At the time of this writing, this capability is only applicable to incidents for Premium capacity. More information is here.</p> <p>Recommendation: A mail-enabled group for Power BI Support should be used for this purpose.</p>

Workspace Settings

Property	Description
Create workspaces (new workspace experience)	<p>Specifies which Power BI Pro users in the organization are permitted to create new workspaces.</p> <p>Recommendation: For creation of workspaces, follow the same type of approach that your organization uses for creation of Teams and SharePoint sites. Most large companies do restrict the creation of workspaces somewhat in order to improve consistency and use of conventions – if that is the case, there should be a very short response time for responding to a workspace request (if too much of a delay is involved, users will over-use sharing functionality, or use existing workspaces which were created for a different purpose which creates other issues). You may consider automating the approval process with the Power Platform.</p>
Use datasets across workspaces	<p>Permits report designers to create reports in one workspace that connect to datasets in a different workspace, assuming the requisite Build permission is granted for the dataset.</p> <p>Recommendation: Highly recommended this be enabled for all users. As discussed earlier in this whitepaper, it becomes important to decouple datasets from reports to achieve data reusability. Also, if you use data workspaces separate from reporting workspaces (for the purpose of managing who is permitted to edit and manage access for datasets separately from managing who is permitted to edit and distribute reports), this tenant setting allows that scenario to work well.</p>

Information Protection Settings

Property	Description
Connect Power BI to Microsoft Information Protection sensitivity labels (step 1)	<p>Allows content published to the Power BI service to use the sensitivity labels set up for Microsoft Information Protection.</p> <p>This setting will be removed in the near future, and only the step 2 will be necessary for customers to specify.</p> <p>Recommendation: Strongly advised to make use of this functionality, as it aids the classification of content based on sensitivity or confidentiality level. When using sensitivity labels, your organization should educate users on their meaning and your policies surrounding them to minimize risk of accidental or uninformed mis-use or oversharing of sensitive data.</p>
Allow users to apply sensitivity labels for Power BI content (step 2)	<p>Specifies which users in the organization are permitted to apply sensitivity labels to content published in the Power BI service.</p> <p>Recommendation: Usually label assignment is permitted by all content authors. Follow the same type of process used for the assignment of labels in Office documents.</p>

Export and Sharing Settings

Property	Description
Share content with external users	<p>Controls whether Power BI allows inviting external users to your organization. Once an external user accepts the invitation, they become an Azure AD B2B guest user in your organization. They appear in “people pickers” throughout the Power BI experience. When the setting is disabled, existing guest users in your organization continue to have access to any items they had access to and continue to be listed in people picker experiences. Additionally, if guests are added through the planned invitation approach, they will also appear in people pickers. To prevent guest users from accessing Power BI, an Azure AD conditional access policy should be used. This setting also controls whether you may subscribe external users to an e-mail subscription (requires Premium capacity).</p> <p>See the B2B whitepaper for the various scenarios in which external content collaboration and distribution could work.</p> <p>Recommendation: This setting should be aligned with the organization’s security policy. If collaboration with users outside of the organization is prohibited or highly unusual, this should be disabled to reduce risk. However, collaboration across organizations is becoming more and more common so most companies do allow external sharing of content to a certain extent.</p> <p>Note that there are additional options in Azure Active Directory, such as conditional access, for limiting what guest users can do.</p>
Publish to web	<p>Controls whether content authors are permitted to distribute reports and dashboards on the public internet using embed codes. The menu options that users see change based on this setting, and are documented here.</p> <p>Recommendation: Since Publish to Web is intentionally insecure, this setting default does not allow users to create new publish to web embed codes, but tells users to contact the power bi admin to get permission., and very selectively enabled for only those persons who have a legitimate need to publish data publicly.</p>

Property	Description
Copy and paste visuals	<p>Allows some or all users to copy an image of a dashboard tile or report visual to your clipboard, which would allow it to be pasted elsewhere (such as an e-mail or a document). A link back to the original visual is included in the pasted image.</p> <p>Recommendation: Should usually be enabled for all users without any limitations for maximum productivity. To handle specific confidentiality scenarios: the copy capability may be disabled for individual visuals in an individual report (within the Visual Formatting pane for the specific visual).</p>
Export data	<p>Allows several capabilities:</p> <ul style="list-style-type: none"> ○ Ability to export data from tiles or visualizations to CSV file ○ Use of Analyze in Excel ○ Use of Power BI service live connection ○ Ability to download PBIX files <p>Recommendation: Should usually be enabled for all users without any limitations for maximum productivity. If there is a concern with the extent of data exports, use data from the audit log to determine who is over-using data exports and try to educate them on more efficient methods. Disabling this setting also prevents users from using Live Connections, which is highly encouraged to facilitate data reusability, so this setting should generally be left enabled. Unlike Excel, PowerPoint, and PDF exports, the sensitivity label does not currently carry forward to the CSV file format.</p>
Export to Excel	<p>Permits exporting Power BI reports to Excel.</p> <p>Recommendation: Should usually be enabled for all users without any limitations for maximum productivity. The Microsoft Information Protection features can be used to carry the sensitivity labels to exported Excel files.</p> <p>If there is a concern about an over-reliance on Excel, use data from the audit log to determine who is over-using this feature and try to educate them on more efficient methods.</p>
Export reports as PowerPoint presentations or PDF documents	<p>Permits exporting Power BI reports to PowerPoint or PDF files. For PowerPoint, each report page becomes an image on a slide in the exported file.</p> <p>Recommendation: Should usually be enabled for all users without any limitations for maximum productivity. The Microsoft Information Protection features can be used to carry the sensitivity labels to exported PPTX and PDF files.</p>

Property	Description
Export reports as image files	<p>Permits exporting Power BI reports to PNG files.</p> <p>Recommendation: Should usually be enabled for all users without any limitations for maximum productivity. Unlike Excel, PowerPoint, and PDF exports, the sensitivity labels do not currently carry forward to the PNG file format. If the report or its underlying dataset is tagged by a sensitivity label, the resulting exported image file will not show the label.</p>
Print dashboards and reports	<p>Controls whether the print options are available when viewing a report or dashboard in the Power BI service.</p> <p>Recommendation: Should usually be enabled for all users without any limitations for maximum productivity. If there is a concern about too much printing, use data from the audit log to determine who is over-using printing and try to educate them on more efficient methods.</p>
Certification	<p>Specifies who in the organization is permitted to certify datasets.</p> <p>Recommendation: The ability to certify content should be restricted to a small group of subject matter experts. There should be a documented process for the certification of content so that it signifies trustworthiness.</p>
Allow external guest users to edit and manage content in the organization	<p>Specifies which external users may edit content. This does also require specific workspace permissions as well. See the B2B whitepaper for additional information.</p> <p>Recommendation: Since most guest users will be read-only, the ability to edit is typically an exception. Therefore, this setting should be disabled, and selectively enabled for external users who have a legitimate need to edit content within your tenant (such as Power BI consultants). Be aware there are still some limitations to using this approach.</p>
Email subscriptions	<p>Allows use of email subscriptions. Applies to the entire organization rather than individual groups.</p> <p>Recommendation: Should usually be enabled for maximum productivity. Subscription activity can be monitored via the audit log.</p>

Property	Description
Featured content	<p>Specifies which users may promote content using the 'Featured' section of the Home page. Featured content is customized recommended content displayed on the Home page for users with access to the promoted content.</p> <p>Recommendation: Since this feature applies broadly to the Home page, it should typically be restricted to a centralized BI team, plus select report publishers who will use this capability judiciously.</p>

Content Pack and App Settings

Property	Description
Publish content packs and apps to the entire organization	<p>Permits publishing of Apps or Organizational Content Packs (which are deprecated and should no longer be used) to everyone in your organization.</p> <p>Recommendation: Since this is applicable to the entire organization, we recommend this be disabled to avoid accidental publishing to the entire organization. It should typically be enabled for a centralized BI team only. Be aware that for apps shared with the entire organization, Power BI restricts access to the app to members of your Azure Active Directory, not to guests.</p>
Create template organizational content packs and apps	<p>Permits creation of Template Apps, which are available to anyone in the organization.</p> <p>Recommendation: Since this is applicable to the entire organization, this capability is typically enabled for a centralized BI team only. Also, note that update of a template app will override and remove sensitivity labels applied on template app artifacts.</p>
Push apps to end users	<p>Allows the ability to automatically install a published app on the users' Apps menu. This saves the app viewers from visiting AppSource to install the app. Disabled by default.</p> <p>Recommendation: Pushing apps automatically can simplify and improve the end-user consumer experience. Enable this capability for the teams that you know are using apps for broad distribution, and who have been trained in how to use the capability judiciously. It is a very nice convenience for consumers of the app but be cautious not to confuse or irritate users by having too much unexpected content continue to show up on their apps menu.</p>

Integration Settings

Property	Description
Use Analyze in Excel with on-premises datasets	Controls if Analyze in Excel may be used in conjunction with on-premises Analysis Services datasets. Recommendation: Should be enabled to use on-prem Analysis Services datasets to its fullest extent.
Use ArcGIS Maps for Power BI	Permits users to use the ArcGIS mapping visualizations provided by Esri, subject to Esri's terms and conditions. Recommendation: Should be disabled if the organization does not use Esri services for mapping. This is a specific option because data is sent to Esri for processing.
Use global search for Power BI	This setting allows the use of Azure Search Service, in conjunction with Cortana, for users to locate information in published reports and dashboards. Recommendation: Typically, there are no concerns with this being enabled. Content would be very difficult to locate without the search functionality.
Snowflake SSO	Enables use of single-sign-on for Snowflake, a third-party data warehousing platform. Recommendation: Should be enabled if the organization uses Snowflake as a data source and has the desire to use single sign-on. This is a specific option because data is sent to Snowflake for processing.

Power BI Visuals Settings

Property	Description
Allow visuals created using the Power BI SDK (custom visuals)	Users in the organization can add, view, share and interact with visuals imported from AppSource or a file (custom visuals). Visuals allowed in the "Organizational visuals" page are not affected by the setting. Managing Power BI visuals is discussed in more detail later in this section. Recommendation: Should be enabled for all users for the best experience – in conjunction with the use of certified visuals only.

Property	Description
Allow only certified custom visuals (block uncertified)	<p>Users in the organization with permissions to add and use visuals can add and use certified visuals only. Visuals allowed in the "Organizational visuals" page are not affected by the setting, regardless of certification.</p> <p>Managing Power BI visuals is discussed in more detail later in this section.</p> <p>Recommendation: In addition to this setting which controls usage of certified custom visuals in the Power BI service, Group Policy should also be used to ensure the same experience in Power BI Desktop. Any uncertified visual can still be deployed by the admin through "organizational visuals" page.</p>

R Visuals Settings

Property	Description
Interact with and share R visuals	<p>Specifies whether users across the organization can interact with and share visuals created with R scripts. Primarily a legacy setting.</p> <p>Recommendation: Since R visuals can potentially represent data privacy or data leakage concerns, depending on the individual developer, this should be used with a measure of caution.</p>

Audit and Usage Settings

Property	Description
Create audit logs for internal activity auditing and compliance	<p>Permits actions taken in Power BI to be captured in the audit logs in Office 365.</p> <p>Recommendation: Should always be enabled, without exception. The auditing data is crucial to a well-managed and governed Power BI implementation.</p>
Usage metrics for content creators	<p>Permits individuals to view the usage metrics related to the dashboards and reports they published.</p> <p>Recommendation: Should be enabled so that content creators are able to understand the effectiveness of the solutions they create.</p>

Property	Description
Per-user data in usage metrics for content creators	<p>Specifies whether the names and e-mail addresses of users are exposed via the usage metrics.</p> <p>Recommendation: Typically enabled. Hiding names and e-mail addresses from certain content authors would usually be an exception; the ability to contact users directly about how they use the content can be very valuable to improving the content.</p>

Dashboard Settings

Property	Description
Web content on dashboard tiles	<p>Permits the use of embed codes from external sites like YouTube or Twitter.</p> <p>Recommendation: Typically enabled, though if the organization has a more restrictive security policy this could be restricted.</p>
Data classification for dashboards	<p>Configuring this list will provide a pre-defined list of data classifications which users can assign to dashboards.</p> <p>Recommendation: Sensitivity labels in conjunction with Microsoft Information Protection should be used instead of data classifications (which are highly limited and only apply to dashboards).</p>

Developer Settings

Property	Description
Embed content in apps	<p>Controls who may use the REST APIs to configure a custom application to embed Power BI content.</p> <p>Recommendation: Should be enabled only for specific groups, such as custom developers, as appropriate. Can be disabled when the organization needs to prevent users from embedding content outside of Power BI, without the appropriate approvals, using dedicated capacity (via A, P, or EM-series SKUs).</p>

Property	Description
Allow service principals to use Power BI APIs	<p>Controls which service principals (Azure Active Directory apps) have access to tenant-level features, including the ability to use the Power BI APIs via scheduled operations.</p> <p>Recommendation: Should be enabled for a tightly controlled list of service principals.</p>

Dataflows Settings

Property	Description
Create and use dataflows	<p>Allows dataflows to be created in the Power BI service using Power Query Online, and in turn used as a data source by Power BI datasets. Applicable to the entire organization – it is not possible to permit the use of dataflows for specific groups.</p> <p>Recommendation: Should be enabled to encourage the reusability of data cleansing activities to the extent possible.</p>

Template App Settings

Property	Description
Publish template apps	<p>Allows template apps to be published outside of the organization, such as through AppSource.</p> <p>Recommendation: Typically disabled by most organizations, as this is usually only applicable to software vendors who publish template apps to AppSource.</p>
Install template apps	<p>Specifies who may use template apps which are found in AppSource, such as Salesforce or Google Analytics.</p> <p>Recommendation: Should be enabled to encourage maximum user productivity and user satisfaction. Utilization of third-party template apps often informs corporate BI of new data integration and data warehousing needs.</p>

Property	Description
Install template apps not listed in AppSource	<p>Specifies who may use template apps which are published by a software vendor outside of AppSource. If a template app is not available in AppSource, no validations from Microsoft have occurred.</p> <p>Recommendation: Should be disabled. May be enabled on a very restrictive basis if there is a specific vendor – who is trustworthy – and their template app is not available in AppSource.</p>

Q&A Settings

Property	Description
Review questions	<p>Allows dataset owners to review the Q&A (natural language) questions that were asked. This enables a feedback loop regarding actual usage back to the dataset owner.</p> <p>Recommendation: Should be enabled without restrictions so that Q&A can continue to be improved by dataset owners.</p>

Power BI administrators are not automatically granted every permission which can be controlled via tenant settings. For example, a Power BI administrator may not certify a dataset unless they are also part of the group who is permitted to certify via the tenant setting. However, Power BI administrators do possess privileges to edit all tenant settings directly, including group assignments, so it is considered a very high privilege role.

Managing Custom Help Menus

Some of the links available in the Power BI service may be optionally redirected to custom pages. There are four links you can specifically manage:

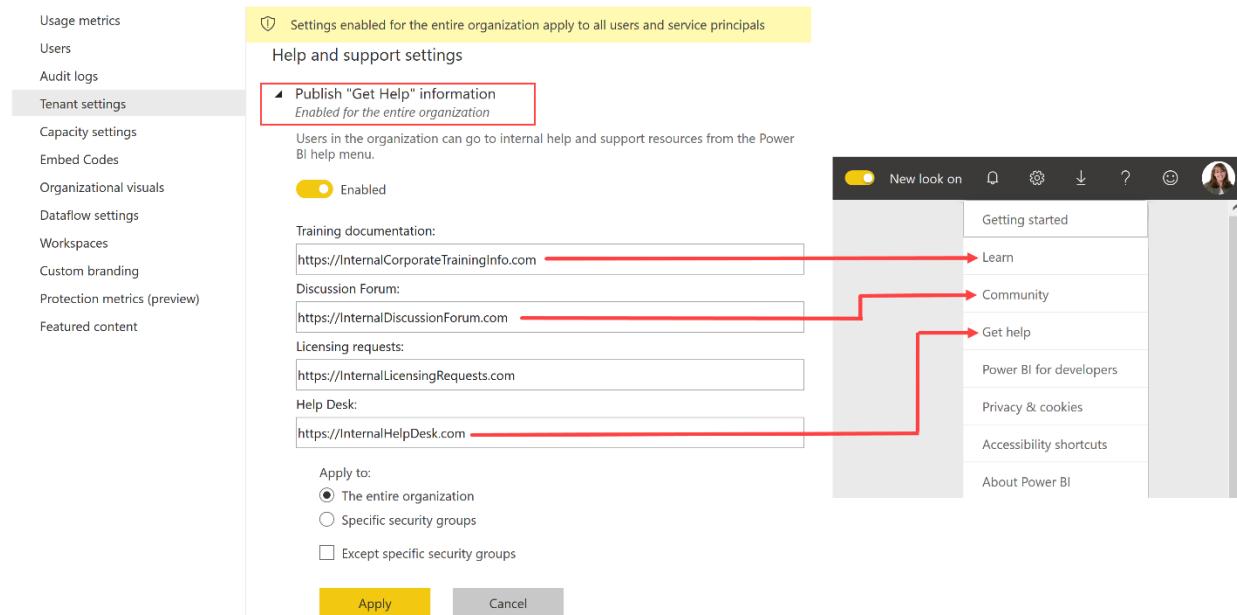
1. Training documentation
2. Discussion forum
3. Help desk
4. Licensing requests

Internal Support Pages

Usage of internal support pages is extremely helpful for organizations who have created support resources to support their internal Power BI community.

Customizing the first three links translates to menu options which are readily accessible in the Power BI service menu:

Admin portal



Usage metrics
Users
Audit logs
Tenant settings
Capacity settings
Embed Codes
Organizational visuals
Dataflow settings
Workspaces
Custom branding
Protection metrics (preview)
Featured content

Help and support settings

▪ Publish "Get Help" information
Enabled for the entire organization

Users in the organization can go to internal help and support resources from the Power BI help menu.

Enabled

Training documentation:

Discussion Forum:

Licensing requests:

Help Desk:

Apply to:
 The entire organization
 Specific security groups
 Except specific security groups

Apply Cancel

New look on Getting started Learn Community Get help Power BI for developers Privacy & cookies Accessibility shortcuts About Power BI

Figure 85 - Customizable URLs in the tenant settings

Licensing Requests

The fourth custom URL, with respect to handling licensing requests, is helpful for organizations who need to govern assignment of user licenses a specific way. See the discussion in [Section 4](#) for additional details about using this URL, and the implications.

Governing the Use of Power BI Visuals and Organizational Visuals

Power BI administrators are responsible for managing approved organizational visuals. Because of the nature of AppSource and visuals from files (custom visuals), they do an inherent level of risk. To exercise caution, there should be a thought-out decision regarding how to govern custom visuals.

There are several types of visuals:

Core visuals	The default core visuals are built into Power BI Desktop and the Power BI service. These are 100% approved Microsoft code and trustworthy.
Certified visuals from AppSource	<p>Certified visuals are created by third-party software vendors or Power BI community members, and they have been additionally tested and inspected by the Power BI team.</p> <p>Although the certification process does not represent a full security guarantee, the following tests are performed during Microsoft's certification process:</p> <ul style="list-style-type: none">• Code reviews• Static code analysis• Data leakage• Data fuzzing• Penetration testing• Access XSS testing• Malicious data injection• Input validation• Functional testing
Non-certified visuals from AppSource	These visuals are created by third-party software vendors or Power BI community members. They passed the AppSource validation tests.
Private organizational custom visuals	These visuals are created internally by an organization or purchased from a vendor due to highly specific or proprietary data requirements. No validations are performed by Microsoft.

There are several aspects of managing Power BI visuals from AppSource and private visuals:

1. **Tenant Settings** in the Power BI service admin portal (discussed in the previous section).
 - a. Allow visuals created using the Power BI SDK (custom visuals) – allows use of AppSource visuals and visuals from file for some or all users.
 - b. Allow only certified visuals – restricts use of visuals from file, and limits use of AppSource visuals to those which are certified by Power BI.
2. **Group Policy** to allow only AppSource certified visuals in Power BI Desktop. This ensures the report author experience in Power BI Desktop aligns with the Power BI service.
3. **Organizational Visuals** in the Power BI service admin portal. This is also referred to as the [Organizational Store](#).

Frequently asked questions about Power BI visuals can be found [here](#).

Additional information about governing the use of custom visuals can be found [here](#).

Managing Use of Power BI Visuals in the Power BI Service

In the tenant settings, there are two settings which control the use of visuals:

Admin portal

Usage metrics
Users
Audit logs
Tenant settings
Capacity settings
Embed Codes
Organizational visuals
Dataflow settings
Workspaces
Custom branding
Protection metrics (preview)
Featured content

Power BI visuals

Allow visuals created using the Power BI SDK
Enabled for the entire organization

Users in the organization can add, view, share, and interact with visuals imported from AppSource or from a file. Visuals allowed in the "Organizational visuals" page are not affected by this setting. [Learn more](#)

Enabled

Apply to:

The entire organization
 Specific security groups
 Except specific security groups

Add and use certified visuals only (block uncertified)
Unapplied changes

Users in the organization with permissions to add and use visuals can add and use certified visuals only. Visuals allowed in the "Organizational visuals" page are not affected by this setting, regardless of certification. [Learn more](#)

Enabled

Apply **Cancel**

Apply **Cancel**

Figure 86 - Tenant settings for managing Power BI visuals

If you choose to disallow the use of all custom visuals in the tenant settings because it's too broad as compared to your risk profile, you can still register specifically allowed visuals on an individual basis by using the "Organizational visuals" page. This alternative approach is discussed later in this section.

Managing Use of AppSource Visuals and Visuals from a File in Power BI Desktop

The tenant settings discussed in the previous section only apply once the .pbix file has been published to the Power BI service. Therefore, there is possibility that a report author could use an AppSource visual or visual from a file in Power BI Desktop, only to find out it is disallowed in the Power BI service. This is frustrating for report authors, so it is a situation that we want to avoid.

Power BI Desktop supports the use of Group Policy for aligning the behavior in Power BI Desktop to that in the Power BI service for AppSource visuals and visuals from a file. The Group Policy attributes and values for visuals can be found [here](#). Note that this only applies to machines which are managed by the organization.

 Even with the settings, it's important to have transparent documentation for the report authors so they understand why decisions like this are made, and so they understand why functionality they read about online may or may not work for them.

Note that certain features for AppSource visuals are *not* supported unless the visual is certified. Examples include displaying the visual within report subscriptions or exports to PPTX.

When an update to a visual is released in AppSource, it will automatically propagate to all Power BI reports and dashboards using that visual. Even if a user imports a specific (older) .pbviz file in Power BI Desktop, it will be overwritten by the latest version from AppSource. Usually this convenience is desirable, but if additional control is required for managing the specific version then organizational visuals (discussed next) are a good option to consider (will require the visual to have a unique ID in order to not be overwritten by the AppSource version).

Managing Use of Organizational Visuals

Organizational visuals are usually private or proprietary visuals which are not available in AppSource. They may be developed internally or purchased from a third-party vendor.

Alternatively, they could be publicly available visuals from AppSource. If a visual is not certified, but has been approved for use within the organization, this is a useful way to distribute the visual. There are future items on the release plan to [treat organizational visuals as out-of-the-box visuals](#), and for [managing AppSource visuals](#).

In the Power BI admin portal, the organizational visuals page allows an administrator to distribute specific visuals to all users throughout the organization:

Admin portal

NAME	PUBLISHED BY	LAST UPDATED	ACTIONS
Corporate Custom Scatterplot	Payton Adamsen	Feb 27, 2020	

Figure 87 - Organizational visuals management in the Power BI admin portal

A Power BI administrator uploads a private visual file (a .pbviz file) which has been developed, evaluated, tested, and approved for use within the organization. Once a visual has been added, report authors will be able to import it to a Power BI Desktop file. The user selects “Import from AppSource” followed by “My Organization.” If an organizational visual is deleted from the Organizational visuals page, it will become disabled in all reports and dashboards which use it. If the organizational visual is updated, all reports and dashboards will be automatically updated.

Using the organizational visuals ensures that all Power BI users within the tenant are using the same version. This approach, though it requires administrator involvement, is significantly easier to manage than the alternative: if content authors use .pbviz files not in conjunction with the organizational visuals store (or AppSource), every Power BI Desktop file would need to be opened and edited in order to propagate an update.



Power BI visual certification guidelines are good practices to follow

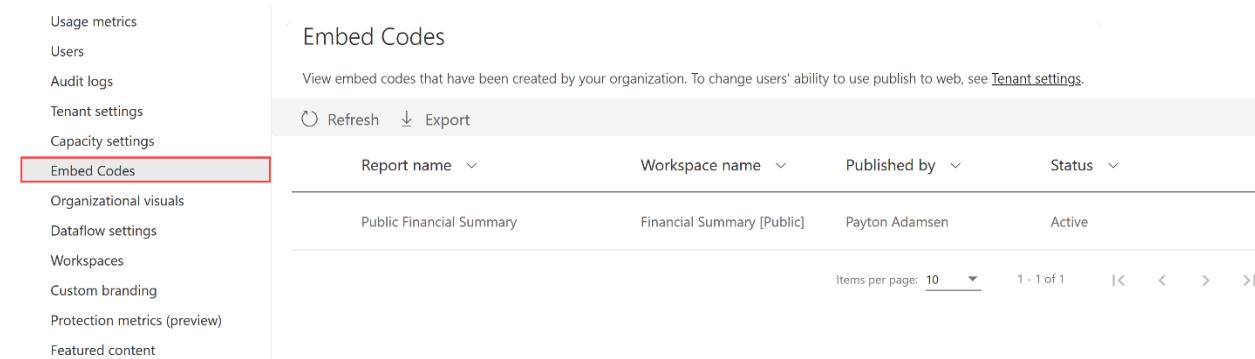
If you are developing a visual, even if you do not intend to make the visual publicly available, we recommend following as many of the [guidelines for getting a custom visual certified](#) as possible. Once your visual is ready to be made available to users in your organization, deploy it through the organizational visuals page.

Managing Embed Codes for Publicly Available Reports

One capability that is available to all users, even those with a Power BI Free license, is the Publish to Web capability. We do **recommend this capability be disabled** for all users and only enabled very selectively on request. This feature is intended for data journalism and other public visuals.

All reports which have an embed code associated with them, across all workspaces, can be viewed in the Power BI admin portal:

Admin portal



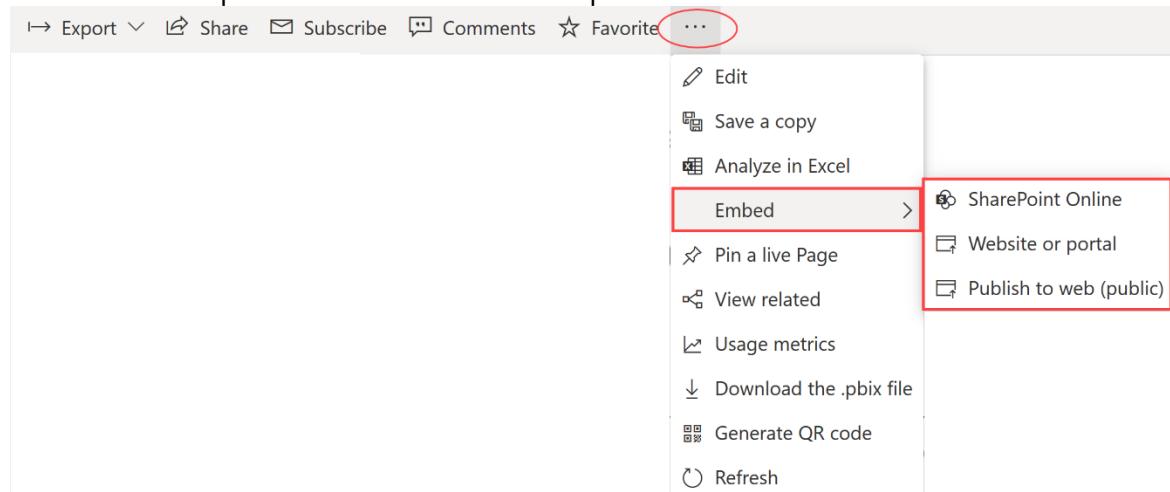
Report name	Workspace name	Published by	Status
Public Financial Summary	Financial Summary [Public]	Payton Adamsen	Active

Figure 88 - Embed codes management in the Power BI admin portal

The status shown above can be Active, Not Supported, Blocked, or Infringed. The meaning of these codes can be found [here](#).



Although you may feel that an internal website is "safe enough" to use the Publish to Web capability, that technique is strongly discouraged. It is a much more secure practice to use the [API embedding](#) functionality, or use the [no-code embed option](#) which is available on the "Website or portal" embed menu for a report or dashboard:



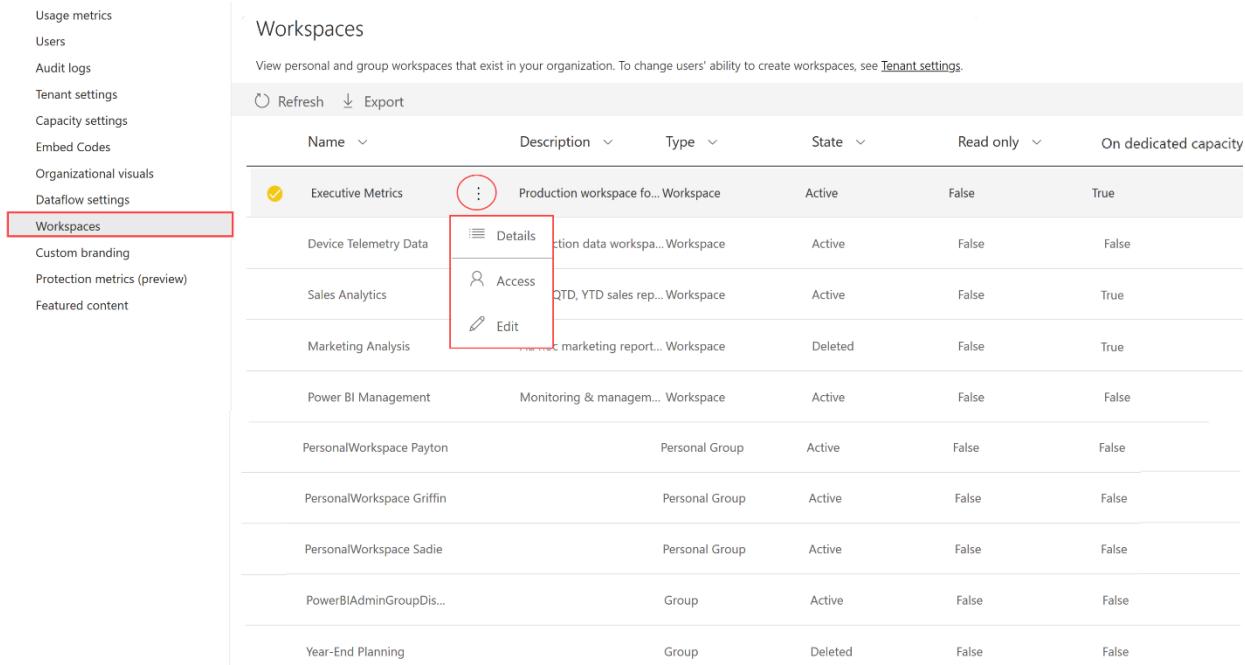
- Edit
- Save a copy
- Analyze in Excel
- Embed >
- Pin a live Page
- View related
- Usage metrics
- Download the .pbix file
- Generate QR code
- Refresh

Figure 89 - Embed menu options available for a report or dashboard in the Power BI service

Managing Workspaces

The [workspaces page](#) in the admin portal allows Power BI administrators to review what workspaces are present in the Power BI tenant:

Admin portal



Name	Description	Type	State	Read only	On dedicated capacity
Executive Metrics	Production workspace fo...	Workspace	Active	False	True
Device Telemetry Data	ction data workspa...	Workspace	Active	False	False
Sales Analytics	QTD, YTD sales rep...	Workspace	Active	False	True
Marketing Analysis	Marketing report...	Workspace	Deleted	False	True
Power BI Management	Monitoring & managem...	Workspace	Active	False	False
PersonalWorkspace Payton	Personal Group	Active	False	False	False
PersonalWorkspace Griffin	Personal Group	Active	False	False	False
PersonalWorkspace Sadie	Personal Group	Active	False	False	False
PowerBIAdminGroupDis...	Group	Active	False	False	False
Year-End Planning	Group	Deleted	False	False	False

Figure 90 - Workspace management in the Power BI service

Workspace access may be edited directly from this page (as well as from the Power BI REST APIs). There are times this can be extremely helpful – such as when a Power BI administrator needs to assist while someone is out of the office.

 **The Power BI administrator role is a high privilege role**
Access to a workspace can be granted not only for other users, but to Power BI administrators themselves. We recommend having an audit process in place to review workspace permission changes.

Managing Custom Branding

A logo, cover image, and theme color may be set for [custom branding](#) of the Power BI service, in alignment with corporate standards.

<p>The use of branding is not only about attractiveness. The use of consistent branding is helpful for users in the organization to identify corporate content.</p> <p>Custom branding is also a significant help for B2B users to recognize which tenant they are in.</p>	<p>Admin portal</p> <p>Usage metrics Users Audit logs Tenant settings Capacity settings Embed Codes Organizational visuals Dataflow settings Workspaces Custom branding Protection metrics (preview) Featured content</p> <p>Custom branding</p> <p>Customize the look of Power BI for your whole organization. Learn more</p> <p>Logo</p> <p>For best results, upload a logo that's saved as a .png, 10 KB or smaller, and at least 200 x 30 pixels.</p> <p><input type="file"/> Upload Delete</p> <p>Cover image</p> <p>For best results, upload a cover image that's saved as a .jpg or .png, 1 MB or smaller, and at least 1920 x 160 pixels.</p> <p><input type="file"/> Upload Delete</p> <p>Theme color</p> <p>#06547F</p> <p>Remove custom branding</p> <p>Preview Publish</p>
--	--

Figure 91 - Custom branding settings in the Power BI admin portal

Managing Protection Metrics

The protection metrics page provides a high-level overview of how many objects are assigned to each sensitivity label. An overview of sensitivity labels is included in [Section 11](#).

Admin portal

Usage metrics
Users
Audit logs
Tenant settings
Capacity settings
Embed Codes
Organizational visuals
Dataflow settings
Workspaces
Custom branding
Protection metrics (preview)
Featured content

Data protection metrics report (preview)

This report shows how Power BI sensitivity labels help protect your content.

Power BI data protection is integrated with Microsoft Cloud App Security [Learn more](#)

[Open the Cloud App Security portal to view more metrics](#)

Data Protection Summary

Total sensitivity labels for all over time

Highly Confidential Internal Use None Restricted

Feb 2020 Mar 2020

Last 7 days Last 30 days Last 90 days

Sensitivity labels in your organization today

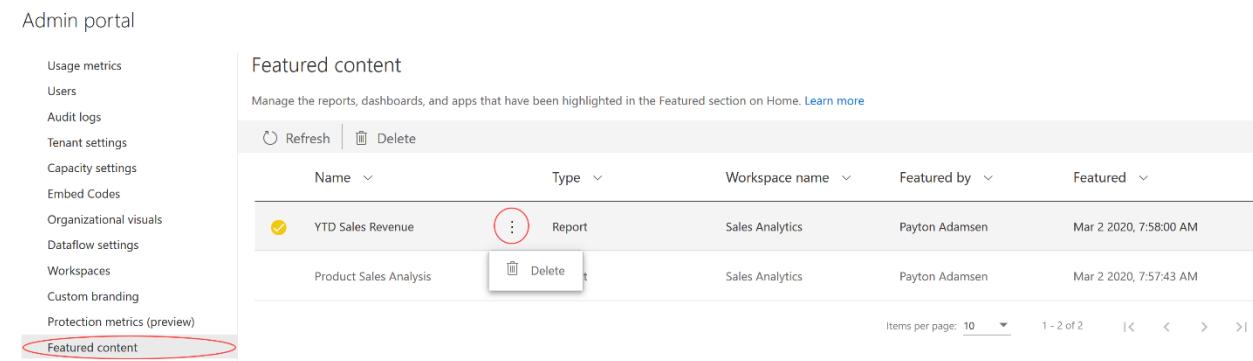
Category	Total	Label	Count	Percentage
Dashboards	11	Highly Confidential	2	18.18%
Dashboards	11	Internal Use	9	81.82%
Reports	28	Highly Confidential	3	10.71%
Reports	28	Internal Use	15	53.57%
Reports	28	None	8	28.57%
Datasets	16	Highly Confidential	1	6.25%
Datasets	16	Internal Use	2	12.5%
Datasets	16	None	11	68.75%
Dataflows	9	Highly Confidential	1	11.11%
Dataflows	9	Internal Use	8	88.89%

Figure 92 - Data protection metrics report in the Power BI admin portal

Managing Featured Content

The [featured content page](#) provides an organizational-wide view of all the reports, dashboards, and apps that have been promoted to the Featured section on Power BI Home.

Admin portal



Name	Type	Workspace name	Featured by	Featured
YTD Sales Revenue	Report	Sales Analytics	Payton Adamsen	Mar 2 2020, 7:58:00 AM
Product Sales Analysis	Report	Sales Analytics	Payton Adamsen	Mar 2 2020, 7:57:43 AM

Figure 93 - Featured content management in the Power BI admin portal

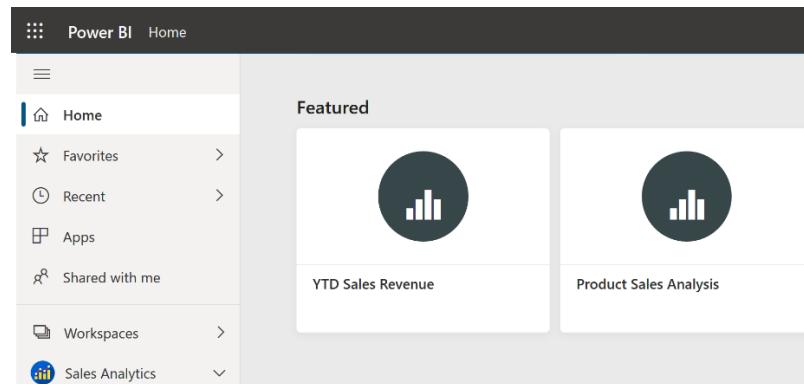


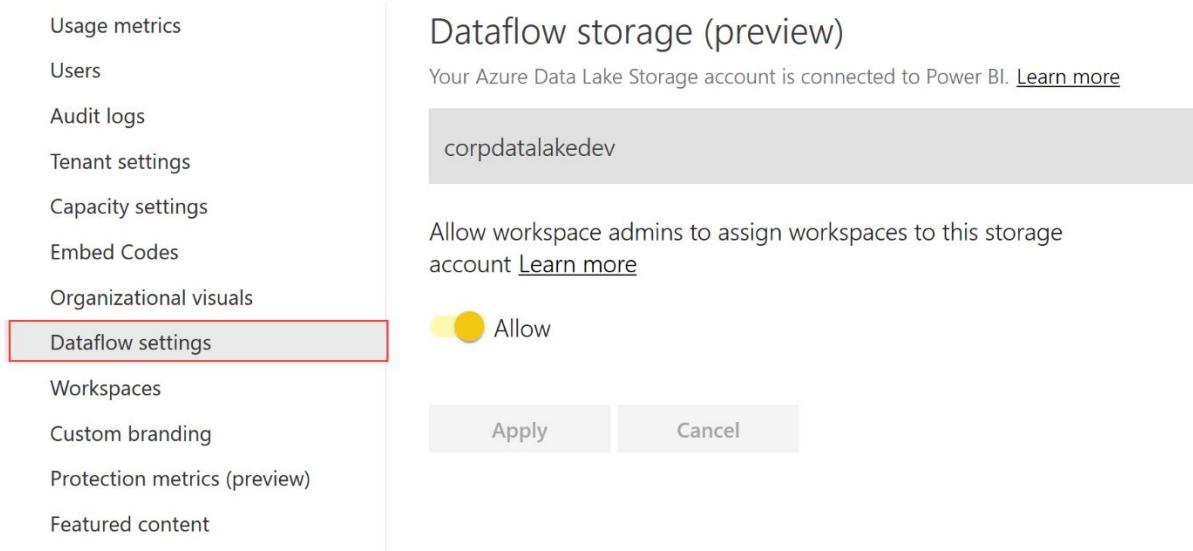
Figure 94 - Featured content on the Home page in the Power BI service

The Power BI administrator may delete an item from featured content when appropriate. Users who are authorized to set featured content are defined in the tenant settings.

Managing Dataflow Settings

Organizations who have an existing Azure Data Lake Storage Gen 2 account they would like to use for storage of all, or some, dataflows may set the account in the [dataflow settings](#). See [Section 3](#) for additional discussion, including limitations to be aware of.

Admin portal



The screenshot shows the 'Dataflow storage (preview)' settings in the Power BI Admin portal. On the left, a sidebar lists various settings: Usage metrics, Users, Audit logs, Tenant settings, Capacity settings, Embed Codes, Organizational visuals, Dataflow settings (which is highlighted with a red border), Workspaces, Custom branding, Protection metrics (preview), and Featured content. The main panel displays the 'Dataflow storage (preview)' configuration. It shows a connection to 'Your Azure Data Lake Storage account is connected to Power BI. [Learn more](#)' with the account name 'corpdatalakedev'. Below this, there is a description: 'Allow workspace admins to assign workspaces to this storage account [Learn more](#)'. A toggle switch is set to 'Allow', indicated by a yellow circle. At the bottom of the panel are 'Apply' and 'Cancel' buttons.

Figure 95 - Dataflow storage settings in the Power BI admin portal

Monitoring Usage of Power BI

Monitoring the activity which occurs in Power BI is highly recommended. Apart from compliance, there are a variety of other reasons to monitor usage, including:

- To see which users are making frequent use of Power BI and which users are not. If Power BI is not being used as expected, it will be necessary to investigate why. More training may be appropriate. Alternatively, it may be that published content is not meeting the actual user needs. Or, perhaps reports are slow to render or respond and need tuning to improve the user experience.
- To identify commonly used sources and types of data. Frequently a corporate BI environment can be improved upon by learning what is occurring via self-service BI initiatives.
- To be aware of where the most critical and commonly used content arises, so support can be provided accordingly.
- To determine which workspaces are used the most, potentially being candidates for moving into Power BI Premium capacity.
- To look for suspicious usage patterns. For instance, users who are exporting significant amounts of data.

- To understand how published content is being used. For instance, perhaps users are exporting data because the reports as designed are not meeting expectations.
- To identify what type of devices are being used to view reports.
- To identify when changes occur, and by whom.

Power BI Service Usage

There are several types of auditing related to Power BI:

Type of Auditing	Scope	Targeted Towards	Where to Locate	Retention Period
Usage metrics	Individual reports + dashboards	Report authors and editors	Usage metrics icon in the page header for report or dashboard	30 days
Admin usage metrics	Power BI tenant	Power BI administrator	Power BI admin portal	90 days
Power BI activity log	Power BI tenant	Power BI administrator	PowerShell	30 days
Office 365 unified audit log	Power BI tenant + other Microsoft 365 services in Audit.General	Microsoft 365 administrator	M365 Admin Center or API or PowerShell	90 days

Each of the above 4 categories are discussed in the following sections.

Usage Metrics for Reports and Dashboards

The usage metrics applicable to individual reports and dashboards are available from a built report available in the Power BI service. Report authors who have contributor, member, or admin permissions for the workspace may view the usage metrics report.

Figure 96 - Menu option to view usage metrics report for a report in a workspace

The usage metrics reports are targeted towards report authors so they can see usage patterns and assess the impact of the solutions they have published. Up to 30 days of history may be viewed, including the most recent fully complete day.

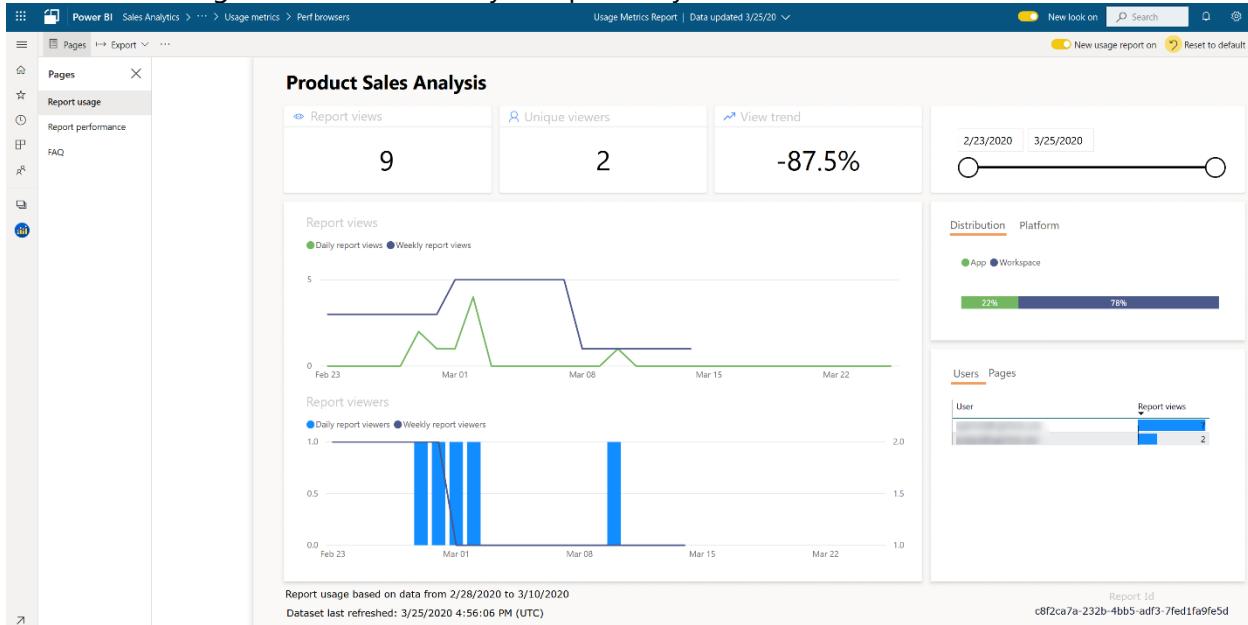


Figure 97 - Example of usage metrics report

[Usage metrics reporting for the new workspace experience](#) is significantly improved from what is available for [usage metrics in the classic workspace experience](#). Previous client-side issues with how data was collected are also rectified with the improved usage metrics report, which means that the usage metrics reports (for content in the new workspace experience) will match the Power BI activity log and Office 365 unified audit log.

A report author may save and customize a copy of the usage metrics report as discussed in [the documentation](#). One benefit of customization is that the filter can be updated to view usage auditing for all reports or dashboards in a workspace rather than just one. Using this approach, the broadest range is still a workspace and the history element is still 30 days. For retrieving tenant-wide activity, see the Power BI Activity Log discussed later in this section.

Usage metrics are captured for activity which occurs within the Power BI service. Usage metrics are only available for commercial cloud customers, not for [national cloud](#) customers.



This type of usage reporting can be disabled in the tenant settings in the Power BI admin portal. However, we recommend enabling usage metrics for the entire organization to allow content authors to see the impact of their work.

The second tenant-level setting allows the activity for individual users to be identified for the authors. This can be helpful for the author to know who to go speak with if they have questions. If per-user data is disabled, the metrics appear as "Unnamed users."

Usage metrics for content creators

Enabled for the entire organization

Users in the organization can see usage metrics for dashboards and reports they created.



Apply to:

The entire organization

Specific security groups

Except specific security groups

Apply

Cancel

Per-user data in usage metrics for content creators

Enabled for the entire organization

Usage metrics for content creators will expose display names and email addresses of users who are accessing content.



Apply to:

The entire organization

Specific security groups

Except specific security groups

Apply

Cancel

Figure 98 - Tenant settings for usage reporting

Admin Usage Metrics

The admin usage metrics is a capability which is on the [release plan to receive improvements](#). The objective of the improvements is to provide Power BI administrators with a tenant-wide view of all activity to manage and govern the Power BI environment.

The current usage metrics page in the admin portal provides a built-in usage metrics dashboard which focuses on displaying high-level metrics for activity across the entire Power BI tenant.

This dashboard differs from the usage metrics reporting discussed in the previous section because it traverses beyond the scope of one report or one workspace.

This dashboard provides a quick high-level view of the most popular dashboards, reports, users, and groups but it does have several limitations:

- o This dashboard cannot be modified or customized
- o The tiles do not drill-through to an underlying report
- o There are no filters or interactivity

Until the new admin usage metrics view is released, to see additional detail we recommend using the Power BI activity log which is discussed next.

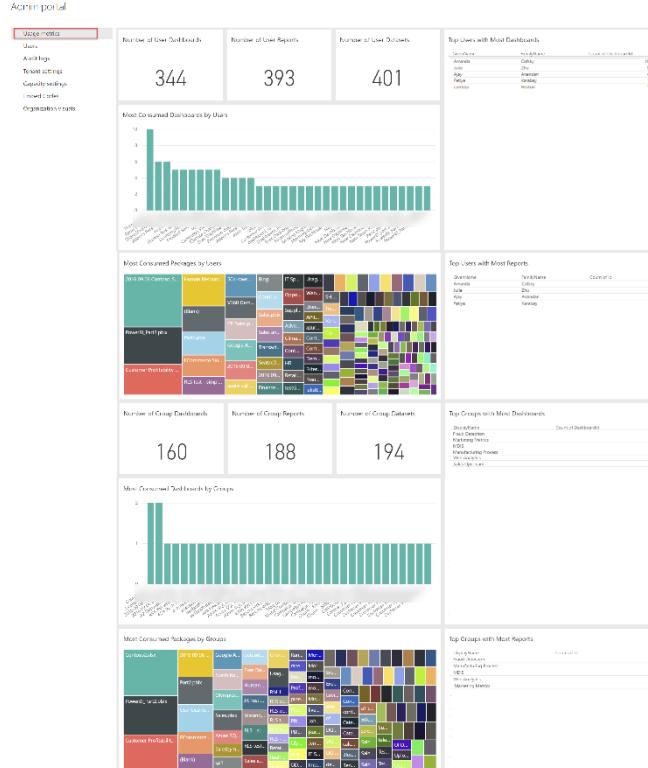


Figure 99 - Admin usage metrics report

Power BI Activity Log

The Power BI Activity Log is the newest addition to the options available for monitoring activity in the Power BI service. You can obtain data one of two ways:

- a. The PowerShell cmdlet to obtain data is "[Get-PowerBIActivityEvent](#)" which is available as part of the [Power BI Management Module](#).
- b. The [ActivityEvents](#) REST API.

The advantages of using the Power BI Activity Log, as compared to the Office 365 Unified Audit Log (which is discussed next) include:

- ✓ Power BI administrator permissions can be used to query the Power BI Activity Log, whereas higher-level Microsoft 365 permissions required to view the unified audit log.
- ✓ Only Power BI activity is returned, rather than all activity from the Audit. General events in the unified audit log. This reduces the amount of data and simplifies the process if the only data you require is from Power BI.
- ✓ Results are not limited in the amount of rows the PowerShell cmdlet can return (as is the case with the older Search-UnifiedAuditLog PowerShell cmdlet), and the continuation token to page through each batch of results are handled as a convenience if you use the PowerShell cmdlet.

There are over 100 types of activities captured by the Power BI activity log, which are documented [here](#). In addition to user activity for understanding usage patterns, there are certain administrative events you may wish to monitor, such as:

- Updated organization's Power BI settings (i.e., tenant settings are updated)
- Changed Power BI gateway admins
- Deleted Power BI gateway
- Update capacity resource governance settings
- Updated capacity admin
- Removed workspace from a capacity



Terminology Tips

In the audit log data, workspaces are referred to as folders.

In the Power BI REST API, workspaces are referred to as groups.

In PowerShell, V1 workspaces are called groups (new workspaces are called workspaces).

Office 365 Unified Audit Log

Prior to the introduction of the Power BI Activity Log, the recommended way to obtain auditing data was via the unified audit log in the Office 365 Security and Compliance Center. However, that approach involves additional effort plus elevated Microsoft 365 and Exchange Online permission levels beyond the Power BI Administrator role.



At this time, we recommend using the Power BI Activity Log which was discussed in the previous section, unless you have an existing investment in data extraction from the unified audit log in Office 365 and/or you need audit data from more services in addition to Power BI. A comparison of functionality can be found [here](#).

Power BI integrates its logs with the unified audit log. This is an advantage because every individual service within Microsoft 365 does not have to implement its own unique way of logging.

There are a few ways you can access this data:

1. **Office 365 Security & Compliance Center.** The [audit log can be searched](#), and results may be exported to a CSV file. Although convenient to have a user interface, this approach is highly limited because:
 - a. It is a manual process, relying on a user interface. That makes this option best for occasional or targeted use.

- b. There are too many events within the Power BI category to choose them all using the user interface. If you try to choose the entire Power BI category, an error is received.
- c. The export is limited to 1,000 rows. Auditing data can be quite voluminous.

Date	IP address	User	Activity	Item	Detail
2020-03-25 ...	174.108.30...		Viewed Power BI report	Usage Metrics - Product Sales Analysis	
2020-03-25 ...	174.108.30...		Viewed Power BI report	Usage Metrics - Product Sales Analysis	
2020-03-25 ...	174.108.30...		Created Power BI report	Usage Metrics - Product Sales Analysis	
2020-03-26 ...	174.108.30...		Requested Power BI dataset refresh	Usage Metrics Report	
2020-03-25 ...	174.108.30...		Viewed Power BI report	Usage Metrics Report	

Figure 100 - Audit log search in the Microsoft 365 Security & Compliance center

2. **Office 365 Management Activity API:** If you intend to extract data from the unified audit log rather than the Power BI activity log, using the [Office 365 Management Activity API](#) is the recommended approach. It is recommended for programmatically downloading data from Microsoft 365 on a schedule because it is more scalable. A few considerations:
 - The API can be called from within PowerShell or the tool of your choice.
 - The API requires using an Azure Active Directory app (service principal) for OAuth authentication which has [permission to retrieve data from the API](#). Specifically, it needs ActivityFeed.Read and ServiceHealth.Read from the O365 Management API, plus User.Read.All from Microsoft Graph.
 - Per the [API schema](#), filtering to Power BI audit events can be done with an AuditLogRecordType of 20 (once all of Audit.General events have been obtained, because that is the lowest granularity of filtering beyond the UTC datetime).
 - By default, the API returns 24 hours of data. A maximum of 7 days history can be retrieved.
 - The API returns content blobs, and each content blob contains individual audit records. For large organizations, there could be thousands of content blobs per day.
 - The article on [troubleshooting the Office 365 Management API](#) is very helpful. FAQs are also available [here](#).
3. **PowerShell cmdlet:** Using the [Search-UnifiedLog](#) cmdlet within PowerShell returns records from the unified audit log as well. A few considerations:
 - Note this is a different, older, PowerShell cmdlet than was discussed in the previous section.

- We do not recommend that you use this cmdlet except for occasional queries, or for very small companies, because it is not scalable. Although you could implement a looping strategy to overcome the 5,000-row limit, it is generally recommended that using this PowerShell cmdlet be reserved for occasional or ad hoc export operations rather than automated exports on a schedule.
- Using this technique does make more history available as compared to the API in option 2. However, due to limits imposed to protect the health of the system, retrieving large amount of history can be extremely challenging in large organizations due to throttling.
- A lot of bloggers have talked online about this option, so scripts are readily available online, but again we do not recommend using this on a regular basis.

Summary of Recommendations for Extracting + Storing Audit Log Data for the Entire Power BI Tenant

If you only need Power BI audit data: We recommend looking at using the Power BI Activity Log first for the reasons stated above. Generally, using the PowerShell cmdlet is easier than using the API. Both are discussed [here](#). Consider using a domain service account for the authentication, rather than an individual Power BI administrator.

If you intend to extract data from multiple services beyond Power BI: We recommend using the Office 365 Management Activity API, as it is the recommended option for regularly scheduled programmatic data extracts from the unified audit log. Plan to invest some time in building this out, as it takes some effort to arrive at a production-ready script. More information is discussed [here](#).

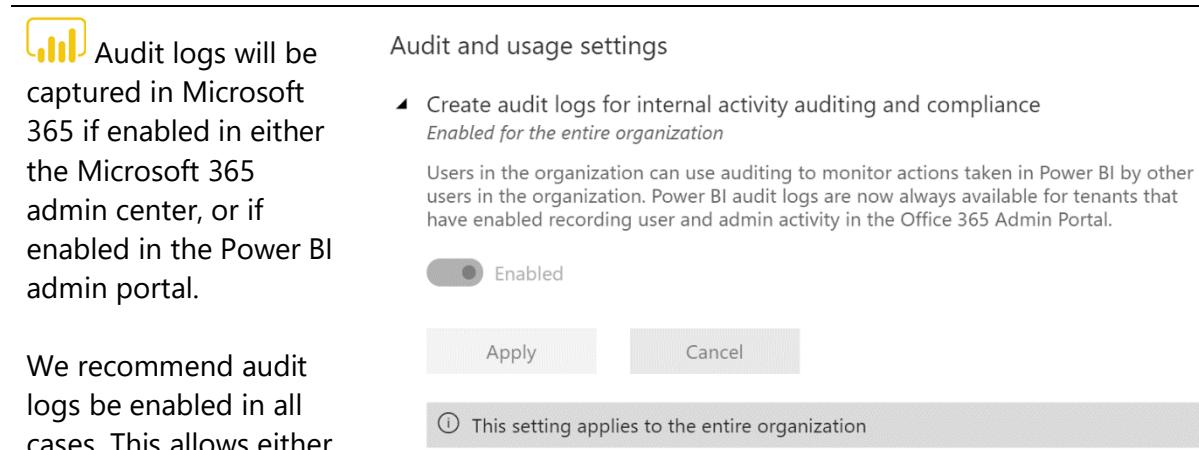
Timing for extracting data: We recommend extracting one day at a time, and generally 1-2 days back from the current day. There is the possibility for late-arriving events, so this minimizes the possibility some audit events will be missed.

Retaining the raw data: As Power BI continues to gain new features, additional [audit events](#) will be captured that do not exist today. For this reason, we recommend retaining the raw JSON data so that if the original raw data needs to be re-parsed in the future it can be done. A data lake or blob storage is commonly for retention of raw data.

Storing history: We recommend planning to export the audit data to a location where the history can be stored. Retaining up to 2 years of history is valuable so that trending and changes over time can be analyzed beyond the default 90-day retention period. Additionally, many organizations use the stored audit log for compliance audits or to respond to security investigations. The storage could be a database, a data lake, or object storage depending on your preference. The original data is in JSON format. Parsing it out to a database table does add additional data integration work, yet makes the data analysis process significantly easier. Once

the audit log history has been written to the data storage of your choice, and curated into a data model suitable for reporting, then Power BI can be effectively used to handle all the reporting and analytical capabilities.

Augment the audit log data: The activity log data can be correlated with other sources to augment the analysis, such as Power BI workspace tenant inventory, workspace security, Azure Active Directory user data, and so on. Integrating the data with these other areas greatly improves the value and usefulness of the audit data.



Audit logs will be captured in Microsoft 365 if enabled in either the Microsoft 365 admin center, or if enabled in the Power BI admin portal.

We recommend audit logs be enabled in all cases. This allows either the Power BI Activity Log or the Office 365 Unified Audit Log to accumulate data.

Audit and usage settings

- Create audit logs for internal activity auditing and compliance
Enabled for the entire organization

Users in the organization can use auditing to monitor actions taken in Power BI by other users in the organization. Power BI audit logs are now always available for tenants that have enabled recording user and admin activity in the Office 365 Admin Portal.

Enabled

Apply Cancel

(i) This setting applies to the entire organization

Figure 101 - Tenant setting for audit logs

Power BI Report Server Usage

The capabilities for tracking report executions for Power BI Report Server are consistent with how auditing is handled for SQL Server Reporting Services server running in native mode. Specifically, a database view within the Report Server catalog database called "[ExecutionLog3](#)" can be queried with T-SQL commands. The execution log captures information to determine things such as:

- How often a report is executed
- Duration of report execution, including time spent for data retrieval vs. processing the report vs. rendering the report
- What output formats are being used to render a report
- The extent that e-mail subscriptions are used
- Parameter values selected when rendering a report
- Data size and row counts returned from a report

Because Power BI Report Server is server-based product (whereas the Power BI service is a cloud-based software-as-a-service), there are also additional logs for monitoring and managing the environment which are discussed in [this documentation](#).

Automating Power BI Management

In addition to the user interface discussed elsewhere in this whitepaper, there are multiple options for managing Power BI in a programmatic way:

- [PowerShell cmdlets](#)
- [Power BI REST APIs](#)
- [.NET SDK](#) (applicable more to Power BI Embedded than for administration)

Using the PowerShell cmdlets is generally the easiest to get started with, because some complexity has been obfuscated. However, only a subset of the API functionality is available currently with PowerShell cmdlets. Therefore, you will likely find yourself using a combination of both.

Power BI Management PowerShell Cmdlets

The [Power BI Management Module](#) is a collection of PowerShell modules for handling a variety of tasks.

We recommend installing all the modules, so they are all available. It is also important to keep the installed modules updated on a regular basis. At a minimum, you will need the Profile module + the child module containing the cmdlet(s) you're interested in.

Description	Module Name
Rollup module for Power BI Cmdlets	MicrosoftPowerBIMgmt
Admin module for Power BI Cmdlets	MicrosoftPowerBIMgmt.Admin
Capacities module for Power BI Cmdlets	MicrosoftPowerBIMgmt.Capacities
Data module for Power BI Cmdlets	MicrosoftPowerBIMgmt.Data
Profile module for Power BI Cmdlets	MicrosoftPowerBIMgmt.Profile
Reports module for Power BI	MicrosoftPowerBIMgmt.Reports
Workspaces module for Power BI	MicrosoftPowerBIMgmt.Workspaces

Figure 102 - PowerShell modules available

Resources related to this set of modules are available on [GitHub](#). If you are deeply interested in this functionality, following the [issues list in GitHub](#) can be very educational. There are also opportunities to [contribute code](#) to the open source repository.

Common administrative tasks which can be automated with this module include things such as:

- Compiling a list of workspaces and their properties [Get-PowerBIWorkspace]
- Compiling a list of objects per workspace [Get-PowerBIReport, Get-PowerBIDashboard, Get-PowerBIDataset, Get-PowerBIDataflow]
- Extracting Power BI activity log events [Get-PowerBIACTivityEvent] (discussed in previous section)
- Authenticating to the Power BI service [Connect-PowerBIServiceAccount] (there are aliases for this as well)
- Executing a REST API command [Invoke-PowerBIRestMethod]
- Enabling customer-managed key encryption for Premium capacity [Add-PowerBIEncryptionKey]

The Power BI Management Module can be used not only by administrators, but also by technical user who wish to automate certain aspects of managing their content:

For administrators: You can set the `-Scope` to 'Organization' in order to return metadata for the entire tenant (such as a list of workspaces).

For technical users: Setting the `-Scope` to 'Individual' (or omitting it) will return metadata based on your permissions (such as workspaces you have permission to).

Since authentication `[Connect-PowerBIServiceAccount]` is done with a domain account, it is simpler to get started (versus interacting directly with the APIs using an AAD app).

A very common scenario is to use the Power BI Management Module for calling a REST API, because the REST APIs offers many more options than the PowerShell cmdlets do directly. The advantage of this for administrators is simplified authentication, coupled with the flexibility to call any of the APIs. Following is a common flow:

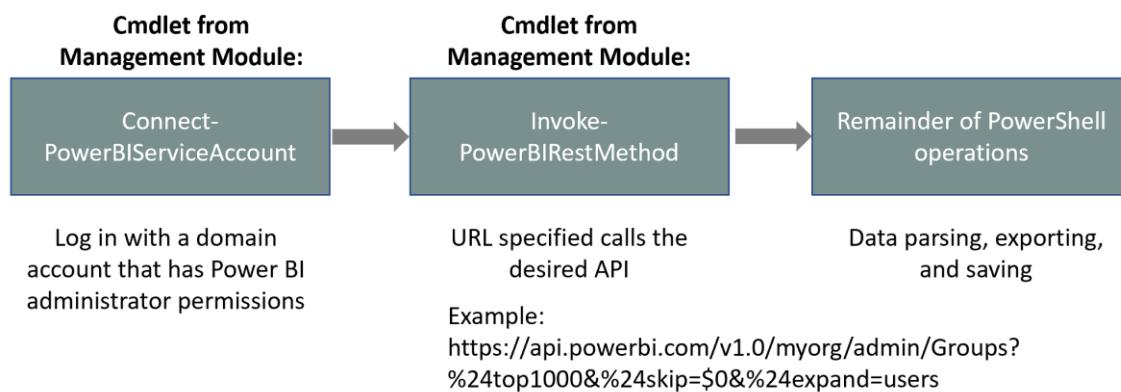


Figure 103 - High-level flow for using the PowerShell management module for authentication and to call a Power BI API

Additional information for getting started using the PowerShell cmdlets:

- [PowerShell Cmdlets, REST APIs, and .NET SDK for Administration](#)
- [PowerShell Cmdlets Reference](#)
- [Announcing APIs and PowerShell Cmdlets for Administrators](#)
- [Working with PowerShell in Power BI](#)
- [PowerShell Gallery for Power BI Management Module](#)

Power BI Data Gateway PowerShell Cmdlets

A set of PowerShell cmdlets is available for tasks involved with managing the On-Premises Data Gateway, such as:

- Retrieve list of gateway clusters [Get-DataGatewayCluster]
- Check status for a gateway [Get-DataGatewayClusterDatasourceStatus]
- Update attributes for a gateway [Set-DataGatewayCluster]
- Add data gateway users [Add-DataGatewayClusterUser]
- Define which users in the organization are authorized to register new gateways [Set-DataGatewayInstaller]
- Define policy for if personal gateways may be installed [Set-DataGatewayTenantPolicy]

 The minimum PowerShell version supported for the data gateway module is PowerShell Core 6.2.2. This is different than the Power BI Management Module discussed in the last section, which permits a minimum version of PowerShell 3.0 or use of PowerShell Core.

Like the Power BI Management Module, the Data Gateway Module also allows a -Scope of 'Organization' or 'User.' Use of the organization scope provides a tenant-wide view when executed by a Power BI administrator.

There is not currently a cmdlet for unattended gateway installations. Data gateway updates are typically released every month, and we recommend staying as current as possible with the data gateway version. See additional information in [Section 7](#).

Additional information for getting started using the data gateway cmdlets:

- [Data Gateway PowerShell Reference](#)
- [PowerShell Gallery for Data Gateway PowerShell Module](#)
- [On-Premises Data Gateway Management Via PowerShell](#)

Power BI REST APIs

The [Power BI REST APIs](#) provide service endpoints for embedding, administration, and user resources. Currently, the following groups of Power BI REST APIs are available:

Operation group	Description
Admin	Operations for working with administrative tasks.
Apps	Operations for working with Apps.
Available Features	Operations that return available features.
Capacities	Operations for working with capacities.
Dashboards	Operations for working with dashboards.
Dataflow Storage Accounts	Operations for working with dataflow storage accounts.
Dataflows	Operations for working with dataflows.
Datasets	Operations for working with datasets.
Embed Token	Operations for working with embed tokens.
Gateways	Operations for working with gateways.
Groups	Operations for working with groups.
Imports	Operations for working with imports.
Push Datasets	Operations for working with push datasets.
Reports	Operations for working with reports.
Users	Operations for working with users.

Figure 104 - Types of operations available with the Power BI REST APIs

In addition to data management and working with embedded content, the Power BI REST APIs can be used to handle many common administrative activities. For example:

- Export a report
- Clone a report to another workspace
- Trigger a data refresh for a dataset
- Check refresh history for a dataset
- Grant user permissions to a workspace
- Take ownership of a dataset
- Update credentials for a data source registered to a data gateway

There are significantly more options available with the REST APIs currently than through PowerShell cmdlets.

Resources with documentation and sample scripts include:

- [Power BI REST API Reference](#)
- [Power BI Embedded REST API Reference](#)
- [Power BI REST API with 'Try It' Tool](#)
- [What Can Developers Do with the Power BI API?](#)
- [Power BI Developer Center](#)

Using Power Automate

In addition to the Power BI REST APIs, integration with Power Automate can be helpful for certain data management and administrative tasks.

One example is using a Flow in Power Automate to start a [data refresh operation](#). This is a useful technique when there is another dependency before the refresh may begin. When the data refresh is not scheduled in the Power BI service, we recommend including the refresh information within your dataset documentation.

For additional ideas on using Power Automate for process automate, search for "Power BI" in the [Power Automate templates](#).

PowerShell Cmdlets for Power BI Report Server

The PowerShell Gallery contains a set of [cmdlets which can be installed for managing Power BI Report Server](#). The PowerShell cmdlets can be used for various tasks such as:

- Export reports
- Deploy new or updated reports
- Deploy reports through the environments (dev/test/prod)
- Modify connection strings and passwords stored for data sources
- Create a folder
- Back up the encryption key

Resources with documentation and sample scripts include:

- [PowerShell Utilities for Managing Report Server on GitHub](#)
- [PowerShell Modules for Power BI Report Server from the PowerShell Gallery](#)

Microsoft 365, Azure Active Directory, and Microsoft Graph REST APIs

In addition to the Power BI REST APIs discussed previously, there are also REST APIs from Microsoft 365, Azure Active Directory and Microsoft Graph which are very helpful for managing user and licensing aspects of Power BI, such as:

- Locating full username and department
- Compiling a list of users assigned to a group
- Assigning Power BI Pro licenses
- Finding users who have and have not signed into Power BI
- Get information for sensitivity labels

Resources with documentation and sample scripts include:

- [Using Power BI Audit Log and PowerShell to Assign Power BI Pro Licenses](#)
- [Microsoft 365 Management APIs](#)
- [Microsoft Graph APIs](#)
- [Azure AD Graph API](#)

Managing Power BI Premium

Management of Power BI Premium is well documented in the [Power BI Premium documentation](#) and the [Power BI Premium Deployment whitepaper](#). This section provides a very brief introductory overview.

Additional discussion of Power BI Premium is also in [Section 3](#) of this whitepaper.

Creating Capacities in the Power BI Service

Before being able to take advantage of Premium resources which have been purchased, one or more capacities need to be assigned in the Power BI admin portal.

The capacities defined in the Power BI service do not have to be allocated exactly 1:1 with the capacity nodes that were purchased. They can be “carved up” differently if desired. This is depicted in the following image, in which 32 purchased v-cores are allocated to 3 capacities in the Power BI admin portal:

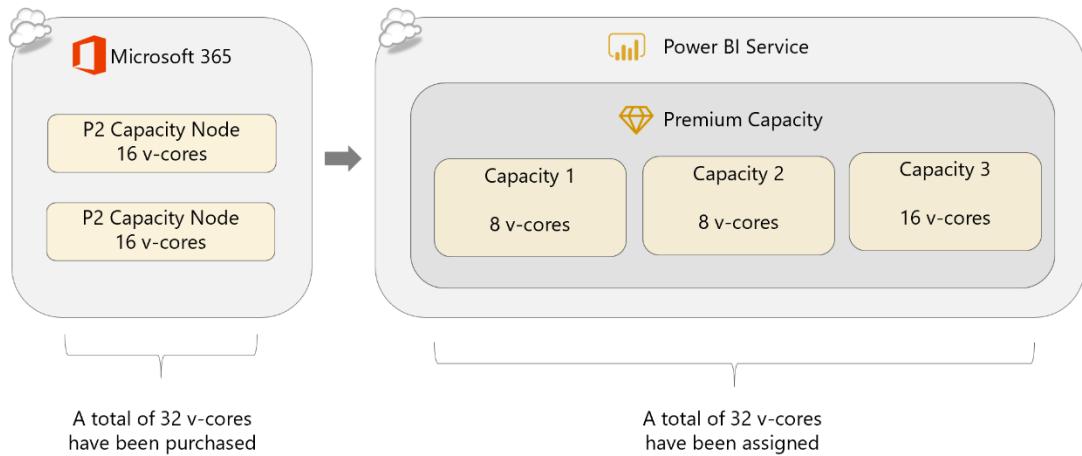


Figure 105 - Depiction of allocating capacities differently in the Power BI service than how they were purchased

Assigning Workspaces to Power BI Premium Capacity

Once a capacity exists in the Power BI admin portal, one or more workspaces must be assigned to it to take advantage of its benefits. This can happen in several ways:

- Assign an individual workspace to a capacity.
- Supply a list of users or security groups and assign all workspaces of which these users are administrators to a capacity.
- Assign all workspaces to a capacity, making it the default storage location for all workspaces.

Workspaces may be moved from one Premium capacity to another, or they may be moved out of Premium capacity back to Power BI shared capacity.

If the entire organization's Power BI content can fit into a Premium capacity, then it may make sense to do that. That situation is very rare. Many companies tend to leave My Workspace for each user in shared capacity, and only consider certain critical workspaces for Premium capacity.

Some organizations use one capacity for hosting 'official' content. This capacity is managed more closely than others, including an approval process to move a workspace in or out.

For the 32 v-cores discussed a few moments ago, workspaces could be assigned to each capacity node to balance out specific workloads:

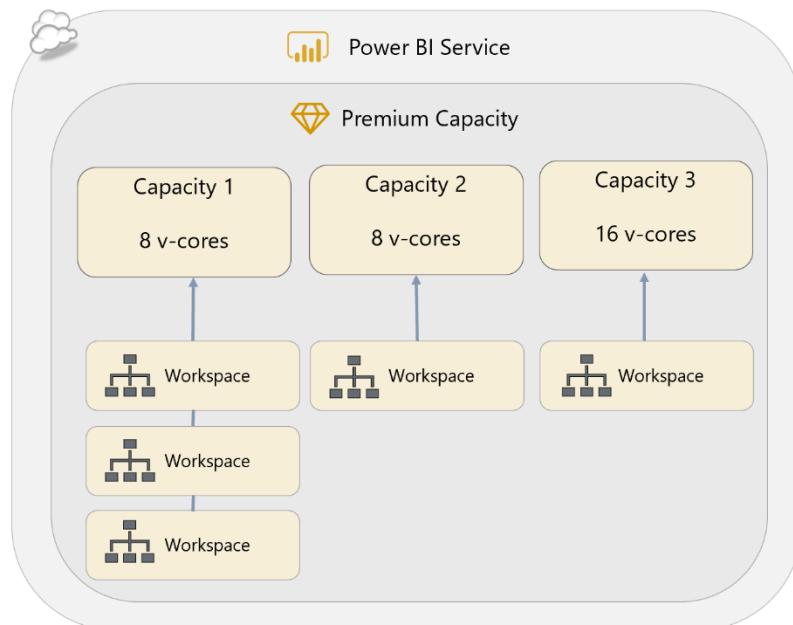


Figure 106 - Depiction of assigning workspaces to separate capacity nodes

For certain workspaces that have extremely unpredictable and variable usage, you may choose to purposely overprovision a capacity node. This technique only works well when there is little chance of significant activity happening concurrently across the workspaces. In this situation, we recommend you closely monitor the capacity health even more closely than usual.

Assigning Capacity Admins and Capacity Assignment Permissions

Purchasing Power BI Premium capacity brings two more sets of permissions that may be granted to a user within a capacity: capacity administrators and assignment permissions.

Capacity Administrator

A Power BI Premium capacity administrator can perform the following tasks within a capacity:

- Assign and remove workspaces to/from a capacity
- Adjust workload settings
- Grant capacity assignment permissions
- Resize and delete capacity nodes (if v-cores are available)
- Monitor capacity usage

A capacity administrator is assigned within the scope of one single capacity within the Power BI admin portal (ex: to Capacity 1). All Microsoft 365 global administrators and Power BI administrators implicitly have capacity administrator rights.

A capacity administrator does not have permissions in Microsoft 365 to change the SKU or billing. Their permissions extend only to managing the capacity settings within the Power BI service.

If workspaces from multiple functional teams or divisions are assigned to a capacity, it may be helpful to have a member from each of the teams as a capacity administrator. Alternatively, you may elect to centralize all capacity administrator responsibilities.

Users with Assignment Permissions

Capacity assignment permissions can be delegated to certain users (who must have a Power BI Pro license). This allows them to assign workspaces to a capacity, provided they are also an administrator of that workspace.

This is helpful functionality in order to delegate responsibility for moving workspaces in and out of capacity. If you prefer instead to implement a workflow, or perhaps require a ticket in a centralized tracking system, then you will not make use of this permission setting.

Configuring Workloads

A Premium capacity can run 4 types of workloads. The datasets workload is always enabled. Additional AI, dataflows, and paginated reports workloads can be enabled.

The Premium capacity administrator has responsibility for adjusting the workload settings to make the best use of the capacity resources.

Workloads

AI - Active

Your workload is ready to use.

On

Max Memory (%)

Allow usage from Power BI Desktop

On

Allow building machine learning models

On

Enable Parallelism for AI Requests

On

DATASETS - Active

Your workload is ready to use.

On

Max Memory (%)

Query Memory Limit (%)

Query Timeout (seconds)

Max Intermediate Row Count

Max Result Row Count

Max Offline Dataset Size (GB)

Automatic page refresh (Preview)

Off

Minimum refresh interval

Minutes

DATAFLOWS - Active

Your workload is ready to use.

On

Max Memory (%)

Enhanced Dataflows Compute Engine (Preview)

Off

Container Size (Mb)

Compute engine memory (%) (Preview)

PAGINATED REPORTS - Active

Your workload is ready to use.

On

Max Memory (%)

Figure 107 - Capacity workload settings in the Power BI admin portal

More information about configuring workloads is available:

- [Configure Workloads in a Premium Capacity](#)

Power BI Premium Capacity Metrics App

There are several things which need to be specifically monitored for the purpose of ensuring capacity health. The Premium Capacity Metrics App is helpful for gaining visibility into the metrics which are important, such as CPU and memory.

Admin portal

- Usage metrics
- Users
- Audit logs
- Tenant settings

Capacity settings

- Embed Codes
- Organizational visuals
- Dataflow settings
- Workspaces
- Custom branding
- Protection metrics (preview)
- Featured content

CHECK OUT THE LATEST VERSION OF THE POWER BI PREMIUM CAPACITY METRICS APP

The app provides up to seven days of comprehensive system, refresh, and query metrics history for your Premium capacities

[Get the app now](#) [Watch a video](#) [Dismiss](#)



Figure 108 - Prompt to get the Power BI Premium Capacity Metrics App

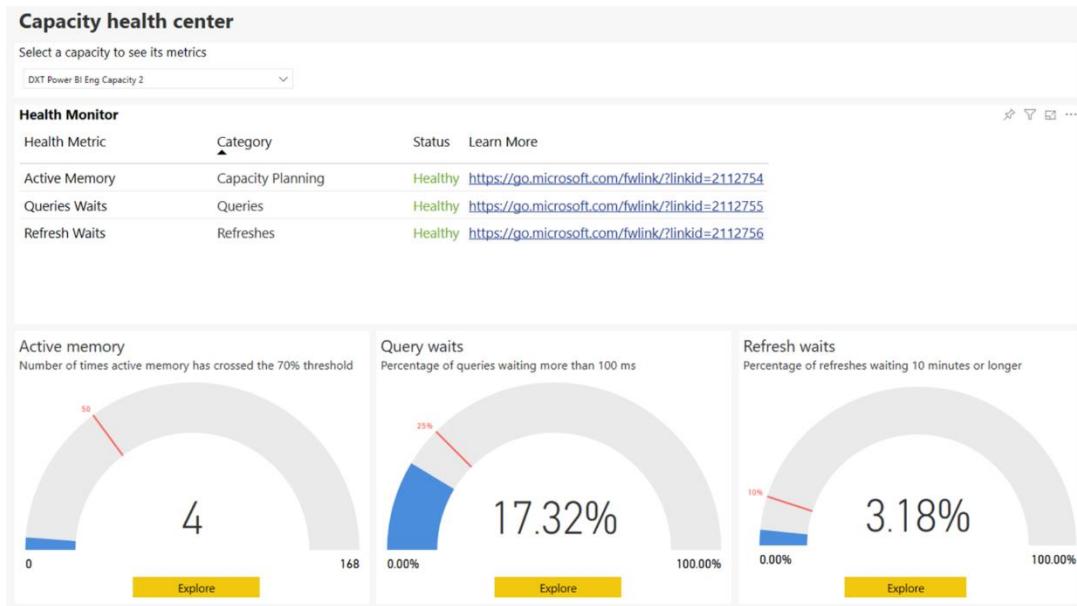


Figure 109 - Example report from the Power BI Premium Capacity Metrics App

Further details on what to monitor, and how to interpret the data presented in the Premium Capacity Metrics App is documented as follows:

- [Power BI Premium Metrics App](#)
- [Monitor Capacities in the Admin Portal](#)
- [Premium Capacity Scenarios](#)

Power BI Premium Capacity Overload Alerts

When the Power BI service detects that the capacity is overloaded, an e-mail can be sent as discussed in [this announcement](#).

Deploying Installers and Applications

There are several downloads available directly within the Power BI service:

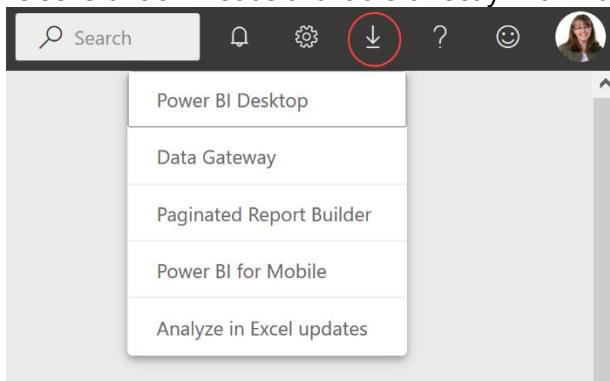


Figure 110 - Download menu in the Power BI service

Power BI Desktop

Nearly all users who develop Power BI content will require Power BI Desktop to be installed on a Windows PC (which is recommended over building reports in the Power BI service using a browser). Power BI Desktop is available in 32-bit and 64-bit versions. The 64-bit version is strongly recommended: Power BI uses an in-memory database engine and the 32-bit version cannot address enough memory to handle larger data volumes.

 The minimum requirements for a PC to run Power BI Desktop can be found [here](#). We recommend providing content creators with well-equipped PCs, particularly RAM for dataset authors.

There is typically a new release of Power BI Desktop every month. We recommend having a process for testing a new release with existing .pbix files, followed by deployment to developers' desktops as soon as possible after each release.

For compatibility reasons, it is important that all report developers use the same release of Power BI Desktop. It may not be possible to open a .pbix file created with a newer version of Power BI Desktop, and there may be issues opening a .pbix file downloaded from the Power BI service from an older version of Power BI Desktop.

There are 3 ways to install Power BI Desktop:

Microsoft Store

Pros:

- ✓ Updates are automatically installed in the background, without requiring the PC owner to act
- ✓ All user PCs will be in sync, removing compatibility issues
- ✓ Administrative privileges are not required
- ✓ Download sizes are smaller and more efficient because only changed components are included
- ✓ Can be integrated with an organization's rollout policies via Microsoft Store for Business

Cons:

- ✗ Requires Windows 10
- ✗ Does not provide for the same degree of control as System Center Configuration Manager
- ✗ Only applies to Power BI Desktop (does not apply to Power BI Desktop Optimized for Report Server, nor Power BI Report Builder)
- ✗ Not all IT departments permit the use of Microsoft Store

Installer EXE

Available from download menu in the Power BI service or <https://aka.ms/pbiSingleInstaller>

Pros:

- ✓ Timing controlled by the owner of the PC
- ✓ Supports numerous versions of Windows

Cons:

- ✗ Requires administrative rights on the machine to install
- ✗ If performed manually by each PC owner, this introduces the risk that all user machines will not be synchronized, leading to compatibility issues
- ✗ Every update requires action by the owner of the PC (monthly releases plus when bug fix releases occur)

Organizational IT deployment tools such as Microsoft System Center Configuration

Manager (SCCM)

Pros:

- ✓ Offers the most control for automation and IT-managed installations for many PCs
- ✓ An efficient option for system administrators responsible for deploying Power BI Desktop in multiple languages to users across the organization
- ✓ Updates are automatically installed in the background, without requiring the PC owner to act
- ✓ All user PCs will be in sync, removing compatibility issues

Cons:

- ✗ More complex, requiring IT oversight
- ✗ If using a tool which requires installation with an .msi file, it will need to be extracted from the .exe

If your organization is using virtual desktop technology such as Citrix or App-V, there are a couple of additional considerations:

- While there are no known issues with running Power BI Desktop in these environments, users working on large amounts of data or doing complex processing can put significant load on CPU or memory on the virtual infrastructure. Ensure that appropriate resources are allocated for each user.
- When Power BI Desktop is set up for users who have already been assigned a Power BI license, you may want to disable the sign-in form that shows to users by default. Setting the following registry key through Group Policy will prevent the form from being displayed:

*Key: HKEY_CURRENT_USER\Software\Policies\Microsoft\Microsoft Power BI Desktop
valueName: ShowLeadGenDialog*

No subscription or license is required to install or use Power BI Desktop. However, users of Power BI Desktop will be routinely prompted to log in, so having a Power BI license (Free or Pro) is the best user experience. Certain features which integrate with the Power BI service (such as publishing a .pbix, get data from the Power BI service, or certain AI features) do require the user to be signed in to work.

Power BI Report Builder

The [Power BI Report Builder](#), also referred to as Paginated Report Builder, is a tool intended for report authors who create paginated reports and deploy them to the Power BI service (to a workspace in Premium capacity).

Power BI Report Builder is available as an MSI file from the download menu in the Power BI service or the [Microsoft Download Center](#). It is not currently available in the Microsoft Store.

Power BI Desktop Optimized for Report Server

[Power BI Desktop Optimized for Report Server](#) is a tool for report authors who will be publishing Power BI reports to Power BI Report Server. This is a separate version of Power BI Desktop which aligns with the release cadence of Power BI Report Server, which is usually 3 times per year. It is important for authors to use this version to avoid compatibility issues with Power BI Report Server.

It is possible for both Power BI Desktop, as well as Power BI Desktop Optimized for Report Server, to be installed on the same PC. There are some implications regarding default behavior for .pbix files, which is documented [here](#).

Power BI Desktop Optimized for Report Server is available as an MSI file, 32-bit or 64-bit, from the advanced download options in Power BI Report Server or the Microsoft Download Center. It is not currently available in the Microsoft Store.

Power BI Mobile Apps

The [Power BI mobile apps](#) are available for iPhone, iPad, Android phone, and Android tablet.

We recommend assessing mobile user needs at the beginning of a Power BI project, because some users will strongly prefer to consume Power BI content through the mobile apps rather than through a browser. This flexibility can make a big difference in adoption rates and user satisfaction.

Power BI App for Windows 10 Devices

For users with a touchscreen, such as Surface Pro, the [Power BI Mobile App for Windows](#) offers a slightly different user experience which some users may prefer. It is also an excellent choice for displaying reports which auto-rotate pages on display monitors (for instance, changing each page every X seconds on a hallway monitor).

See [Section 9](#) for additional discussion on using the Windows 10 app for delivering content.

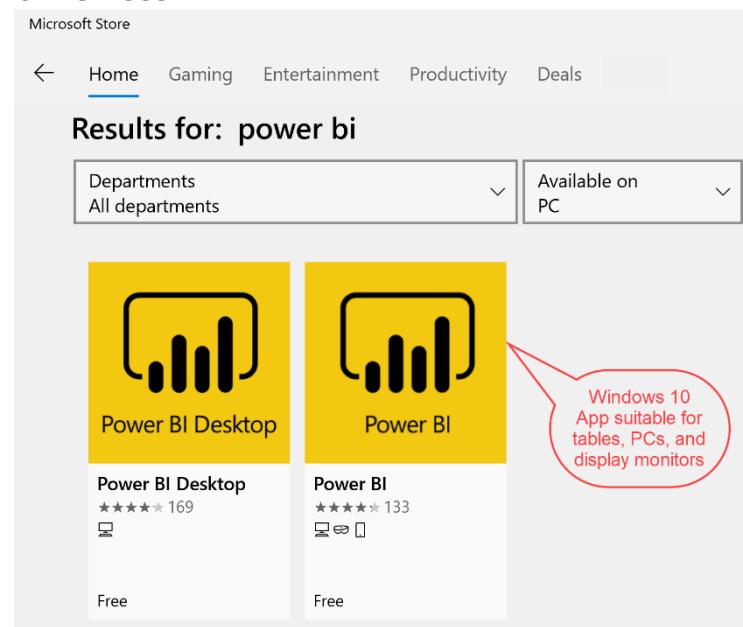


Figure 111 - Apps available in the Microsoft Store

Analyze In Excel Provider

The Excel experts in the organization will appreciate being able to use the [Analyze in Excel](#) capability, discussed earlier in this document:

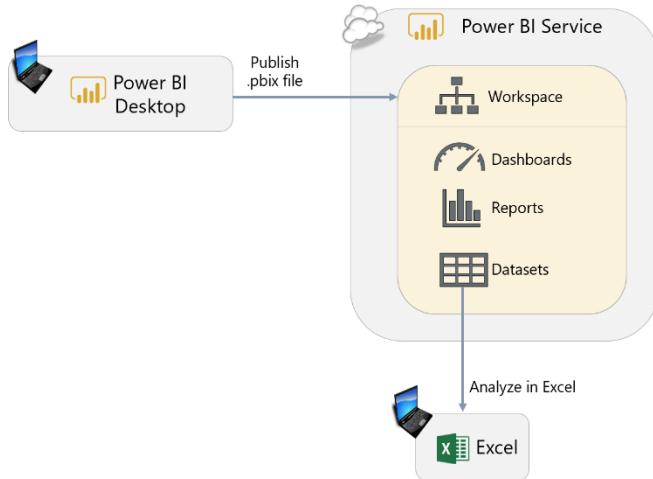


Figure 112 - Diagram of using Analyze in Excel with a dataset in the Power BI service

Use of Analyze in Excel requires installation of the Microsoft Analysis Services OLE DB Provider for Excel. These libraries do need to be updated from time to time, as discussed in [this troubleshooting page](#).

Usage of Analyze in Excel can be prevented via group policy.

Data Gateway

The data gateway appears on the download menu in the Power BI service. In most situations, the On-Premises Data Gateway is installed in standard mode on a server cluster and is managed by IT. In some situations, it may be appropriate to install a data gateway in personal mode for certain Power BI users. A personal gateway is usually installed on an individual user's PC to facilitate data refresh in the Power BI service when a standard mode gateway is unavailable.

The data [gateway installer](#) is an EXE file. Refer to [Section 7](#) for further details and recommendations related to the data gateway.

Other Potential Installers

For all Power BI functionality to work seamlessly, additional items may also need to be installed on user machines (and/or gateway machines), such as:

- Custom drivers associated with certain data sources
- [Microsoft Access Database Engine 2010 Redistributable](#) when using MS Access as a data source

Managing Power BI Mobile Apps with Intune

The Power BI mobile apps can be managed and deployed to users' mobile devices using [Microsoft Intune](#) or third party mobile device management (MDM) solutions (provided that an [app configuration file](#) may be used).

Intune supports app protection policies such as:

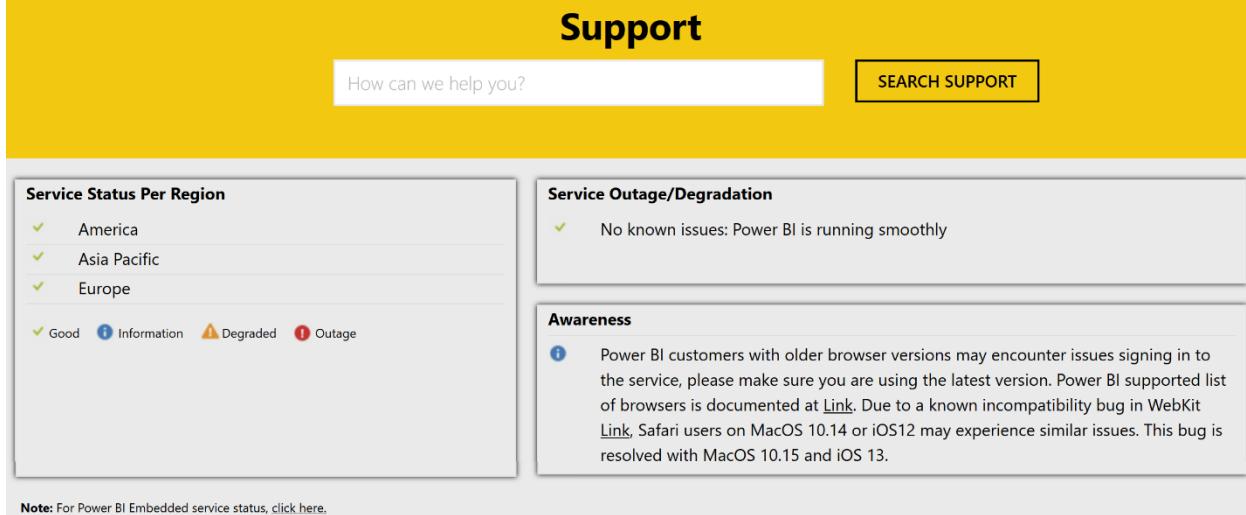
- Requiring an access PIN every time the mobile app is launched
- Specifying how data is handled
- Encrypting application data when the app is not in use
- Selectively wiping data if a device is lost
- Restricting cut, copy, paste between apps
- Printing org data

Some users may want to make [adjustments to interaction settings](#), such as a single tap vs. double tap on report visuals. Organizations may also want to [configure standard settings](#) for interactions.

Monitoring Service Health

Power BI Support Site

Status for the Power BI service is posted on the Power BI Support site located at:
<https://powerbi.microsoft.com/en-us/support/>.



The screenshot shows the Power BI Support page with a yellow header containing the word 'Support'. Below the header is a search bar with the placeholder 'How can we help you?' and a 'SEARCH SUPPORT' button. The main content area is divided into two sections: 'Service Status Per Region' and 'Service Outage/Degradation'. The 'Service Status Per Region' section lists regions with a green checkmark: America, Asia Pacific, and Europe. Below this list are four status indicators: 'Good' (green), 'Information' (blue), 'Degraded' (yellow), and 'Outage' (red). The 'Service Outage/Degradation' section contains a single item with a green checkmark: 'No known issues: Power BI is running smoothly'. At the bottom of the page, there is a note: 'Note: For Power BI Embedded service status, click here.'

Figure 113 - Example of Power BI Support page

Informational messages for broad awareness, such as the one shown above, are also made available on the support site.

Issues relating to the sovereign/national clouds (US Govt, Germany, China) are typically not included in the above status page. Rather, any issues there are communicated in the Microsoft 365 admin portal.

Azure Status

Status for Power BI Embedded comes from the Azure status page located at:
<https://azure.microsoft.com/en-us/status/>.

The screenshot shows the Microsoft Azure Status page. At the top, there is a header with the Microsoft Azure logo, 'My account', and 'Portal'. Below the header, the title 'Azure status' is displayed, with a note 'Last updated 50 seconds ago'. A link 'Go to Azure Service Health' is present. On the right, there is a 'HELPFUL LINKS' section with links to 'Status history & Root Cause Analysis (RCAs)', 'Set up automatic service health alerts', and 'Service Level Agreements (SLAs)'. Below this, there is a 'Refresh every' dropdown set to '2 minutes'. A legend indicates status levels: 'Good' (green checkmark), 'Information' (blue info icon), 'Warning' (orange triangle), and 'Critical' (red X). A table below shows service health across regions for various products, with most entries showing a green checkmark.

Region	Americas	Europe	Asia Pacific	Middle East and Africa	Azure Government	Azure China						
PRODUCTS AND SERVICES	*NON-REGIONAL	EAST US	EAST US 2	CENTRAL US	NORTH CENTRAL US	SOUTH CENTRAL US	WEST CENTRAL US	WEST US	WEST US 2	CANADA EAST	CANADA CENTRAL	BRAZIL SOUTH
Power BI Embedded		✓	✓	✓	✓	✓		✓	✓		✓	✓

Figure 114 - Example of Azure status page

Microsoft 365 Admin Center

Additionally, the Microsoft 365 Admin Center (at <https://admin.microsoft.com>) also displays service health information when the severity of the event is high enough to warrant displaying it in Microsoft 365:

The screenshot shows the Microsoft 365 Admin Center Service Health page. The left sidebar includes links for Home, Users, Groups, Roles, Resources, Billing, Support, Settings, Setup, Reports, and Health, with 'Health' selected and highlighted with a red box. The main content area is titled 'Service health' and shows the status of various services: Exchange Online (1 advisory), SharePoint Online (2 advisories), OneDrive for Business (Healthy), Planner (Healthy), Power Apps in Microsoft 365 (Healthy), and Power BI (Healthy). A message center at the bottom left shows '22 items'.

Figure 115 - Example of Microsoft 365 service health page

The [service health history](#), for up to 30 days, can be found on the History page in the Microsoft 365 service health area as well. The history includes information such as a full description of the event, scope of impact, root cause, and any next steps. If the event is completed, the duration will be available; if the event is active, the estimated time to resolve will be available.

Firewall Settings/Whitelisted URLs

Use of the Power BI service involves internet connectivity. This requires certain endpoints to be reachable by organizational users. The URLs which are required to be whitelisted are documented [here](#).

 For the best user experience, we recommend making sure that organizational users can reach community, documentation, and feedback sites.

There are also firewall considerations for the On-Premises Data Gateway which are documented [here](#).

High Availability and Disaster Recovery

As a fully managed cloud service, Microsoft takes responsibility for these aspects. [This article](#) explains how business continuity is delivered to customers of the Power BI service.

Section 11. Power BI Security and Data Protection

This section covers security and data protection aspects of Power BI. The focus of this section is not on duplicating the technical content which is already available online. Rather, the focus here is a brief overview of key functionality for awareness.

Following are the key pieces of documentation related to security:

- Whitepaper on [Power BI Security](#)
- Documentation on [Power BI Security](#)
- [Data protection in Power BI](#)
- Whitepaper titled "[Supporting Your EU GDPR Compliance Journey With Microsoft Power BI](#)"
- Security documentation available in the [Service Trust Portal](#)
- Whitepaper on [distributing Power BI content to external guest users via Azure Active Directory B2B](#)

The datasets, dataflows, reports, and dashboards in Power BI potentially contains some of the most sensitive information your organization possesses. Managing who has access to this data and how it is shared around your organization is, as a result, one of the most important ongoing tasks in overseeing Power BI.

Integration with Azure Active Directory

Since Power BI uses Azure Active Directory (AAD) for authentication, it can use AAD's full range of functionality.

[AAD Conditional Access policies](#) can be configured to enforce policies (including in the Power BI mobile app) such as:

- Requiring multi-factor authentication for some or all users
- Allowing only devices that comply with organizational policies
- Allowing connectivity from a specific network or IP range(s)
- Blocking connectivity from non-domain-joined machines
- Blocking connectivity for risky sign-ons
- Allowing certain types of devices to connect
- Conditionally allowing or denying access to Power BI for some users

It may be difficult to find a balance between fairly open access and overly restrictive access which can be frustrating for users. The best strategy depends upon the regulatory and compliance requirements facing the organization. You may consider beginning with a cautious access policy, while remaining responsive to the needs of the business users and adjusting restrictions where appropriate for segments of users.

Managing Access to the Power BI Service

Access to the Power BI service requires each user to be assigned a Power BI Free or Power BI Pro license. By default, any user is permitted to sign up for a Power BI Free license or to sign up for a Pro BI Pro trial. Techniques to change, automate, and/or introduce workflow to the process, are discussed in the [Section 4](#) of this whitepaper.

Power BI supports Azure B2B (Business-to-Business). External users, for instance from a customer or partner company, can be invited as guest users for collaboration purposes in Power BI or other Azure and Office 365 services. The [B2B whitepaper](#) is the best resource for getting started with strategies for content collaboration and distribution for external users.

Managing Access to Content in the Power BI Service

Workspace Security

As previously discussed in [Section 9](#), in most cases workspaces should be treated as an area where groups of users can collaborate. The new workspace experience permits assigning permissions to:

- Security groups
- Mail-enabled security groups
- Distribution lists
- Office 365 groups
- Individual users

 We recommend using groups whenever possible for security, rather than individual users. For decentralized organizations, and those with self-service users who manage their own security, this involves user education. Data from the REST APIs (for current state of workspace users) and Power BI activity log (for changes) is invaluable for auditing security assignments.

There are [four roles in the new workspace experience](#):

- Admin
- Member
- Contributor
- Viewer

Capability	Admin	Member	Contributor	Viewer
Update and delete the workspace.	X			
Add/remove people, including other admins.	X			
Add members or others with lower permissions.	X	X		
Publish and update an app.	X	X		
Share an item or share an app. ¹	X	X		
Allow others to reshare items. ¹	X	X		
Feature apps on colleagues' Home	X	X		
Feature dashboards and reports on colleagues' Home	X	X	X	
Create, edit, and delete content in the workspace.	X	X	X	
Publish reports to the workspace, delete content.	X	X	X	
Create a report in another workspace based on a dataset in this workspace. ¹	X	X	X	
Copy a report. ²	X	X	X	
View and interact with an item. ³	X	X	X	X
Read data stored in workspace dataflows	X	X	X	X

1. Contributors and Viewers can share items in a workspace if they have Reshare permissions.
2. To copy a report, and to create a report in another workspace based on a dataset in this workspace, you need to meet additional criteria:

- You need a Power BI Pro license. See the next section, [Licensing](#), for details.
- You need Build permission for the dataset. For datasets in this workspace, the people with the Admin, Member, and Contributor roles have Build permission through their workspace role.

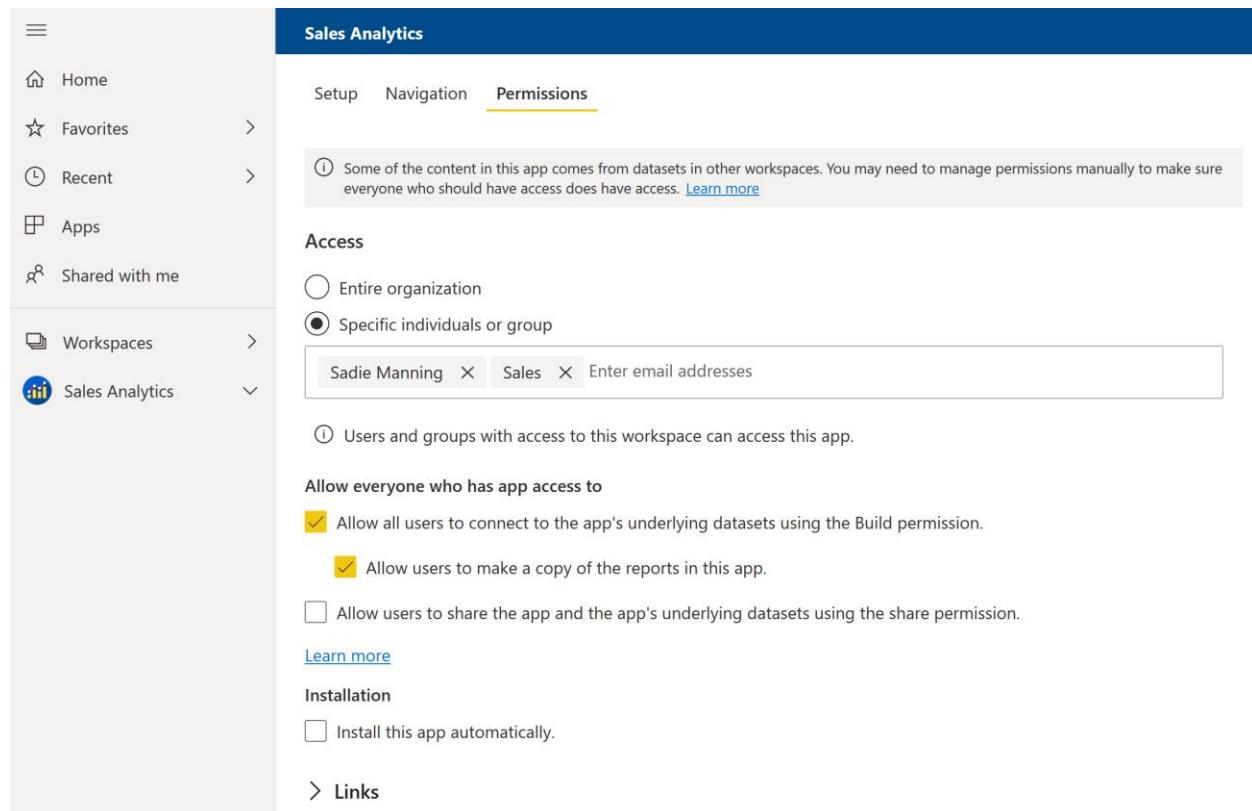
3. Even if you don't have a Power BI Pro license, you can view and interact with items in the Power BI service if the items are in a workspace in a Premium capacity.

Figure 116 - Capabilities for each workspace role

If a user belongs to multiple roles, the permissions are additive.

App Security

During the publishing process for an app, permissions are also assigned. This allows for content to be made available to a wider audience of users. All app permissions are always read-only.



The screenshot shows the 'Sales Analytics' app's permissions page in the Power BI service. The 'Permissions' tab is active. The 'Access' section shows that 'Specific individuals or group' is selected, with 'Sadie Manning' and 'Sales' listed. Under 'Allow everyone who has app access to', two checkboxes are checked: 'Allow all users to connect to the app's underlying datasets using the Build permission' and 'Allow users to make a copy of the reports in this app'. There is also an unchecked checkbox for 'Allow users to share the app and the app's underlying datasets using the share permission'. A 'Learn more' link is available. The 'Installation' section has an unchecked checkbox for 'Install this app automatically'. A 'Links' section is at the bottom.

Figure 117 - Permissions page when publishing an app in the Power BI service

There are two choices for who can see the published app:

1. **Entire organization.** This is only likely to be appropriate in very small organizations, or for information that is widely distributed (for instance, key metrics published throughout the company). There is a tenant setting to allow who is permitted to publish content organization-wide. Usually, this capability is reserved for centralized IT or BI teams, or for certain individuals who are approved to publish for the entire organization. (Note that for apps, the entire organization does not include external users.)
2. **Specific individuals or groups.** Groups are recommended; managing individual users can become tedious and error-prone. The same group or distribution list can be reused to control access to multiple apps, which saves a lot of time setting up access rights for apps and minimizes the chances of error. Using groups also removes the need to republish an app merely for a permissions change.



The separate layer of security available with apps is a great convenience. This allows you to think through the strategy for workspace access separate from read-only app access. Strategies and suggestions are discussed further in [Section 9](#).

In the screenshot above, note there are additional security-related options which are very important:

- **Allow all users to connect to the app's underlying datasets using the Build permission.** The [Build permission](#) allows users to build their own reports and dashboards from the underlying dataset. When publishing an app with this selected, it does adjust the permissions for the underlying dataset (if the dataset is in the same workspace).
- **Allow users to make a copy of the reports in the app.** This setting allows user to essentially use 'save as' functionality to save their own copy of a report which can be customized. For corporate branded reports which are validated, or have a trustworthiness indicator, this option should be used with caution.
- **Allow users to share the app and the app's underlying datasets using the share permission.** This setting controls whether users of an app are able re-share with additional users. This is a [newer setting](#) which offers an additional level of control.

The implications of workspace and app security are also discussed in [Section 9](#), which focuses on content collaboration, sharing, and distribution.

Sharing of Dashboards and Reports

As was discussed in [Section 9](#), sharing operations are distinctly separate from workspace and app security covered above. Since [sharing](#) is handled for individual reports and dashboards, organizational **users should be educated to use sharing as the exception rather than the rule**. App and workspace security should be used whenever possible, rather than sharing. However, since sharing does offer flexibility to set permissions independently for one object, it is a helpful option to have.

In addition to a read permission that a sharing operation grants, it may also grant re-share rights to the recipient of the share.

Dataset Permissions

Each dataset has a set of permissions, including read, re-share, and build.

Read permissions are assigned along with a sharing operation for a dashboard or report, to ensure the underlying data can be read.



The read permission provides a user access to any data in the dataset. If you desire to limit which rows of data a user can see in a dataset, use row-level security (RLS).

The [build permission](#) on a dataset is extremely important in self-service BI implementations. It permits report authors to use the published dataset for the [creation of new reports](#). This is highly encouraged because it allows report authors to reuse an existing dataset without duplicating the data unnecessarily.

Subscription Permissions

Each [subscription](#) has a set of e-mail recipients. The subscription also allows read access to the underlying content. As discussed in [Section 9](#), an image is included in the e-mail so a report subscription should not be created on a report page which could potentially include highly sensitive data.

Location of Original Data Files

The location of original Power BI files, prior to being published, should be considered. Since data is stored and/or displayed within the files, they should be factored into security plans alongside plans for securing the Power BI service. We recommend using OneDrive or SharePoint, with versioning enabled.

Data Protection and Privacy

In addition to Power BI, related data resides many other places as well: Office applications, OneDrive, SharePoint, Teams, Exchange e-mail, among others. For this reason, Power BI integrates with Microsoft Information Protection and Cloud App Security for organizational data protection.

Microsoft Information Protection is not a single product: it is a framework which broadly categorizes numerous products and integrated capabilities.

Unified Sensitivity Labels

Sensitivity labels are a key foundational aspect of Microsoft's approach to data protection.

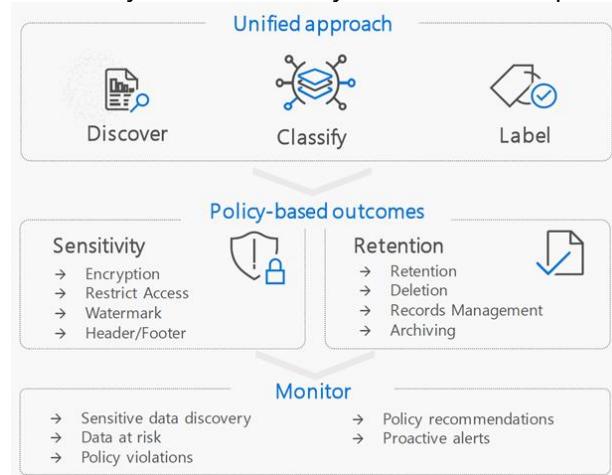


Figure 118 - Microsoft's 3-tier approach to data protection

There is generally a 5-step process for implementing unified labels:

Step 1: Define Labels and Data Protection Policies

The first step is a planning process, usually driven by a data governance committee. This is where sensitivity levels (also called data classifications) are defined, such as:

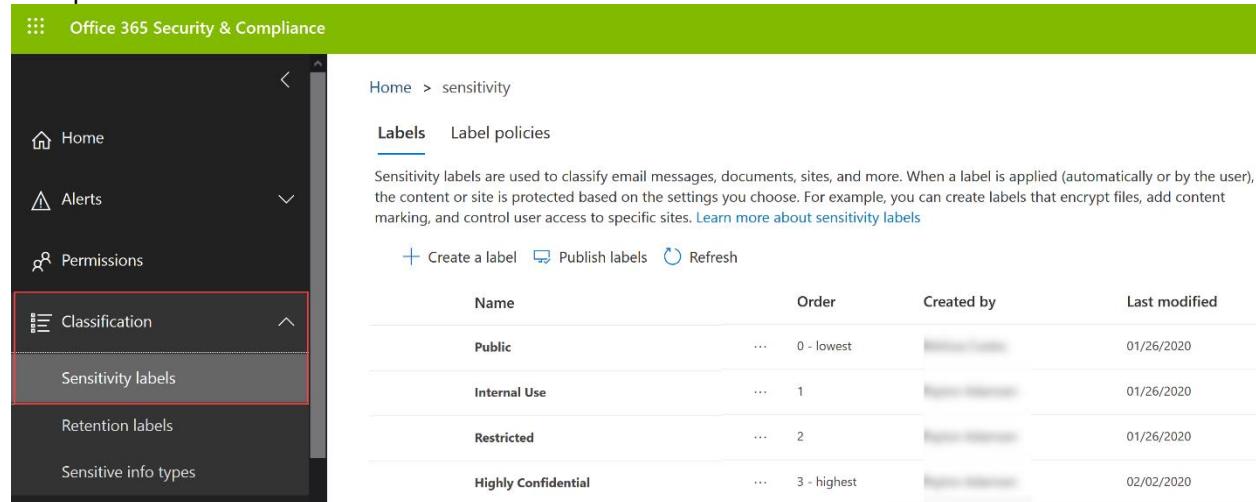
- Public
- Internal Use
- Restricted
- Highly Confidential

The actual sensitivity labels should be defined by your organization, based on needs. In addition to identifying the appropriate levels, data policies should be prepared for each level including what is (and is not) allowed for each label. The data policies should be transparently available to the user community so users understand what can (and cannot) happen with data categorized a certain way.

For reasons like this, some organizations may request users acknowledge a data protection policy before being granted access to Power BI.

Step 2: Create Sensitivity Labels and Data Protection Policies

The unified sensitivity labels are created and managed in the Microsoft 365 Security & Compliance center:



Name	Order	Created by	Last modified
Public	0 - lowest		01/26/2020
Internal Use	1		01/26/2020
Restricted	2		01/26/2020
Highly Confidential	3 - highest		02/02/2020

Figure 119 - Sensitivity labels page in the Microsoft 365 Security and Compliance Center

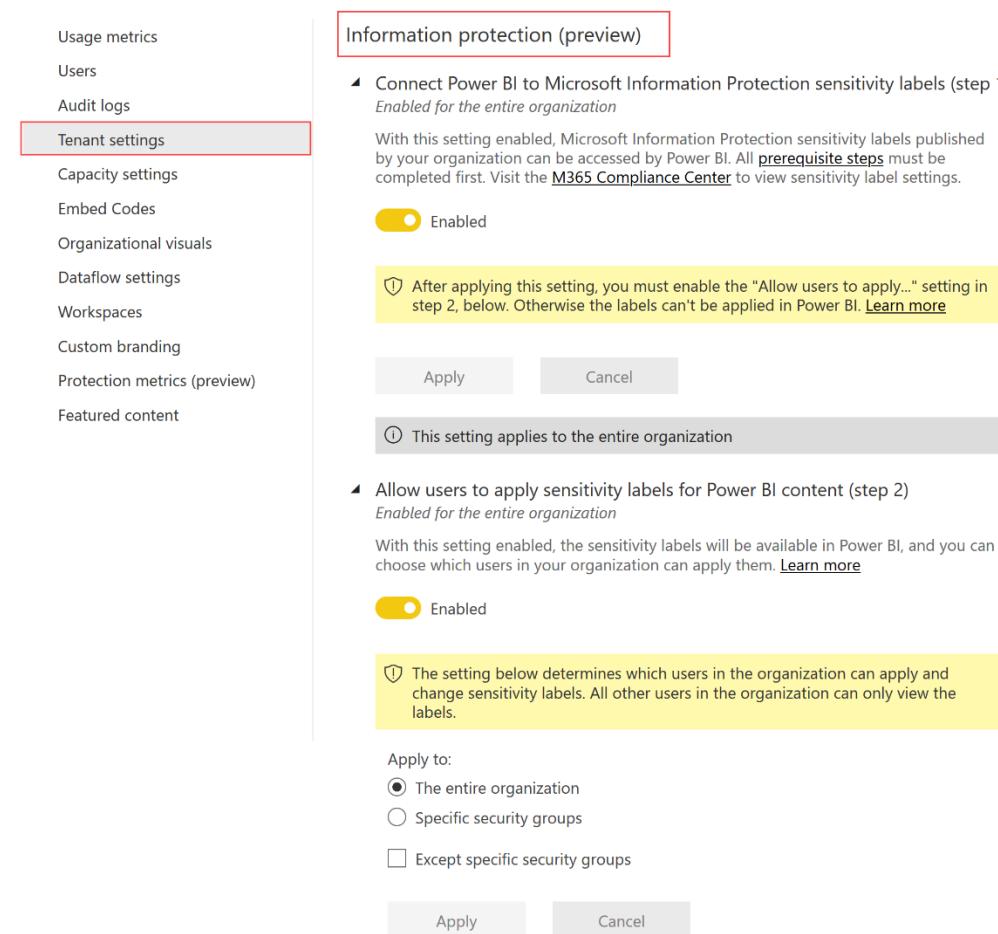
See [this post](#) for more details about the settings available for sensitivity labels. Currently, the use of encryption is not supported with Power BI content.

Once the labels exist, [label policies](#) define aspects such as which users and groups may use the labels.

Step 3: Enable in Power BI Tenant Settings

In the [tenant settings](#), use of the sensitivity labels needs to be enabled, and the users who are permitted to set labels defined.

Admin portal



Information protection (preview)

◀ Connect Power BI to Microsoft Information Protection sensitivity labels (step 1)
Enabled for the entire organization

With this setting enabled, Microsoft Information Protection sensitivity labels published by your organization can be accessed by Power BI. All [prerequisite steps](#) must be completed first. Visit the [M365 Compliance Center](#) to view sensitivity label settings.

Enabled

⚠ After applying this setting, you must enable the "Allow users to apply..." setting in step 2, below. Otherwise the labels can't be applied in Power BI. [Learn more](#)

Apply Cancel

ⓘ This setting applies to the entire organization

◀ Allow users to apply sensitivity labels for Power BI content (step 2)
Enabled for the entire organization

With this setting enabled, the sensitivity labels will be available in Power BI, and you can choose which users in your organization can apply them. [Learn more](#)

Enabled

⚠ The setting below determines which users in the organization can apply and change sensitivity labels. All other users in the organization can only view the labels.

Apply to:

The entire organization

Specific security groups

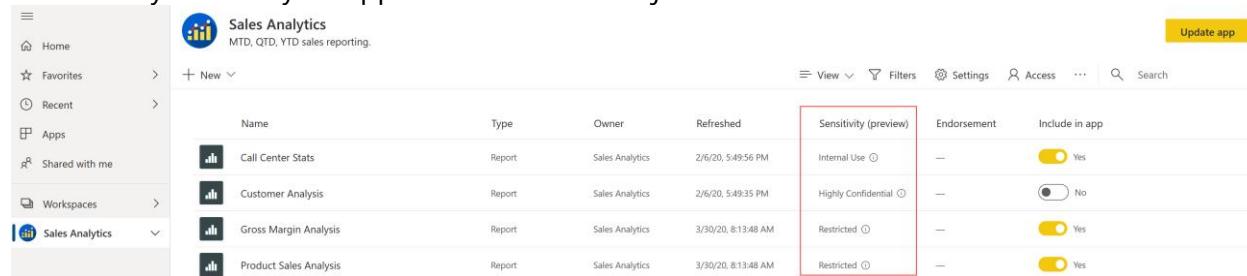
Except specific security groups

Apply Cancel

Figure 120 - Tenant settings in the Power BI service related to information protection

Step 4: Apply Labels to Content

A sensitivity label may be applied to individual objects in the Power BI service:



Home

Sales Analytics MTD, QTD, YTD sales reporting.

Update app

Favorites > + New >

Recent >

Apps

Shared with me

Workspaces >

Sales Analytics

Name Type Owner Refreshed Sensitivity (preview) Endorsement Include in app

Call Center Stats	Report	Sales Analytics	2/6/20, 5:49:56 PM	Internal Use	—	<input checked="" type="button"/> Yes
Customer Analysis	Report	Sales Analytics	2/6/20, 5:49:35 PM	Highly Confidential	—	<input checked="" type="button"/> No
Gross Margin Analysis	Report	Sales Analytics	3/30/20, 8:13:48 AM	Restricted	—	<input checked="" type="button"/> Yes
Product Sales Analysis	Report	Sales Analytics	3/30/20, 8:13:48 AM	Restricted	—	<input checked="" type="button"/> Yes

Figure 121 - Reports in a workspace which have sensitivity labels assigned

When content is exported from the Power BI service, the sensitivity label will be retained with it when the output is Excel, PowerPoint, or PDF. It appears to the user like a tag. Export to any other format does not retain the label which was applied. Note that the ability to export to Excel, PowerPoint or PDF is controlled via a tenant setting in the admin portal:

- ▶ Export to Excel
Enabled for the entire organization
- ▶ Export reports as PowerPoint presentations or PDF documents
Enabled for the entire organization

Figure 122 - Tenant settings in the Power BI admin center for exports

At the time of this writing, usage of the label in Power BI is **informational**. Taking additional actions and enforcement within Power BI is additional functionality we can anticipate over time. This makes the transparent data policies mentioned in step 1 even more important. See [here](#) for additional considerations to be aware of.

Step 5: Monitor, Enforce, and Adjust Data Protection Policies

In the Power BI admin portal, a summary report of content with sensitivity labels is available for Power BI administrators:

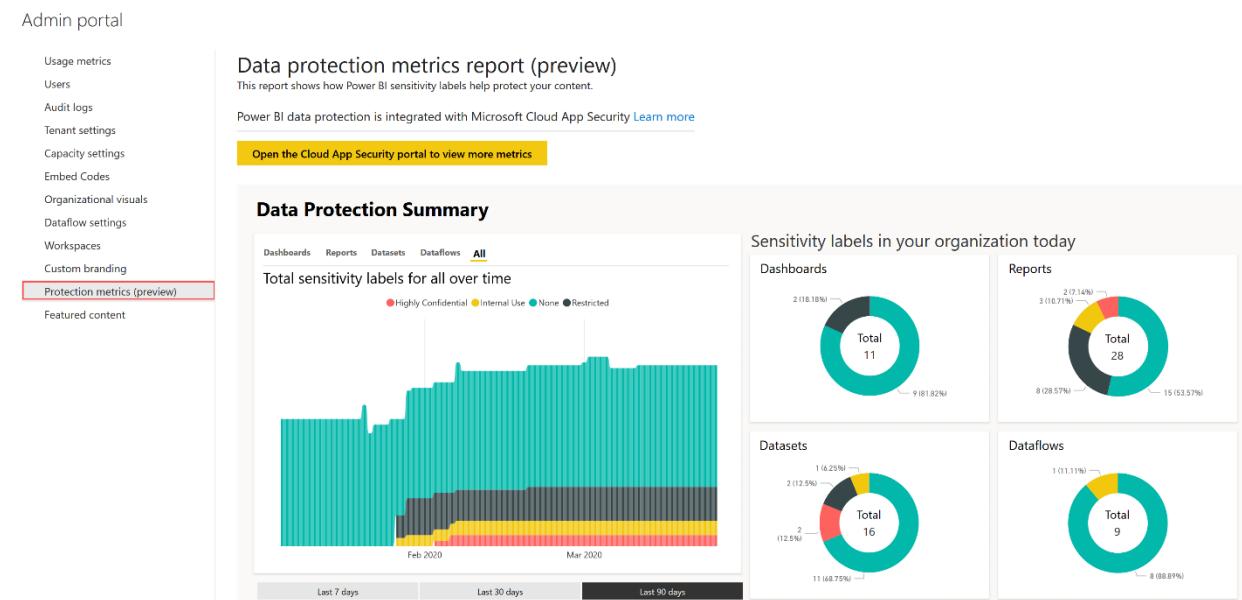


Figure 123 - Data protection metrics report available in the Power BI admin center

Cloud App Security contains additional information for monitoring of Power BI:

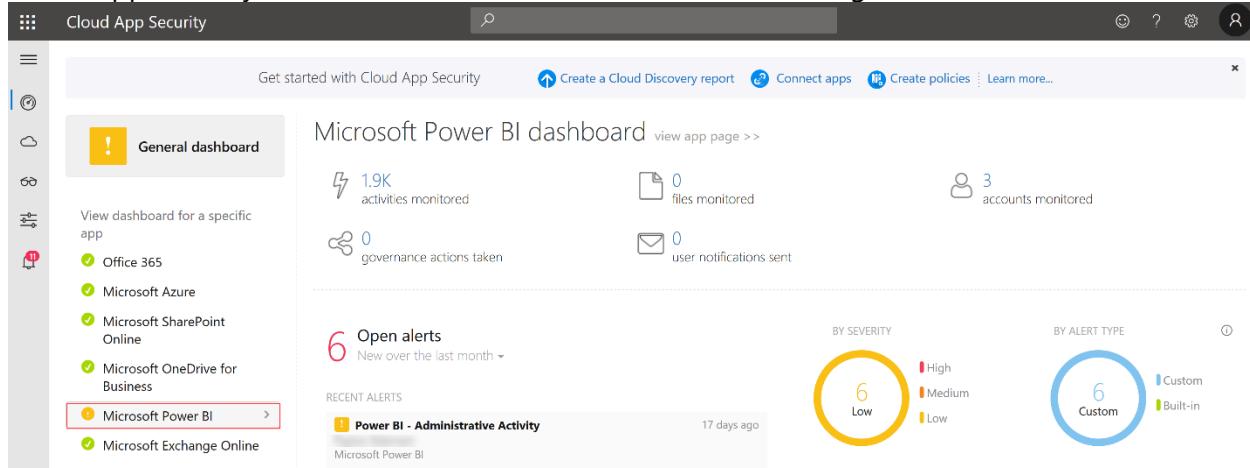


Figure 124 - Cloud App Security monitoring for the Power BI service

You can use Cloud App Security to define custom alerts based on certain types of activities. For instance, you could create an [activity policy](#) which monitors when a Power BI tenant setting (i.e., administrative activity) change occurs and generates [an alert](#). For tenant setting changes, it does not track the before and after value of the setting, nor the group membership change, but it does note a change occurred.

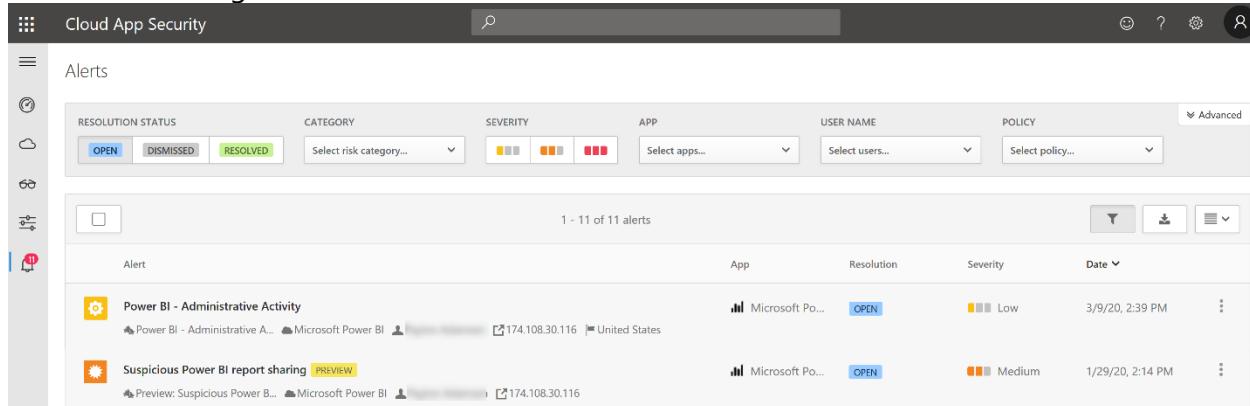
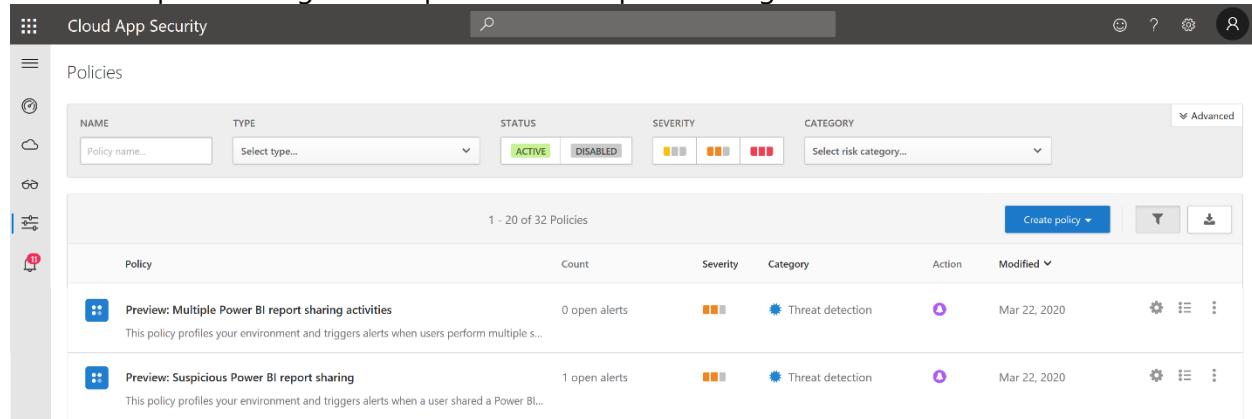


Figure 125 - Activity policies with alerts defined in Cloud App Security

Data Protection Risk Detection

[Cloud App Security](#) can be used to detect anomalous user activity which represents a potential risk.

There are built-in policies to do real-time risk analysis for detecting situations such as suspicious Power BI report sharing or multiple Power BI report sharing activities:



The screenshot shows the 'Policies' section of the Cloud App Security interface. At the top, there are filters for 'NAME', 'TYPE' (set to 'Select type...'), 'STATUS' (set to 'ACTIVE'), 'SEVERITY' (set to 'Low'), and 'CATEGORY' (set to 'Select risk category...'). A search bar and a 'Create policy' button are also present. The main table lists two policies:

Policy	Count	Severity	Category	Action	Modified
Preview: Multiple Power BI report sharing activities This policy profiles your environment and triggers alerts when users perform multiple s...	0 open alerts	Low	Threat detection	Mar 22, 2020	More options
Preview: Suspicious Power BI report sharing This policy profiles your environment and triggers alerts when a user shared a Power BI...	1 open alerts	Low	Threat detection	Mar 22, 2020	More options

Figure 126 - Built-in policies in Cloud App Security for real-time risk analysis

Data Privacy Levels in M Queries

One easy-to-miss feature around data security is the concept of data privacy levels for loading data. In certain scenarios, the M queries that Power BI uses to load data into datasets may be able to gain a significant performance boost by sending data from one data source to another.

For example, a user may have two data sources: an Excel workbook and a table in a SQL Server database, and want to be able to filter the data in the SQL Server table using a value found in the Excel workbook. The most efficient way to do this is for Power BI to generate a SQL query with a WHERE clause to perform the necessary filtering on the server (known as "query folding"). However, to do this, the WHERE clause of this query will have to contain a value taken from the Excel workbook, and if this is the case a DBA monitoring the queries would be able to see this value. If the Excel workbook contains sensitive data, this may represent a breach of security: the DBA is able to see values that he or she may not have permission to see. The alternative is for Power BI to download the entire contents of the SQL Server table and perform the filtering itself, something that is likely to be more inefficient and slower, but at the same time is a better security practice.

Power BI dataset authors can set data privacy levels for each of the data sources they use in the Query Editor of Power BI Desktop, and this controls whether data from one data source is sent to another. All report creators should be made aware of this feature and its implications if they are working with sensitive data. However, the privacy level set in Power BI Desktop does not carry over to the Power BI service. Once a dataset has been published to the Power BI service, its

privacy level is defined within the data gateway (for on-premises sources, or sources within a virtual network).

Dashboard Data Classification

The Power BI service has the ability to assign [data classification tags](#) to dashboards. This could include data classifications such as Confidential or Low Business Impact. It's also possible to use the classifications for subject matter areas.

 We recommend using the unified sensitivity labels associated with Microsoft Information Protection, rather than the dashboard classifications which are restricted to dashboards only and have limited functionality.

Compliance Certifications

A great number of major industry certifications are available. The current reference for compliance certifications can be found in the [Microsoft Trust Center](#).

Securing Data at Rest

There are several types of data at rest including, but not limited to:

- The data stored in the dataset
- Metadata for reports and dashboards
- Visual caching of dashboard tiles
- Credentials to connect to source data
- Usage tracking data

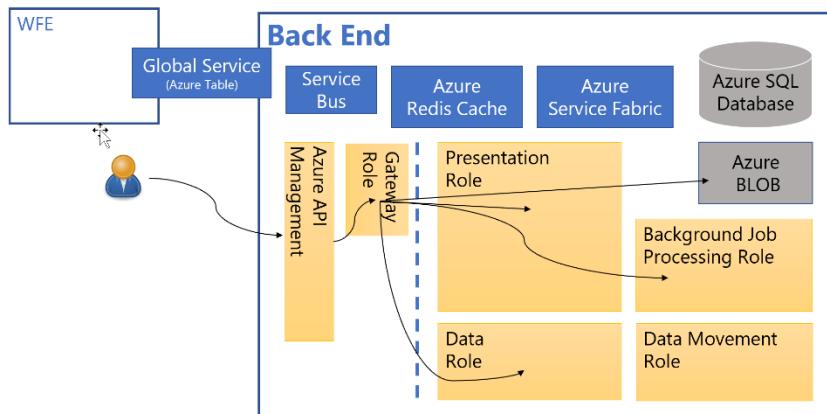


Figure 127 - High-level overview of the web front end and back end for supporting Power BI

Power BI uses two repositories for storing and managing data and metadata: Azure Blob Storage and Azure SQL Database. The exception to this is large datasets, which are instead stored in Azure Premium Files.

Encryption keys for Azure Blob Storage are stored and encrypted in a separate location in the Power BI service. The process is fully managed by an internal Microsoft service.

The encryption keys for Azure SQL Database are managed by Transparent Data Encryption (TDE).

Dataset Row-Level Security

For situations where different users should only see some of the data in a dataset, Power BI [row-level security](#) allows filters to be applied on the rows in any table in that dataset. For example, if a table in a dataset contains sales data for different countries, it may be a requirement that users can only see the sales information for their country in any reports they view. Row-level security allows a report developer to create a single report that, when viewed by users in the USA, only shows sales information for the USA; when viewed by users in Germany only shows sales in Germany, and so on.

Row-level security can work in one of two ways:

1. In the most common scenario, a report developer creates multiple security roles, gives each role a filter to view a specific slice of data, and then assigns users or security groups to those roles.
2. It is also possible to create a dynamic security role, where a single role can display different slices of data to different users using DAX expressions that can determine who is trying to connect to the report and filter the data appropriately. Dynamic security is more complex to set up but can avoid the need to create and maintain large numbers of roles with different permissions. When using dynamic security, users and the data they map to are part of the dataset.

If row-level security is applied in a dataset, it is crucial that it is properly tested. This is especially true when complex DAX expressions are used in the role definitions. As a best practice, we recommend using the DAX UserPrincipalName() function in row-level security expressions as it will yield the same results when testing results in Power BI Desktop and the Power BI service.



Shared datasets become even more important when row-level security is involved

If the same data is used in multiple datasets, row-level security must be applied consistently across every dataset. The use of a single (or fewer) shared datasets can significantly reduce the overhead of managing row-level security, as well as reduce the chance for errors and inconsistencies.

Securing Data at the Data Source Level with SSO

Since creating, managing and maintaining row-level security can be difficult across multiple datasets, you may consider applying security at the data source level in certain circumstances.

If Power BI uses a data source in DirectQuery mode, in conjunction with [single sign-on](#), it becomes possible to use the built-in security functionality from the database platform. Using single sign-on with DirectQuery requires configuration of the data gateway in standard mode.



When connecting to a PaaS service such as Azure SQL Database or Azure SQL Data Warehouse, you will also need to permit connectivity to the database for users who connect from Power BI Desktop. This is handled by either opening firewall ports for the relevant IP address range for an office, or by including the Azure SQL Database in the corporate virtual network (i.e., Azure vNet).

Data Storage for Imported Datasets (<10 GB)

For imported datasets, the data is encrypted in Azure Blob Storage. For push datasets, the data is encrypted in Azure SQL Database.

Data Storage for Large Imported Datasets (>10 GB)

When [large datasets are enabled for Power BI Premium](#), these datasets are allowed to grow (via incremental refresh) beyond the initial 10 GB upload limit. To facilitate this, data is stored on Azure Premium file storage. Premium file shares provide consistent high performance and low latency. Azure storage encryption is enabled for all accounts.

Storage for Metadata and Visual Cache

The metadata related to a Power BI subscription (such as report/dashboard layouts, recent data sources, workspaces, tenant & organizational information) as well as visual caching of tiles and visuals are encrypted in Azure Blob Storage and Azure SQL Database. For Analysis Services source data, there is also a metadata reference to Azure SQL Database.

Storage of Credentials to Connect to Source Data

Data source credentials used by the Power BI service are encrypted in Azure SQL Database.

Storing Data in a Specific Geographic Region

The [multi-geo feature of Power BI Premium](#) or [Power BI Embedded](#) helps multinational customers address regional, industry-specific, or organizational data residency requirements. This permits customers to deploy content to datacenters in regions other than the home region of the Power BI tenant.

Securing Data in Transit

When data is actively being used or accessed, it is considered in transit. This includes activities such as interacting with a report or refreshing a dataset. Power BI uses HTTPS, TCP/IP and TLS to ensure data is encrypted and maintains integrity during the transport.

When the data is held in-memory within the Power BI service, it is unencrypted to allow for report display to occur. If data becomes cached, it is encrypted and stored in Azure SQL Database (some artifacts can also be stored in Azure Redis Cache unencrypted for a limited period).

Within a client's web browser, data is cached only for the duration of the session. The cache is destroyed when the user navigates the browser tab out of the Power BI service, or closes the browser window.

More details are covered in the [Power BI Security Whitepaper](#).

Securing the Connection Between Data in Azure and Power BI

As discussed in [this post](#) (which mentions Azure SQL Database, but is applicable to other data services as well), the easiest way to allow connectivity is the "Allow access to Azure services" setting. However, doing this may open connectivity more broadly than the corporate security policy permits.

The use of virtual networks within Azure is generally supported for use with Power BI. When a cloud service has been added to a virtual network, then a data gateway will be required for connectivity from the Power BI service.

Currently, use of Power BI dataflows with an Azure Data Lake Storage Gen 2 account within a virtual network is not supported.

Securing the On-Premises Data Gateway

Data source credentials are encrypted using asymmetric encryption so that they cannot be decrypted in the cloud. The credentials are sent to the gateway machine, where they are decrypted when the data sources are accessed.

The data gateway uses data compression, as well as transport encryption.

Data Transmitted Outside of the Power BI Service

If map visuals are displayed on reports which use [Bing Maps](#) or [Esri](#), those services do transmit data out of the Power BI service.

Additionally, as discussed in [Section 10](#), some custom visuals also may transmit data to an external resource.

ExpressRoute

Power BI supports the use of ExpressRoute, which allows you to create a private network connection from your organization to Power BI (or using an internet service provider's colocation facility). ExpressRoute bypasses the public internet, providing for additional security and connectivity performance.

Protecting Data on Mobile Apps

The Power BI mobile app supports the following configuration scenarios:

- Data protection settings
- Interaction settings for [users](#) and [organizational standards](#)
- Report Server configuration

Azure Active Directory

Azure AD conditional access rules, discussed earlier in this section, are also applicable to the mobile apps.

Intune

With [Intune](#), you can set up [app protection policies](#) to configure protection for access and data. Certain behaviors inside of the application may be blocked or restricted, such as:

- Force a pin code
- Block jailbroken devices
- Restrict cut/copy/paste actions
- Require data encryption

Custom Mobile Device Management (MDM)

MDM configuration is available for solutions which support using an [app configuration file](#).

See [this post](#) for additional security information related to Power BI mobile.

Securing Power BI Report Server

Access to content in Power BI Report Server is based on its independent role-based security model. Permissions can be applied at three levels:

- **Portal access.** This controls which users have access to Power BI Report Server.
- **Folder level.** This is the ideal level to control access because it provides control over all the content in the folder (including any subfolders) with a single setting.
- **File level.** At the level of individual pieces of content such as a single Power BI report. In most cases, this level of granularity should be used sparingly. It can make managing permissions tedious and error-prone.

A full description of all of the built-in roles available in Power BI Report Server is available [here](#). In summary:

- The **Content Manager** role provides full permissions to publish and delete content and manage other users in Power BI Report Server. This is conceptually like the administrator of a workspace in the Power BI service.
- The **Publisher** role allows users to publish content to Power BI Report Server. This is similar to the contributor of a workspace in the Power BI service.
- The **Browser** role only allows users to view content in Power BI Report Server. This is similar to these types of capabilities from the Power BI service:
 - Being a viewer of a workspace, or
 - Having a report or dashboard shared with you, or
 - Being granted access to an app

Managing Encryption Keys

Azure Premium Encryption Key

The use of customer-managed encryption keys, along with specifying a specific geographic region, allows customers an additional level of control for data which resides in Premium capacity.

When the customer chooses to manage the encryption key, the RSA key is integrated with [Azure Key Vault](#). This feature is known as BYOK: Bring Your Own Key.

Additional information:

- [Announcing General Availability of Bring Your Own Key \(BYOK\) for Power BI Premium](#)
- [Bring Your Own Encryption Keys for Power BI](#)

On-Premises Data Gateway Recovery Key

As discussed in [Section 7](#), a recovery key is assigned at the time the data gateway is installed. The recovery key is required if the gateway is to be relocated to another machine, or if the gateway is to be restored. This key cannot be changed or reset, so it is important to store it in a safe location where the Power BI gateway administrators can access it. A corporate password vault is an ideal choice for this.

Encryption Keys for Gateway Credentials

For on-premises data sources, the encryption key does not leave the on-premises gateway infrastructure. The encryption key for the Data Movement Service remains stored in the customer's data gateway. Power BI cannot access or intercept the key value.

For cloud data sources, the encryption key is stored in Azure SQL Database and encrypted with Transparent Data Encryption (TDE).

Encryption Key for Power BI Report Server

The Report Server uses encryption keys to for securing credentials and connection information. Encryption is supported through a combination of public, private, and symmetric keys which are used to protect sensitive data.

Power BI Report Server Product Key

As discussed earlier in this whitepaper, one feature of Power BI Premium is access to (optionally) deploy to Power BI Report Server. The product key can be located in the Power BI admin portal: Admin portal

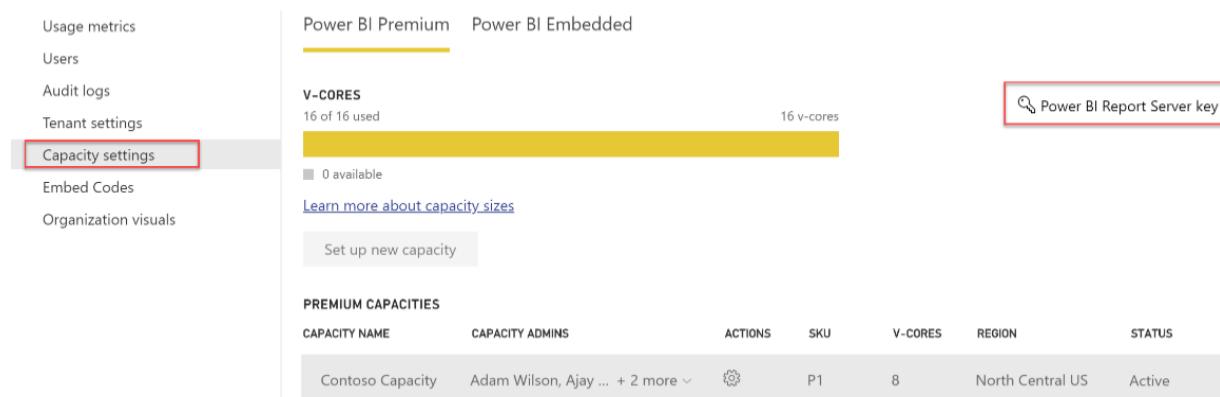


Figure 128 - Capacity settings in the Power BI admin portal

Power BI in the National Clouds

Currently, Power BI is available in 3 national clouds: [US Government, China, and Germany](#). The data location, oversight, and management of those data centers is handled in compliance with local/governmental regulations.

There are some feature differences between the commercial cloud and each national cloud. The US Government cloud is [discussed in this post](#).

Section 12. Power BI Deprecated Items

Following are the features and functionality which are deprecated, or on a deprecation path, at the time of this writing.

Deprecated feature:	A feature which has been announced as discontinued and will be fully removed from the product in the future. Customers should be actively planning a migration and discontinuing use of any deprecated features.
Feature on a deprecation path:	A feature which will be formally deprecated at a future date. New development is strongly discouraged.

Power BI Publisher for Excel

The Power BI Publisher for Excel add-in is [not supported](#) in the new workspace experience. We recommend all customers discontinue use of this add-in as part of the migration from classic workspaces.

Classic Workspaces in the Power BI Service

With the introduction of the new workspace experience, discussed earlier in this whitepaper, classic workspaces (i.e., Office 365 group-backed workspaces) are [now on a deprecation path](#). We encourage all customers to make a plan for [upgrading all existing workspaces](#) to the new workspace experience as soon as it is practical.

Old Filter Experience

The new filter experience was introduced via an opt-in experience: through a report setting in Power BI Desktop, and the 'new look' in the Power BI service. When the new filter experience is generally available, the old experience will no longer be an option.

Windows 7 and .NET 4.5 Support for Power BI Desktop

Support for installing Power BI Desktop on Windows 7 with .NET 4.5 will become deprecated in the future. Minimum requirements are documented [here](#).

Ribbon in Power BI Desktop

The new ribbon experience in Power BI Desktop has become the default. In the future, the option to go back to the old menu will be removed. The quick access toolbar will not be available in the new ribbon.

Windows 10 Mobile

The Power BI mobile app for phones using Windows 10 Mobile is expected to reach [end of support](#) March 2021. At that time, app users will no longer receive security updates, non-security hotfixes, free assisted support options, or online technical content updates from Microsoft.

TLS 1.0 and 1.1 Support

Support for [Transport Layer Security \(TLS\) versions 1.0 and 1.1](#) will be discontinued for all Microsoft online services, including the Power BI service. To prepare for this change, make sure all users who rely on Power BI are using supported operating systems and browser versions.

Power BI APIs .NET SDK v2

With the release of [Power BI APIs .NET SDK v3](#), the previous version will be retired. The Power BI APIs .NET SDK v2 is now deprecated and is expected to be retired in February 2022. All customers are advised to migrate to the latest version as soon as possible. New APIs will only be added to the Power BI APIs .NET SDK v3.

Get Data Experience in the Power BI Service

Connecting to databases or systems, via Get Data, in the Power BI service is on a deprecation path. The capabilities to connect to data, transform and clean data, as well as creation of a dataset, relationships, and calculations are all fully supported in Power BI Desktop. All customers are advised to use Power BI Desktop for this capability.

Specifically, with regard to connecting to Analysis Services models: now that the unified dataset discovery experience for Power BI is done through [shared datasets](#), the ability to connect to Analysis Services models registered via the On-Premises Data Gateway (accessible via Get Data>Databases in the Power BI service) is no longer available. Instead, an Analysis Services model should be published directly to the Power BI service, and used as shared dataset.

Organizational Content Packs

Distributing content to other users via organizational content packs is on a deprecation path. [Power BI Apps](#) are the replacement functionality and should be used going forward. Third party apps (formerly referred to as third party content packs) from AppSource are still fully supported.

MSI Installer for Power BI Desktop

Power BI Desktop now ships as a [single .exe installation package](#) which contains all supported languages. For administrators who manage Power BI Desktop installations for multiple languages, this simplifies the deployment process. The former language-specific .msi packages are discontinued as of September 2019.

Custom Visuals API Version 1.1 and Older

The custom visuals API is currently on [version 2.6](#). Version 1.2 and above is [supported](#).

Certain Behaviors of Data Analysis eXpressions (DAX)

As the in-memory VertiPaq storage engine evolves over time, sometimes changes are introduced which impact the behavior of DAX. One such example is the use of SUMMARIZE for aggregations: [SUMMARIZECOLUMNS](#) should be used instead.

Certain Color Classes within Report Theme JSON Files

There are several color classes within a report theme which are deprecated, as documented [here](#).

Azure Workspace Collections

Azure Workspace Collections are deprecated, and should be [migrated](#) to the replacement service, [Power BI Embedded](#). Although it is still listed in Azure as a service, customers are now actively blocked from using Power BI Workspace Collections with one exception: customers with an Enterprise Agreement signed before June 2017 have until the end of that agreement to migrate to Power BI Embedded.

Power BI CLI

The Power BI command line interface (CLI), for managing workspace collections, is deprecated and no longer supported. Automation capabilities for Power BI can be invoked via [PowerShell cmdlets](#) and/or the [REST APIs](#) for Power BI Embedded.

Section 13. Support, Learning, and Third-Party Tools

It is inevitable that, at some point, you will encounter an issue that you will need assistance with resolving. The Microsoft community has an abundance of resources available which are discussed in this section.

Due to the pace of change, be wary of community content (such as blogs and videos) which are older than a few months.

Power BI Release Plan (Roadmap)

The Power Platform release plan describes new features for Power BI, Power Apps, Power Automate, AI Builder, Power Virtual Agents, as well as the Common Data Model and Data Integration. The release plan, with tentative release dates, may be found online and may also be downloaded as a PDF file. Release plans align with the biannual planning cycle for the Power BI product team.

-  Release plan for Dynamics 365 and Power Platform: <https://docs.microsoft.com/en-us/dynamics365/release-plans/>

Support and Help

Power BI Documentation

Hub for information and documentation about using Power BI.

-  10 tips for getting help: <https://docs.microsoft.com/en-us/power-bi/service-tips-for-finding-help>
-  Power BI documentation: <https://aka.ms/powerbi-docs>

Power BI Guidance and Best Practices

Recommended practices for Power BI from the Customer Advisory Team (CAT). CAT works with enterprise customers daily and are an exceptional source of information.

-  Power BI guidance documentation: <https://docs.microsoft.com/en-us/power-bi/guidance/>

Power BI Support, Service Outages, and Degradations

Although the Power BI service is extremely reliable, there may be times when the service is unavailable or certain features do not work as expected. The support page will tell you about any outages or degradations that Microsoft is currently aware of and the timelines for getting them resolved. You can also open a support ticket from this page, which is available to users with paid Power BI Pro licenses.

-  Power BI support: <https://support.powerbi.com>
-  Power BI service: <https://powerbi.microsoft.com/en-us/support/>
-  Power BI Embedded: <https://azure.microsoft.com/en-us/status/>

Reporting Issues

If you suspect a bug, the best place to report the problem is the Power BI issues forum. Before you post, please search the forum to check whether a similar issue has been posted. If you have a new issue, make sure to include full details of the steps needed to reproduce the problem and screenshots of any error messages that you receive.



Issues: https://community.powerbi.com/t5/Issues/ct-p/PBI_Comm_Issues

Getting Help with Power BI Development

If you are having trouble with any aspect of Power BI development or configuration, then the Power BI forums are a great place to get advice, usually from members of the Power BI community. Please search the forums to see whether anyone else has asked the same question before you post. You will increase the chances of getting your problem solved if you post in the right forum for your issue, you are polite and courteous, and you supply all the necessary technical details and examples in a succinct manner.



Discussion forum: <https://community.powerbi.com/t5/Forums/ct-p/Forums>

How to get your question answered quickly:



<https://community.powerbi.com/t5/Community-Blog/How-to-Get-Your-Question-Answered-Quickly/ba-p/38490>

Ideas for Improving Power BI

On the Power BI Ideas site (also known as User Voice) you can suggest improvements, add new ideas, vote for and/or comment on existing ideas, and view status of ideas (such as started, or under review). As always, please check to see whether a similar idea has already been suggested before you post. The Power BI product team does factor in the most requested items into their bi-annual planning cycle. The more votes an issue receives the more likely that it will be implemented.



Ideas: <https://ideas.powerbi.com/forums/265200-power-bi-ideas/filters/top>

Finding a Consultant or Partner to Accelerate Your Power BI Implementation

Sometimes, building or extending an analytics solution requires bringing onboard a professional with deep knowledge in Power BI to architect the first set of content and do a knowledge transfer to your team. It may also be helpful to involve a partner when the underlying data infrastructure needs enhancements to be able to take full advantage of Power BI. Microsoft has a rich ecosystem of [partners](#), which can result in reducing the time and cost required to build an analytics solution. High quality [consulting services offerings](#) that have clear deliverables and predictable prices make it easier to start quickly.



Power BI partners: <https://powerbi.microsoft.com/en-us/partners/>

Learning More About Power BI

Power BI Team Blog and Videos

Power BI is a very flexible and effective set of products and services that continually evolves and improves. There is always more to learn about. The Power BI blog has regular roundups of new updates and features, technical tips, and announcements. To keep your knowledge up-to-date, you should closely follow all posts on the official Power BI blog. Some of the .pbix files used in the monthly release videos are also posted on GitHub.

 Power BI official team blog: <https://powerbi.microsoft.com/en-us/blog/>

 GitHub: <https://github.com/Microsoft/powerbi-desktop-samples>

Power BI Adoption Framework

The Power BI adoption framework provides a set of guidance, practices, and resources to help organizations build a data culture, establish a Power BI center of excellence, and manage Power BI effectively.

 Documentation: <https://github.com/pbaf/powerbiadoption>

 Video series:
<https://www.youtube.com/playlist?list=PL1N57mwBHTN0UZbEgLHTA1yxqPlae3B90>

Guy In a Cube YouTube Channel

The Guy In a Cube videos are an excellent resource available for you to learn about Power BI. Guy in a Cube are members of the Customer Advisory Team (CAT) at Microsoft who post multiple times per week with roundups of new features and new community content, as well as technical tips and suggestions.

 Guy In a Cube: <https://www.youtube.com/guyinacube>

Power BI Documentation

The official Power BI documentation site is very detailed and updated on a regular basis. It should be your primary reference when you need to learn about a feature of Power BI.

 Power BI documentation home: <https://powerbi.microsoft.com/en-us/documentation/>

Power BI Community Site

The Power BI community site includes a community knowledgebase, forums for Q&A, announcements, and several galleries such as the Themes Gallery, Data Stories Gallery, R Scripts Showcase, Quick Measures Gallery, among others.

 Power BI community: <http://community.powerbi.com/>

Power BI User Groups

Power BI user groups have been established in many places around the world. User groups are a great place to hear presentations by Power BI experts, talk to other Power BI users, and potentially even recruit new staff.



Power BI user group: <https://www.pbiusergroup.com>

Guided Learning

For introductory training on Power BI, the videos in the Guided Learning area of the Power BI site are useful place to start. Videos and written instructions are available in short increments.



Guided learning: <https://powerbi.microsoft.com/en-us/guided-learning/>

Community and Third-Party Tools

Following are some tools contributed by the vibrant Power BI user community. These are independent tools not supported by Microsoft, some of which are free, and others are paid products.

DAX Studio

DAX Studio is a client tool for executing, analyzing, and troubleshooting DAX queries.

Contributed and supported by: [SQLBI](#)



DAX Studio: <http://daxstudio.org/>

DAX Formatter

The DAX Formatter tool improves the readability and consistency of DAX code.

Contributed and supported by: [SQLBI](#)



DAX Formatter: <http://www.daxformatter.com/>

Vertipaq Analyzer

Vertipaq Analyzer is a helpful tool for analyzing tables and columns in a data model (Power BI, Power Pivot, or Analysis Services tabular), for the purpose of optimizing performance.

Contributed and supported by: [SQLBI](#)



Vertipaq Analyzer: <https://www.sqlbi.com/tools/vertipaq-analyzer/>

Tabular Editor

Tabular Editor is a lightweight editor alternative to using SQL Server Data Tools for creation of a tabular model.

Contributed and supported by: [SQLBI](#)



Tabular Editor: <https://www.sqlbi.com/tools/tabular-editor/>

Power BI Sentinel

Power BI Sentinel extends the capabilities of the Power BI service with change tracking, backups, auditing, and other features.

Supported by: [Purple Frog Systems Ltd](#)



Power BI Sentinel: <https://www.powerbisentinel.com/>

Scrims and Layouts

Scrims and Layouts are both solutions for improving productivity and quality of reports created by Power BI authors.

Supported by: [Power BI Tips](#)



Scrims: <https://powerbi.tips/2019/12/introducing-scrims/>

Layouts: <https://powerbi.tips/tools/layouts/>

Report Theme Generator

The Report Theme Generator is a web-based tool which allows a user to specify properties and download into a JSON file to be used in a Power BI theme.

Contributed and supported by: [Power BI Tips](#)



Report Theme Generator: <https://powerbi.tips/tools/report-theme-generator-v3/>

Power BI Comparer

The Power BI Comparer compares a Power BI file with another previous version.

Contributed and supported by: [Imke Feldmann](#)



DAX Studio: <https://www.thebiccountant.com/2019/09/14/compare-power-bi-files-with-power-bi-comparer-tool/>

Power BI Field Finder

The Power BI Field Finder tool assists dataset authors with documenting a model and finding unused measures and columns.

Contributed and supported by: [Stephanie Bruno](#)



Power BI Field Finder: <https://powerbi.tips/2020/01/power-bi-field-finder/>

Power BI Helper

The Power BI Helper aids in removing unused data elements from a model, finding dependencies, and in documenting a model.

Contributed and supported by: [RADACAD](#)



Power BI Field Finder: <https://radacad.com/power-bi-helper>

Power BI Documenter

Power BI Documenter is a tool which auto-generates documentation from Power BI Desktop, for the purpose of identifying and documenting data usage, visuals, and development practices.

Contributed and supported by: [Data Vizioner](#)



Power BI Documenter: <https://www.datavizioner.com/#introduction>

Power Update

Power Update is a tool which moves supports scheduled refresh operations for datasets in Power BI Desktop or Excel on a local machine.

Contributed and supported by: [Power On BI](#)



Power Update: <https://poweronbi.com/power-update/>

Turbo.net Power BI Desktop Application

The Turbo.net tools allow applications to run on any desktop. Their version allows Power BI Desktop to run on an Apple Mac.

Contributed and supported by: [Turbo.net](#)



Turbo.net: <https://turbo.net/run/powerbi/powerbi>