

Power BI Monitoring 101



2021-12

Rui Romano

Senior Program Manager | PBI CAT

Microsoft

 @RuiRomano

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

 <https://ruiromanoblog.wordpress.com>



Lives in Porto, Portugal

+15 years working on Microsoft
Data Stack

 Power BI & Data Architecture



Slides & Content:

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

Power BI Monitoring 101

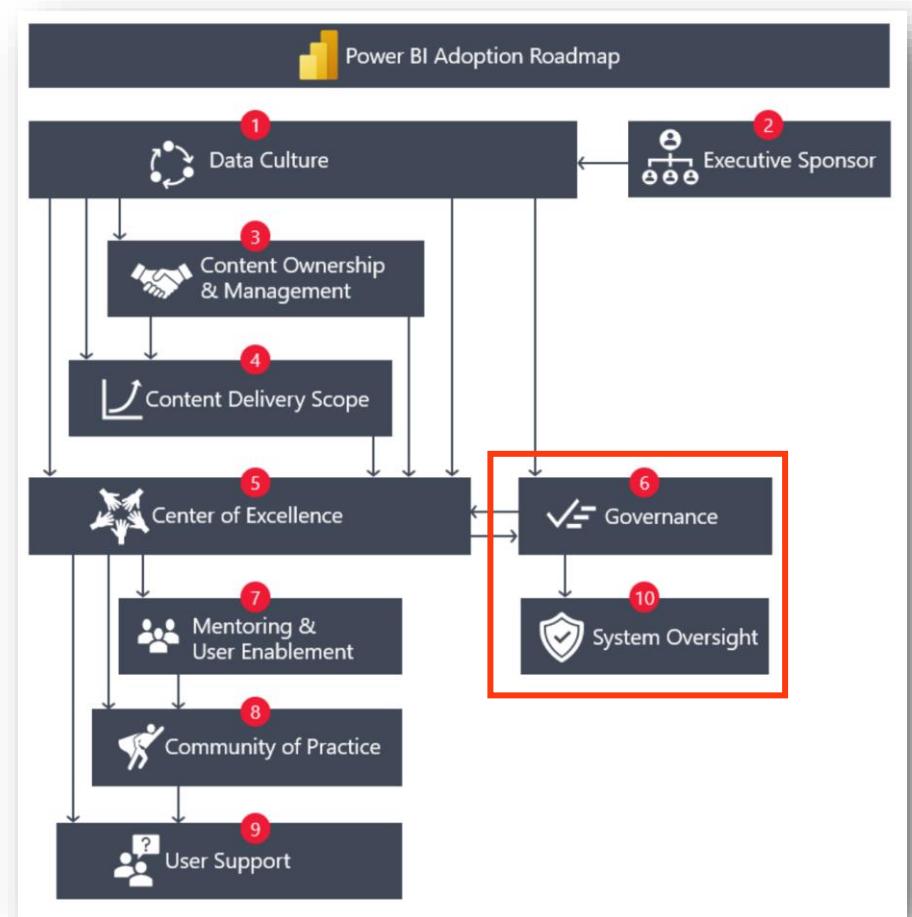
- Why you need monitoring?
- Demo
- How to do it? – Fast Paced ☺
Focus on the possibilities not the how...

Why Monitoring?

Power BI Monitoring is one of the pillars for a good Power BI Governance, allows you to:

- Focus on what really matters
- Detect Patterns & react to them
- Get Insights & Anticipate issues (ex: wrong sharing, non authorized export)
- Enforce guidelines (ex: naming convention)

Power BI Adoption Roadmap



Can you answer these questions?

- Who are most active users?
- Which Reports/Workspaces/Datasets/Apps are used every day?
- How is content being shared & distributed? Any content on My Workspace?
- Do your users access from Browser/Mobile/Excel? Which browser?
- Top used DataSource's? FileSystem/OneDrive/DataLakes? SQL? Oracle?
- How many **distinct users**? Per Month? Per day? Per hour?
- 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 the 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...

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



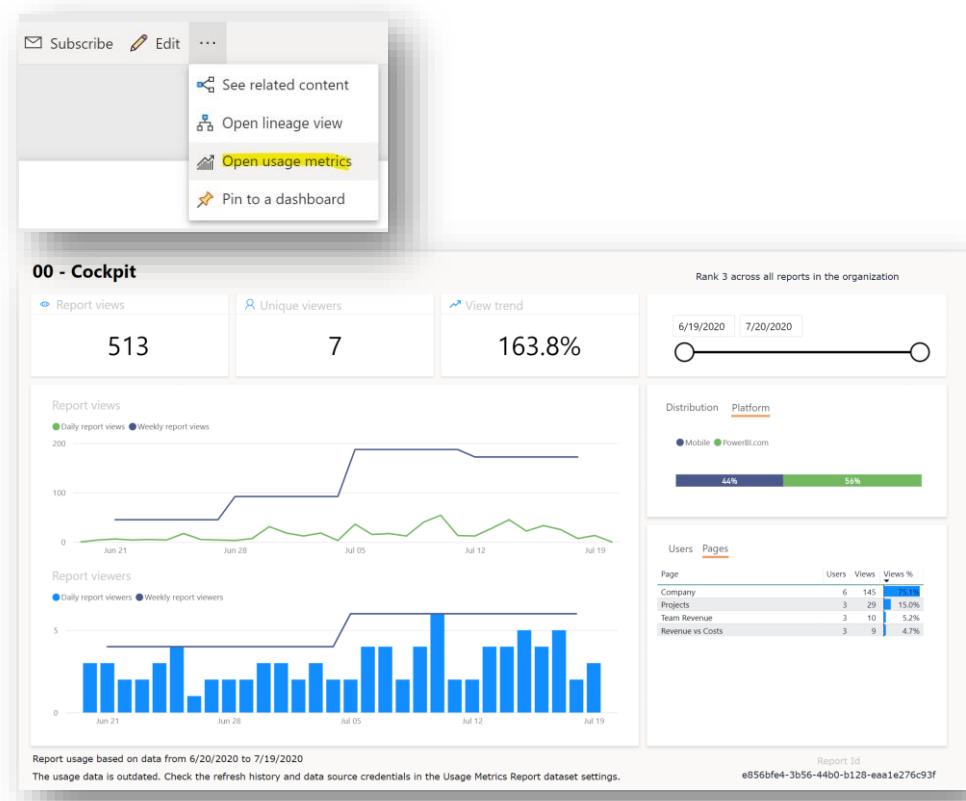
Whats available Out of the Box?



Report & Admin Usage

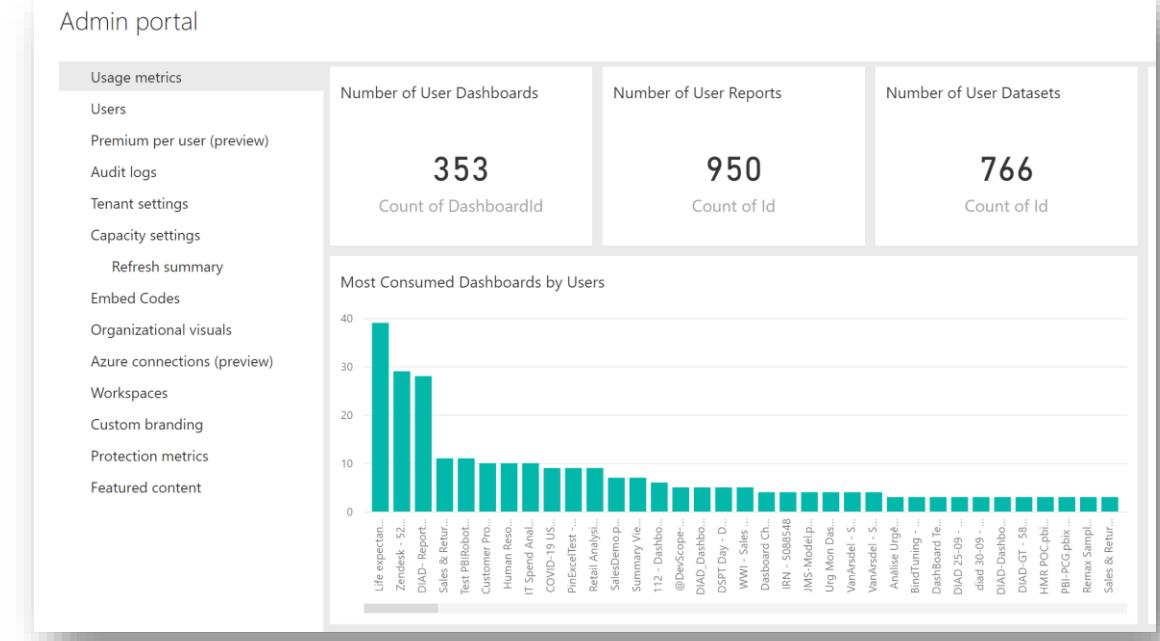
Power BI Report Usage

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



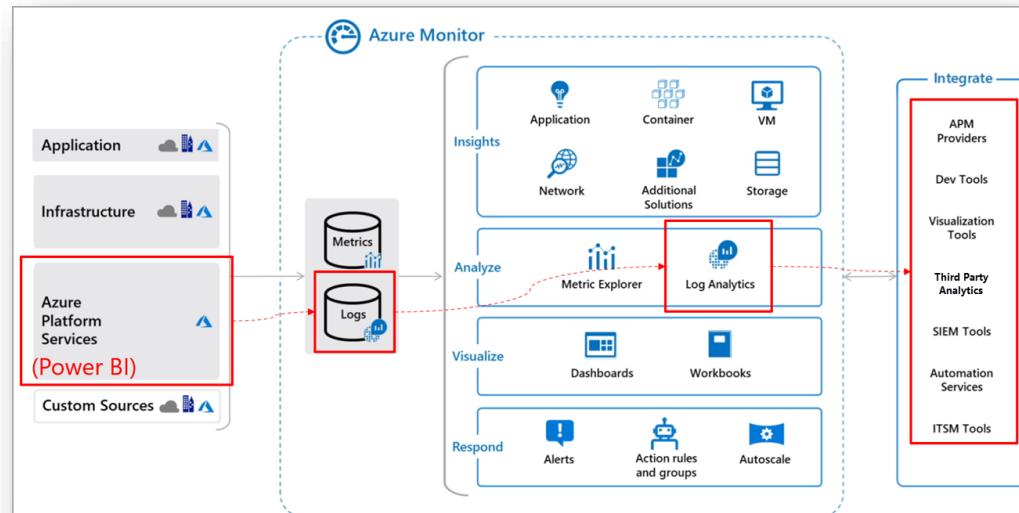
Usage Metrics in Admin Portal

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



Power BI & Log Analytics (BYOLA)

- Dataset level – Queries, Refresh Commands,...
- Very useful to debug & deep dive track performance issues
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



Power BI Analysis Services Engine Workspace summary

Settings

Storage

Log Analytics

Subscription: a1 [redacted] 7e

Resource group: rg-byola

Log Analytics workspace: admir [redacted]

Configured by Admin Admin on 2020-10-12T12:00:00Z

Disconnect from Azure

Logs

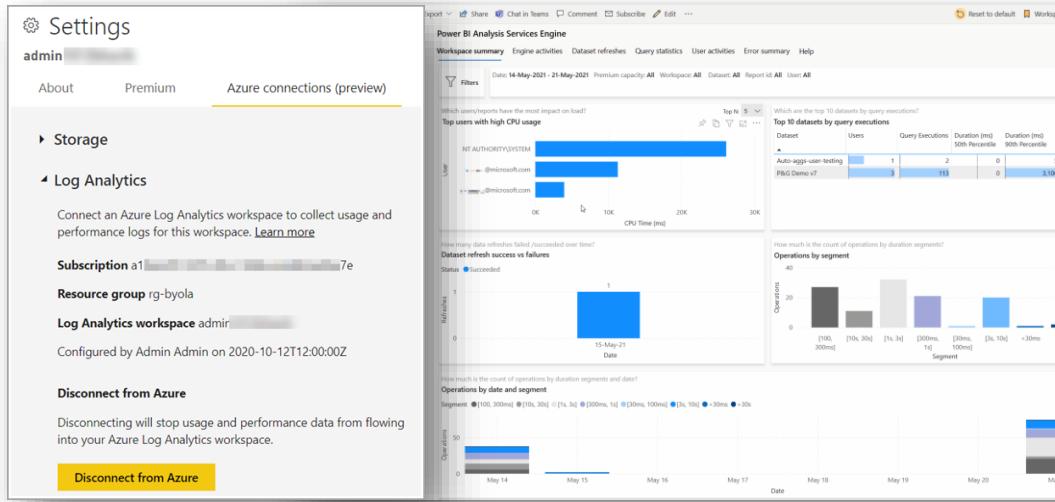
New Query 1*

```

1 PowerBIDatasetsWorkspace
2 | where TimeGenerated > ago(30d)
3 | where OperationName == 'QueryEnd'
4 | summarize avg(DurationMs) by format_datetime(TimeGenerated, 'yyyy-MM-dd'), ArtifactName
    
```

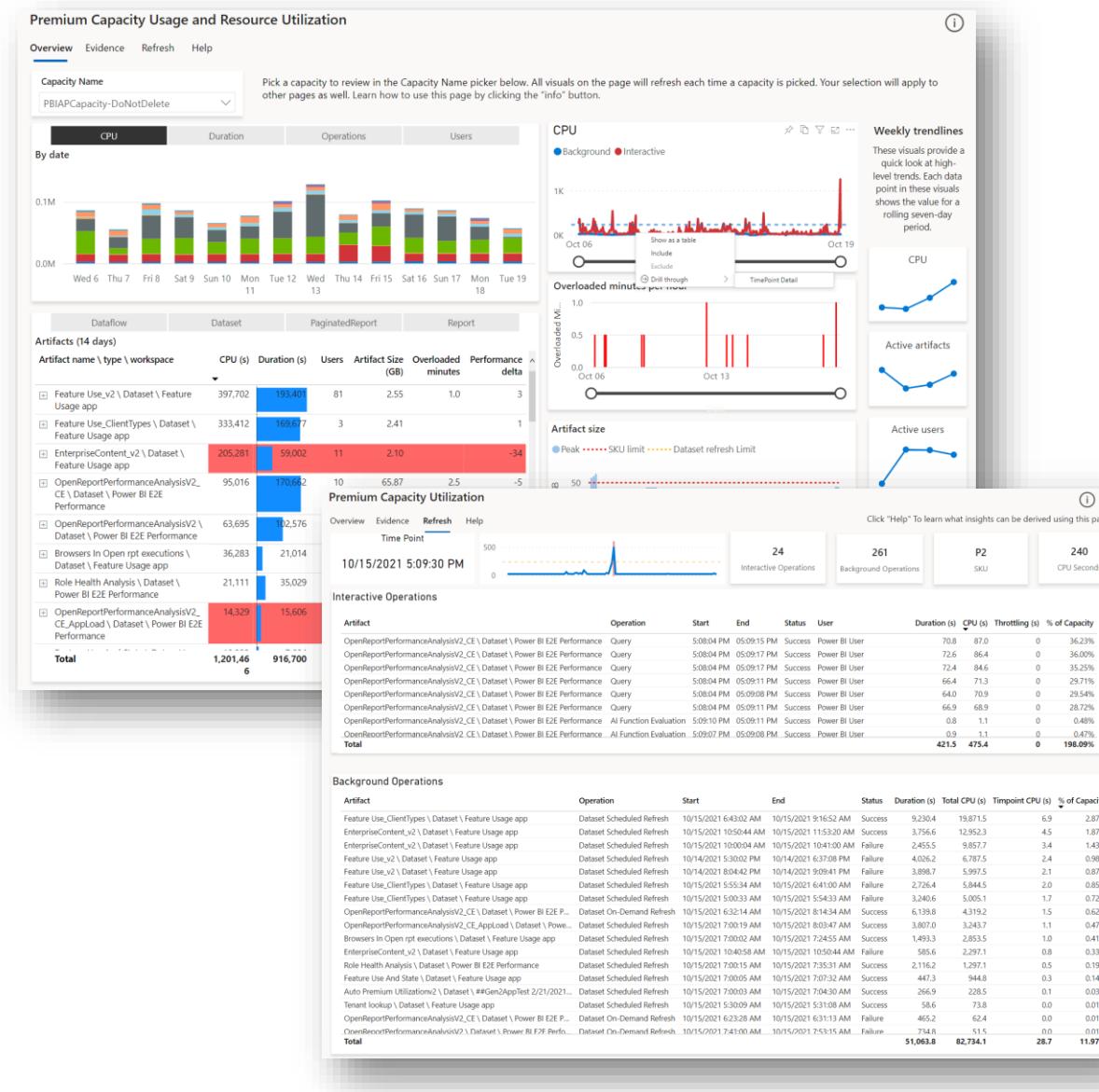
Completed

TimeGenerated	ArtifactName	avg_DurationMs
2021-11-16	Contoso	468.2
2021-11-16	Contoso-Partitioned	2,064.714



Premium Capacity Monitoring

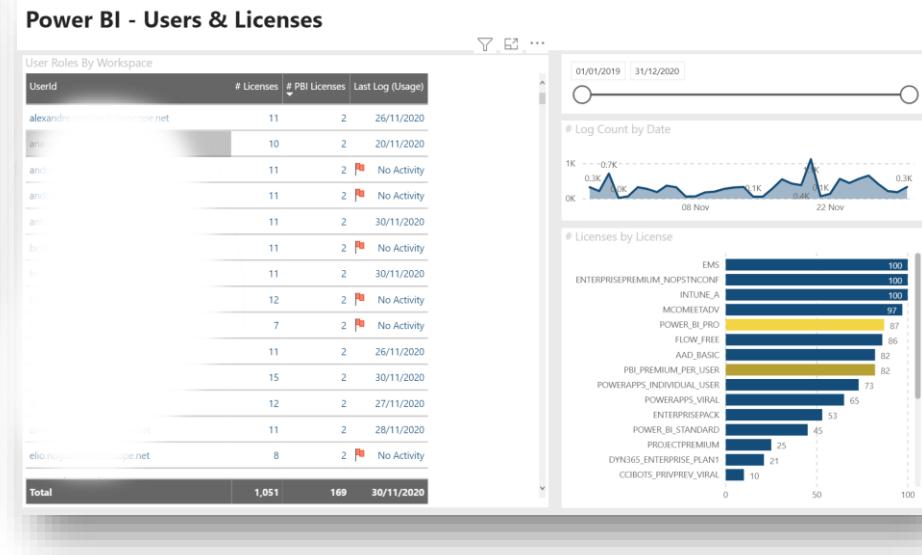
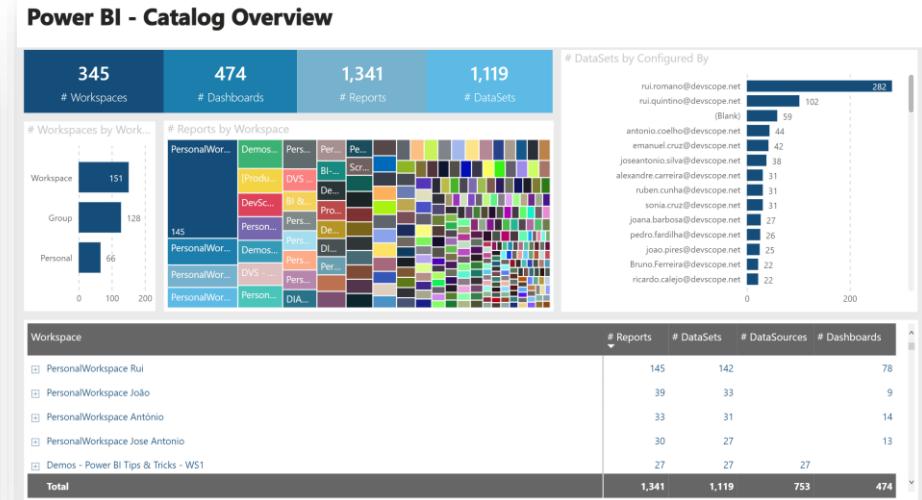
- Power BI Capacity Metrics App
- Documentation on the measures/fields through tooltips
- Near Real-Time
- Drill to detail and understand the CPU spends on 30s cycles
- 14 days of data



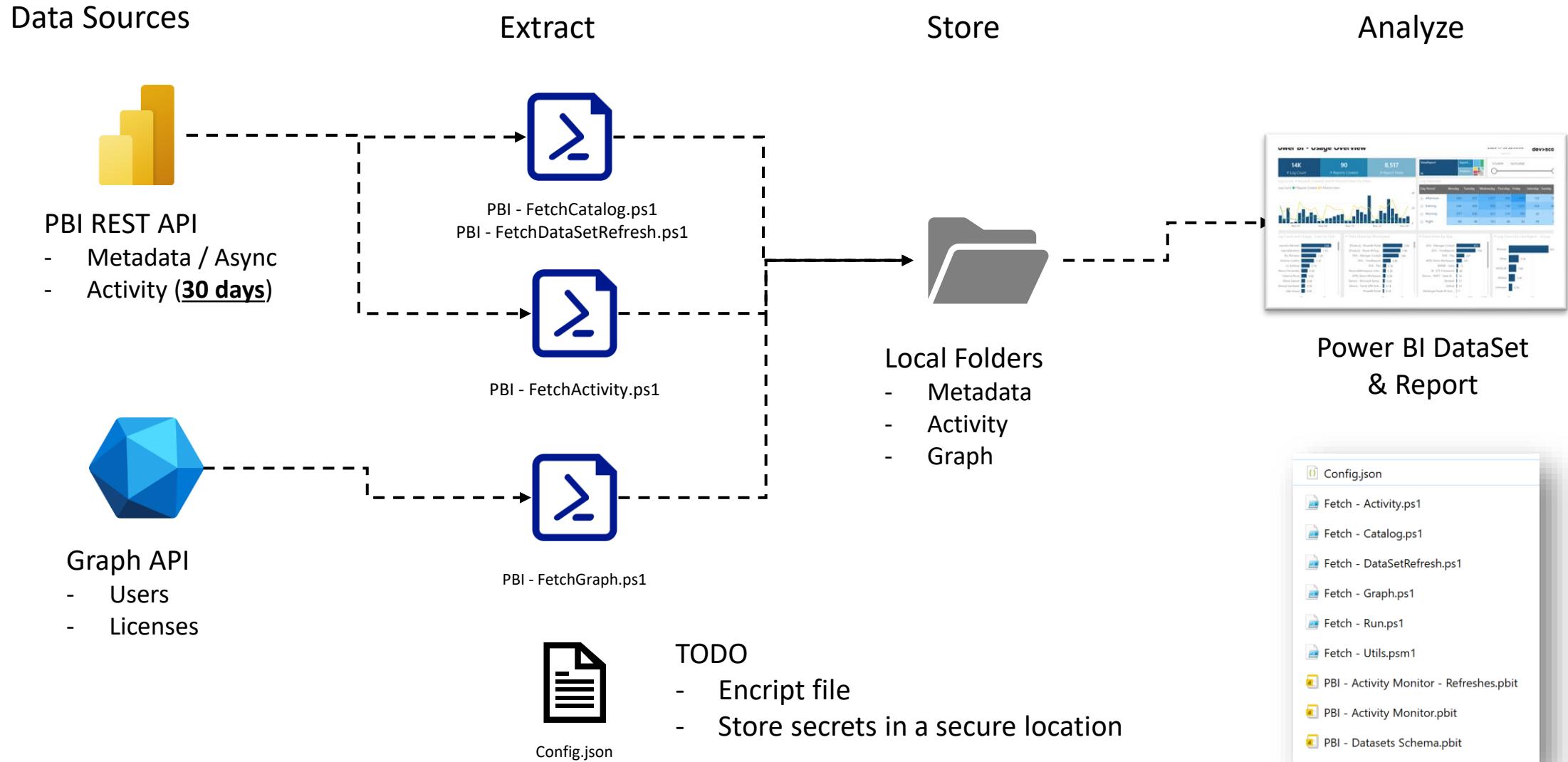
Custom Activity & Catalog Monitoring



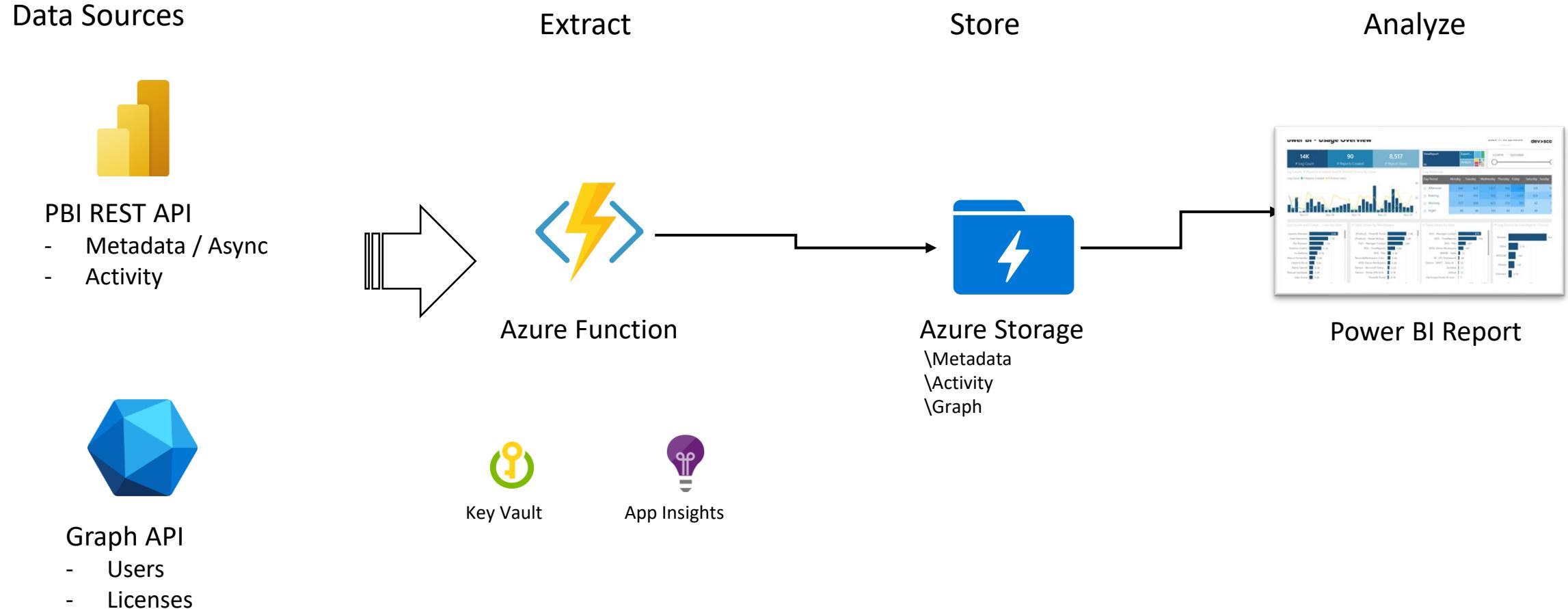
DEMO – Power BI Monitor



How? – Simplest



How? – Recommended (there are other possibilities)



API's Overview

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

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 ①

Yes No

2

Admin API settings

Allow service principals to use read-only Power BI admin APIs
Enabled for a subset of the organization

Web apps registered in Azure Active Directory (Azure AD) will use an assigned service principal to access read-only Power BI Admin APIs without a signed in user. To allow an app to use service principal authentication, its service principal must be included in an allowed security group. By including the service principal in the allowed security group, you're giving the service principal read-only access to all the information available through Power BI admin APIs (current and future). For example, Power BI user names and emails, dataset and report detailed metadata. [Learn more](#)

Enabled

Apply to:

The entire organization

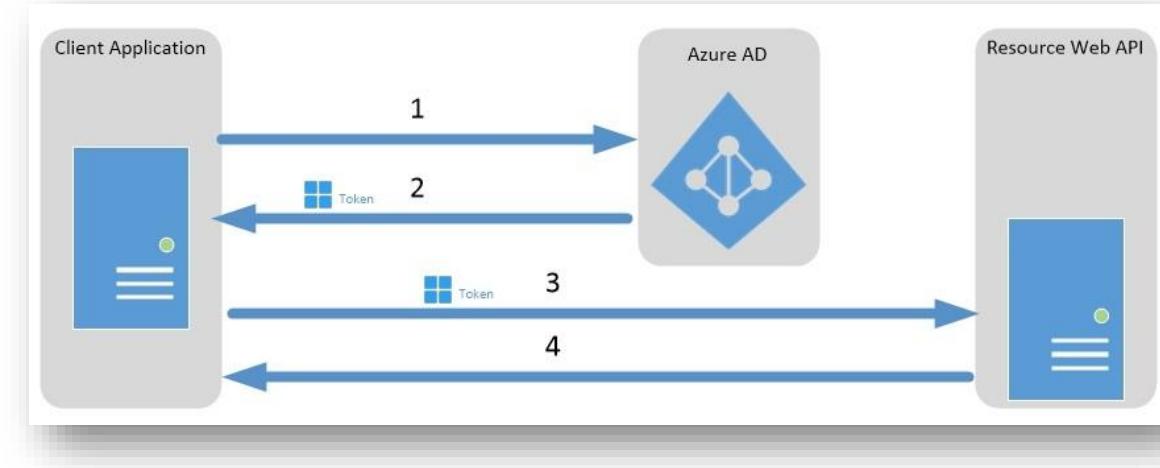
Specific security groups

PBISettings-ServicePrincipalsAdmin X Enter security groups

Apply Cancel

API Authentication

- OAuth 2.0
- Authentication Flows:



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

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

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



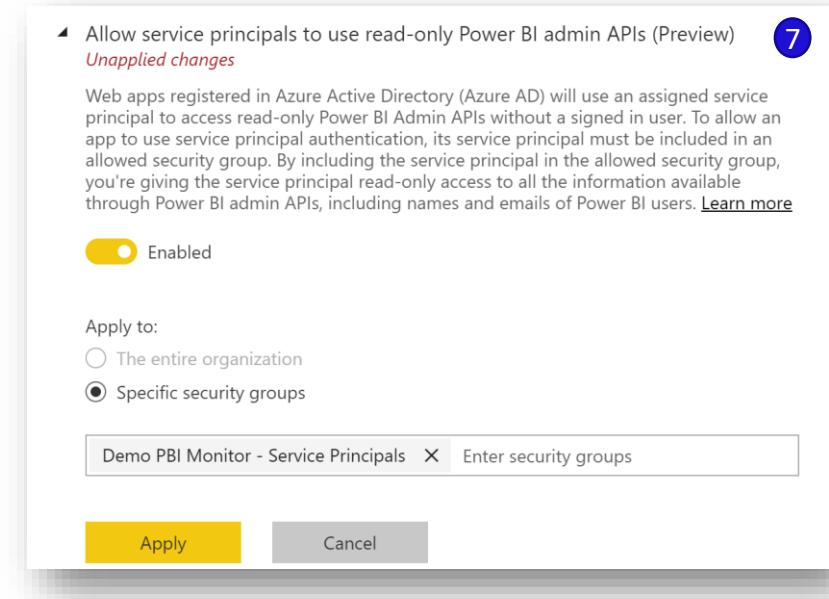
```
HTTP/1.1 200 OK
Content-Type: application/json
Cache-Control: no-store
Pragma: no-cache

{
  "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. Enable Power BI Tenant Settings (using the Security Group)
 - 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

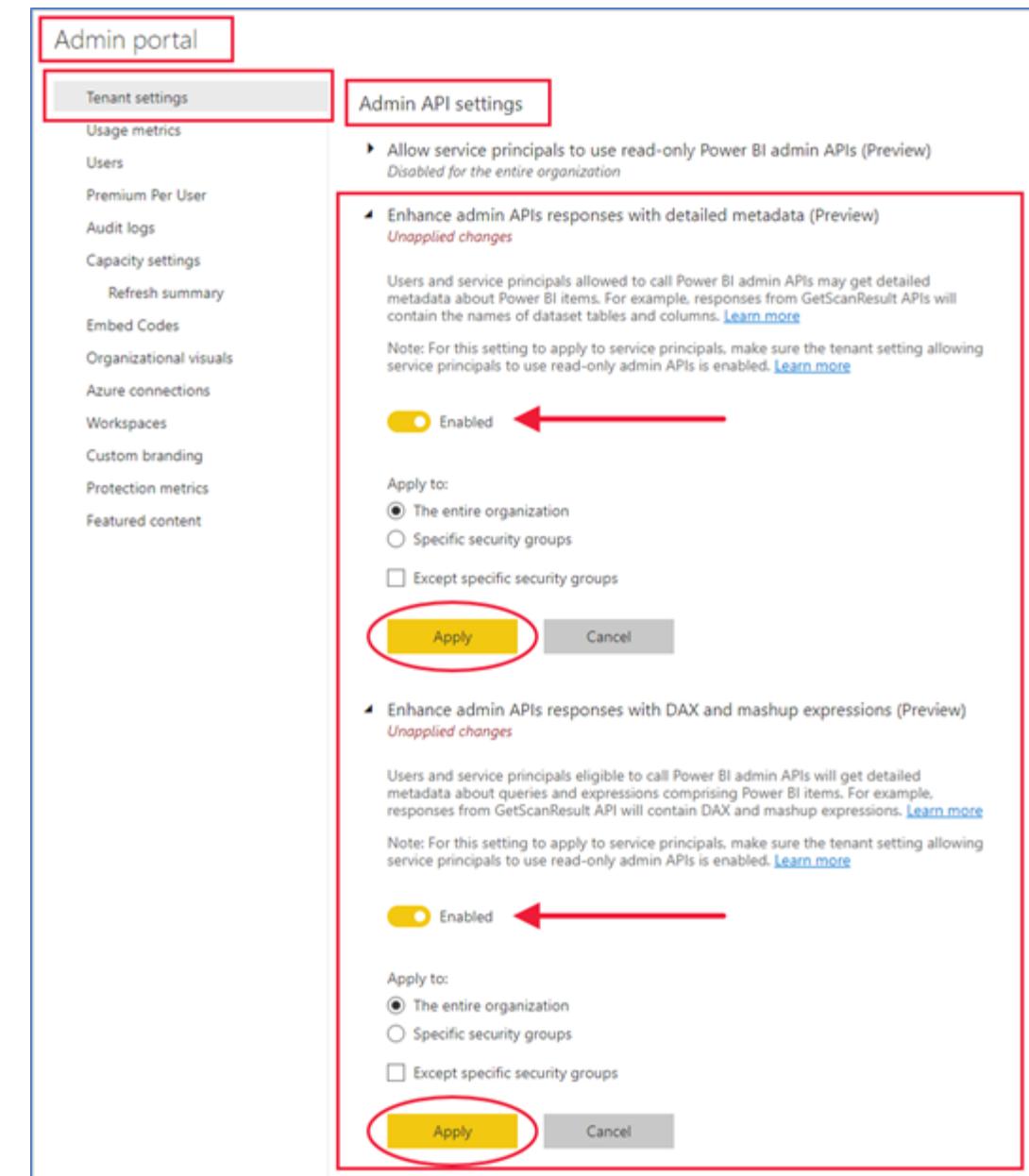


+ Add a permission ✓ Grant admin consent for Rui Romano AD (MVP Subscription) 8

API / Permissions name	Type	Description
<input checked="" type="checkbox"/> Microsoft Graph (3)		
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 - optional

- Add Service Principal to “Allow service principals to use read-only Power BI admin API’s”
- Enable “Enhance admin APIs responses with detailed metadata”
Turns on the caching flow and enhances API responses with low-level metadata (for example, name and description) for tables, columns, and measures.
- Enable “Enhance admin APIs responses with DAX and mashup expressions”
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

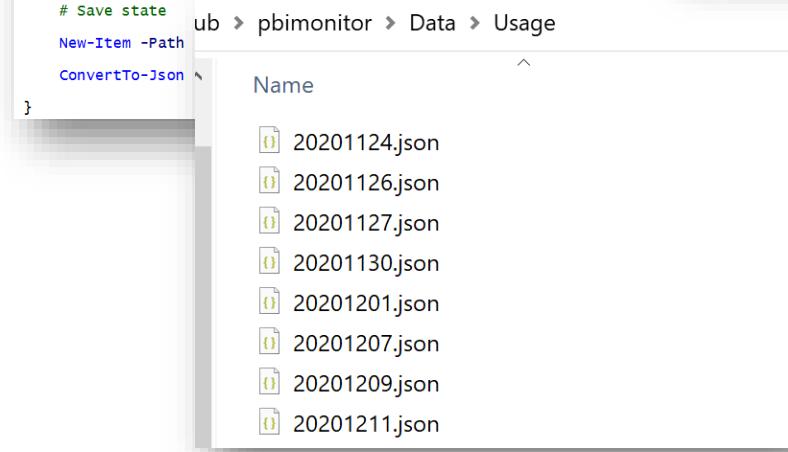
- IT/BI Admin Support
- Extract Data
- Data Store
- Power BI DataSet
- Power BI Report

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***

```
: > @Repos > Github > pbimonitor > {} Config.json
1  {
2      "OutputPath": ".\\Data",
3      "ServicePrincipal": {
4          "AppId": "[PUT YOUR APP ID HERE]",
5          "AppSecret": "[PUT YOUR APP SECRET HERE]",
6          "TenantId": "[PUT YOUR TENANT ID HERE]",
7          "Environment": "Public"
8      }
9  }
```

```
# Gets audit data for each day
while ($pivotDate -le [datetime]::UtcNow) {
    Write-Host "Getting audit data for: $($pivotDate.ToString("yyyyMMdd"))"
    $outputFilePath = ("$outputPath\{0:yyyyMMdd}.json" -f $pivotDate)
    $activityAPIUrl = "admin/activityevents?startDateTime='$(($pivotDate.ToString("s"))'&endDateTime='$(($pivotDate.AddDays(1)).ToString("s"))'"
    # Get-PowerBIAuditEvent was having memory issues on large tenants
    $result = Invoke-PowerBIRestMethod -Url $activityAPIUrl -method Get | ConvertFrom-Json
    $audits = @{$result.activityEventEntities}
    while($result.continuationToken -ne $null)
    {
        $result = Invoke-PowerBIRestMethod -Url $result.continuationUri -method Get | ConvertFrom-Json
        if ($result.activityEventEntities)
        {
            $audits += @{$result.activityEventEntities}
        }
    }
    if (!$audits -is [array]) {
        $audits = @{$audits}
    }
    if ($audits.Count -gt 0) {
        Write-Host "'$($audits.Count)' audits"
        New-Item -Path (Split-Path $outputFilePath -Parent) -ItemType Directory -Force -ErrorAction SilentlyContinue
        ConvertTo-Json @($audits) -Compress -Depth 5 | Out-File $outputFilePath -force
    }
    else {
        Write-Warning "No audit logs for date: $($pivotDate.ToString('yyyy-MM-dd'))"
    }
    $state.Activity.LastRun = $pivotDate.Date.ToString("o")
    $pivotDate = $pivotDate.AddDays(1)
    if ($config.StorageAccountConnStr -and (Test-Path $outputFilePath))
        Write-Host "Writing to Blob Storage"
        $storageRootPath = "$($config.StorageAccountContainerRootPath)/$($outputFilePath)"
        Add-FileToBlobStorage -storageAccountConnStr $config.StorageAccountConnStr -storageContainerName $config.StorageAccountContainerName $storageRootPath
    }
    # Save state
    New-Item -Path
    ConvertTo-Json }
```



```
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>
```

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")

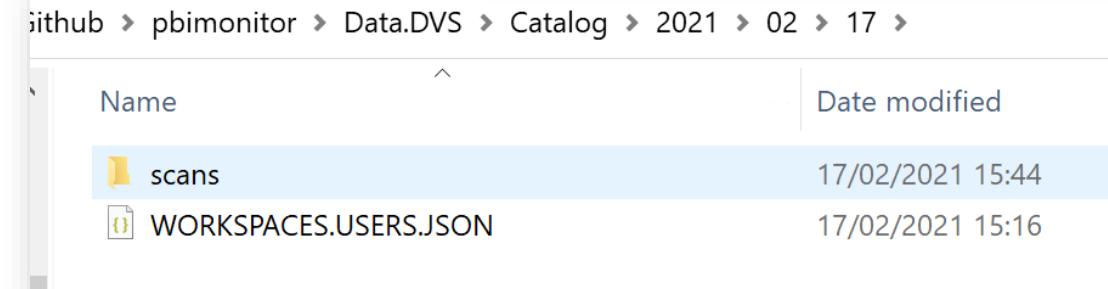
$skip = 0
$batchCount = 100
$workspacesScanRequests = @()

# call GetInfo to request workspace scan in batches of 100 (throttling after 500 calls per hour) https://github.com/microsoft/powerbi-admin-dotnet/blob/main/src/PowerBIAPI/PowerBIAPI/PowerBIAPIClient.cs#L104
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 -Method Post -Uri $modifiedRequestUrl -Body ($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/rest/api/power-bi/admin/get-workspace-scan-status
while(@( $workspacesScanRequests |? status -in @("Running", "Notstarted")))
{
    Start-Sleep -Seconds 1
}

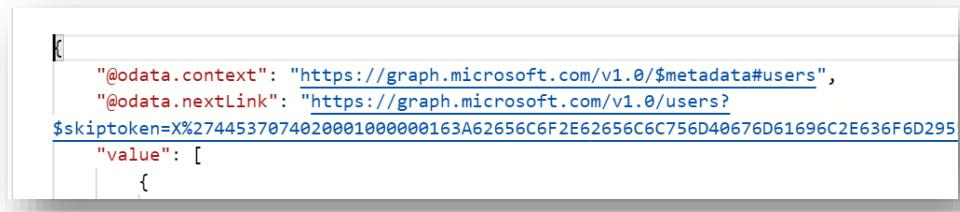
PS C:\@Repos\Github\pbimonitor> C:\@Repos\Github\pbimonitor>
Fetching 5000 /admin/workspaces
Getting workspaces to scan
Since: 2021-02-18T00:00:00.0000000Z
Modified workspaces: 5
Requesting workspace scan: 100 / 5
Waiting for scan results...
Scan 'bbf36328-5210-4006-bd45-830cf22e2793' : 'succeeded'
Scan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

```



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

```
# Get Scan results (500 requests per hour) - https://docs.microsoft.com/en-us/rest/api/power-bi/admin/get-activity-events
foreach ($workspaceScanRequest in $workspacesScanRequests)
{
    Wait-On429Error -tentatives 1 -sleepSeconds $throttleErrorSleepSeconds -script {
        $scanResult = Invoke-PowerBIRestMethod -Url "admin/workspaces/scanResult/$($workspaceScanRequest.id)" -Method Get
        Write-Host "Scan Result'$($scanStatus.id)' : '$($scanResult.workspaces.Count)'"
        $outputFilePath = "$scansOutputPath\$($workspaceScanRequest.id).json"
        $scanResult | Add-Member -MemberType NoteProperty -Name "scanCreatedDateTime" -Value $(Get-Date).ToString("yyyy-MM-ddTHH:mm:ssZ")
        ConvertTo-Json $scanResult -Depth 10 -Compress | Out-File $outputFilePath -force
    }
}
```



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 PowerQuery editor open. On the left, the 'Queries [41]' pane is visible, listing various catalog and usage queries. A red arrow points from the 'FilesProxy' query in the main editor area to the 'FilesProxy' entry in the 'Queries [41]' pane. The main editor area displays the 'FilesProxy' query code:

```
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
```

Below the code, a diagram illustrates the 'FilesProxy' query structure. It shows a central 'Folder Proxy' node connected to 'Local Folder', 'Sharepoint List', and 'M Query 1' (with an 'OR' connector) and 'M Query N' (with another 'OR' connector). A note 'Slow Query!' is placed near the Sharepoint List connection.

On the right, a preview table shows 11 rows of data with columns: Content, Filename, Date, and FileType. The data consists of binary files scanned on 10/02/2021.

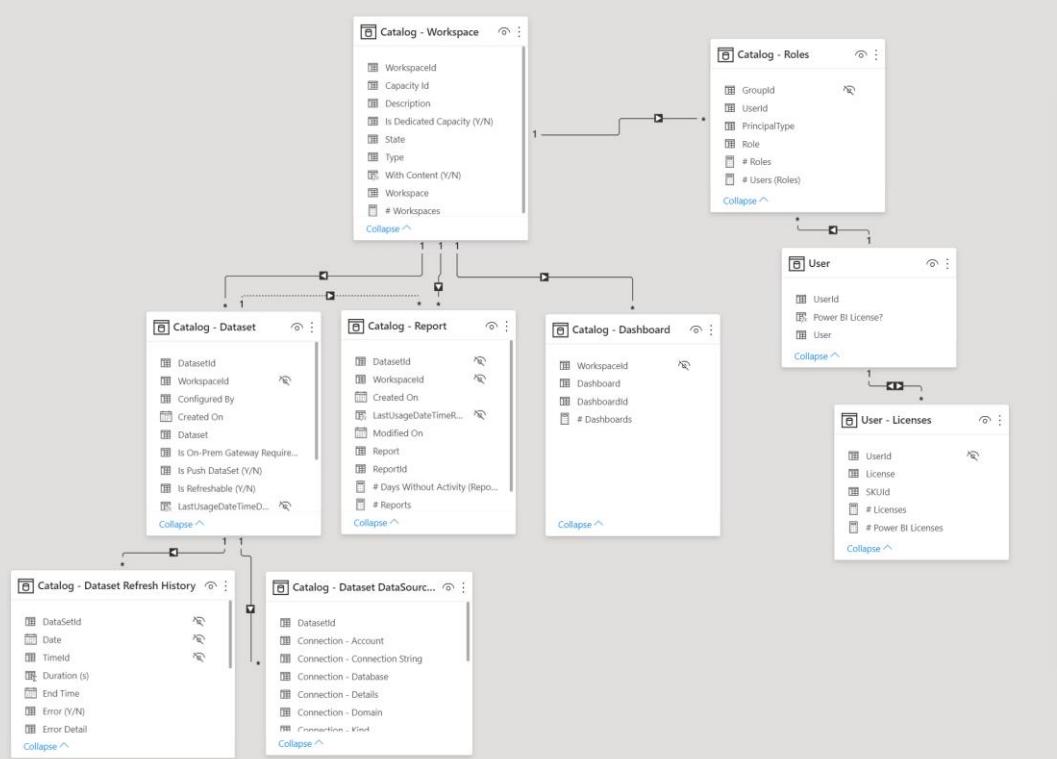
Content	Filename	Date	FileType	
1	Binary	0224742b-abc1-4ddb-8e35-7cda46b31b14.json	10/02/2021	scan
2	Binary	03b400bb-0473-4e16-a5d5-62052443ca05.json	10/02/2021	scan
3	Binary	05f460dd-3944-4fb4-a607-db5c6b8a6a1d.json	10/02/2021	scan
4	Binary	09c9baf6-6df7-4bf5-b7a3-77d277527c49.json	10/02/2021	scan
5	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 in the Power BI interface. It displays a hierarchical tree structure of queries:

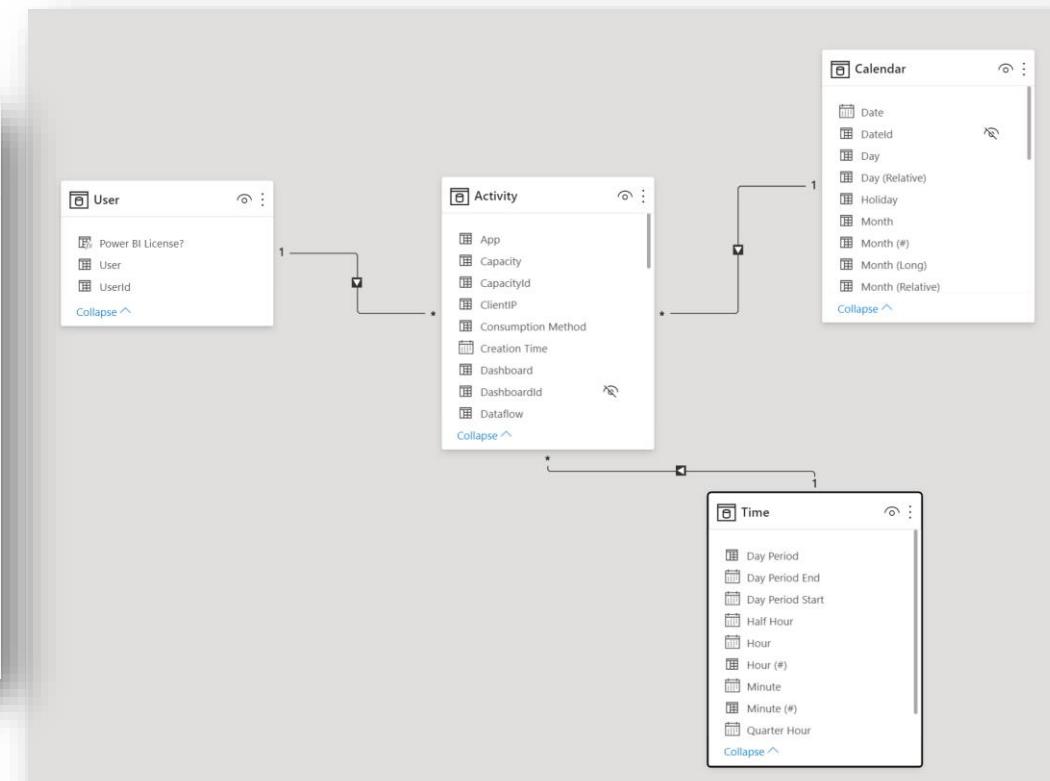
- Global Parameters [1]
 - TimezoneOffset (0)
- Calendar [5]
- Catalog [13]
 - Catalog-Files
 - Catalog-Files-Last
 - Catalog-Scans
 - Catalog - Workspaces RAW
 - Catalog - Dataset RAW
 - Catalog - DataSource
 - Catalog - DatasetDataSource
 - Catalog - Report
 - Catalog - Dashboard
 - Catalog - Dataset
 - 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...
 - PBICatalogDataFolder
 - PBIACTivityDataFolder
 - GraphDataFolder
 - Catalog-Files from Folder
 - Activity-Files from Folder
 - Graph-Files from Folder
 - Other Queries [2]
 - FilesProxy
 - System

Power BI – DataSet, 2 Models in 1

Catalog



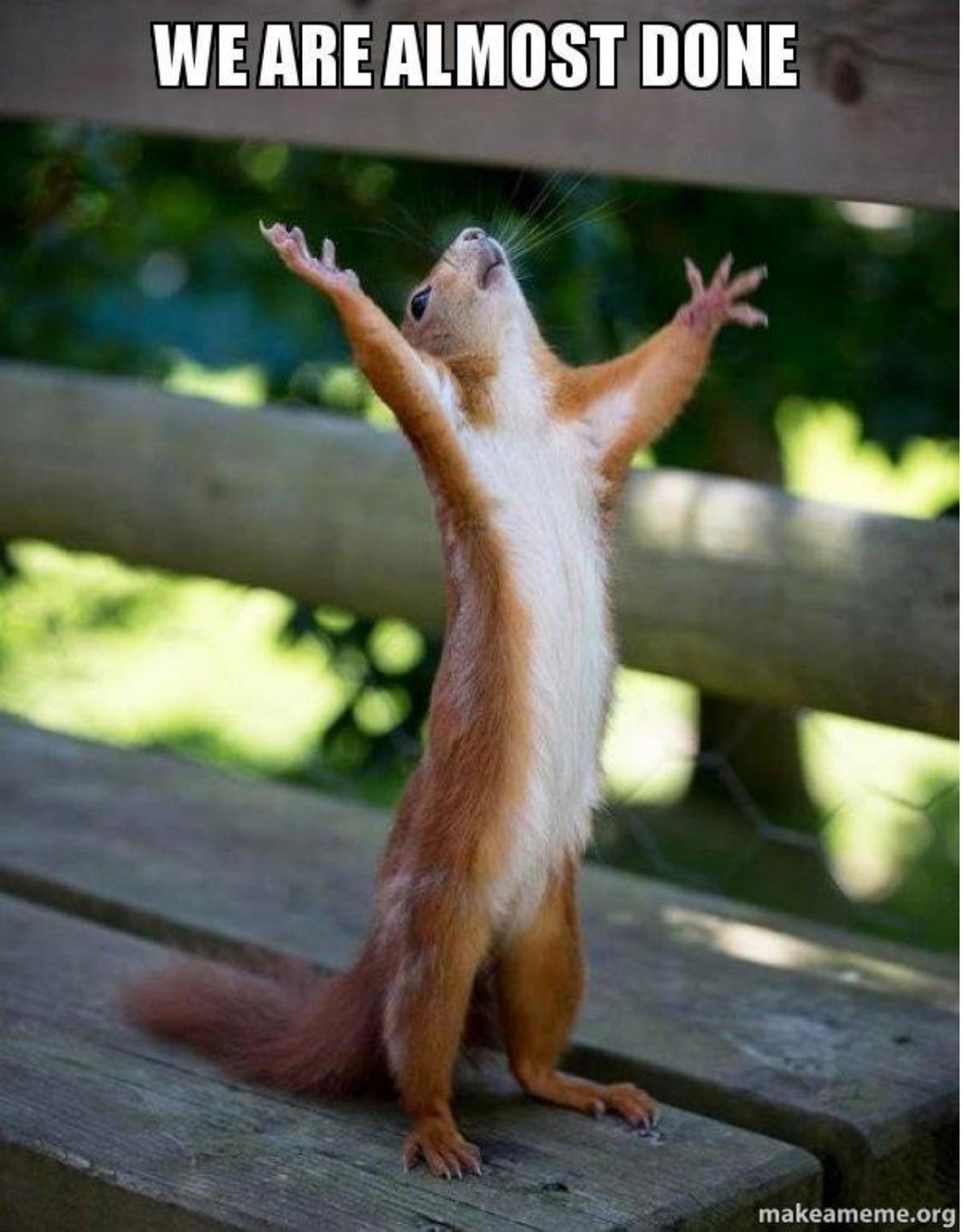
Activity



Power BI – DataSet Main Measures

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

WE ARE ALMOST DONE

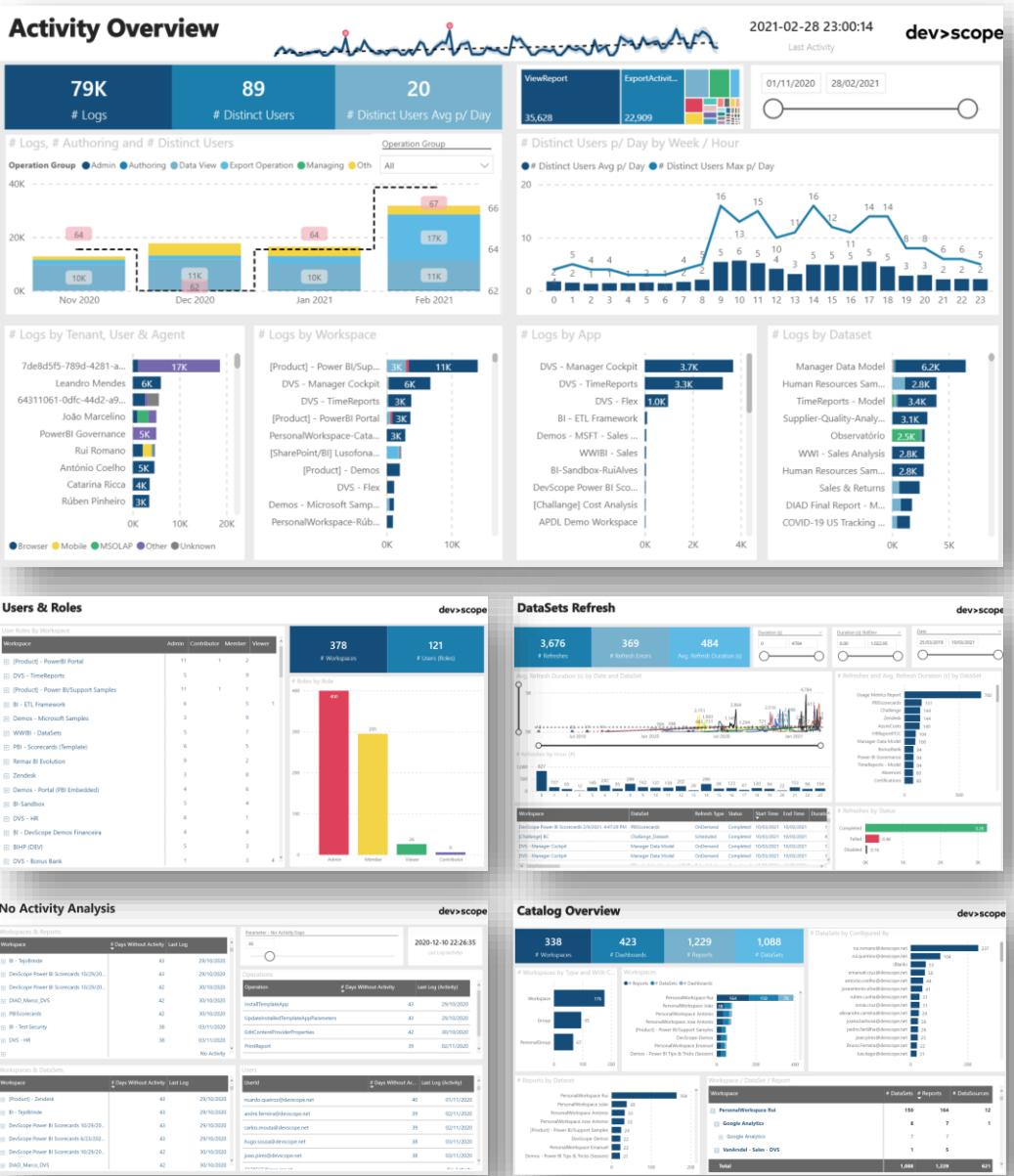


milestones

- ✓ 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





■ milestones

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

Thanks!

 @RuiRomano

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

 <https://ruiromanoblog.wordpress.com>