



>>>>>>>

dev>scope

# Rui Romano

+ 15 years "playing" with data

Head of BI Team @ **DevScope** – OPorto, **Portugal**

#DataArchitect #DataEngineer #PowerBIAddict  
#MSFTBI #DataPlatformMVP #PBIPortugal

 Rui.Romano@DevScope.net

 [@RuiRomano](https://twitter.com/RuiRomano)

 <https://www.linkedin.com/in/ruiromano/>

 <https://ruiromanoblog.wordpress.com>

 <https://www.meetup.com/Power-BI-Portugal>



**dev>scope**





# Power BI Monitoring 101

- Why you need monitoring?
- Demo
- How to do it? – Fast Paced ☺  
*Focus on the possibilities not the how...*

# Can you answer these questions?

- Who are most active users?
- Which Reports/Workspaces/Datasets/Apps are mostly used?
- Are Personal Workspaces being used? How is content being shared?
- Do your users access from Browser/Mobile/Excel? Which browser?
- Top used DataSource's? FileSystem/OneDrive? SQL? Sharepoint? DataLake?
- How many **distinct users**? Per Month? Per day? Per hour?
- DataSets Refreshing trends/average/errors?
- DataSets/Reports not used in more than 1 year?
- How many users **with license** don't use Power BI in more than 3 months?
- Are developers following Dataset naming conventions?

# NO? You need to stop Guessing and start Knowing

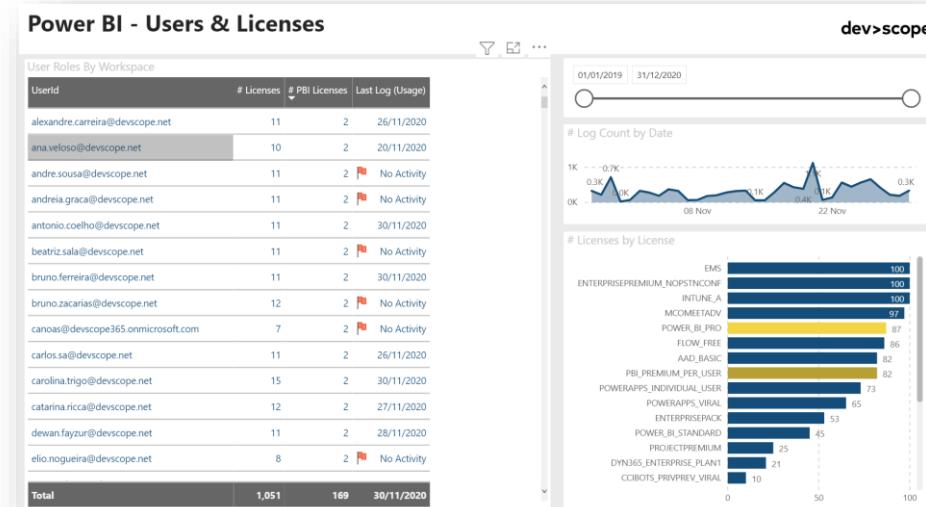
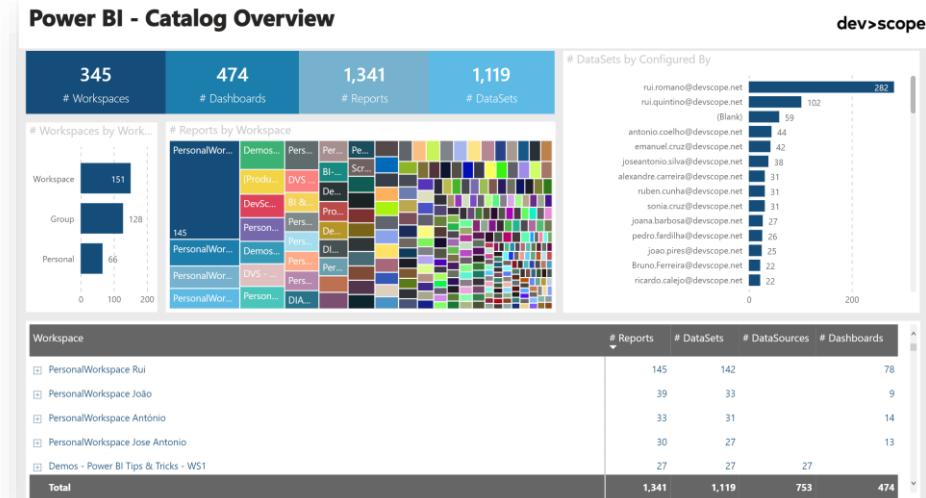
You are “driving blind” and certainly, a **BIG Reality Check** is ahead of you...

Monitoring is one of the main pillars of a good Power BI Governance strategy

This session goal is to motivate you to look at this data, it's always important to do the “BI of the BI”!



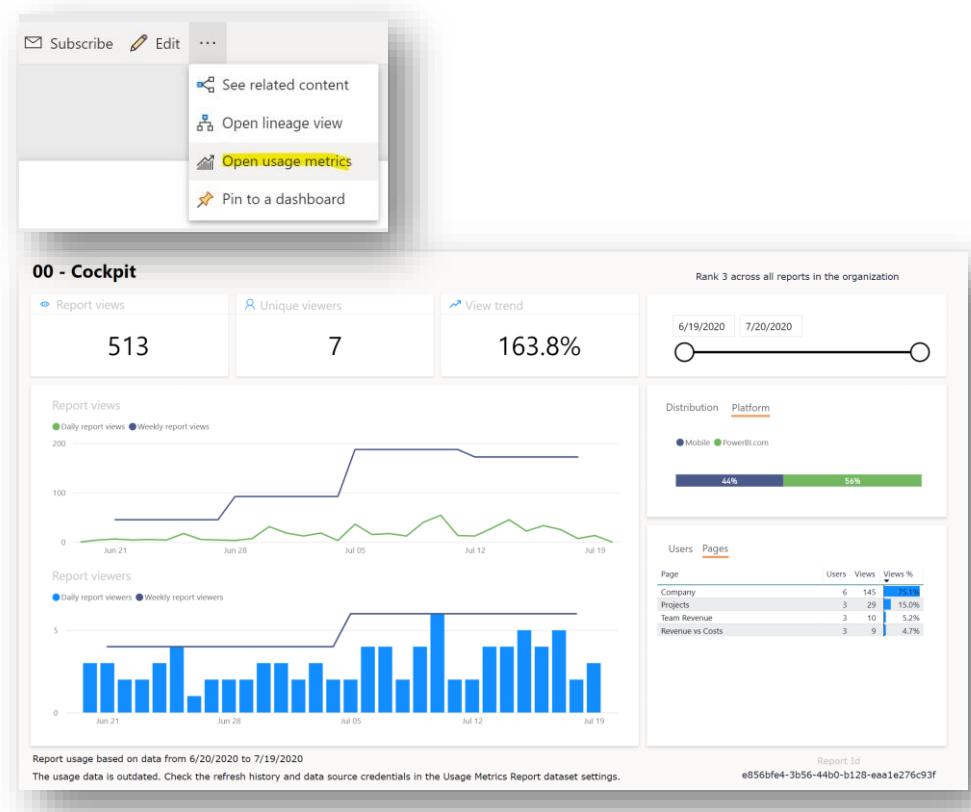
# DEMO – Power BI Monitor



# What is available Out of the Box?

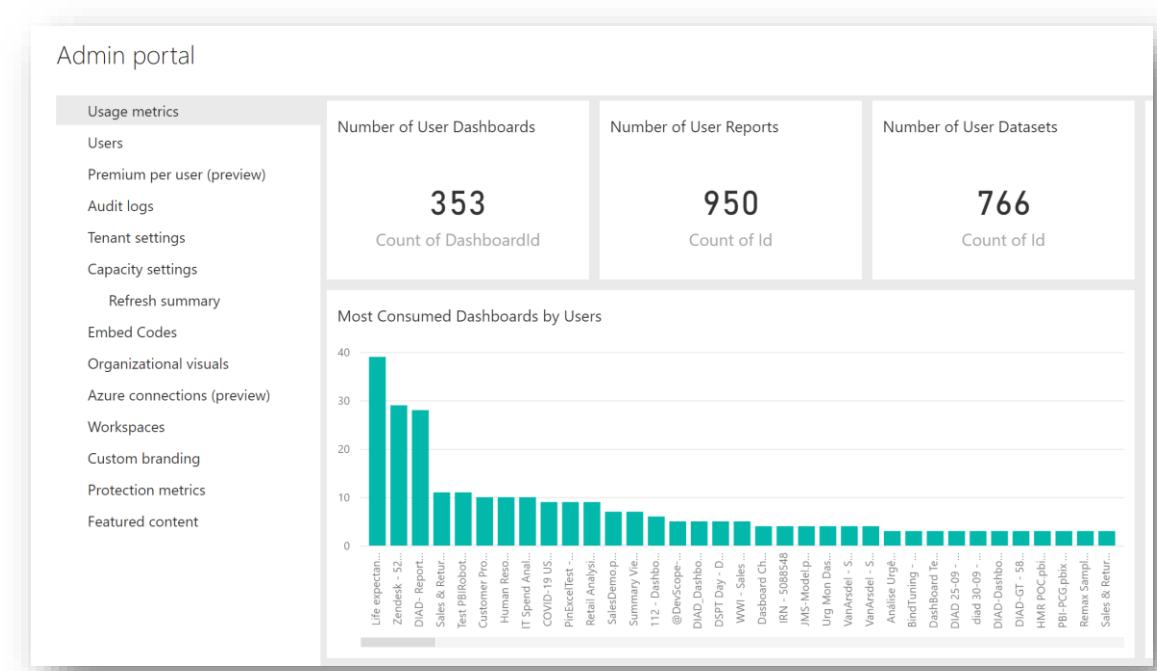
## Power BI Report Usage

- Workspace level
- Page metrics
- Client Telemetry data (time to open report)
- 7 days



## Usage Metrics in Admin Portal

- Tenant Level
- Zero Interactivity and Customization
- No Refresh Control

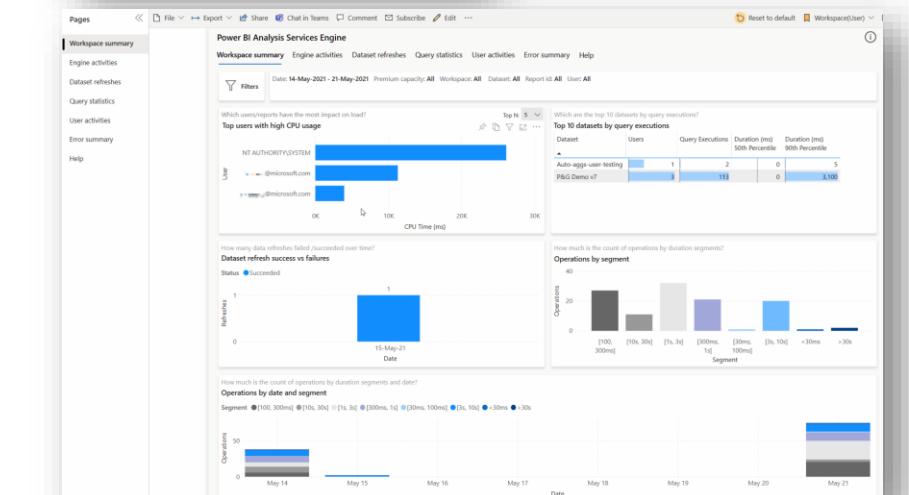
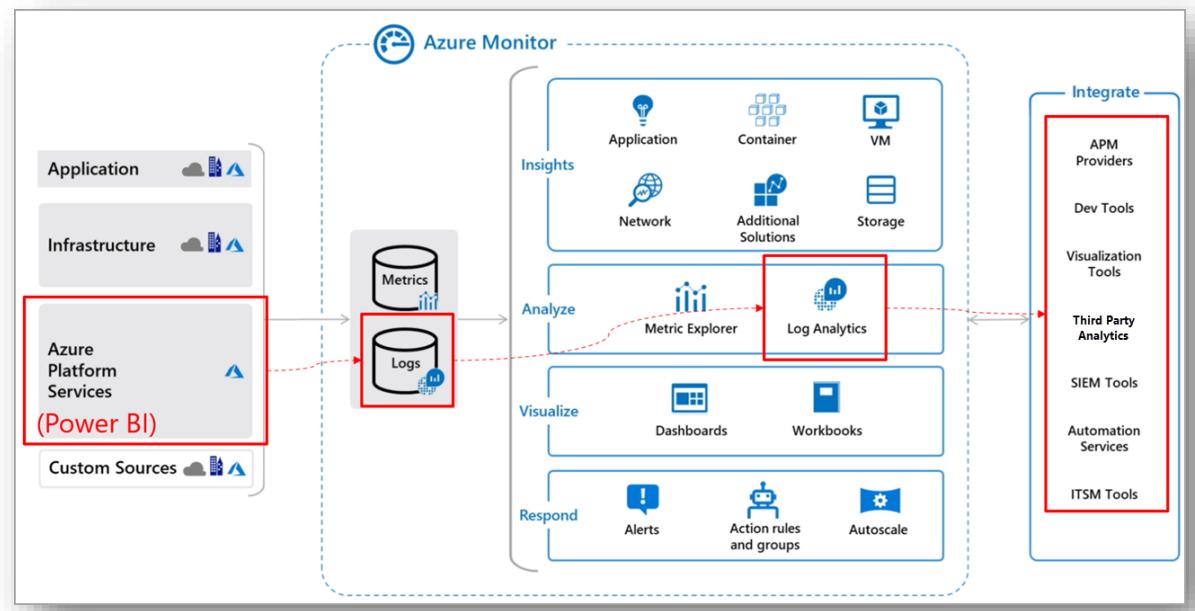


# What is available Out of the Box?

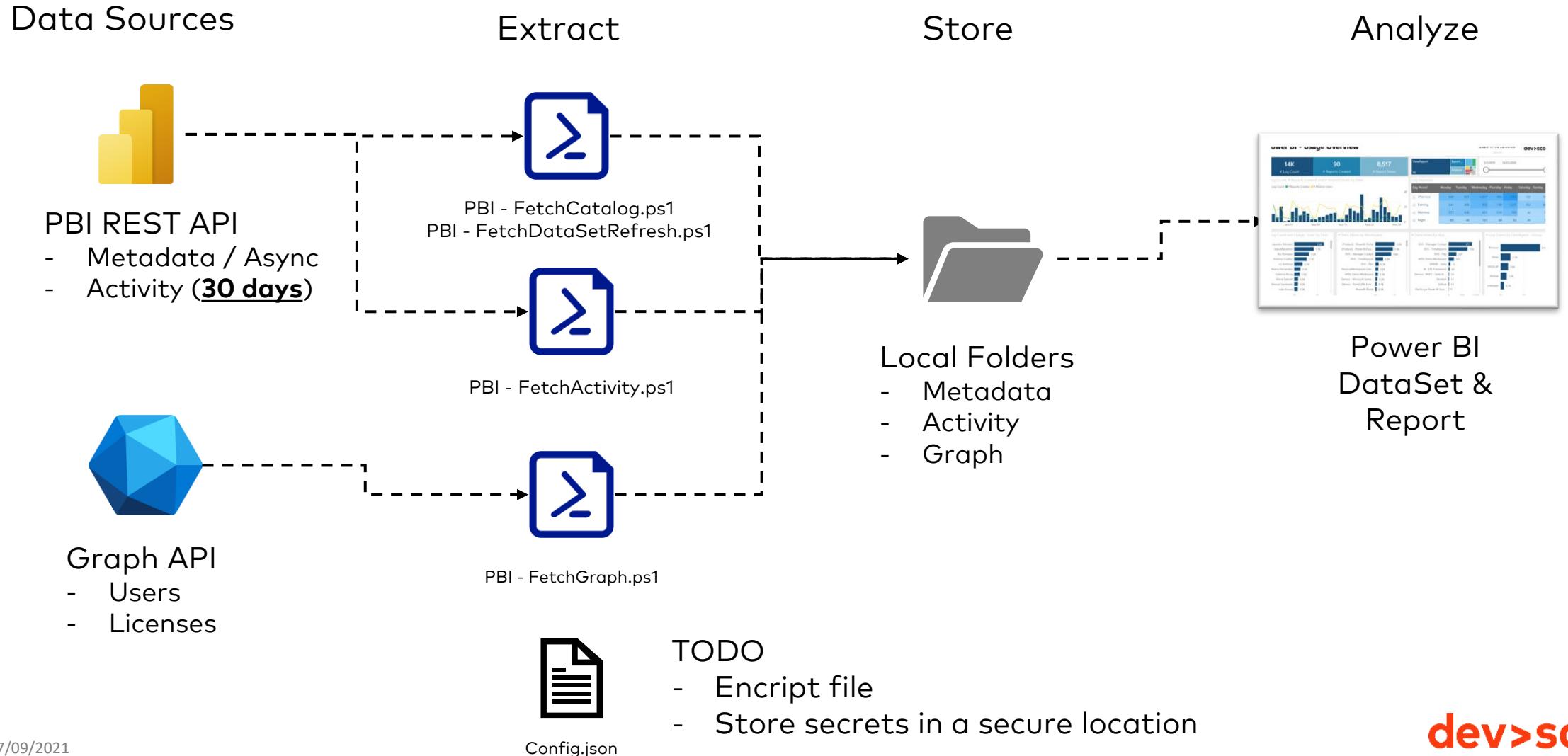
## Power BI & Log Analytics

- Dataset level – Queries, Refresh Commands,...
- Similar to [Azure AS Diagnostics](#)
- Very useful to track performance issues on a large tenant
  - Ex: Correlating a usage spike to the DAX queries
- Pre-Built [Power BI Report Template](#)
- **Preview Limitations**
  - Cannot connect multiple workspaces to same Log Analytics

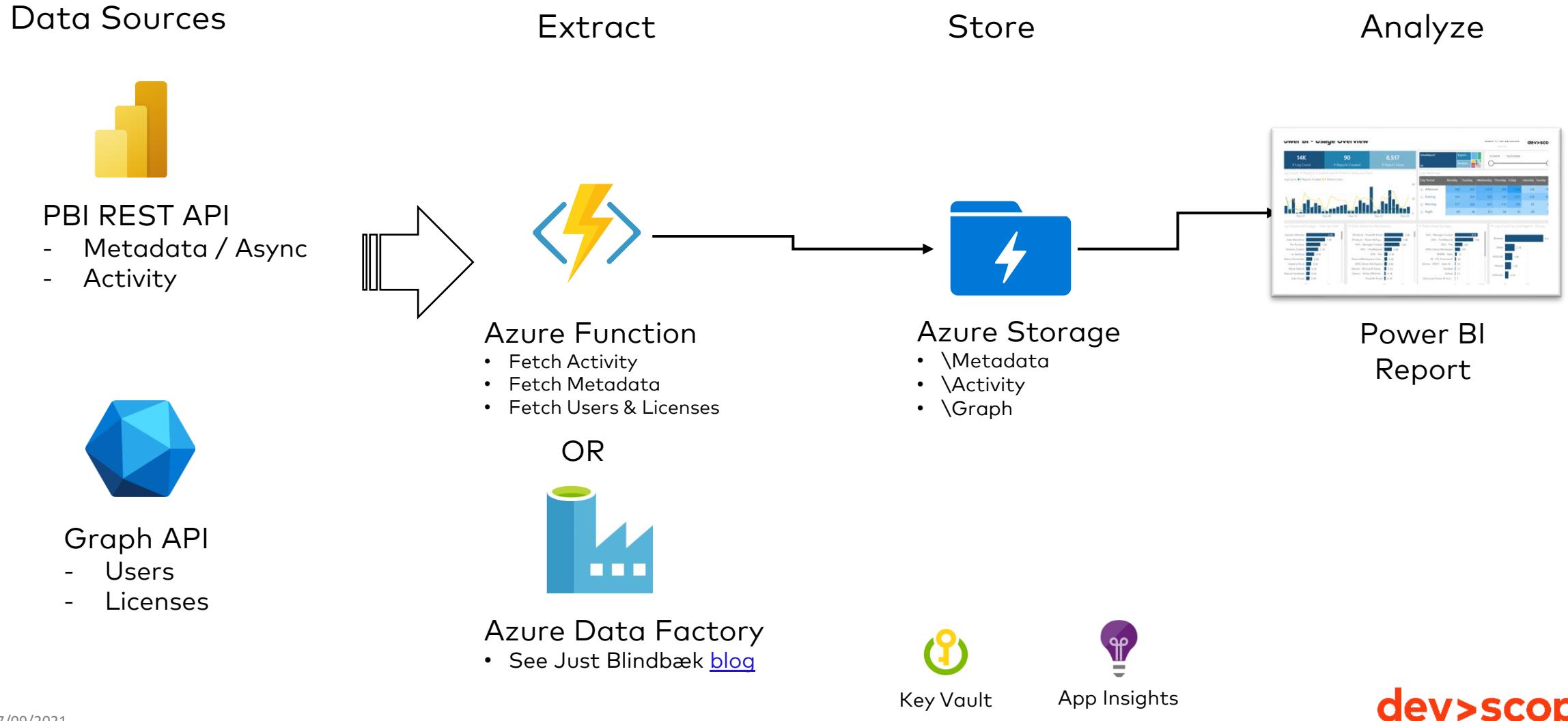
The screenshot shows the 'Settings' page for a workspace named 'admin'. The 'Azure connections (preview)' tab is selected. Under 'Log Analytics', it shows a connection to a workspace named 'admin' in a subscription and resource group. A yellow 'Disconnect from Azure' button is visible at the bottom.



# Architecture – Simplest and Easy to Share

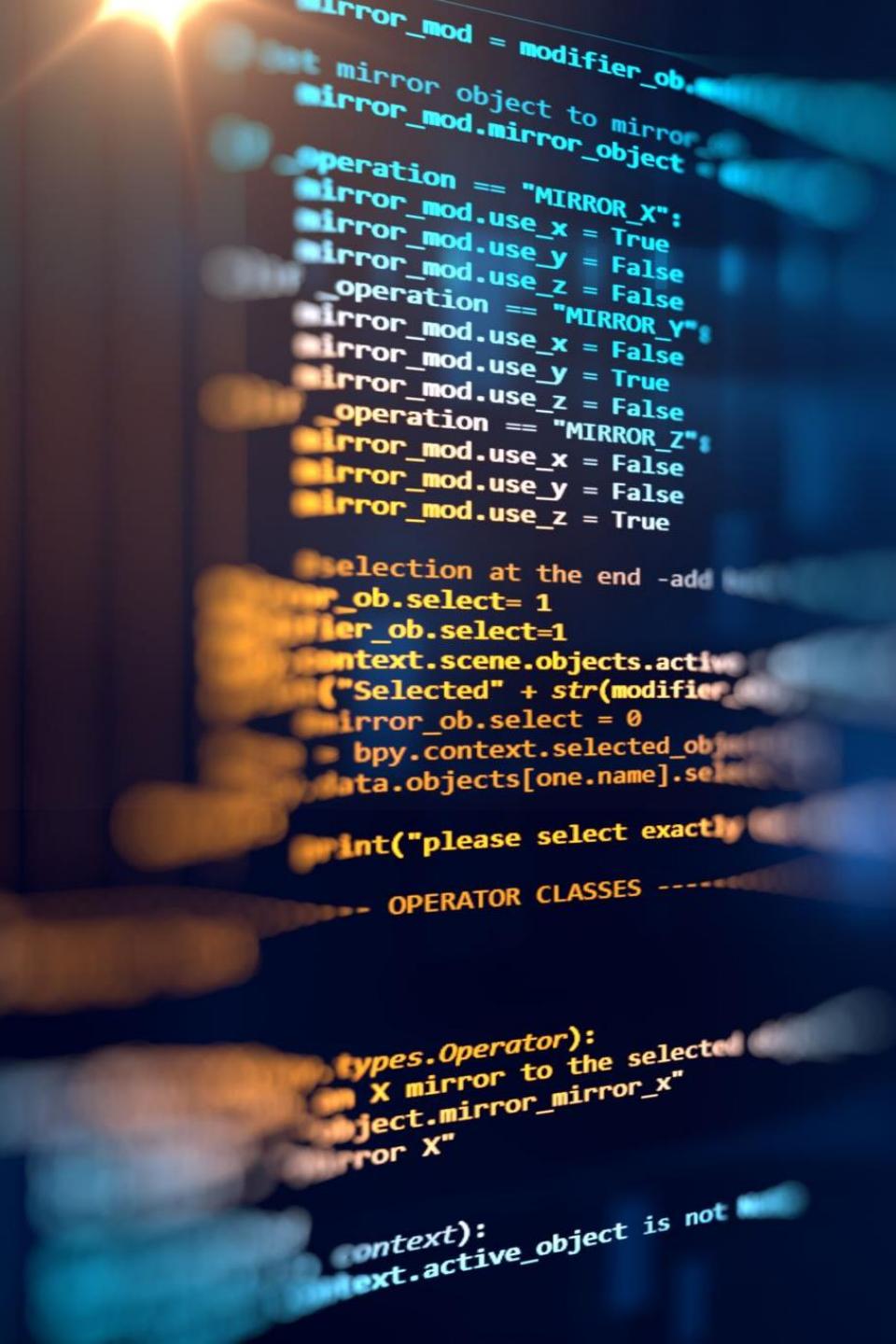


# Architecture – Recommended (one of many possibilities)



# API's Overview

Scope	Resource	API
Activity	Power BI Activity Logs	<a href="#">Admin API - Activity Events</a>
Power BI Metadata	Workspaces DataSets Reports Dashboards Permissions Schema & Lineage	<ol style="list-style-type: none"> <li>1. <a href="#">Admin Scan API – GetModifiedWorkspaces</a></li> <li>2. <a href="#">Admin Scan API – PostWorkspaceInfo</a></li> <li>3. <a href="#">Admin Scan API – GetScanStatus (loop)</a></li> <li>4. <a href="#">Admin Scan API – GetScanResult</a></li> </ol>
	RefreshHistory	<a href="#">Admin API - GetGroupsAsAdmin</a> + <a href="#">Expand DataSets</a> <a href="#">Dataset API - Get Refresh History</a>
Users & Licenses	Users & Licenses	<a href="#">Graph API – Users</a>
	Licenses Details	<a href="#">Graph API – SubscribedSKUs</a>



# Session Content

Download here:

- <https://github.com/RuiRomano/pbimonitor>

A screenshot of a GitHub repository page for 'RuiRomano / pbimonitor'. The repository is public and contains one branch ('main') and no tags. The commit history shows 16 commits from 'RuiRomano bug fix' made 1 minute ago, including changes to '.gitignore', 'Config.json', 'LICENSE', and various Python files like 'FetchActivity....', 'FetchAll.ps1', 'FetchCatalog....', 'FetchDataSet...', 'FetchGraph.ps1', 'Refreshes.pb1', 'Datasets Schema.pb1', 'PBIMonitoring101.pdf', 'README.md', and 'Setup - PreRequisites.ps1'. The commits are dated from 2 months ago to 1 minute ago.

Commit	Date	Message
b693df3	1 minute ago	RuiRomano bug fix
.gitignore	2 months ago	new content2
Config.json	2 months ago	new content2
LICENSE	3 months ago	Initial commit
PBI - Activity Monitor - FetchActivity....	1 minute ago	bug fix
PBI - Activity Monitor - FetchAll.ps1	8 days ago	fetch all script
PBI - Activity Monitor - FetchCatalog....	8 days ago	fetch all script
PBI - Activity Monitor - FetchDataSet...	8 days ago	fetch all script
PBI - Activity Monitor - FetchGraph.ps1	8 days ago	fetch all script
PBI - Activity Monitor - Refreshes.pb1	28 days ago	Switch to MicrosoftPowerBIMgmt
PBI - Activity Monitor.pb1	1 minute ago	bug fix
Datasets Schema.pb1	28 days ago	Switch to MicrosoftPowerBIMgmt
PBIMonitoring101.pdf	28 days ago	Switch to MicrosoftPowerBIMgmt
README.md	28 days ago	Update README.md
Setup - PreRequisites.ps1	8 days ago	fetch all script

# Requirements

- Be a Power BI Administrator (or friend to one 😊)
- Permissions to create an Azure AD Application / Service Principal
- Permission to create an Azure AD Security Group

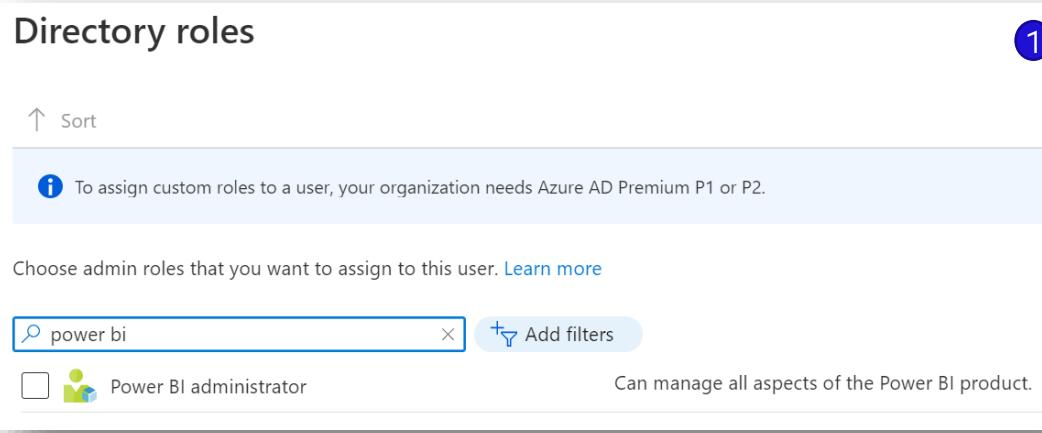
Directory roles

↑ Sort

To assign custom roles to a user, your organization needs Azure AD Premium P1 or P2.

Choose admin roles that you want to assign to this user. [Learn more](#)

Power BI administrator Can manage all aspects of the Power BI product.



1 Rui Romano AD (MVP Subscription) | User settings ... 2

Azure Active Directory

Overview Preview features Diagnose and solve problems

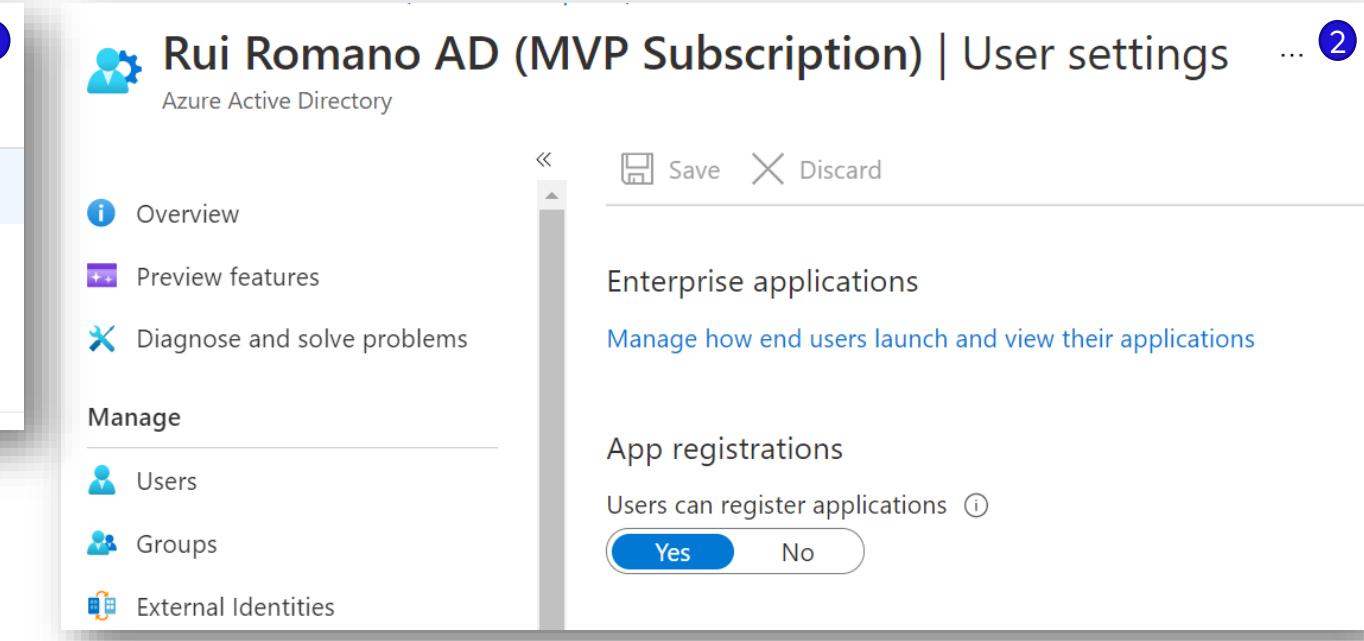
Manage

Users Groups External Identities

Save Discard

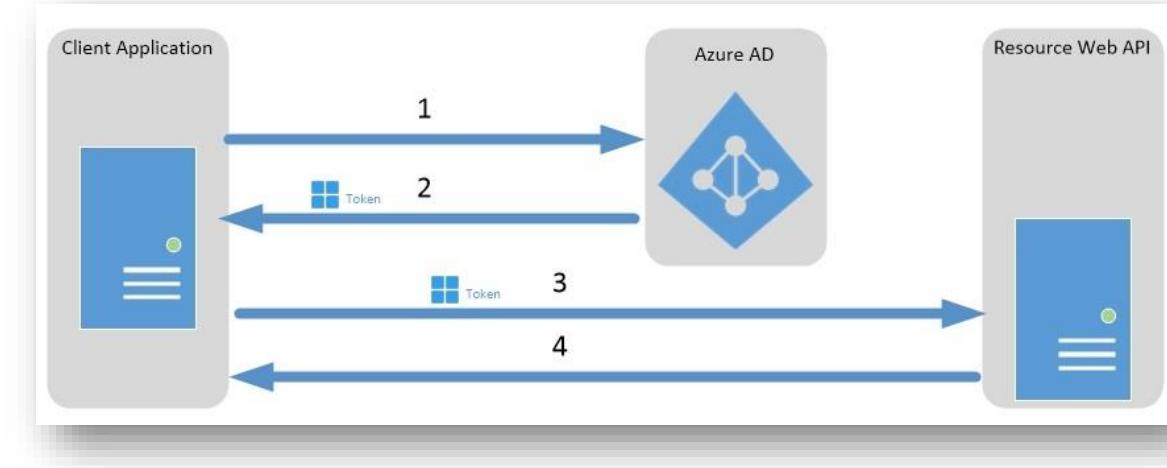
Enterprise applications Manage how end users launch and view their applications

App registrations Users can register applications ⓘ Yes No



# API Authentication

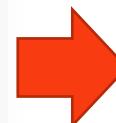
- OAuth 2.0
- Authentication Flows:



Authentication Flow	Interactive	Requirements
<a href="#"><u>Client Credentials (Recommended)</u></a>	No	Azure AD Service Principal
<a href="#"><u>Auth Code</u></a>	Yes	Power BI Account

```
POST /token HTTP/1.1
Host: authorization-server.com

grant_type=client_credentials
&client_id=xxxxxxxxxx
&client_secret=xxxxxxxxxx
```



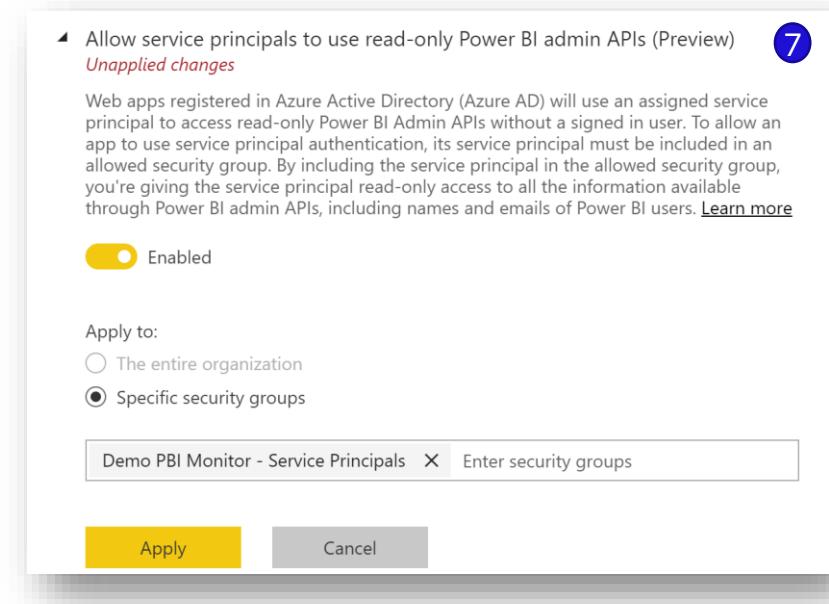
```
HTTP/1.1 200 OK
Content-Type: application/json
Cache-Control: no-store
Pragma: no-cache

{
  "access_token": "MTQ0NjJkZmQ5OTM2NDE1ZTZjNGZmZjI3",
  "token_type": "bearer",
  "expires_in": 3600,
  "refresh_token": "IwOGYzYTlmM2YxOTQ5MGE3YmNmMDFkNTVk",
  "scope": "create"
}
```

# Service Principal Step by Step

1. Go to [Azure AD Active Directory](#)
2. Go to [App Registrations](#) and create a new App (leave defaults)
3. Generate a new App Secret
4. Save the [App Id](#), [App Secret](#) & [Tenant Id](#)
5. [Create an Azure AD Security Group](#)
6. Add the Service Principal to the Security Group as a member
7. Authorize the Security Group in Power BI Admin Portal
  - Allow service principals to use Power BI APIs
  - Allow service principals to use read-only Power BI admin APIs
8. **Optional** - Authorize the Service Principal to Access Graph API

Note: You don't need to add any Power BI API Permissions



API / Permissions name	Type	Description
<b>Microsoft Graph (3)</b>		
Organization.Read.All	Application	Read organization information
User.Read	Delegated	Sign in and read user profile
User.Read.All	Application	Read all users' full profiles

# Power BI Tenant Settings

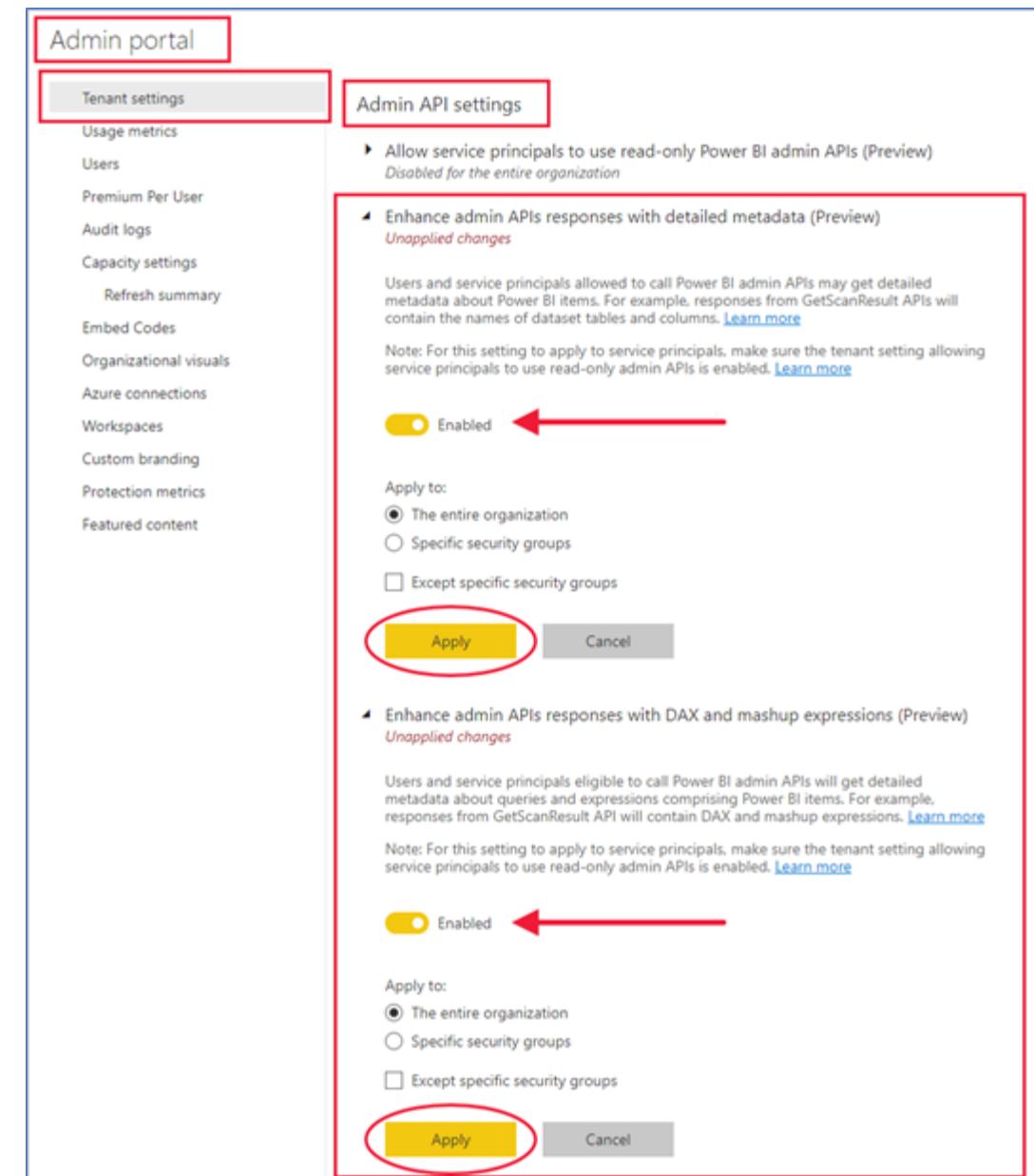
Enable the following Tenant Settings:

- Enhance admin APIs responses with detailed metadata (Preview)

Turns on the caching flow and enhances API responses with low-level metadata (for example, name and description) for tables, columns, and measures.

- Enhance admin APIs responses with DAX and mashup expressions (Preview)

Allows the API response to include DAX expressions and Mashup queries.  
This setting can only be enabled if the first setting is also enabled.





## milestones

- ✓ IT/BI Admin Support
- ❑ Extract Data
- ❑ Data Store
- ❑ Power BI DataSet
- ❑ Power BI Report

# Script PBI - FetchActivity.ps1

- Incrementally fetch Power BI Activity  
Uses [PowerBIPS](#) module
- Uses the Power BI Admin API:  
[Get Activity Events](#)
- Can only go back **30 days**

```

write-Host "Getting OAuth Token"
$credential = New-Object -TypeName System.Management.Automation.PSCredential -ArgumentList $config.ServicePrincipal
Connect-PowerBIServiceAccount -ServicePrincipal -Tenant $config.ServicePrincipal.TenantId -Credential $credential
if ($config.Activity.LastRun)
{
    $pivotDate = [datetime]::Parse($config.Activity.LastRun).ToUniversalTime()
}
else
{
    $config | Add-Member -NotePropertyName "Activity" -NotePropertyValue @{"LastRun" = $null} -Force
    $pivotDate = [datetime]::UtcNow.Date.AddDays(-30)
}

# Gets audit data daily
while($pivotDate -le [datetime]::UtcNow)
{
    write-Host "Getting audit data for: '$($pivotDate.ToString("yyyyMMdd"))'"
    $outputFilePath = ("$outputPath\{0:yyyyMMdd}.json" -f $pivotDate)
    $audits = Get-PowerBIActivityEvent -StartTime $pivotDate.ToString("s") -EndTime $pivotDate.AddHours(2)
    if (!$audits -is [array])
    {
        $audits = @($audits)
    }
    if ($audits.Count -gt 0)
    {
        write-Host "'$($audits.Count)' audits"
        New-Item -Path (split-path $outputFilePath -Parent) -ItemType Directory
        ConvertTo-Json @($audits) -Compress -Depth 5 | Out-File $outputFilePath
    }
    else
    {
        write-warning "No audit logs for date: '$($pivotDate.ToString("yyyyMMdd"))'"
    }
}

```

```

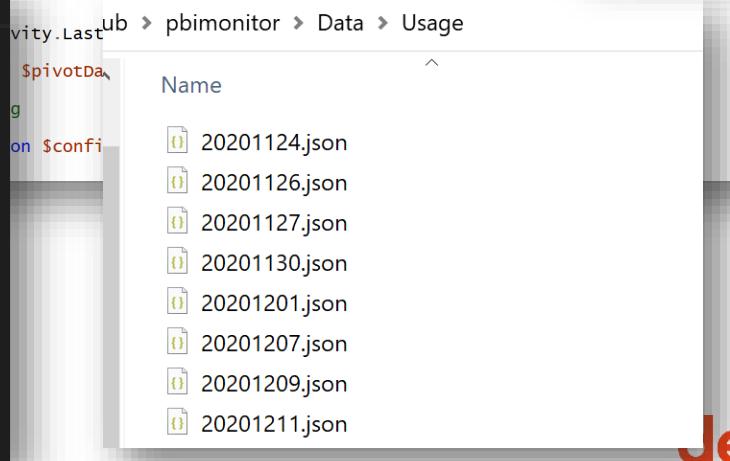
PS C:\@Repos\Github\pbimonitor> C:\@Repos\Github\pbimonitor>
Getting OAuth Token
Getting audit data for: '20210218'
'1322' audits
Getting audit data for: '20210219'
'223' audits
PS C:\@Repos\Github\pbimonitor>

```

```

1  {
2      "ServicePrincipal": {
3          "AppId": "",
4          "AppSecret": "",
5          "TenantId": ""
6      },
7      "Catalog": {
8          "LastRun": "2021-02-18T00:00:00.000000Z"
9      },
10     "Activity": {
11         "LastRun": "2021-02-18T00:00:00.000000Z"
12     }
}

```



# Script – PBI - FetchCatalog.ps1

- Snapshot the entire tenant metadata:  
Workspaces (personal included)  
DataSets  
DataSources  
Reports  
Dashboards  
Permissions – All objects  
Lineage  
Dataset Schema
- Uses the new Admin [Async API](#) - Faster & Incremental
  1. GetModifiedWorkspaces
  2. PostWorkspaceInfo
  3. GetScanStatus (loop)
  4. GetScanResult
- Missing on Async API:  
Workspace Users + Roles
  - [Admin Get Groups](#) + \$expand=users
  - Admin Apps

```

write-Host "Getting workspaces to scan"
$modifiedRequestUrl = "admin/workspaces/modified"
if ($config.Catalog.LastRun -and !$reset)
{
    $modifiedRequestUrl = $modifiedRequestUrl + "?modifiedsince=$($config.Catalog.LastRun)"
}
else
{
    $config | Add-Member -NotePropertyName "Catalog" -NotePropertyValue @{"LastRun" = $null} -Force
}

write-Host "Reset: $reset"
Write-Host "Since: $($config.Catalog.LastRun)"

# Get Modified Workspaces since last scan
$workspacesModified = Invoke-PowerBIRestMethod -url $modifiedRequestUrl -Method Get | ConvertFrom-Jso
if (!$workspacesModified)
{
    write-Host "No workspaces modified"
}

write-Host "Modified workspaces: $($workspacesModified.count)"
$config.Catalog.LastRun = [datetime]::UtcNow.Date.ToString("o")

$skip = 0
$batchCount = 100
$workspacesScanRequests = @()
# call GetInfo to request workspace scan in batches of 100 (throttling after 500 calls per hour) https://github.com/microsoft/powerbi-admin-dotnet
do
{
    $workspacesBatch = @($workspacesModified | Select -First $batchCount -skip $skip)
    if ($workspacesBatch)
    {
        Write-Host "Requesting workspace"
        $bodyStr = @{
            "workspaces" = @($workspacesBatch)
            "getInfoDetails" = "lineage=true&status=true"
        }
        $workspacesScanRequests += (Invoke-PowerBIRestMethod -url $modifiedRequestUrl -Method Post -Body ($bodyStr | ConvertTo-Json))
        $skip += $batchCount
    }
} while($workspacesBatch.count -ne 0 -and $workspacesScanRequests.Count -lt 500)

# Wait for scan to execute - https://docs.microsoft.com/en-us/power-bi/admin/get-workspace-scan-status
while(@( $workspacesScanRequests |? status -in @("Running", "Notstarted")))
{
    Start-Sleep -Seconds 1
}

PS C:\@Repos\Github\pbimonitor> C:\@Repos\Github\pbimonitor>
Fetching 5000 /admin/workspaces
Getting workspaces to scan
Since: 2021-02-18T00:00:00.0000000Z
Modified workspaces: 5
Requesting workspace scan: 100 / 5
Waiting for scan results...
Scan 'bbf36328-5210-4006-bd45-830cf22e2793' : 'succeeded'
Scan Result 'bbf36328-5210-4006-bd45-830cf22e2793' : '5'
Elapsed: 9.0469164s
PS C:\@Repos\Github\pbimonitor>

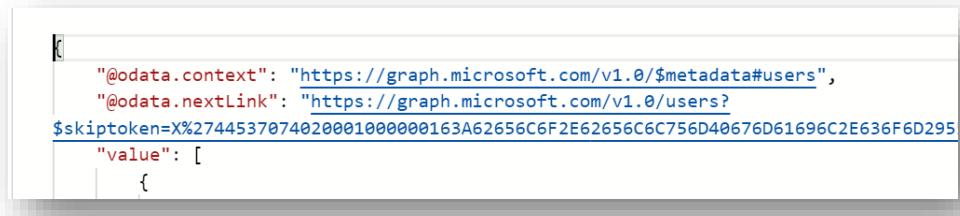
```

github > pbimonitor > Data.DVS > Catalog > 2021 > 02 > 17 >

Name	Date modified
scans	17/02/2021 15:44
WORKSPACES.USERS.JSON	17/02/2021 15:16

# Script – PBI - FetchGraph.ps1

- Snapshot tenant:
  - Users & Assigned Licenses
  - Tenant Subscribed SKUs
- Uses the Microsoft Graph API
  - Users
  - SubscribedSkus
- Paginated API



```

New-Item -ItemType Directory -Path $outputPath -ErrorAction SilentlyContinue | Out-Null
# Get the authentication token
$authToken = Get-AuthToken -resource "https://graph.microsoft.com" -appid $servicePrincipal.AppId -
$graphUrl = "https://graph.microsoft.com/beta"
Write-Host "Getting Users from Graph"
$users = Read-FromGraphAPI -accessToken $authToken -url "$graphUrl/users?$select=id,mail,companyName"
$filePath = "$outputPath\Graph.Users.json"
$users | ConvertTo-Json -Compress -Depth 5 | Out-File $filePath -Force
Write-Host "Getting SKUs from Graph"
$skus = Read-FromGraphAPI -accessToken $authToken -url "$graphUrl/subscribedskus?$select=id,capabilities"
$filePath = "$outputPath\Graph.SKUs.json"
$skus | ConvertTo-Json -Compress -Depth 5 | Out-File $filePath -Force
  
```

ithub > pbimonitor > Data.DVS > Graph > 2021 > 02 > 17

Name	Date modified
subscribedSkus.json	17/02/2021 18:50
users.json	17/02/2021 18:50

# Script – PBI - FetchDataSetRefresh.ps1

- There is no Admin API to get DataSetRefresh History
- Ensure the service principal is a member of every workspace to monitor, manually or [script](#)
- Loop all datasets and call “Refreshes” Api that get the latest refreshes for the dataset

```

foreach($workspace in $Workspaces)
{
    $item++
    Write-Host "Processing workspace: '$($workspace.Name)' $item/$total"
    Write-Host "Datasets: $($workspace.datasets.Count)"

    $refreshableDatasets = @($workspace.datasets | ? { $_.isRefreshable -eq $true -and $_.addRowsAPIEnabled -eq $false})
    Write-Host "Refreshable Datasets: $($refreshableDatasets.Count)"

    foreach($dataset in $refreshableDatasets)
    {
        try
        {
            Write-Host "Processing dataset: '$($dataset.name)'"
            Write-Host "Getting refresh history"

            $dsRefreshHistory = Invoke-PowerBIRestMethod -url "groups/$($workspace.id)/datasets/$($dataset.id)/refreshes"
            $dsRefreshHistory = $dsRefreshHistory.value

            if ($dsRefreshHistory)
            {
                $dsRefreshHistory = $dsRefreshHistory | select *, @{Name="datasetId"; Expression={ $dataset.id }}, @{Name="group"; Expression={ $workspace.name }}, @{Name="configuredBy"; Expression={ $dataset.configuredBy }}
                $dsRefreshHistoryGlobal += $dsRefreshHistory
            }
        }
        catch
        {
            "value": [
            {
                "refreshType": "ViaApi",
                "startTime": "2017-06-13T09:25:43.153Z",
                "endTime": "2017-06-13T09:31:43.153Z",
                "serviceExceptionJson": "{\"errorCode\":\"ModelRefreshFailed_CredentialsNotSpecified\"}",
                "status": "Failed",
                "requestId": "11bf290a-346b-48b7-8973-c5df149337ff"
            }
            ]
        }
    }
}

```

# API Throttling

- Power BI & Graph API's have throttling enabled
- Handle the exception "429 Too Many Requests"

## Admin - Get Activity Events

Service: Power BI REST APIs

API Version: v1.0

Returns a list of audit activity events for a tenant.

**Note:** Activity logging isn't supported for Microsoft Cloud Deutschland. The user must have administrator rights (such as Office 365 Global Administrator or Power BI Service Administrator) to call this API or authenticate via service principal.

This API allows 200 requests per hour at maximum.

1 | HTTP/1.1 429 Too Many Requests  
2 | Content-Type: text/html  
3 | Retry-After: 3600

```
        }  
    }  
    catch [System.Net.WebException]  
{  
        $ex = $_.Exception  
        $statusCode = $ex.Response.StatusCode  
        if ($statusCode -eq 429)  
        {  
            $waitSeconds = [int]::Parse($ex.Response.Headers["Retry-After"])  
            Write-Host "429 Throttling Error - Need to wait $waitSeconds seconds..."  
            Start-Sleep -Seconds ($waitSeconds + 5)  
        }  
    }  
}
```



## milestones

- ✓ IT Admin Support
- ✓ Extract Data
- ✓ Data Store
- Power BI DataSet
- Power BI Report

# Power BI - PowerQuery

- Timezone offset, all dates are UTC
- Proxy Query to reference all files, easy switch data source (ex: folder or data lake)

The screenshot shows the Power BI interface with the 'FilesProxy' query editor open. On the left, the 'Queries [41]' pane is visible, showing a tree structure of queries categorized under 'Catalog [13]', 'Usage [3]', 'Files - FileSystem [7]', and 'Other Queries [2]'. A red arrow points from the 'Catalog-Files' node in the tree to the 'Source' part of the query code. Another red arrow points from the 'FilesProxy' node in the tree to the 'Content' column of a preview table on the right. The preview table has columns: Content, Filename, Date, and FileType. It lists 11 rows of binary files, all scanned on 10/02/2021.

```
let
    Source = [ActivityFiles = "#Activity-Files from Folder"
    , CatalogFiles = "#Catalog-Files from Folder"
    , GraphFiles = "#Graph-Files from Folder"
    ]
    //Source = [ActivityFiles = "#Activity-Files from BlobStorage"
    //, CatalogFiles = "#Catalog-Files from BlobStorage"
    //, GraphFiles = "#Graph-Files from BlobStorage"
    //]
in
```

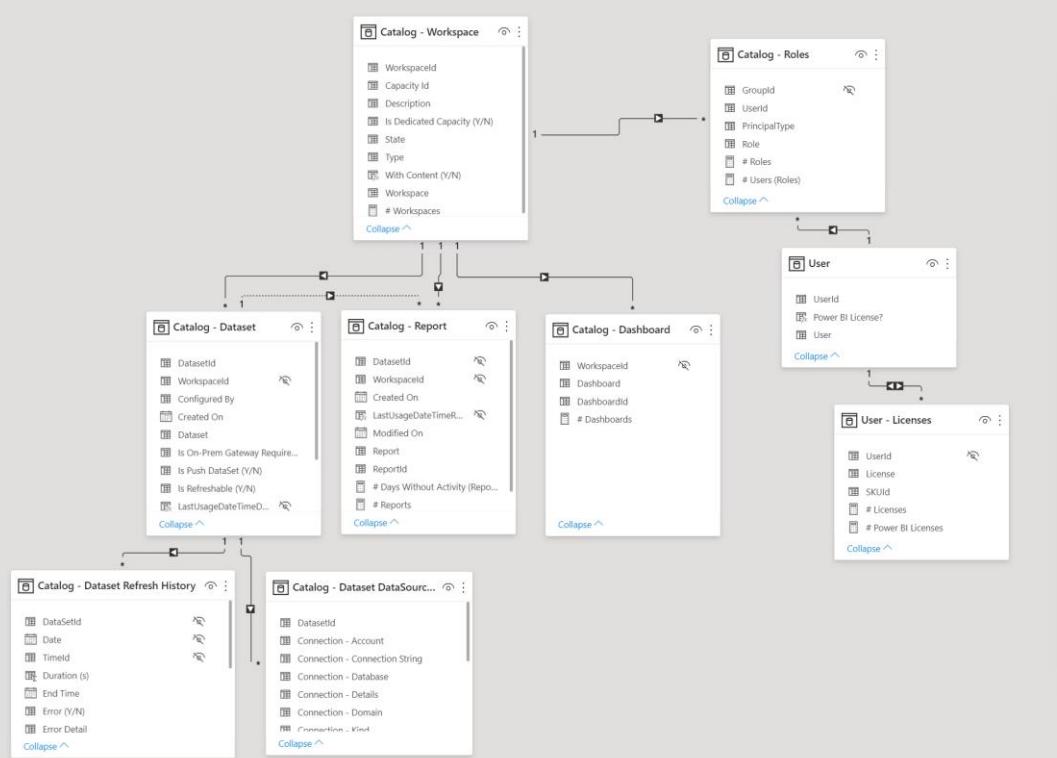
Content	Filename	Date	FileType	
1	Binary	0224742b-abc1-4ddb-8e35-7cda46b31b14.json	10/02/2021	scan
2	Binary	03b400bb-0473-4e16-a5d5-62052443ca05.json	10/02/2021	scan
3	Binary	05f460dd-3944-4fb4-a607-db5c6b8a6a1d.json	10/02/2021	scan
4	Binary	09c9baf6-6df7-4bf5-b7a3-77d277527c49.json	10/02/2021	scan
5	Binary	0bef91e1-9d41-4cd3-9a43-ccc19857b41c.json	10/02/2021	scan
6	Binary	0c0fed54-fb15-4fc4-bca8-8bb062fb5259.json	10/02/2021	scan
7	Binary	0ef464ae-b606-4095-87d6-2285536300da.json	10/02/2021	scan
8	Binary	156a0679-a542-45c7-b1df-bb89181d1d86.json	10/02/2021	scan
9	Binary	18200c8c-7b91-4945-b483-e4190293ea2b.json	10/02/2021	scan
10	Binary	1a75f1d8-57f7-4d87-94db-ad656166b104.json	10/02/2021	scan
11	Binary	1b3eerb5-efda-4b37-965f-1e7fb7f50ac4.json	10/02/2021	scan

The screenshot shows the 'Queries [40]' pane on the right side of the Power BI interface. It lists various global parameters, catalog items, usage metrics, and specific file system entries. The 'Catalog-Files' item is highlighted with a yellow bar at the top of the list. The 'FilesProxy' entry is also visible in the 'Files - FileSystem' section.

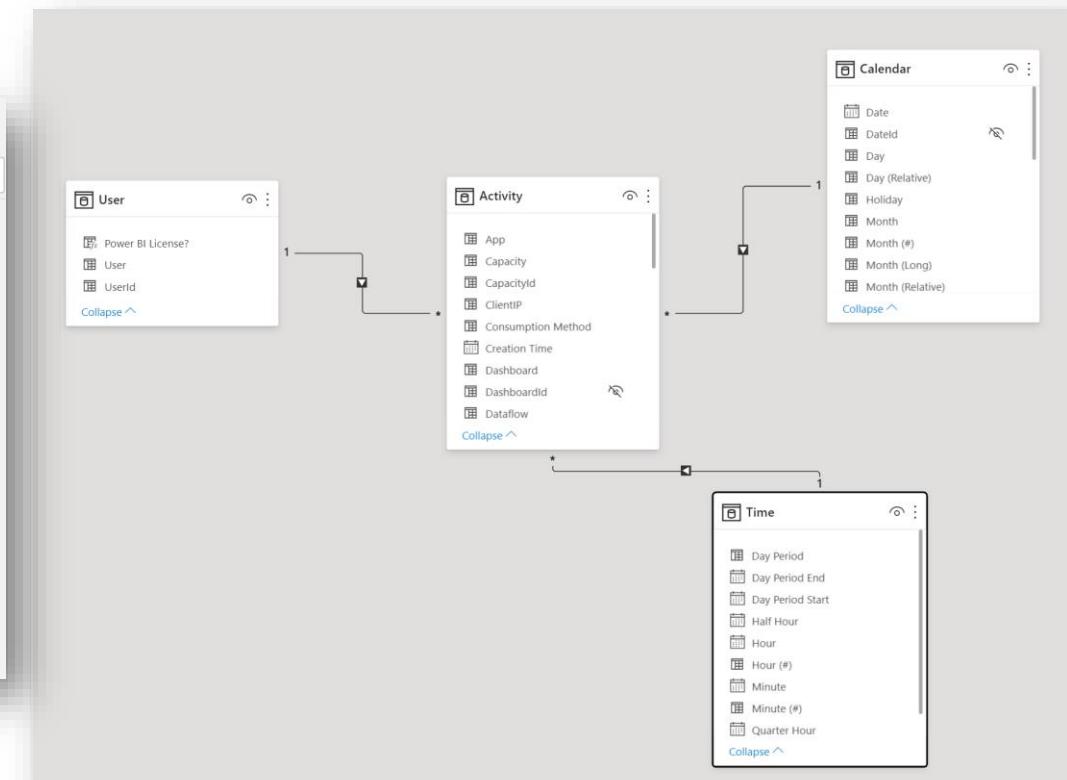
- Global Parameters [1]
  - TimezoneOffset (0)
- Calendar [5]
- Catalog [13]
  - Catalog-Files
  - Catalog-Files-Last
  - Catalog-Scans
  - Catalog - Workspaces RAW
  - Catalog - Dataset RAW
  - Catalog - DataSource
  - Catalog - DatasetDataSource
  - Catalog - Report
  - Catalog - Dashboard
  - Catalog - Dataset
  - Catalog - Workspace
  - Catalog - Refresh History
  - Catalog - Roles
- Usage [3]
  - Activity-Files
  - Activity-RAW
  - Activity
- Users [6]
  - Graph-Files
  - Graph-Files-Last
  - O365SKUs
  - Users - RAW
  - User
  - User - Licenses
- Files - FileSystem [7]
  - DataFolder (C:\@Repos\Github\pbimonitor\Data.D...
  - PBI\CatalogDataFolder
  - PBI\ActivityDataFolder
  - GraphDataFolder
  - Catalog-Files from Folder
  - Activity-Files from Folder
  - Graph-Files from Folder
  - FilesProxy
  - System

# Power BI - DataSet, 2 Models in 1

Catalog



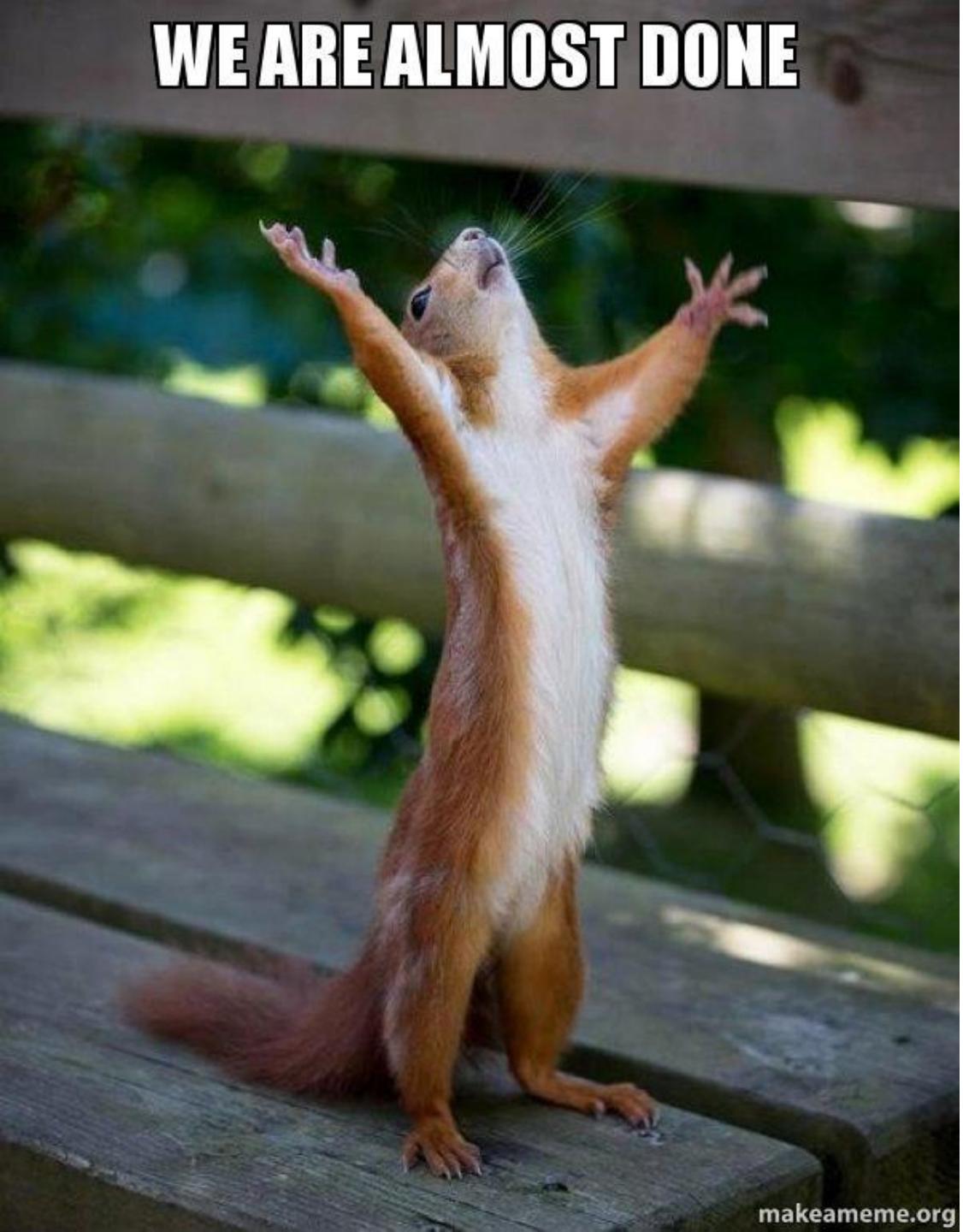
Activity



# Power BI – DataSet Main Measures

Catalog	Activity
# Reports	# Logs
# DataSets	<b># Logs from Excel</b>
# Reports	# Report Views
...	<b># Reports Created</b>
<b># Days without Activity</b>	<b># Distinct Users by Day</b>
# Licenses	<b># Data Views</b>
# Users	Activity Grouping => DataViews, Authoring, Admin

# WE ARE ALMOST DONE



## ■ milestones

- ✓ IT Admin Support
- ✓ Extract Data
- ✓ Data Store
- ✓ Power BI DataSet
- ❑ Power BI Report

# Power BI - Report

- Base theme, easy company branding
- Drillthrough for detail
- Look at No Activity
- Quick search for a report/dataset by id / name
- Make use of **advanced AI features:**  
Explain Increase/Decrease  
Anomaly Detection  
Forecast



dev>scope



## ■ milestones

- ✓ IT Admin Support
- ✓ Extract Data
- ✓ Data Store
- ✓ Power BI DataSet
- ✓ Power BI Report

## Rui Romano

- ❖ rui.romano@devscope.net
- ❖ linkedin.com/in/ruiromano
- ❖ @ruiromano
- ❖ <https://ruiromanoblog.wordpress.com>



dev>scope

