



# MICROSOFT PURVIEW ADVANCED RICH REPORTS FOR POWER BI AND SENTINEL

BASED ON OFFICE 365 MANAGEMENT API, MICROSOFT  
GRAPH API, AIP SERVICE API AND OFFICE 365 EXCHANGE  
ONLINE API

(MARCH 2023)



<https://aka.ms/MPARR-GitHub>

Sebastián Zamorano A.  
ISD Senior Consultant



# GRATITUDE

*My special gratitude to all my colleagues that helped and supported me on this new release. Special thanks to Grzegorz Berdzik, our black belt in this script; Dominik Kot who share some ideas and present me to Grzegorz; Stephan Carsten that who put some order in this script. And others that contact me constantly to ask for this, they make a lot of pression and now is here.*

*Additional thanks to Florian Boigner who is guide me on the Power BI path, as a good instructor.*

*We cannot forget to Waliid Elmorsy who is the owner of the original script used for all of this.*

*And several other people that is believe on me and this crazy idea.*

*Really thanks to all  
Sebastián Andrés Zamorano Andrade*



# AGENDA

WHAT WILL WE HAVE FOUND HERE?

## AGENDA

What will we have found here?

- Samples OF Queries and reports that can be created
- General concepts
- Prerequisites
- Baseline configuration
- Power BI configuration
- Advanced Settings



# DASHBOARDS

## SOME SAMPLE REPORTS

# DASHBOARD - OVERVIEW

## Dashboard for Unified Labeling behavior

### Unified Labeling

Overview

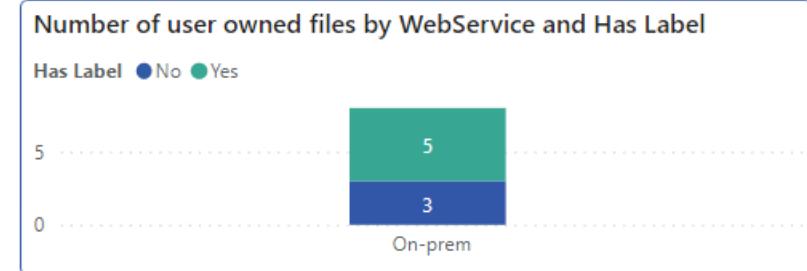
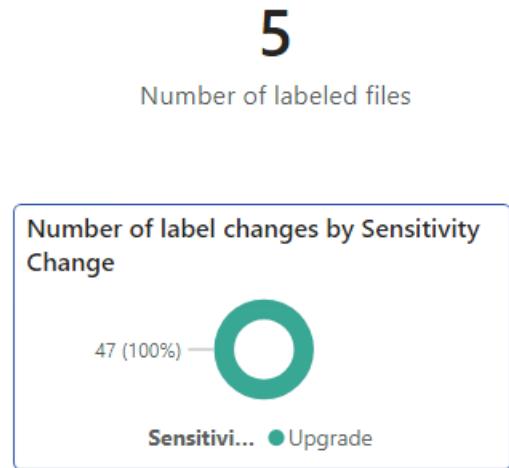
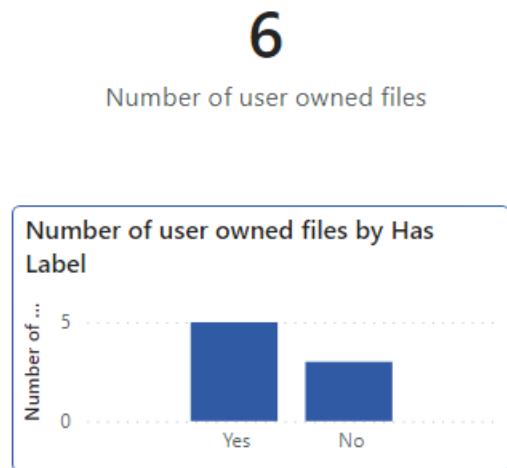
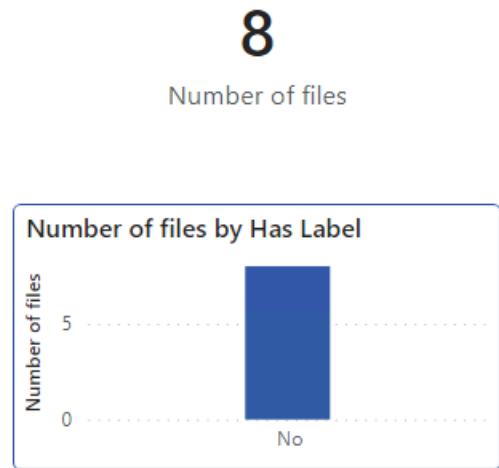
Filtering

Department

HR

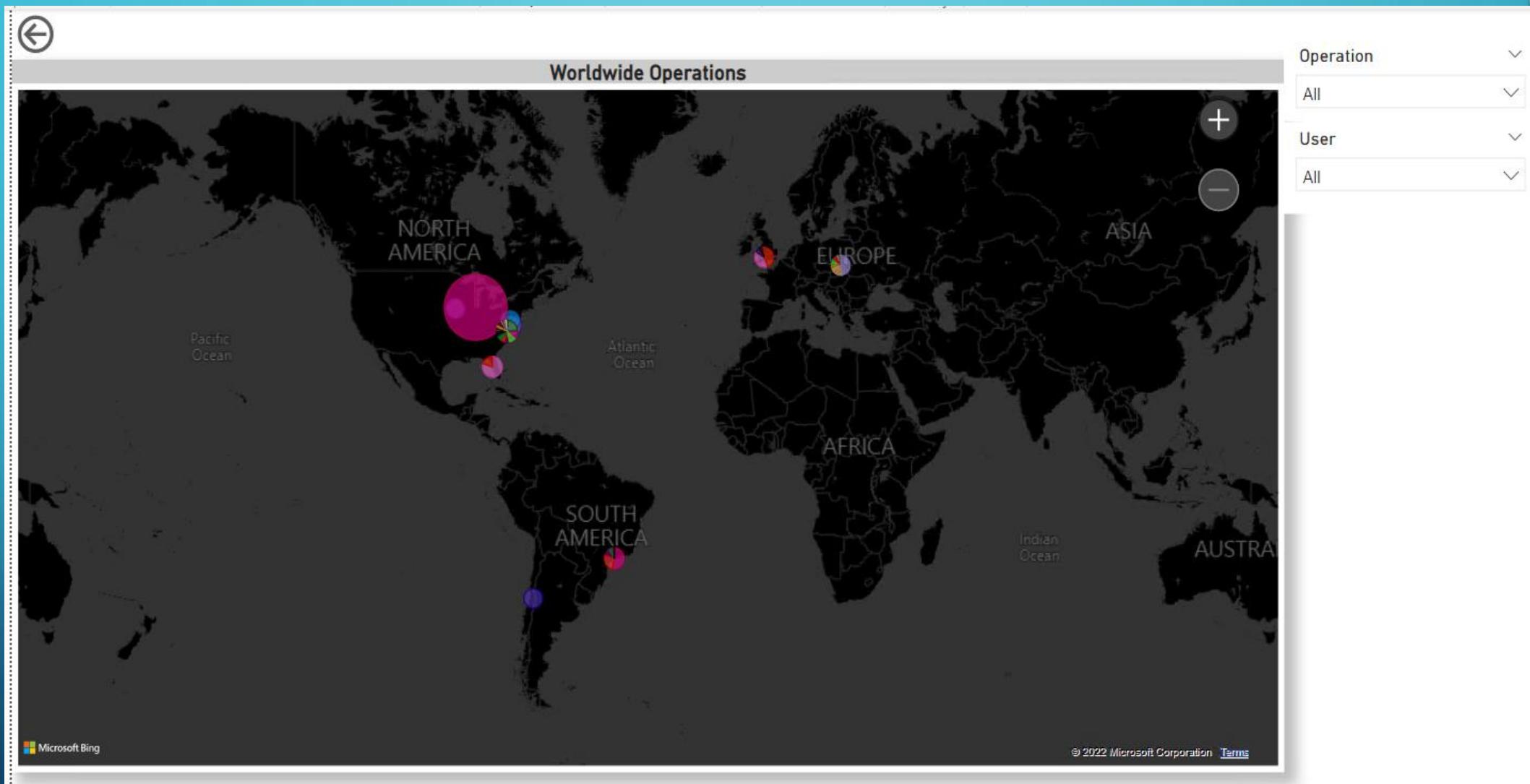
Country\_s

Poland



# DASHBOARD – GEO LOCATION

## Dashboard for audit activities



## Dashboard for DLP activities



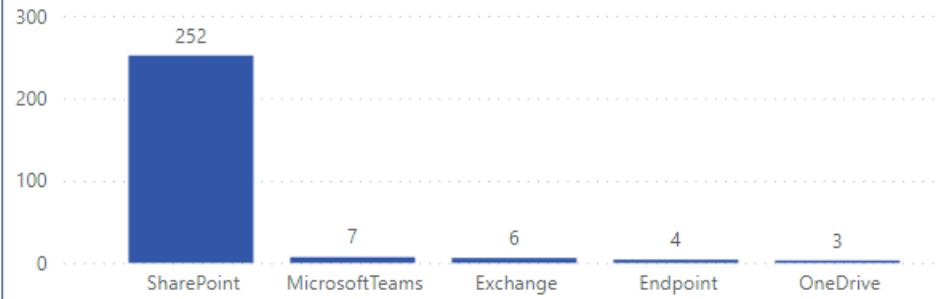
**272**

Number of DLP rule matches

### Number of DLP rule matches by Department\_s



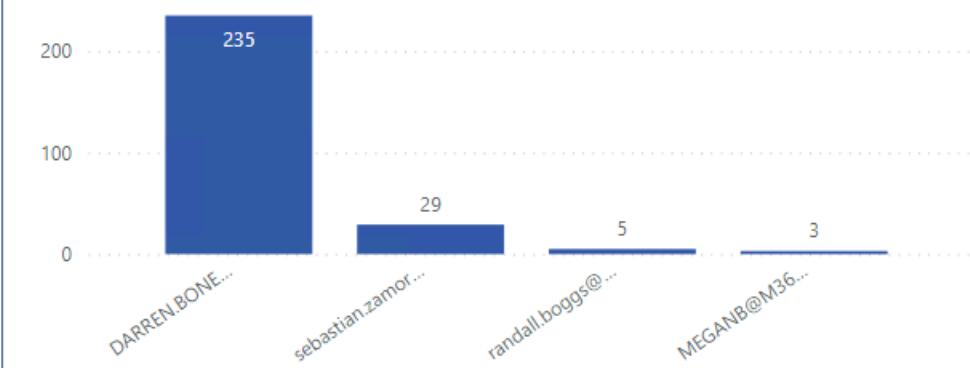
### DLP rule matches by workload



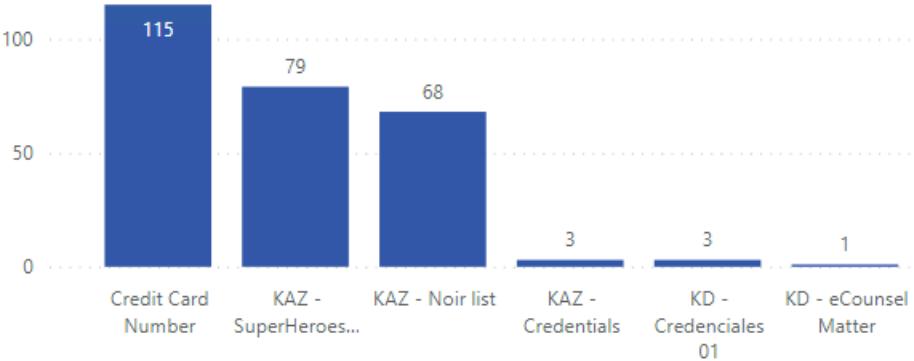
### DLP rule matches by date



### DLP rule matches by User



### DLP rule matches by SIT Detected

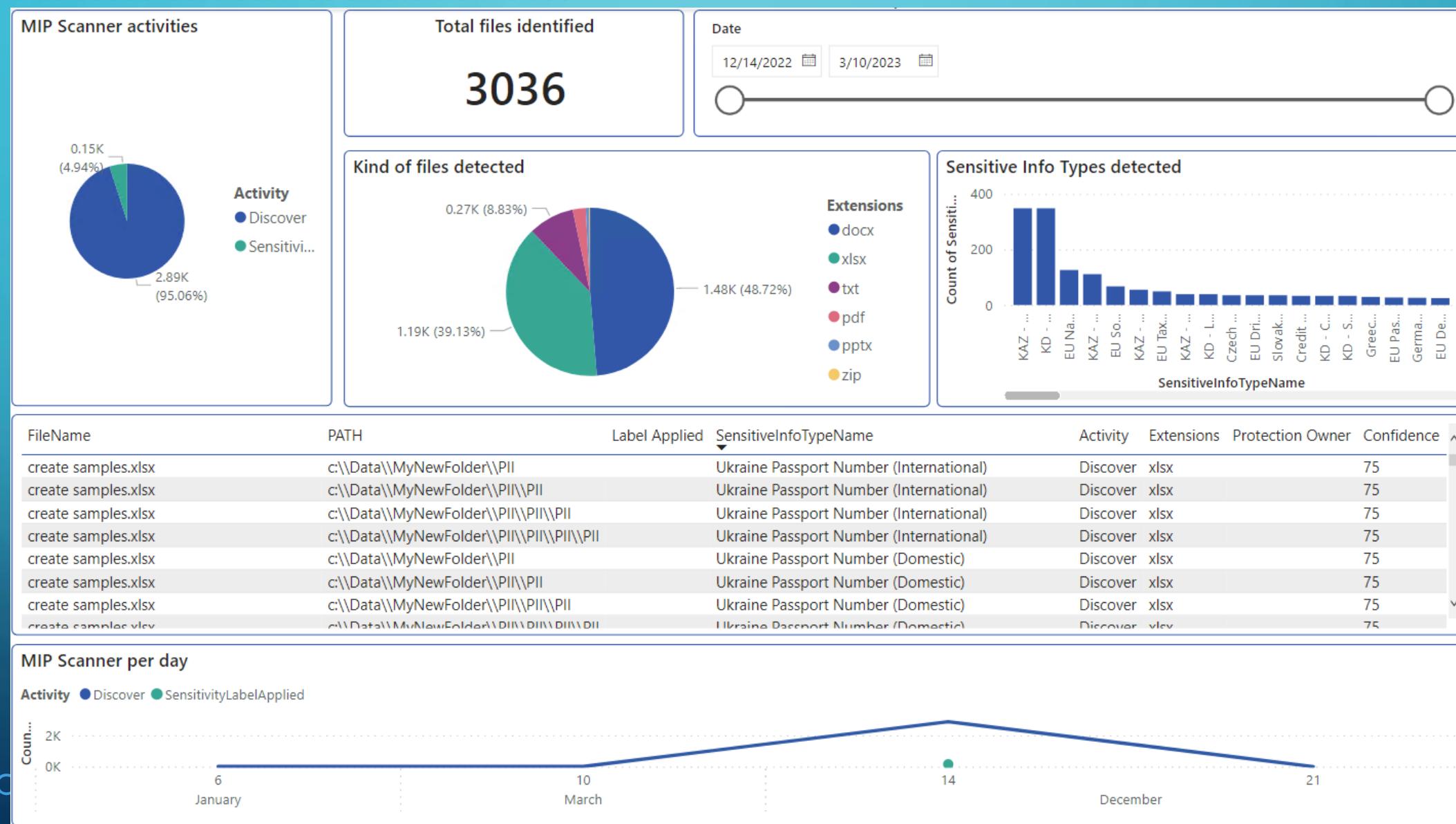


Right click on SIT Detected for drill-down to the details page



# DASHBOARD – AIP SCANNER

## Dashboard for MIP Scanner activities

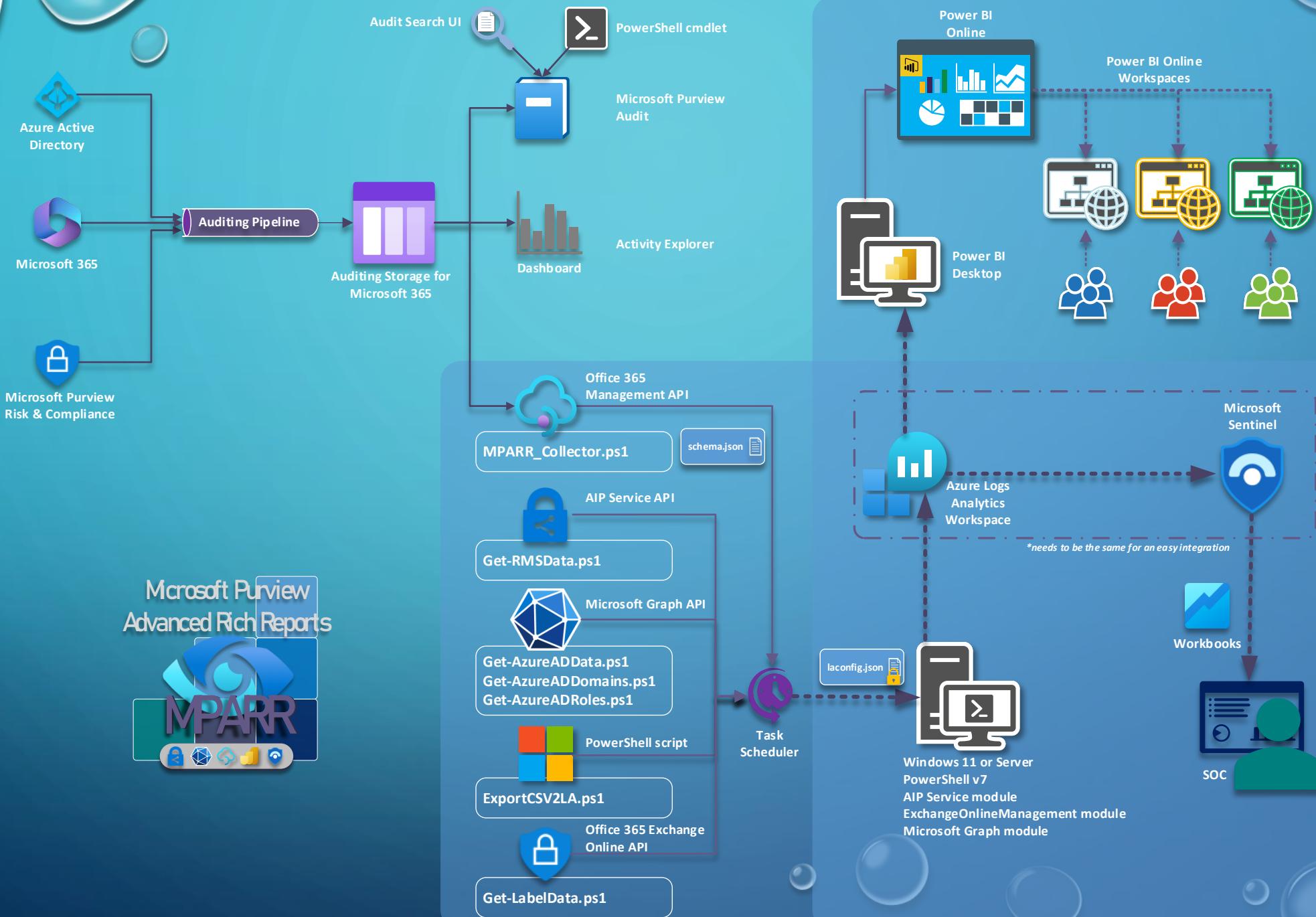


# SOME STUFFS TO TAKE ON MIND

Capabilities to implement previously reports shown depends to implement MPARR as your solution to collect all the information available through Office 365 Management API, and supporting APIs, if some of the scripts are not implemented, the reports can be not implemented in the right way.



# MPARR ARCHITECTURE



# CHECKLIST

## HAVE I ALL TO OBTAIN RESULTS?

**Certs** folder: To execute the **Get-xxxxx** scripts we need set some permissions based on certificates, inside of this folder you will find a **CreateCertificate.ps1** script, used for that function.

**Logs** Folder: Here the **timestamp.json** file is set (this file is used for **export\_logs.ps1** script) and additional any error is record here (The default folder is located on C:\APILogs)

**RMSLogs** Folder: Used by **Get-RMSData.ps1** script, this is used to process data collected from AIP Service API.

**Support** Folder: Contains the document used to create the table with service plan and friendly names, that can be downloaded from [here](#)

**Laconfig.json**: this file contains the keys and data require to execute the next scripts; the data related to keys can be encrypted.

**Schemas.json**: this file contains the names of the content blobs used by the Office 365 Management API, and the collector takes the information from these ones, if some information cannot be collected, the value needs to be change from True to False.

**MPARR\_Collector.ps1**: Script used to obtain the data from Office 365 management API and send to Logs Analytics, to use with parameters through Task Scheduler a new file needs to be created called **run\_me.ps1**

**ExportCSV2LA.ps1**: Script used to export the CSV file (located on Support folder) with service plan to Logs Analytics (required to execute on-demand)

**Get-AzureADData.ps1**: Script used to collect data from Azure AD using Microsoft Graph API, permissions are added to Azure AD App for unattended execution, can be modified to add or remove certain Azure AD attributes (Can be execute through Task Scheduler Monthly, depending on users' rotation)

**Get-AzureADDomains.ps1**: Script used to collect data from Azure AD using Microsoft Graph API, permissions are added to Azure AD App for unattended execution, this script permit to collect all the domains registered at Tenant.

**Get-AzureADRoles.ps1**: Script used to collect data from Azure AD using Microsoft Graph API, permissions are added to Azure AD App for unattended execution, this script permit to collect all the administrator roles assigned to user accounts.

**Get-LabelData.ps1**: Script used to obtain Display Name and ID for labels, require ExchangeOnlineManagement PowerShell Module, permissions are added to Azure AD App for unattended execution (can be executed on-demand)

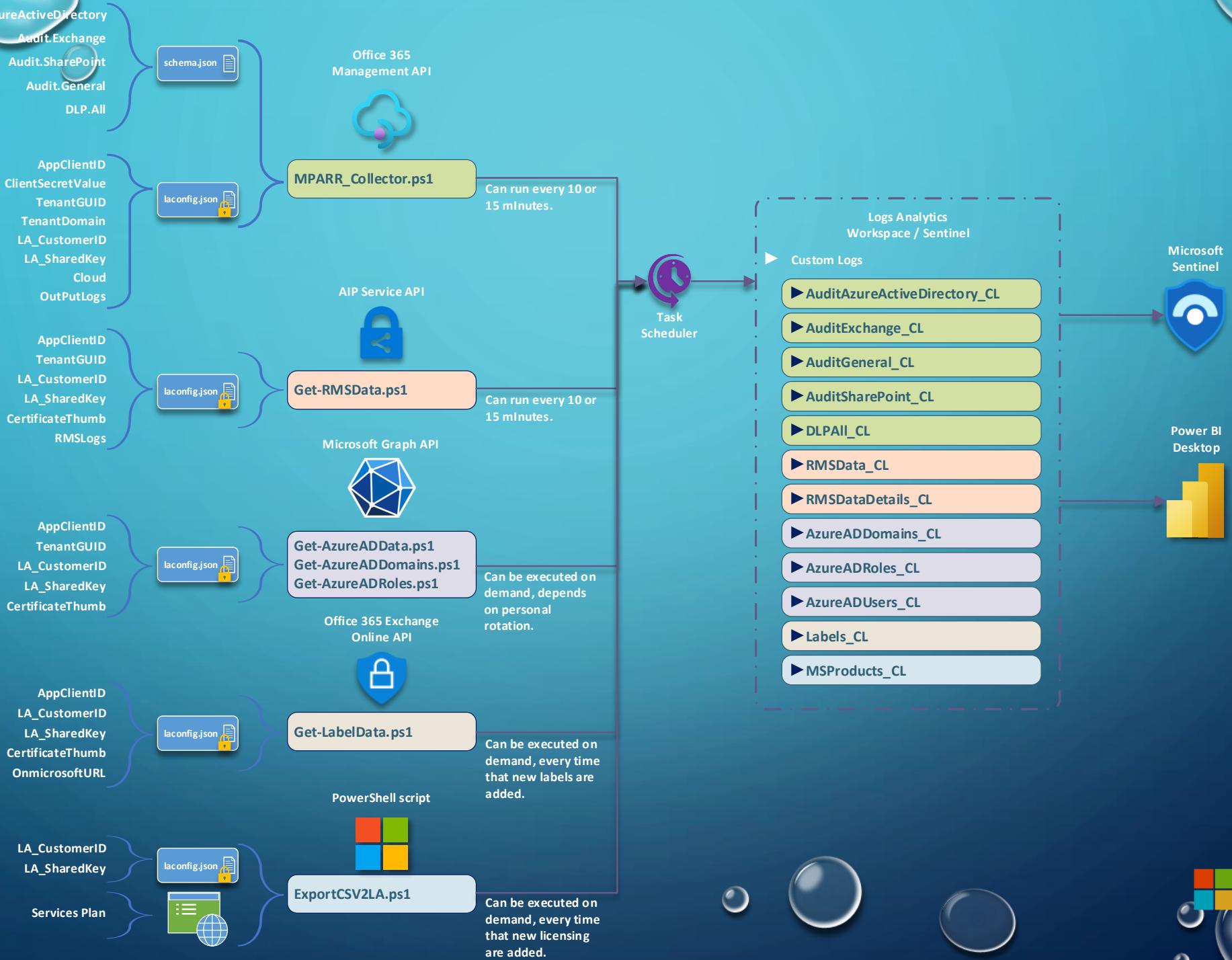
**Get-RMSData.ps1**: Script used to obtain information about external access to protected files, require AIPService PowerShell Module, permissions are added to Azure AD App for unattended execution , can be execute with the same timing as **export\_logs.ps1**

**Run\_me.ps1**: script used to call **export\_logs.ps1** with parameters to use with task scheduler

Name	Date modified	Type	Size
Support	19/12/2022 7:34 PM	File folder	
RMSLogs	12/03/2023 7:21 PM	File folder	
Logs	19/12/2022 6:53 PM	File folder	
Certs	19/12/2022 7:05 PM	File folder	
run_me.ps1	12/12/2022 5:56 PM	Windows PowerShell Sc...	1 KB
mparr_collector.ps1	10/03/2023 8:54 PM	Windows PowerShell Sc...	28 KB
Get-RMSData.ps1	26/01/2023 8:18 PM	Windows PowerShell Sc...	17 KB
Get-LabelData.ps1	24/01/2023 2:07 PM	Windows PowerShell Sc...	9 KB
Get-AzureADData.ps1	20/01/2023 2:16 PM	Windows PowerShell Sc...	8 KB
Get-AzureADDomains.ps1	03/03/2023 8:55 PM	Windows PowerShell Sc...	10 KB
ExportCSV2LA.ps1	16/11/2022 4:47 PM	Windows PowerShell Sc...	7 KB
schemas.json	08/02/2023 8:41 PM	JSON File	1 KB
laconfig.json	19/12/2022 6:54 PM	JSON File	1 KB



# DETAILED ARCHITECTURE



# DETAILED ARCHITECTURE

Logs Analytics  
Workspace / Sentinel

Custom Logs

- ▶ AuditAzureActiveDirectory\_CL
- ▶ AuditExchange\_CL
- ▶ AuditGeneral\_CL
- ▶ AuditSharePoint\_CL
- ▶ DLPAII\_CL

Used to collect all the data from Office 365 Management API, according to this documentation.

The principal information collected permit to create all the dashboards based on the Operation field. The information collected permit to identify activities from end-users, admin users, and services accounts

- ▶ RMSData\_CL
- ▶ RMSDataDetails\_CL

Used to collect data from AIP Service API, this permit to collect information related to external access to protected documents, and information related to denied access to protected documents from internal or external access.

This information permit create dashboards for Unified Labeling to identify access denied over protected documents for internal or external users, and detailed information for access to protected documents from external users.

- ▶ AzureADDomains\_CL
- ▶ AzureADRoles\_CL
- ▶ AzureADUsers\_CL

Used to collect data from Microsoft Graph API, this information permit to identify Users, Admin roles, Domains registered on the Tenant, and Azure AD attributes.

This information permit to identify activities from specific roles, identify the domains registered on the Tenant, this permit to identify external access, and add filters based on Azure AD Attributes, like as, Department, Location, Countries, and all the attributes available and populated.

- ▶ Labels\_CL

Used to collect data from Office 365 Exchange Online API, this information permit identify Labels Names and Priority for these labels.

Used to create reports for Unified Labeling, collecting Display Name of the labels and Parent Display Name, this last is for Sub-Labels, this permit to show the names of the Labels on Unified Labeling reports, the display name is not available under Office 365 Management API.

- ▶ MSProducts\_CL

This table collect the information related to friendly names for licensing.

Under Azure AD the licensing is showing with an ID or a short name that is not friendly to identify the right one, for that reason this data permit to see the friendly names on reports.

## Prerequisites to implement

- Logs Analytics workspace (Azure subscription)
- Workstation or Server with Internet access
- PowerShell v7 installed on the previous machine
- ExchangeOnlineManagement version 3.0.0 module for PowerShell
- Microsoft.Graph PowerShell module installed
- AIPService Module for PowerShell
- At least Application administrator to create an App under Azure AD
- Compliance administrator role to obtain Sensitivity Labels list
- Power BI desktop
- Open URLs:
  - [https://\\*.aadrm.com](https://*.aadrm.com)
  - [https://\\*.protection.outlook.com](https://*.protection.outlook.com)
  - [https://\\*.azure.com](https://*.azure.com)
  - <Https://manage.office.com> (can be different for GCC, GCCH or DoD tenants)
  - <https://graph.microsoft.com>
  - <https://login.microsoftonline.com>

## GO DIRECTLY FROM HERE TO THE SPECIFIC TOPICS

- Azure AD App configuration →
- Workspace Logs Analytics configuration →
- Configure the Script and set the Task Scheduler Task →
- Select your Tenant type →
- Generate a Certificate to Execute some PowerShell cmdlets unattended →
- Create a Table in Logs Analytics with Display Names for Labels and IDs →
- Generate a Table in Logs Analytics to identify Licensing names →
- Access to Logs Analytics and Export to Power BI consume →
- Configure to use Microsoft Graph API through Azure AD App →
- Get users from Azure AD with attributes and Licensing →
- Connect and consume the data from Power BI →
- Create and consume Power BI templates →
- Geo Location on Power BI based on IP Addresses →

## Create App to recollect data (1/2)

- Open <https://portal.azure.com>
- Look for Azure Active Directory
- Go to “**App registrations**” menu
- Press **New registration**
  - Use any name here
  - Select Accounts in this organizational directory only
  - On Redirect URI, select Web and set <https://localhost>
  - Press Register
- On the new app
  - copy Application (client) ID and Directory (tenant) ID

### To give unattended access

- Then go to Certificates & secrets, press +New Client Secret and set a Description and Expiration time
- After press Add button on the blade a new Key will be created, copy that key under Value column
- Under certificates tab, on the same menu, select Upload certificate
- Select the certificate created on [this Step](#) and add a description.

## Create App to recollect data (2/2)

- On the new app (coming from previous page)
  - Go to API permissions and select +Add a permission:
    - To give access to Office 365 Management API**
      - At the blade open looking for Office 365 Management APIs
      - After select this Application, click on Application permissions
      - Select the 3 options available, check all of them, and then press Add permissions button at the final of that blade
    - To give access to Microsoft Graph**
      - At the blade open looking for Microsoft Graph
      - After select this Application, click on Application permissions
      - Select AuditLog.Read.All, Directory.Read.All, Group.Read.All, Organization.Read.All, User.Read.All
    - To give access to Compliance PowerShell console (Additional permission is required [here](#))**
      - At the blade open looking, at top select the tab “APIs my organization uses”
      - Search for “Office 365 Exchange online”
      - After select this Application, click on Application permissions
      - Under Exchange submenu select “Exchange.ManageAsApp”
    - To give access to AIP Service API (Azure Rights Management Services)**
      - At the blade open looking for Azure Rights Management Services
      - After select this Application, click on Application permissions
      - Select Application.Read.All
    - Press “Add permissions” in all the cases



# AZURE AD

## Steps

The screenshot shows the 'Welcome to Azure Active Directory' page. At the top, there's a search bar with 'Azure acti' and a navigation bar with tabs: All, Services (92), Resources, Resource Groups, Marketplace, and Documentation. The 'All' tab is selected.

In the center, there's a section titled 'Services' with a list of services: Azure Active Directory (selected), Activity log, Security, Azure Arc, Azure Cosmos DB, Azure Databricks, Azure Database for MySQL servers, Automanage – Azure machine best practices, and a 'See all' link.

Below this, there are three cards:

- Start with an Azure free trial**: Get \$200 free credit toward Azure products and services, plus 12 months of popular free services. Buttons: Start, Learn more ↗
- Manage Azure Active Directory**: Manage access, set smart policies, and enhance security with Azure Active Directory. Buttons: View, Learn more ↗
- Access student benefits**: Get free software, Azure credit, or access Azure Dev Tools for Teaching after you verify your academic status. Buttons: Explore, Learn more ↗

Under 'Azure services', there are icons for Create a resource, Azure Active Directory, Azure Information..., Microsoft Purview..., Azure AD B2C, Azure AD Privileged..., Help + support, Microsoft Sentinel, Function App, and More services.

At the bottom, there's a 'Resources' section with tabs Recent (selected) and Favorite. It shows a table with columns Name, Type, and Last Viewed. A message says 'No resources have been viewed recently' with a 'View all resources' button.



## Steps

Home > **Kaz Demos | Overview** Azure Active Directory

Add Manage tenants What's new Preview features Got feedback?

Overview Preview features Diagnose and solve problems

Manage

- Users
- Groups
- External Identities
- Roles and administrators
- Administrative units
- Enterprise applications
- Devices
- App registrations**
- Identity Governance
- Application proxy
- Custom security attributes (Preview)
- Licenses
- Azure AD Connect
- Custom domain names

Overview Monitoring Properties Tutorials

Search your tenant

**Basic information**

Name	Kaz Demos	Users	31
Tenant ID	<a href="#">Copy</a>	Groups	59
Primary domain	kazdemos.org	Applications	7
License	Azure AD Premium P2	Devices	7

**Alerts**

**Upcoming TLS 1.0, 1.1 and 3DES deprecation**

Please enable support for TLS 1.2 on clients(applications/platform) to avoid any service impact.

[Learn more](#)

## Steps

Home > Kaz Demos | App registrations >

### Register an application

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

Office 365 Management API capture ✓

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

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

[Help me choose...](#)

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

Web https://localhost ✓

Register an app you're working on here. Integrate gallery apps and other apps from outside your organization by adding from [Enterprise applications](#).

By proceeding, you agree to the [Microsoft Platform Policies](#) ↗

[Register](#)



## Steps

Home > Kaz Demos | App registrations >

### Office 365 Management API capture

Search (Ctrl+ /) <> Delete Endpoints Preview features

Overview Quickstart Integration assistant

Got a second? We would love your feedback on Microsoft identity platform (previously Azure AD for developer). →

Manage

- Branding & properties
- Authentication
- Certificates & secrets
- Token configuration
- API permissions
- Expose an API
- App roles
- Owners
- Roles and administrators
- Manifest

Essentials

Display name	: <a href="#">Office 365 Management API capture</a>
Application (client) ID	: 827e1a00-3a00-459c-9104-2a00cf18b4a
Object ID	: 59817ef1-3d21-44a4-a0dd-27b4b09bf57c
Directory (tenant) ID	: ac1a...00-3a00-459c-9104-2a00cf18b4a

Supported account types : [My organization only](#)

Welcome to the new and improved App registrations. Looking to learn how it's changed from App registrations (Legacy)? [Learn more](#)

Starting June 30th, 2020 we will no longer add any new features to Azure Active Directory Authentication Library (ADAL) and Azure AD Graph. We will continue to support the existing features and fix critical security and stability issues.

Get Started Documentation

Build



# Steps

Home > Kaz Demos | App registrations > Office 365 Management API capture

## Office 365 Management API capture | Certificates & secrets

Search (Ctrl+ /) Got feedback?

Credentials enable confidential applications to identify themselves to the authentication service when receiving tokens at a web addressable location (using an HTTPS scheme). For a higher level of assurance, we recommend using a certificate (instead of a client secret) as a credential.

Application registration certificates, secrets and federated credentials can be found in the tabs below.

**Client secrets (0)** **Federated credentials (0)**

A secret string that the application uses to prove its identity when requesting a token. Also can be referred to as application password.

+ New client secret

Description	Expires	Value ⓘ	Secret ID
No client secrets have been created for this application.			

Add a client secret

Description	Collector
Expires	24 months

## Steps

Home > Kaz Demos | App registrations > Office 365 Management API capture

## Office 365 Management API capture | Certificates & secrets

Search (Ctrl+ /) Got feedback?

Got a second to give us some feedback? →

Credentials enable confidential applications to identify themselves to the authentication service when receiving tokens at a web addressable location (using an HTTPS scheme). For a higher level of assurance, we recommend using a certificate (instead of a client secret) as a credential.

Application registration certificates, secrets and federated credentials can be found in the tabs below.

Certificates (0) Client secrets (1) Federated credentials (0)

A secret string that the application uses to prove its identity when requesting a token. Also can be referred to as application password.

New client secret

Description	Expires	Value ⓘ	Copied	Secret ID
Collector	8/18/2024	IOs8	dkGTrkr ...	d58bb7c0-e119-46c0-8da6-247d493b4f14

Troubleshooting

New support request

## Steps

Home > Kaz Demos | App registrations > Microsoft Purview Reports

### Microsoft Purview Reports | Certificates & secrets

Search

Got feedback?

Overview

Quickstart

Integration assistant

#### Manage

Branding & properties

Authentication

Certificates & secrets

Token configuration

API permissions

Expose an API

App roles

Owners

Roles and administrators

Manifest

#### Support + Troubleshooting

Troubleshooting

New support request

Credentials enable confidential applications to identify themselves to the authentication service when receiving tokens at a web addressable location (using an HTTPS scheme). For a higher level of assurance, we recommend using a certificate (instead of a client secret) as a credential.

Application registration certificates, secrets and federated credentials can be found in the tabs below.

Certificates (1) Client secrets (3) Federated credentials (0)

Certificates can be used as secrets to prove the application's identity when requesting a token. Also can be referred to as public keys.

Upload certificate

Thumbprint	Description	Start date	Expires	Certificate ID
180 [REDACTED] 7917...	Microsoft Purview	10/11/2022	10/11/2023	dc535e74-a958-4eea-...



# Steps

The screenshot shows the Azure AD portal interface. On the left, there's a sidebar with navigation links like Home, App registrations, Office 365 Management API capture, Overview, Quickstart, Integration assistant, Manage, Branding & properties, Authentication, Certificates & secrets, Token configuration, API permissions (which is selected), Expose an API, App roles, Owners, Roles and administrators, and Manifest.

The main content area is titled "Office 365 Management API capture | API permissions". It displays a table of configured permissions:

API / Permissions name	Type	Description	Admin consent req...	Status
Microsoft Graph (1)	User.Read	Delegated	Sign in and read user profile	No

A message at the bottom says: "To view and manage permissions and user consent, try Enterprise applications."

To the right, a modal window titled "Request API permissions" is open. It shows the "Office 365 Management APIs" section and asks: "What type of permissions does your application require?". It has two options: "Delegated permissions" (selected) and "Application permissions". A note states: "Your application needs to access the API as the signed-in user." The "Application permissions" section is described as: "Your application runs as a background service or daemon without a signed-in user."

## Steps

## Request API permissions

All APIs

Office 365 Management APIs  
<https://manage.office.com/> Docs

What type of permissions does your application require?

Delegated permissions  
Your application needs to access the API as the signed-in user.

Application permissions  
Your application runs as a background service or daemon without a signed-in user.

Select permissions expand all

Start typing a permission to filter these results

Permission	Admin consent required
ActivityFeed (2)	
ActivityFeed.Read ⓘ Read activity data for your organization	Yes
ActivityFeed.ReadDlp ⓘ Read DLP policy events including detected sensitive data	Yes
ServiceHealth (1)	
ServiceHealth.Read ⓘ Read service health information for your organization	Yes

## Request API permissions

Microsoft Graph  
<https://graph.microsoft.com/> [Docs](#)

What type of permissions does your application require?

**Delegated permissions**  
Your application needs to access the API as the signed-in user.

**Application permissions**  
Your application runs as a background service or daemon without a signed-in user.

Select permissions [expand all](#)

Start typing a permission to filter these results

Permission	Admin consent required
> AccessReview	
> Acronym	
> AdministrativeUnit	
> AgreementAcceptance	
> Agreement	
> APIConnectors	
> AppCatalog	
> Application	
> AppRoleAssignment	
> ...	

## Request API permissions

< All APIs  
Office 365 Exchange Online  
https://ps.outlook.com

What type of permissions does your application require?

**Delegated permissions**  
Your application needs to access the API as the signed-in user.

**Application permissions**  
Your application runs as a background service or daemon without a signed-in user.

Select permissions expand all

Start typing a permission to filter these results

Permission	Admin consent required
full_access_as_app ⓘ Use Exchange Web Services with full access to all mailboxes	Yes
Calendars	
Contacts	
Exchange (1)	
Exchange.ManageAsApp ⓘ Manage Exchange As Application	Yes
IMAP	
Mailbox	

## Steps

## Request API permissions

[All APIs](#)

Azure Rights Management Services  
<https://aadrm.com/> [Docs](#)

What type of permissions does your application require?

**Delegated permissions**  
Your application needs to access the API as the signed-in user.

**Application permissions**  
Your application runs as a background service or daemon without a signed-in user.

Select permissions [expand all](#)

Start typing a permission to filter these results

Permission	Admin consent required
<input checked="" type="checkbox"/> Application.Read.All ⓘ Read all service configuration and log data for the Azure Information Protection service.	Yes
<b>Content</b>	
<input type="checkbox"/> Content.DelegatedReader ⓘ Read protected content on behalf of a user	Yes
<input type="checkbox"/> Content.DelegatedWriter ⓘ Create protected content on behalf of a user	Yes
<input type="checkbox"/> Content.SuperUser ⓘ Read all protected content for this tenant	Yes
<input type="checkbox"/> Content.Writer ⓘ Create protected content	Yes

## Steps

Home > Kaz Demos | App registrations > Microsoft Purview Reports

## Microsoft Purview Reports | API permissions

Grant admin consent confirmation.

Do you want to grant consent for the requested permissions for all accounts in Kaz Demos? This will update any existing admin consent records this application already has to match what is listed below.

Yes  No

Configured permissions

Applications are authorized to call APIs when they are granted permissions by users/admins as part of the consent process. The list of configured permissions should include all the permissions the application needs. [Learn more about permissions and consent](#)

API / Permissions name	Type	Description	Admin consent requ...	Status
Azure Rights Management Services				
Application.Read.All	Application	Read all service configuration and log data for the Azure I...	Yes	<span>Granted for Kaz Demos</span> ...
Microsoft Graph (6)				
AuditLog.Read.All	Application	Read all audit log data	Yes	<span>Granted for Kaz Demos</span> ...
Directory.Read.All	Application	Read directory data	Yes	<span>Granted for Kaz Demos</span> ...
Group.Read.All	Application	Read all groups	Yes	<span>Granted for Kaz Demos</span> ...
Organization.Read.All	Application	Read organization information	Yes	<span>Granted for Kaz Demos</span> ...
User.Read	Delegated	Sign in and read user profile	No	<span>Granted for Kaz Demos</span> ...
User.Read.All	Application	Read all users' full profiles	Yes	<span>Granted for Kaz Demos</span> ...
Office 365 Exchange Online (1)				
Exchange.ManageAsApp	Application	Manage Exchange As Application	Yes	<span>Granted for Kaz Demos</span> ...
Office 365 Management APIs (3)				
ActivityFeed.Read	Application	Read activity data for your organization	Yes	<span>Granted for Kaz Demos</span> ...
ActivityFeed.ReadDlp	Application	Read DLP policy events including detected sensitive data	Yes	<span>Granted for Kaz Demos</span> ...
ServiceHealth.Read	Application	Read service health information for your organization	Yes	<span>Granted for Kaz Demos</span> ...

## Create a workspace in Logs Analytics

- Open <https://portal.azure.com>
- Look for Log Analytics workspaces
- Press +Create
  - Select your Azure Subscription
  - Select or create a Resource group
  - Set a Name related to this workspace
  - Select the Region that best match
  - Press Review + Create, wait until a resume is show
  - Press Create
- Wait until the workspace be success deployed
- Open the new workspace
- Go to Agents management menu, and open Log Analytics agent instructions
  - Copy Workspace ID
  - Copy Primary key

To **integrate** with **Sentinel** the same Logs Analytics workspace is needed to use.

## Steps

The screenshot shows the Azure portal search results for the query "log Analytics". The search bar at the top contains the text "log Analytics". Below the search bar, there are several filter tabs: All (selected), Services (17), Marketplace (7), Documentation (20), Resources (0), and Resource Groups (0). The main search results are categorized under "Services", "Marketplace", and "Documentation".

**Services**

- Log Analytics query packs
- Log Analytics workspaces (highlighted)
- Activity log
- Stream Analytics clusters

**Marketplace**

- Log Analytics Workspace
- Azure Log Analytics Agent Health
- FortiAnalyzer Centralized Log Analytics
- HPE OneView for Azure Log Analytics (v1.4.0)

**Documentation**

- Overview of Log Analytics in Azure Monitor - Azure Monitor
- Create Log Analytics workspaces - Azure Monitor | Microsoft Docs

# LOGS ANALYTICS WORKSPACE

Steps

The screenshot shows the Microsoft Azure Log Analytics workspaces interface. At the top, there's a navigation bar with 'Microsoft Azure (Preview)', a search bar, and a 'Home' link. Below the navigation is the main title 'Log Analytics workspaces'. A toolbar contains buttons for 'Create', 'Open recycle bin', 'Manage view', 'Refresh', 'Export to CSV', 'Open query', and 'Assign tags'. There are also filter options: 'Subscription equals 3 of 48 selected', 'Resource group equals all', 'Location equals all', and a 'Add filter' button. The main area displays columns for 'Name' (sorted by name), 'Resource group' (sorted by resource group), and 'Location' (sorted by location). The 'Name' column has a checkbox icon.



## Steps

Home > Log Analytics workspaces >

## Create Log Analytics workspace

Basics Tags Review + Create

**i** A Log Analytics workspace is the basic management unit of Azure Monitor Logs. There are specific considerations you should take when creating a new Log Analytics workspace. [Learn more](#)

With Azure Monitor Logs you can easily store, retain, and query data collected from your monitored resources in Azure and other environments for valuable insights. A Log Analytics workspace is the logical storage unit where your log data is collected and stored.

**Project details**

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription \* ⓘ Microsoft Azure Internal Consumption (Preview) ...

Resource group \* ⓘ kazdemos.org Create new

**Instance details**

Name \* ⓘ M365Reports ✓

Region \* ⓘ Central US

Review + Create << Previous Next : Tags >

## Steps

The screenshot shows the Microsoft Azure (Preview) portal with the following details:

- Page Title:** Microsoft LogAnalyticsOMS | Overview
- Deployment Status:** Your deployment is complete
- Deployment Details:**
  - Deployment name: Microsoft.LogAnalyticsOMS
  - Subscription: Microsoft Azure Internal Consumption (8c9c38d5-57e...)
  - Resource group: kazdemos.org
  - Start time: 19/8/2022, 16:59:35
  - Correlation ID: fe71f61b-0d3f-4eeb-8ce6-3615b0384a1f
- Next Steps:** Go to resource



Microsoft Azure (Preview) Search resources, services, and docs (G+)

Home > Microsoft.LogAnalyticsOMS | Overview > M365Reports

## M365Reports | Agents management

Log Analytics workspace

Search (Ctrl+ /) <

- Overview
- Activity log
- Access control (IAM)
- Tags
- Diagnose and solve problems

Settings

- Locks
- Agents management
- Legacy agents management
- Custom logs
- Computer Groups
- Data Export
- Linked storage accounts
- Network Isolation
- Tables (preview)

General

- Workspace summary
- Workbooks

Windows servers Linux servers

**i 0 Windows computers connected**  
via Azure Monitor Windows agent  
[See them in Logs](#)

**i 0 Windows computers connected**  
via Log Analytics Windows agent (legacy)  
[See them in Logs](#)

Want to setup the new Azure Monitor agent? Go to '[Data Collection Rules](#)'

[Data Collection Rules](#)

**Log Analytics agent instructions**

### Download agent

Download an agent for your operating system, then install and configure it using the keys for your workspace ID. You'll need the Workspace ID and Key to install the agent.

[Download Windows Agent \(64 bit\)](#)  
[Download Windows Agent \(32 bit\)](#)

Workspace ID: 5703c95e-dec2-4c21-809b-d23d31c4c3c6

Primary key: r5Bt8HlskQroE8zuhXaMh7GZvcheWPuDWKZ5On+NI9Tlv... [Regenerate](#)

Secondary key: T7yhxt/T2td0JBU9sn6hLNW8jwvCNZxLkE5jdkuz4jREa7TA... [Regenerate](#)

### Log Analytics Gateway

If you have machines with no internet connectivity to Log Analytics workspace, download the Log Analytics Gateway to act as a proxy.

## Extend your data retention until 2 years

Under General select Usage and estimated costs and then Data Retention, set Data Retention period.

**MCAR-Kazdemos | Usage and estimated costs**

Log Analytics workspace

Search (Ctrl+ /) Usage details Insights Daily cap Data Retention Help

Your Log Analytics cost depends on your choice of pricing tier, data retention and which solutions are used. Here you can see the estimated monthly cost for each of the available pricing tiers, based on your last 31-days of Log Analytics data ingested. These cost estimates can be used to help you select the best pricing tier based on your data ingestion patterns. These estimates include the 500MB/VM/day data allowances if you are using Microsoft Defender. This page does not reflect your actual billed usage. To view that, use Cost Management (learn more). If you have questions about using this page, contact us. Learn more about Log Analytics pricing.

**Pricing Tiers**

Pay-as-you-go Recommended Tier Per GB

The Pay-as-you-go pricing tier offers flexible consumption pricing in which you are charged per GB of data ingested. There are additional charges if you increase the data retention above the 31 day included retention (or 90 day included retention if using Sentinel on this workspace). Learn more about Log Analytics pricing.

**Estimated costs**

Item type	Price	Monthly usage (last 31 days)	Estimated monthly cost
Log data ingestion	2,76 US\$	0,00 GB	0,00 US\$
Microsoft Defender allowance	0,00 US\$	0,00 GB	0,00 US\$
Log data retention (beyond 31 days)	0,12 US\$	0,00 GB	0,00 US\$
<b>Total</b>			<b>0,00 US\$</b>

This is the current pricing tier.

**Data Retention**

31 days of retention is included with your pricing plan. Longer retention will incur additional charges. Retention can also be configured individually for specific data types.

Data Retention (Days) 730

Retention for Application Insights data types default to 90 days and will get the workspace retention if it is over 90 days. To set the retention on these types to be less than 90 days, set the retention on each of these data types. Learn more.

OK

Billable data ingestion per solution (last 90 days)

Category	Value (kB)
18 ago	~600
12	~350
19	~350

Data ingested per solution (last 90 days)

Category

No data

## Generate a Certificate to Execute some PowerShell cmdlets unattended

- We will need to **create a certificate** to connect to Information Protection services through Azure AD application, to do that the 1<sup>st</sup> step is create that certificate:

- Open an ISE PowerShell console and execute (Select a folder 1<sup>st</sup> to save the 2 files created, example c:\Tmp):

```
# Create certificate  
  
$mycert = New-SelfSignedCertificate -DnsName "kazdemos.org" -CertStoreLocation  
"cert:\CurrentUser\My" -NotAfter (Get-Date).AddYears(1) -KeySpec KeyExchange  
  
# Export certificate to .pfx file  
  
$mycert | Export-PfxCertificate -FilePath mycert.pfx -Password (Get-  
Credential).password  
  
# Export certificate to .cer file  
  
$mycert | Export-Certificate -FilePath mycert.cer
```

- On the 2nd line a Password will be required for PFX file, PowerShell will ask for user and password, but finally only password will be used (add both)
- We need to use the same Azure AD application explained [here](#)
- Under the same Azure AD App we need go to “certificates & secrets” menu and select certificates, then press “Upload certificate” here is required the file .cer, a new blade appear, select the folder icon to looking for the file and open the file.
- After Add copy the Thumbprint value, we will need to add to the Iaconfig.json file
- A script to create the certificate is added with the rest of the scripts under the Certs folder, and called **CreateCertificate.ps1**, this script contains the same previous cmdlets.



# CERTIFICATES CREATION

## STEPS

The screenshot shows a Windows PowerShell ISE window with the title "Administrator: Windows PowerShell ISE". The menu bar includes File, Edit, View, Tools, Debug, Add-ons, and Help. The toolbar has standard icons for file operations. There are two tabs open: "Untitled1.ps1\*(Recovered)" and "Untitled2.ps1\*(Recovered)". The "Untitled1.ps1\*(Recovered)" tab contains the following PowerShell script:

```
1 # Create certificate
2 $mycert = New-SelfSignedCertificate -DnsName "kazdemos.org" -CertStoreLocation "cert:\CurrentUser\My" -NotAfter (Get-Date).AddYears(1)
3
4 # Export certificate to .pfx file
5 $mycert | Export-PfxCertificate -FilePath KazDemosCertificate.pfx -Password (Get-Credential).password
6
7 # Export certificate to .cer file
8 $mycert | Export-Certificate -FilePath KazDemosCert
```

The "Untitled2.ps1\*(Recovered)" tab is currently active. A modal dialog box titled "cmdlet Get-Credential at command pipeline position 1" is displayed, prompting for a user name and password. The user name field contains "NoMatter" and the password field contains a masked password. Buttons for "OK" and "Cancel" are at the bottom.

In the bottom left of the PowerShell window, the status bar displays: "Running script / selection. Press Ctrl+Break to stop. Press Ctrl+B to break into debugger." The bottom right shows the line count "Ln 11 Col 1" and the zoom level "100%".



## STEPS

Home > Kaz Demos | App registrations > Microsoft Purview Reports

## Microsoft Purview Reports | Certificates & secrets

Search Got feedback?

Overview

Quickstart

Integration assistant

### Manage

Branding & properties

Authentication

Certificates & secrets

Token configuration

API permissions

Expose an API

App roles

Owners

Application registration certificates, secrets and federated credentials can be found in the tabs below.

**Certificates (1)** Client secrets (1) Federated credentials (0)

Certificates can be used as secrets to prove the application's identity when requesting a token. Also can be referred to as public keys.

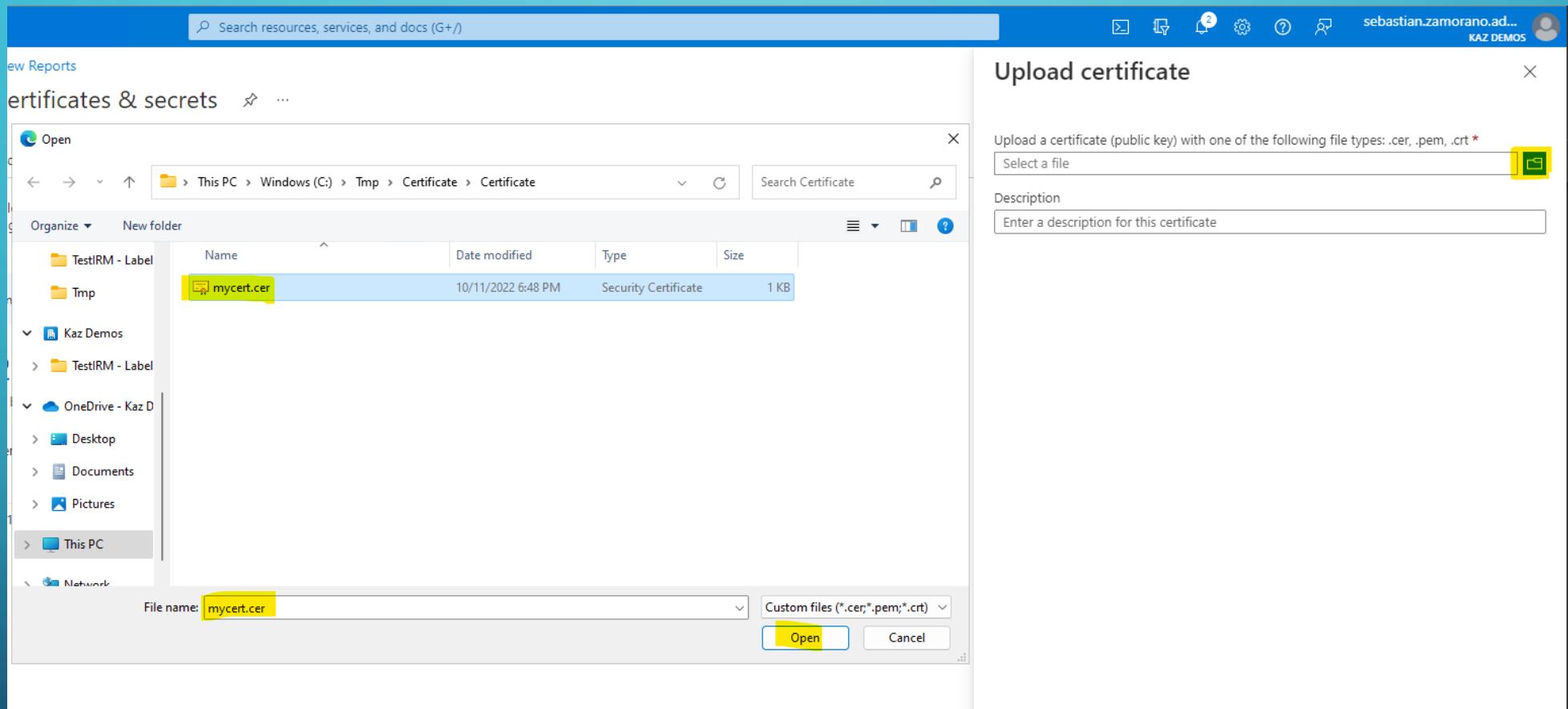
Upload certificate

Thumbprint	Description	Start date	Expires	Certificate ID
1808C721701BE04644F3D9FD107917...	Microsoft Purview	10/11/2022	10/11/2023	dc535e74-a958-4eea-...



# ADD CONFIGURATION TO AZURE AD APP

## STEPS



# ADD CONFIGURATION TO AZURE AD APP

## STEPS

Certificates (1) Client secrets (1) Federated credentials (0)

Certificates can be used as secrets to prove the application's identity when requesting a token. Also can be referred to as public keys.

Upload certificate

Thumbprint	Description	Start date	Expires	Certificate ID
1808C721701BE04644F3D9FD107917...	Microsoft Purview	10/11/2022	10/11/2023	dc535e74-a958-4eea-...

```
new 1 new 2 my_kazdemos_labels.csv new 3 new 4 run_me.ps1 new 5 Get-auditLogs.ps1 export_logs.ps1 Get-LabelData.ps1 laconfig.json ExportCSV2LA.ps1
1 {
2     "EncryptedKeys": "True",
3     "AppClientID": "701a1c2f-2500-429f-8e13-aa51ca5af20",
4     "ClientSecretValue": "01000000d08c9ddf0115d1118c7a00c04fc297eb010000009be602676a73ec489a1075b1c814e1e30000000",
5     "TenantGUID": "a01a113-100e-4ac8-a4c9-52c",
6     "TenantDomain": "kazdemos.org",
7     "LA_CustomerID": "buzo511/-1a38-400j-adaf-a00,000,01-3",
8     "LA_SharedKey": "01000000d08c9ddf0115d1118c7a00c04fc297eb010000009be602676a73ec489a1075b1c814e1e30000000002",
9     "CertificateThumb": "01000000d08c9ddf0115d1118c7a00c04fc297eb010000009be602676a73ec489a1075b1c814e1e300000000",
10    "OnmicrosoftURL": "MC...36.onmicrosoft.com"
11 }
12 }
```

In this case the certificate thumbprint was encrypted using the steps explained in this [slide](#)

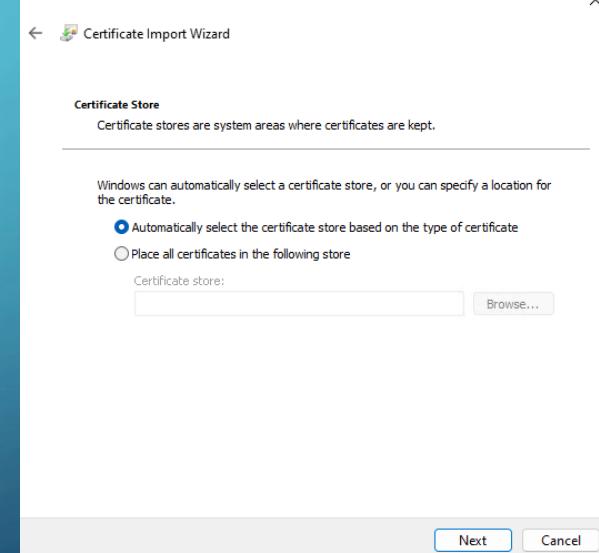
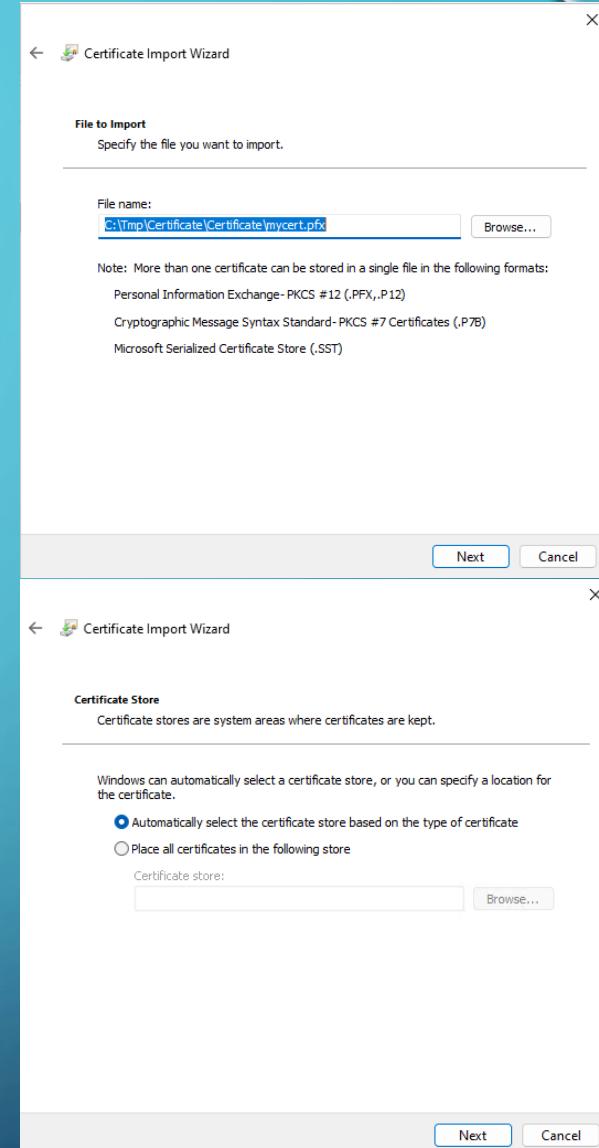
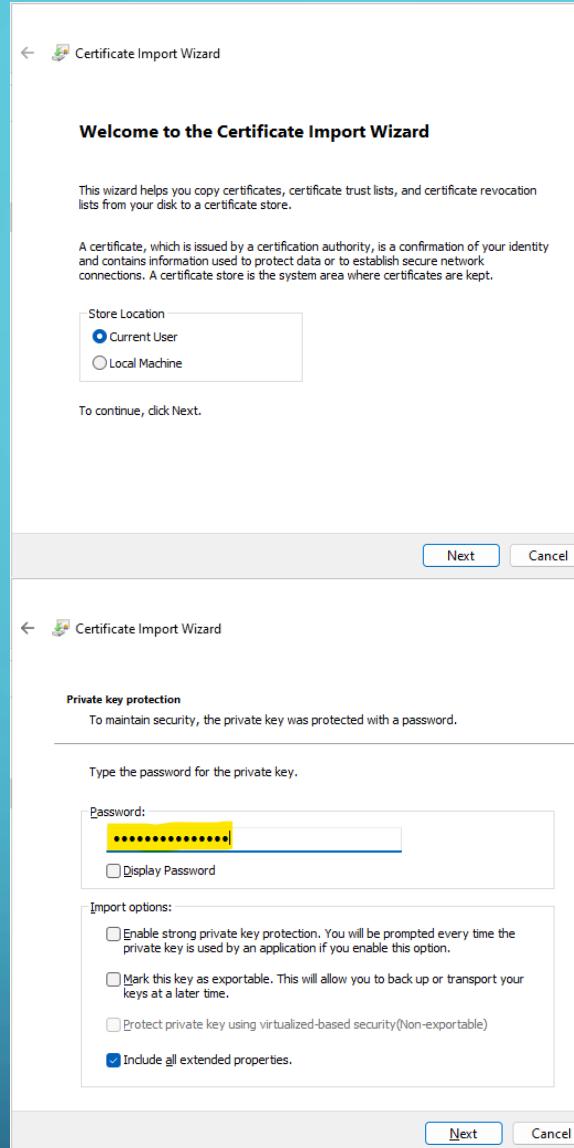
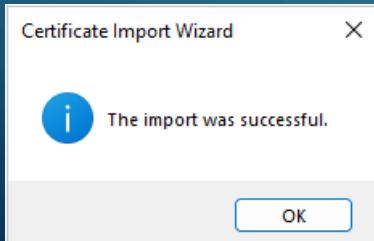
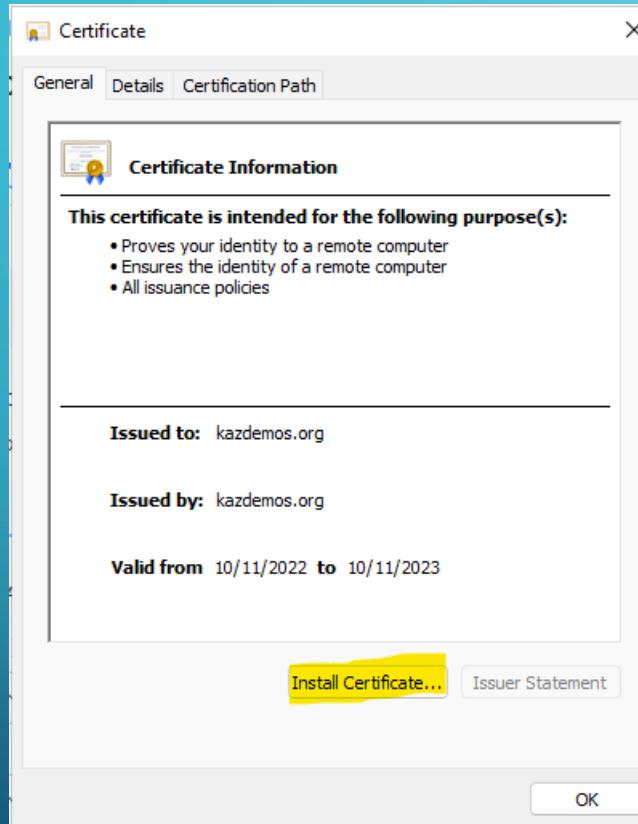


## To execute the script using the certificate

- On the same computer used to execute the scripts is needed to install both certificates previously created
- The certificates needs to be installed on the same account used to execute these scripts:
  - Get-AzureADData.ps1
  - Get-AzureADDomains.ps1
  - Get-AzureADRoles.ps1
  - Get-LabelData.ps1
  - Get-RMSData.ps1 (This one can be executed only using PowerShell 5)
- To install, is required put both certificates on the machine used to execute the scripts and execute both doing double click.
- For each one is required to install the certificate locally.

# INSTALL CERTIFICATE LOCALLY

STEPS



Microsoft

## Compliance administrator role assigned to Azure AD App

To execute Get-LabelData.ps1 script additional permissions are required:

- To give the permissions is required open Azure Active Directory
- Then go to “Roles and administrators” menu
- Search for Compliance Administrator and press the name
- In the new window click over “+ Add assignments”
- Under Add assignments interface, press under “Select member(s)” and looking for the name of the Azure AD App previously created, click it over the name and press select
- Press the Next button, and in the new interface select “Active” under Assignment type, maintain check the option “Permanently assigned” and provide a justification to enable the “Assign” button
- Understanding the complexity to give elevate privileges to an unattended script, a line can be modified on the script to a manual execution, in this case someone with Compliance Administrator role needs to execute this script on PowerShell, this execution is on-demand and is required only when you have modified or new labels, to up to date the reports with this information, or can be required depending the retention period in the Logs Analytics workspace.

```
154
155 Function Export-LabelData() {
156     #
157     #   Name      : Export-LabelData
158     #   Desc      : Extracts data from Get-Label into Log analytics workspace tables for reporting purposes
159     #   Return    : None
160     #
161     <#
162     .NOTES
163     If you cannot add the "Compliance Administrator" role to the Azure AD App, for security reasons, you can comment the line 167 and uncomment the line 166, in that case
164     Someone with "Compliance Administrator" role needs to execute this script, this script is executed on-demand to refresh the label names
165     #>
166     #Connect-IPPSession
167     Connect-IPPSession -CertificateThumbprint $CertificateThumb -AppID $AppClientID -Organization $OnmicrosoftTenant
```

# COMPLIANCE ADMIN ROLE FOR SCRIPT

## Steps

The screenshot shows the Azure portal interface. At the top, there is a search bar with the text "Azure acti". Below the search bar, a navigation bar includes tabs for "All", "Services (92)", "Resources", "Resource Groups", "Marketplace", and "Documentation". The "Services" tab is selected. A sub-menu for "Azure Active Directory" is open, showing options like "Activity log", "Azure Arc", "Azure Databricks", and "Automanage – Azure machine best practices".

**Welcome to Azure Active Directory**

Don't have a subscription? [Get started with a free trial](#)

**Services**

- Azure Active Directory
- Security
- Azure Cosmos DB
- Azure Database for MySQL servers

See all

**Start with an Azure free trial**

Get \$200 free credit toward Azure products and services, plus 12 months of popular [free services](#).

[Start](#) [Learn more](#)

**Manage Azure Active Directory**

Manage access, set smart policies, and enhance security with Azure Active Directory.

[View](#) [Learn more](#)

**Access student benefits**

Get free software, Azure credit, or access Azure Dev Tools for Teaching after you verify your academic status.

[Explore](#) [Learn more](#)

**Azure services**

[Create a resource](#) [Azure Active Directory](#) [Azure Information...](#) [Microsoft Purview...](#) [Azure AD B2C](#) [Azure AD Privileged...](#) [Help + support](#) [Microsoft Sentinel](#) [Function App](#) [More services](#)

**Resources**

[Recent](#) [Favorite](#)

Name	Type	Last Viewed

No resources have been viewed recently

[View all resources](#)



## Steps

Home >

### Kaz Demos | Overview

Azure Active Directory

+ Add Manage tenants What's new Preview features Got feedback?

Overview Preview features Diagnose and solve problems

Manage

- Users
- Groups
- External Identities
- Roles and administrators**
- Administrative units
- Delegated admin partners
- Enterprise applications
- Devices
- App registrations
- Identity Governance
- Application proxy
- Custom security attributes (Preview)
- Licenses

Microsoft Entra has a simpler, integrated experience for managing all your Identity and Access Management needs. Try the new Microsoft Entra admin center! [Learn more](#)

Overview Monitoring Properties Tutorials

Search your tenant

Basic information

Name	Kaz Demos	Users	36
Tenant ID	ac1dff03-7e0e-4ac8-a4c9-9b38d24f062c	Groups	71
Primary domain	kazdemos.org	Applications	9
License	Azure AD Premium P2	Devices	12

Alerts

**Upcoming MFA Server deprecation**  
Please migrate from MFA Server to Azure AD Multi-Factor Authentication by September 2024 to avoid any service impact.  
[Learn more](#)

**Upcoming TLS 1.0, 1.1 and 3DES deprecation**  
Please enable support for TLS 1.2 on clients(applications/platform) to avoid any service impact.  
[Learn more](#)

## Steps

Home > Kaz Demos | Roles and administrators >

### Roles and administrators | All roles ...

Kaz Demos - Azure Active Directory

- « [All roles](#)
- [Diagnose and solve problems](#)
- [Activity](#)
- [Access reviews](#)
- [Audit logs](#)
- [Troubleshooting + Support](#)
- [New support request](#)

+ New custom role <span style="color: #0078d4;">Delete custom role</span> <span style="color: #0078d4;">Download assignments</span> <span style="color: #0078d4;">Refresh</span> <span style="color: #0078d4;">Preview features</span> <span style="color: #0078d4;">Got feedback?</span>		
<p>Get just-in-time access to a role when you need it using PIM. Learn more about PIM →</p>		
<input type="checkbox"/>	B2C IEF Keyset Administrator	Can manage secrets for federation and encryption in the Identity Experience Framework (IEF). <span style="float: right;">Built-in</span>
<input type="checkbox"/>	B2C IEF Policy Administrator	Can create and manage trust framework policies in the Identity Experience Framework (IEF). <span style="float: right;">Built-in</span>
<input type="checkbox"/>	Billing Administrator	Can perform common billing related tasks like updating payment information. <span style="float: right;">Built-in</span>
<input type="checkbox"/>	Cloud App Security Administrator	Can manage all aspects of the Cloud App Security product. <span style="float: right;">Built-in</span>
<input type="checkbox"/>	Cloud Application Administrator	Can create and manage all aspects of app registrations and enterprise apps except App Proxy. <span style="float: right;">Built-in</span>
<input type="checkbox"/>	Cloud Device Administrator	Limited access to manage devices in Azure AD. <span style="float: right;">Built-in</span>
<input checked="" type="checkbox"/>	Compliance Administrator	Can read and manage compliance configuration and reports in Azure AD and Microsoft 365. <span style="float: right;">Built-in</span>
<input type="checkbox"/>	Compliance Data Administrator	Creates and manages compliance content. <span style="float: right;">Built-in</span>
<input type="checkbox"/>	Conditional Access Administrator	Can manage Conditional Access capabilities. <span style="float: right;">Built-in</span>
<input type="checkbox"/>	Customer LockBox Access Approver	Can approve Microsoft support requests to access customer organizational data. <span style="float: right;">Built-in</span>
<input type="checkbox"/>	Desktop Analytics Administrator	Can access and manage Desktop management tools and services. <span style="float: right;">Built-in</span>
<input type="checkbox"/>	Directory Readers	Can read basic directory information. Commonly used to grant directory read access to applications and guests. <span style="float: right;">Built-in</span>
<input type="checkbox"/>	Directory Writers	Can read and write basic directory information. For granting access to applications, not intended for users. <span style="float: right;">Built-in</span>



## Steps

Home >

### Compliance Administrator | Assignments

Privileged Identity Management | Azure AD roles

Add assignments Settings Refresh Export Got feedback?

Manage

Assignments Description Role settings

Eligible assignments Active assignments Expired assignments

Search by member name or principal name

Name	Principal name	Type	Scope	Membership	State	Start time	End time	
Compliance Administrator								
Mou	m@kazdemos.org	User	Directory	Direct	Assigned	-	Permanent	
Darrer	ll@kazdemos.o	User	Directory	Direct	Assigned	-	Permanent	
Vinicic	@kazdemos.org	User	Directory	Direct	Assigned	-	Permanent	
Fem	i@kazdemos.org	User	Directory	Direct	Assigned	-	Permanent	
Nt	N	ui@kazdemo	User	Directory	Direct	Assigned	-	Permanent
N.	I	jl@kazdemos.org	User	Directory	Direct	Assigned	-	Permanent
D	C	ot@kazdemos.org	User	Directory	Direct	Assigned	-	Permanent
Mic.	sports 701	13-d3	Service principal	Directory	Direct	Assigned	10/11/2022, 7:09:36 PM	Permanent
M.	Sen 096	0c0-d	Service principal	Directory	Direct	Assigned	12/12/2022, 5:37:55 PM	Permanent
Test	col 6	-9831-d	Service principal	Directory	Direct	Assigned	10/27/2022, 1:26:04 PM	Permanent

# COMPLIANCE ADMIN ROLE FOR SCRIPT

## Steps

Home > Compliance Administrator | Assignments >

### Add assignments

Privileged Identity Management | Azure AD roles

Membership   Setting

*You can also assign roles to groups now. [Learn more](#)*

Resource  
Kaz Demos

Resource type  
Directory

Select role *①*  
Compliance Administrator

Scope type *①*  
Directory

Select member(s) \* *②*  
No member selected

Next >   Cancel

Select a member

Privileged Identity Management | Azure AD roles

Only groups eligible for role assignment are displayed. [Learn more](#)

MPARR - Collector for Sentinel

M- 096e4-a0 Jef9 Selected

Selected items

MPARR - Collector for Sentinel  
096e4-a0 Jef9 Remove

Select



## Steps

Home > Compliance Administrator | Assignments >

### Add assignments

Privileged Identity Management | Azure AD roles

Membership    Setting

Assignment type ⓘ

Eligible

Active

Maximum allowed assignment duration is permanent.

Permanently assigned

Assignment starts  
12/30/2022  7:57:35 PM

Assignment ends  
06/28/2023  7:57:35 PM

Enter justification \*

Unattended script for Microsoft Purview Advanced Rich Reports (MPARR) collector ✓

## Configure the script

- **Install PowerShell v7**

- Create a local folder under C:\ like as “**Collector**”
- Unzip the file “**MPARR Collector.zip**” in that folder
- Open the file “**laconfig.json**” and complete with this information:
  - **EncryptedKeys:** “False” (To use check [here](#), this can be used to encrypt the Keys stored in this file)
  - **AppClientID:** Application (client) ID from app registration step
  - **ClientSecretValue:** Secret key from Certificates and secrets under app registration step
  - **TenantGUID:** Tenant ID from app registration step
  - **TenantDomain:** Principal domain in the tenant
  - **LA\_CustomerID:** Workspace ID from Logs Analytics (here can be used Sentinel Workspace)
  - **LA\_SharedKey:** Primary key from Logs Analytics (here can be used Sentinel Workspace)
  - **Cloud:** used to select the kind of tenant, this permit uses the right URL for the collector (see notes)
  - **CertificateThumb:** Certificate thumbprint from a self signed certificate
  - **OnmicrosoftURL:** Tenant domain in format <tenant.onmicrosoft.com>
  - **RMSLogs:** This will be used for the Get-RMSdata script to capture information from AIP Service API
  - **OutPutLogs:** Select where your Logs will be recorded
- Save the changes

## Configure the script

- On the same folder
- Open the file “**schemas.json**” and make changes if you want to avoid download certain kind of information:
  - According to [this](#) information, All the information collected on the different Office 365 Management API schemas are collected on 5 content blobs, these are:
    - Audit.AzureActiveDirectory
    - Audit.Exchange
    - Audit.SharePoint
    - Audit.General (includes all other workloads not included in the previous content types)
    - DLP.All (DLP events only for all workloads)
  - To make any changes to avoid certain information the value “True” need to changed to “False”
- In the new version of “**schemas.json**” new filtering capabilities was added to the file, for each Content Blob named previously can be set a filter for “Contains” or “NotContains”, previously the filter only worked with “Contains”, in this case if some operation is not wanted using “NotContains” this can be avoided. This is only on the case that you want to use any of these parameters with the MPARR Collector script:
  - FilterAuditAzureActiveDirectory
  - FilterAuditExchange
  - FilterAuditSharePoint
  - FilterAuditGeneral
  - FilterDLPAll
- Save the changes

## Configure task scheduler

- Open Task Scheduler
- Go to Task Scheduler Library
- Under Actions click **Create Task\***
  - On General tab
  - Set a Name
  - Under Security options change to Run whether users is logged on or not
  - On Triggers tab
  - On Settings select Daily
  - On Advanced settings set Repeat task every in this case 10 minutes, can be set every hour, and for a duration of Indefinitely
  - On Actions tab
  - Click New...
  - Under Program/script looking for PowerShell 7 application, normally is in this path "C:\Program Files\PowerShell\7\pwsh.exe"
  - Under Add arguments select the path to the script like as "C:\Collector\Sentinel\MPARR\_Collector.ps1"\*\*
  - Select OK and local credentials will be required
  - The script can be executed pressing Run option

\*Using the same process can be created a new task under task scheduler to execute the PowerShell script **Get-RMSData.ps1**, is required to execute on PowerShell v5

\*\* Previously was used run\_me.ps1 as principal script used to add the parameter "OutputPath" now is included on laconfig.json file



## Steps

```
1 {  
2     "EncryptedKeys": "False",  
3     "AppClientID": "70f-2f5-4f8-3c20",  
4     "ClientSecretValue": "i5o jRZEVbKs",  
5     "TenantGUID": "a3-7e-48-a9-92c",  
6     "TenantDomain": "kazdemos.org",  
7     "LA_CustomerID": "5d4c-71c-41i-8bf-04",  
8     "LA_SharedKey": "kN IDKC",  
9     "Cloud": "Commercial",  
10    "CertificateThumb": "1800000000000000000000000000000000000000000000000000000000000000",  
11    "OnmicrosoftURL": "M3.onmicrosoft.com",  
12    "RMSLogs": "c:\\Collector\\Script2.0\\RMSLogs\\",  
13    "OutPutLogs": "Logs\\"  
14}  
15
```

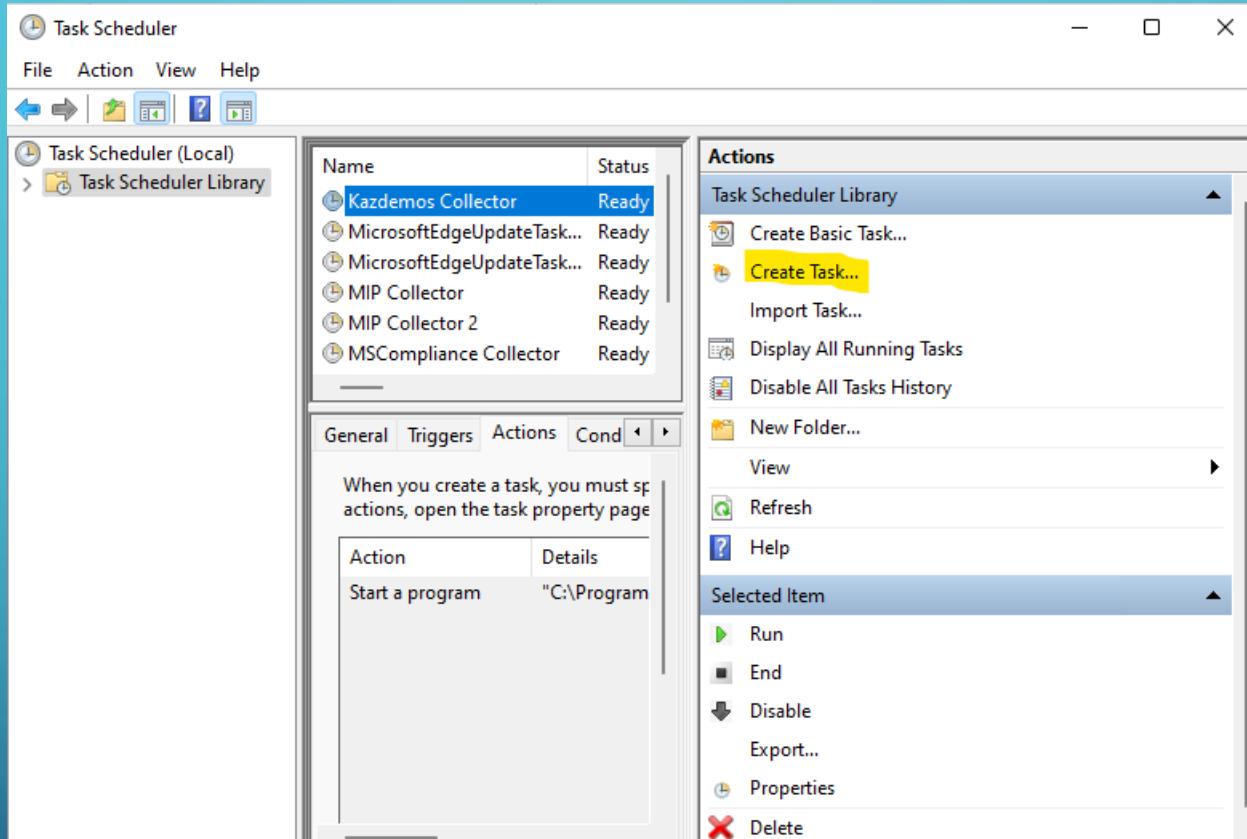
## Steps



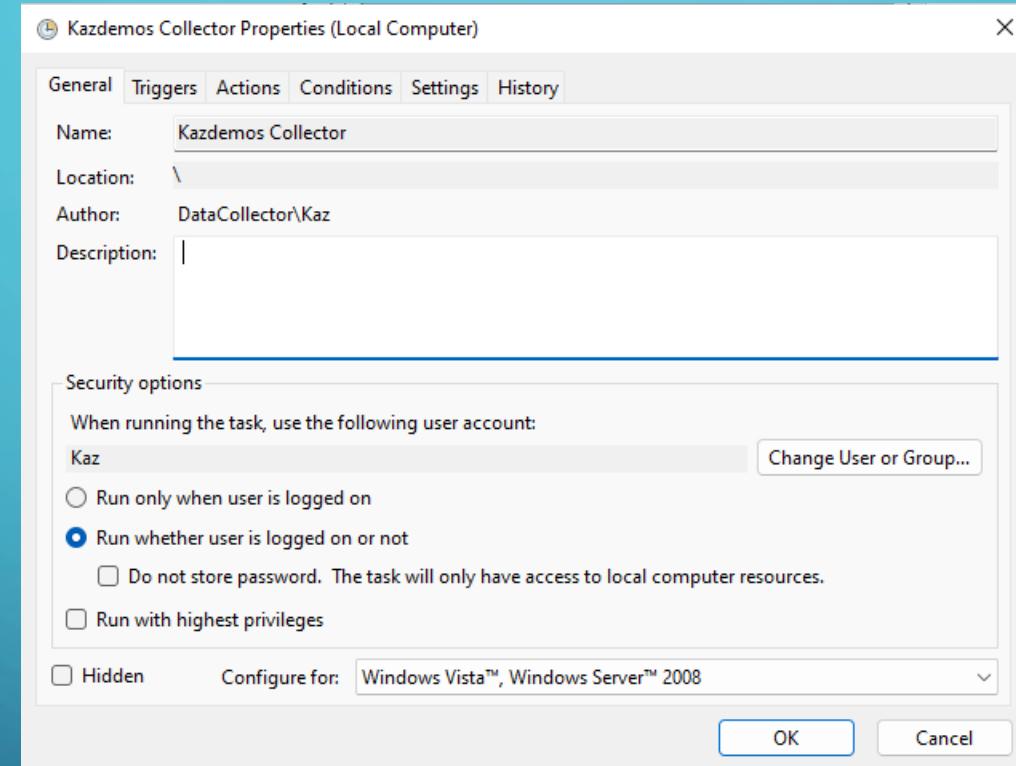
A screenshot of a code editor window showing a JSON configuration file. The file contains settings for auditing various Microsoft services. The code is as follows:

```
1  {
2    "Audit.AzureActiveDirectory": "True",
3    "FilterAuditAzureActiveDirectory": "NotContains",
4    "Audit.Exchange": "True",
5    "FilterAuditExchange": "Contains",
6    "Audit.SharePoint": "True",
7    "FilterAuditSharePoint": "Contains",
8    "Audit.General": "True",
9    "FilterAuditGeneral": "Contains",
10   "DLP.All": "True",
11   "FilterDLPAll": "Contains"
12 }
```

## Steps

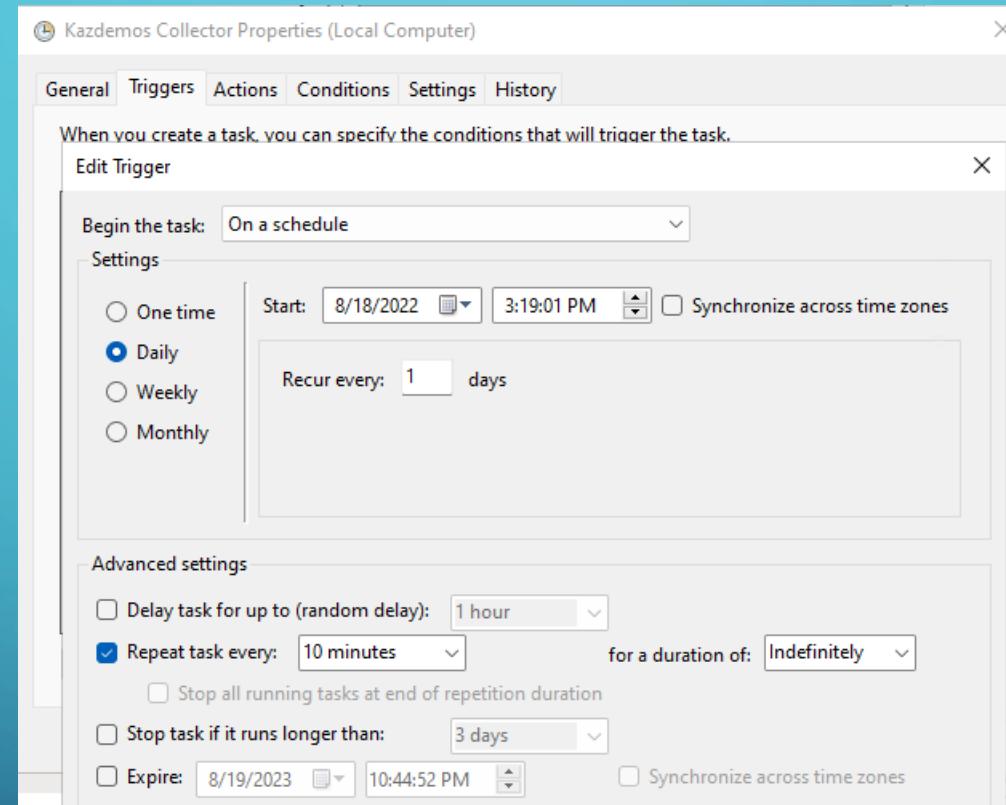


## Steps



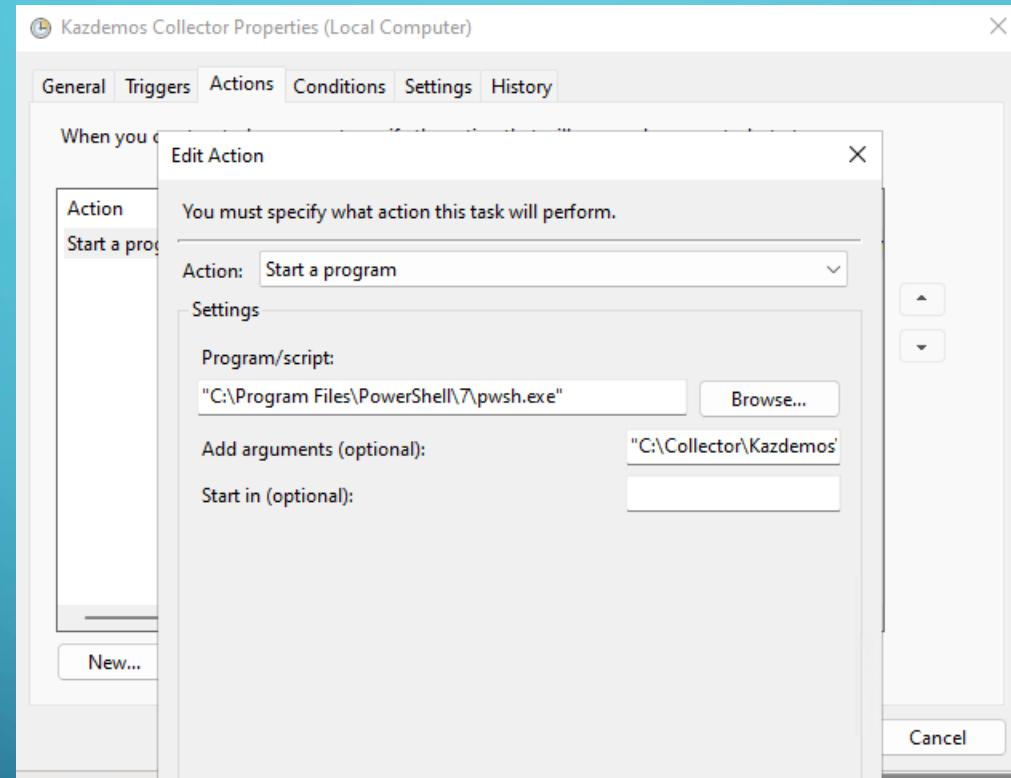
# WORKSTATION OR SERVER

## Steps

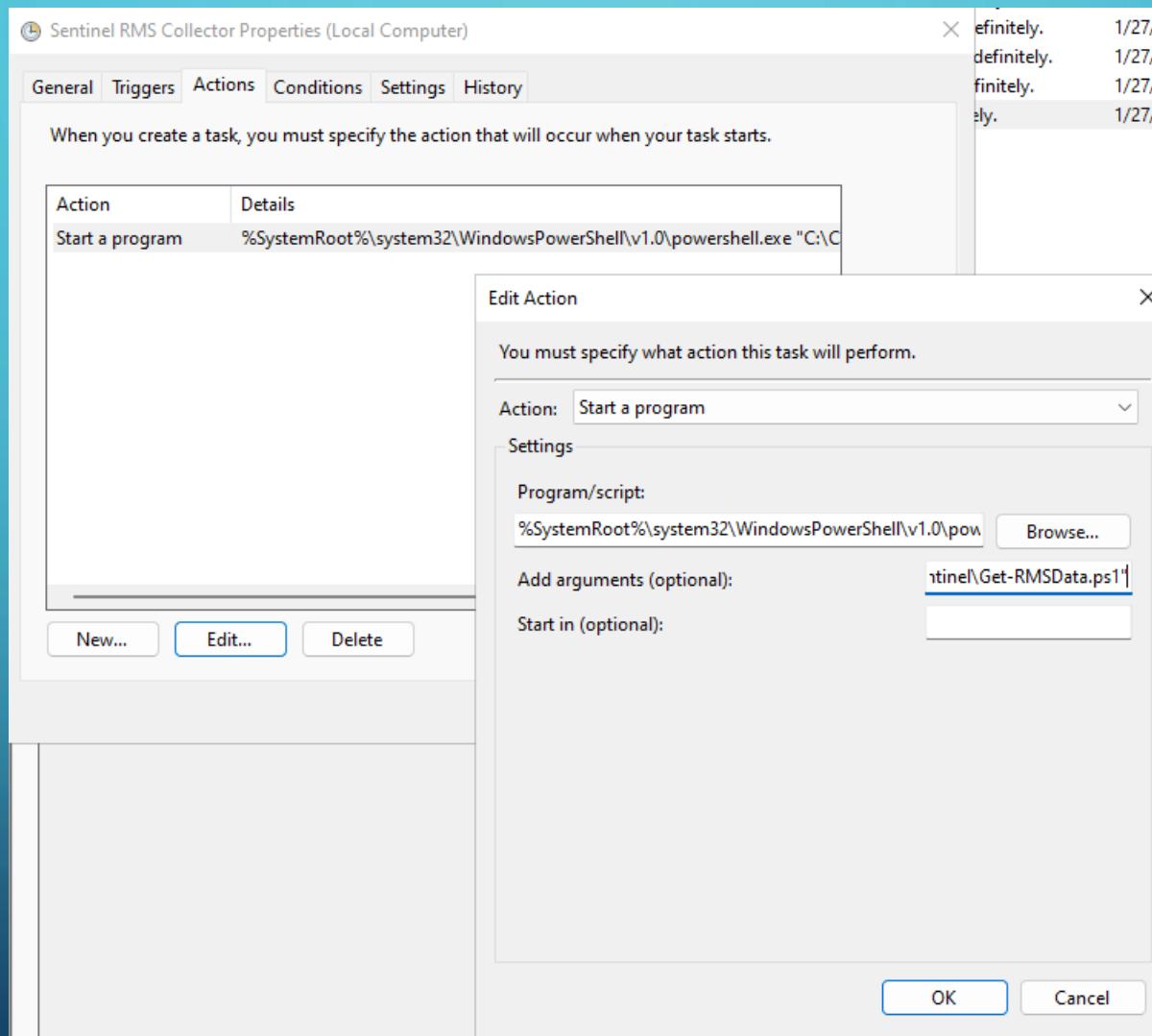


# WORKSTATION OR SERVER

## Steps

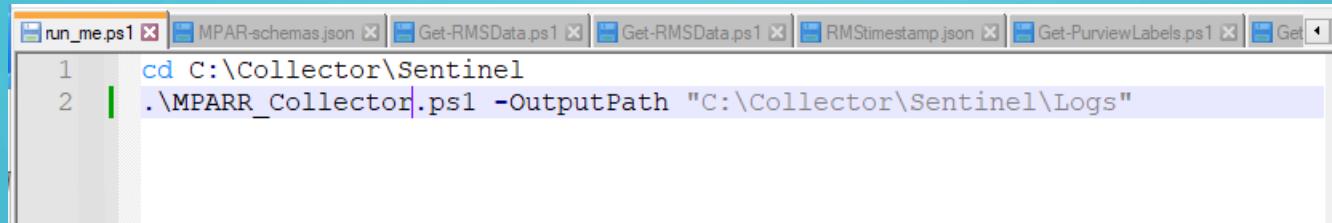


## Steps (Get-RMSData.ps1 needs to be execute with PowerShell v5)



## To execute the script with parameters

- Create a **run\_me.ps1** file in this way, and then replace the script used under Task Scheduler



```
run_me.ps1
1 cd C:\Collector\Sentinel
2 .\MPARR_Collector.ps1 -OutputPath "C:\Collector\Sentinel\Logs"
```

- The new **MPARR\_Collector.ps1** use the variable **OutputPath** inside of **laconfig.json** file, in this case **run\_me.ps1** can be still used to set other parameters.
- ADDITIONAL PARAMETERS ARE AVAILABLE AS A COMMENTS UNDER “**EXPORT\_LOGS.PS1**”, THAT OPTIONS CAN BE USED TO FILTER SOME **SPECIFIC OPERATIONS**, OR CHANGE LOG DIRECTORY, OR SET ONLY TO EXPORT TO LOCAL FILES. THESE FILTERS ARE:
  - FILTERAUDITAZUREACTIVEDIRECTORY
  - FILTERAUDITEXCHANGE
  - FILTERAUDITSHAREPOINT
  - FILTERAUDITGENERAL
  - FILTERDLPALL
- ALL THE FILTER AVAILABLE ARE EXPLAINED ON THE PRINCIPAL SCRIPT HEADER

## To encrypt Azure AD app key, workspace key and Certificate Thumbprint do this:

- Open laconfig.json file and do\* this actions:

- Under "EncryptedKeys" change "False" by "True"

**Copy the value for "ClientSecretValue" and execute on PowerShell this cmdlet:**

- PS C:\> "ClientSecretValue\*\*" | ConvertTo-SecureString -AsPlainText -Force | ConvertFrom-SecureString
- Replace the value on laconfig.json with the new string

**Copy the value for "LA\_Sharedkey" and execute on PowerShell this cmdlet:**

- PS C:\> "LA\_SharedKey\*\*" | ConvertTo-SecureString -AsPlainText -Force | ConvertFrom-SecureString
- Replace the value on laconfig.json with the new string

**Copy the value for "CertificateThumb" and execute on PowerShell this cmdlet:**

- PS C:\> "CertificateThumb\*\*" | ConvertTo-SecureString -AsPlainText -Force | ConvertFrom-SecureString
- Replace the value on laconfig.json with the new string

```

Administrator: PowerShell 7 (x64)
PS C:\Collector\MPIP Demo> "ClientSecretValue**" | ConvertTo-SecureString -AsPlainText -Force | ConvertFrom-SecureString
01000000d08c9ddf0115d1118c7a00c04fc297eb01000009be602676a73ec489a107
000
1a5ca7f6704f1d7f28183860893557d84e4e9255767d6000000027365d67b323ea5r0285f6f992d7f45867338b723fabeec8410848f3c62736c2f7b
ac48cb19ce941c4d5dc7ad3c6fc5f005adfd5b37f
b984400000008f99b24451bd7f87d641150a9fe54f7808135890e72a55674a6a3f8001718fd07155439e4855e24f3fbf86770b453988d618102e265
c0fe8bce1f32b00caa98
PS C:\Collector\MPIP Demo> "LA_SharedKey**" | ConvertTo-SecureString -AsPlainText -Force | ConvertFrom-SecureString
01000000d08c9ddf0115d1118c7a00c04fc297eb01
00038760a927b866b3c635c1b47cf24084ff7a747713608759b99a2c0e77b05a5400000000e80000000020000200000051e1587bf1b011b78
17c3906c2dd1a69eb2094b6397
805222a739b94aff548e571351e3fab8e533be66fe71a387ea72820d142a87c65fef6e24a39d342ea48148e98992b4252eb9d61660f5537dc5ee7e2
3529da83845f39b5c3349a90f9f5b5ada526fef786307062feb8c8f3411026475be999a9841fe48
755cc...1c9fd803401f8a400000067b772862e2d4a791d0fec8f311880526592
070909b6725dec4db1f1add282f3a76260b114868ccae088c28c9c433774d09cfe9b0db7c2d0ffcfc8c58ec35b64b
PS C:\Collector\MPIP Demo>

```

```

C:\Collector\Script2.0\laconfig.json - Notepad++ [Administrator]
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
new 4 run_me.ps1 new 5 Get-auditLogs.ps1 export_logs.ps1 Get-LabelData.ps1 laconfig.json
1 {
2   "EncryptedKeys": "True",
3   "AppClientId": "701af...a3d5f20",
4   "ClientSecretValue": "01000000d08c9ddf0115d1118c7a00c04fc297eb01
5   "TenantGUID": "ac1d...38d24f062c",
6   "TenantDomain": "kazdemos.org",
7   "LA_CustomerID": "bd285...9731c3",
8   "LA_SharedKey": "01000000d08c9ddf0115d1118c7a00c04fc297eb010000
9   "CertificateThumb": "01000000d08c9ddf0115d1118c7a00c04fc297eb010
10  "OnmicrosoftURL": "M3...36.onmicrosoft.com"
11 }
12

```

\*This action must be executed with the same account used to run the script

\*\*Use the key value



## SELECT TENANT TYPE

### SELECT YOUR TENANT TYPE BETWEEN ENTERPRISE, GCC, GCCH OR DOD

- SELECT TENANT TYPE WAS ADDED TO “LACONFIG.JSON” FILE UNDER “CLOUD” ATTRIBUTE

- THE POSSIBLE VALUES ARE:

- **COMMERCIAL** - COMMERCIAL CLOUD
- **GCC** - GOVERNMENT COMMUNITY CLOUD
- **GCCH** - GOVERNMENT COMMUNITY HIGH CLOUD
- **DOD** - DEPARTMENT OF DEFENSE CLOUD

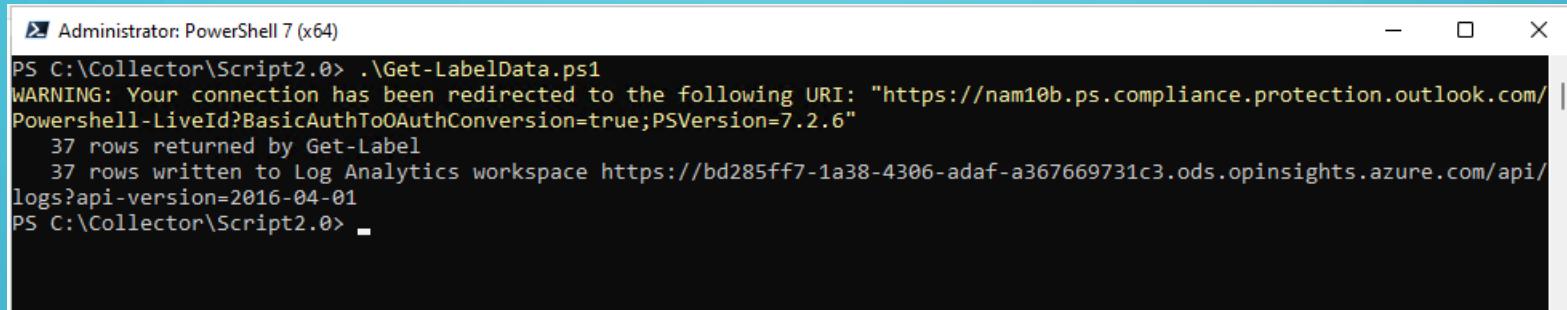


## Obtain your Label list with IDs and send to Logs Analytics

- Please complete first all this [steps](#), Office 365 Exchange Online API is used to obtain the next information
- The data stored on API schemas contains only the ID of the labels, for that reason is required have a Matrix between IDs and Display Names, the script used here take the information from Information Protection Service and Import the data in Logs Analytics.
- To execute this script please [complete these steps](#) 1<sup>st</sup>.
- Copy the script Get-LabelData.ps1 to the same folder where the laconfig.json file is located, validate that you have the last version of this file containing Certificate Thumbprint and Onmicrosoft URL
- If you have an error to execute, validate that the certificate was imported in the machine used to execute this script
- How to consume through Power BI, [click here](#).
- Understanding the complexity to give elevate privileges to an unattended script, a line can be modified on the script to a manual execution, in this case someone with Compliance Administrator role needs to execute this script on PowerShell, this execution is on-demand and is required only when you have modified or new labels, to up to date the reports with this information, or can be required depending the retention period in the Logs Analytics workspace.

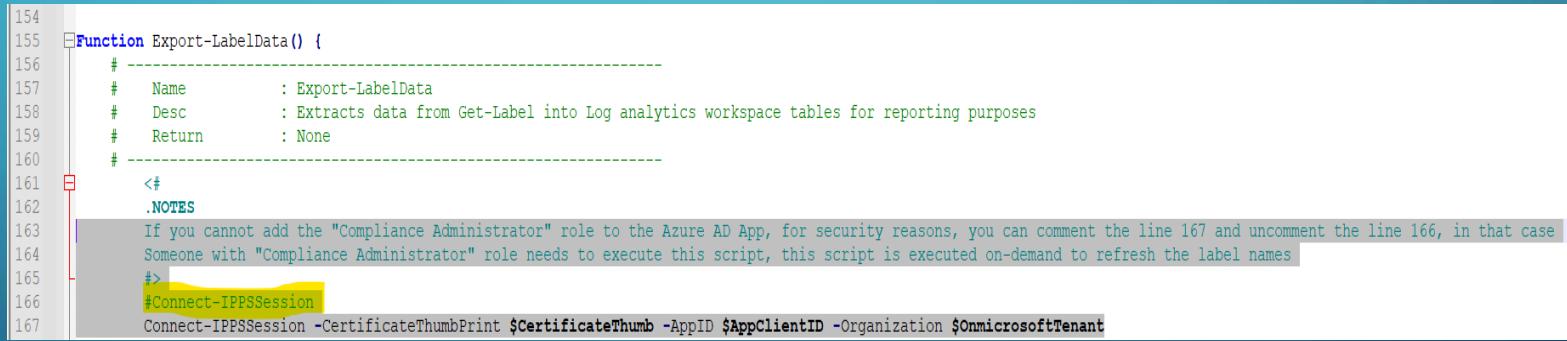
# WORKSTATION - POWERSHELL

## STEPS



Administrator: PowerShell 7 (x64)

```
PS C:\Collector\Script2.0> .\Get-LabelData.ps1
WARNING: Your connection has been redirected to the following URI: "https://nam10b.ps.compliance.protection.outlook.com/
PowerShell-LiveId?BasicAuthToOAuthConversion=true;PSVersion=7.2.6"
    37 rows returned by Get-Label
    37 rows written to Log Analytics workspace https://bd285ff7-1a38-4306-adaf-a367669731c3.ods.opinsights.azure.com/api/
logs?api-version=2016-04-01
PS C:\Collector\Script2.0>
```



```
154
155 Function Export-LabelData() {
156     #
157     #   Name      : Export-LabelData
158     #   Desc      : Extracts data from Get-Label into Log analytics workspace tables for reporting purposes
159     #   Return     : None
160     #
161     <#
162     .NOTES
163     If you cannot add the "Compliance Administrator" role to the Azure AD App, for security reasons, you can comment the line 167 and uncomment the line 166, in that case
164     Someone with "Compliance Administrator" role needs to execute this script, this script is executed on-demand to refresh the label names
165     #>
166     #Connect-IPPSSession
167     Connect-IPPSSession -CertificateThumbprint $CertificateThumb -AppID $AppClientID -Organization $OnmicrosoftTenant
```



## Create a Table in Logs Analytics with Product names and service plan identifiers for licensing

When user licensing data is exported, a special names are used and doesn't match with Known products, these steps helps to create a Matrix for that purpose.

- Download the latest updated product list from [here](#) and save the file in a known folder.

- Using the script ExportCSV2LA.ps1 you can import the data to Logs Analytics, to do that, execute the script in this way:

```
PS C:\Collector> .\ExportCSV2LA.ps1 -FileName ".\Product names and service plan identifiers  
for licensing.csv"-TableName "MSProducts"
```

- After executing the script can take between 5 to 10 minutes to display the New table under Logs Analytics
- How to consume through Power BI, [click here](#).

\*<https://learn.microsoft.com/en-us/azure/active-directory/enterprise-users/licensing-service-plan-reference>

\*\*Depending on the Logs Analytics workspace retention this script will be required to execute continuously(monthly for a 30 days retention period) to don't lose that information

\*\*\* Power BI templates use the table name called "MSProducts" any change to this name, need to make the change on Power BI queries



Microsoft

# IMPORT DATA TO LOGS ANALYTICS

## STEPS

```
Administrator: PowerShell 7 (x64)
PS C:\Collector\Script2.0> .\ExportCSV2LA.ps1 -FileName ".\Product names and service plan identifiers for licensing.csv"
-TableName "MSProducts"
Importing CSV file...
Calculating batch size...
Starting export to LA...
.
Export finished.
PS C:\Collector\Script2.0>
```

The screenshot shows the Microsoft Power BI Data Studio interface. On the left, there's a sidebar with 'New Query 1\*' selected, showing tables like 'MP-Reports', 'Select scope', 'Run' (disabled), 'Time range: Last 24 hours', 'Save', 'Share', 'New alert rule', 'Export', 'Pin to', and 'Format query'. Below that is a 'Tables' section with 'MSProducts\_CL' selected, a search bar, and filter/group buttons. The main area shows a table titled 'MSProducts\_CL' with the following data:

String_Id_s	GUID_g	Service_Plan_Name_s	Service_Plan_Id_g	Service_Plans_Included_Friendly_Names_s
ADV_COMM	e4654015-5daf-4a48-9b37-4f309dd88b	TEAMS_ADVCOMMS	604ec28a-ae18-4bc6-91b0-11da94504ba9	Microsoft 365 Advanced Communications
CDSAICAPACITY	d2dea78b-507c-4e56-b400-39447f4738f8	CDSAICAPACITY	a7c70a41-5e02-4271-93e6-d9b4184d83f5	AI Builder capacity add-on
CDSAICAPACITY	d2dea78b-507c-4e56-b400-39447f4738f8	EXCHANGE_S_FOUNDATION	113feb6c-3fe4-4440-bddc-54d774bf0318	Exchange Foundation
SPZA_IW	8f0c5670-4e56-4892-b06d-91c085d7004f	SPZA	0bf98ed-1dbc-4a97-b246-701754e48b17	APP CONNECT
SPZA_IW	8f0c5670-4e56-4892-b06d-91c085d7004f	EXCHANGE_S_FOUNDATION	113feb6c-3fe4-4440-bddc-54d774bf0318	EXCHANGE FOUNDATION
1g	MCOMEETADV	MCOMEETADV	3e26ee1f-8a5f-4d52-aee2-b81ce45c8f40	Microsoft 365 Audio Conferencing
	AAD_BASIC	AAD_BASIC	c4da7f8a-5ee2-4c99-a7e1-87d2df57f6fe	MICROSOFT AZURE ACTIVE DIRECTORY BASIC
P1	AAD_PREMIUM	AAD_PREMIUM	41781fb2-bc02-4b7c-bd55-b576c07bb09d	AZURE ACTIVE DIRECTORY PREMIUM P1



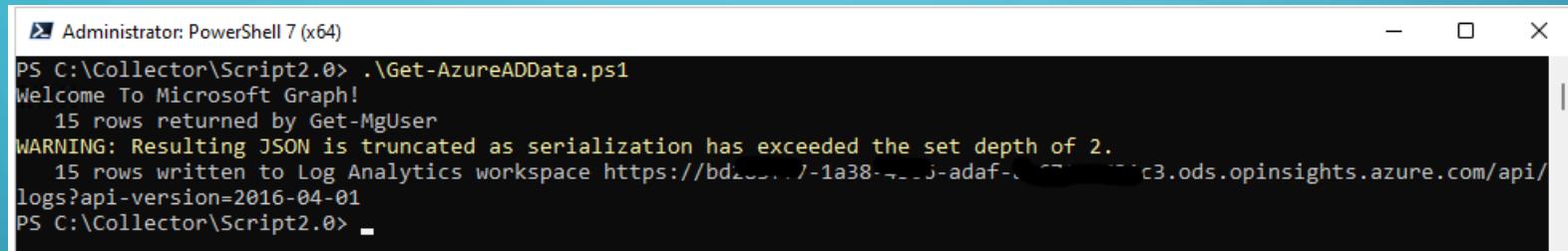
## LIST OF AZURE AD USERS WITH ATTRIBUTES AND LICENSING

- Please complete first all this [steps](#), Microsoft Graph API is used to obtain the next information
- To generate a list of Azure AD users with some attributes like Country, City, Department, Job Title, or Office Location, you need execute the script **Get-AzureADData.ps1** previously check that you are using the latest **laconfig.json** file that contain the attribute **CertificateThumb**
- Execute the script on PowerShell 7 and that is.

```
PS C:\Collector> .\Get-AzureADData.ps1
```

# POWERSHELL - AZURE AD USERS

STEPS



```
Administrator: PowerShell 7 (x64)
PS C:\Collector\Script2.0> .\Get-AzureADData.ps1
Welcome To Microsoft Graph!
    15 rows returned by Get-MgUser
WARNING: Resulting JSON is truncated as serialization has exceeded the set depth of 2.
    15 rows written to Log Analytics workspace https://bd.../1a38-...-adaf-...-c3.ods.opinsights.azure.com/api/logs?api-version=2016-04-01
PS C:\Collector\Script2.0>
```



## LIST OF AZURE AD ROLES AND DOMAINS REGISTERED

- Please complete first all this [steps](#), Microsoft Graph API is used to obtain the next information
- To generate a list of Azure AD Roles, and Domains registered on the Tenant, you need execute the script **Get-AzureADRoles.ps1** and **Get-AzureADDomains.ps1**, previously check that you are using the latest **laconfig.json** file that contain the attribute **CertificateThumb**
- Execute the script on PowerShell 7 and that is.

```
PS C:\Collector> .\Get-AzureADDomains.ps1
```

```
PS C:\Collector> .\Get-AzureADRoles.ps1
```

# POWERSHELL - AZURE AD USERS

## STEPS

```
Administrator: PowerShell 7 (x64)
PS C:\Collector\Sentinel> .\Get-AzureADRoles.ps1
Welcome To Microsoft Graph!
    17 rows returned by Get-MgDirectoryRole
Processing account Conditional Access Administrator 1/17
Processing account Privileged Role Administrator 2/17
Processing account Exchange Administrator 3/17
Processing account Cloud Device Administrator 4/17
Processing account Security Administrator 5/17
Processing account Global Reader 6/17
Processing account Intune Administrator 7/17
Processing account Security Reader 8/17
Processing account User Administrator 9/17
Processing account Reports Reader 10/17
Processing account Compliance Administrator 11/17
Processing account Azure Information Protection Administrator 12/17
Processing account Directory Readers 13/17
Processing account Global Administrator 14/17
Processing account Application Administrator 15/17
Processing account Azure AD Joined Device Local Administrator 16/17
Processing account Cloud App Security Administrator 17/17
    17 rows written to Log Analytics workspace https://f7e2ed[REDACTED]5d953.ods.opinsights.azure.com/api/logs?api-version=2016-04-01
PS C:\Collector\Sentinel> .\Get-AzureADDomains.ps1
Welcome To Microsoft Graph!
    3 rows returned by Get-MgDomains
Processing account M365x089236.onmicrosoft.com 1/3
Processing account kazdemos.org 2/3
Processing account ms-mparr.com 3/3
    3 rows written to Log Analytics workspace https://f7e2ed[REDACTED]5d953.ods.opinsights.azure.com/api/logs?api-version=2016-04-01
PS C:\Collector\Sentinel>
```



## Connect to logs stored in your workspace

- Open <https://portal.azure.com>
- Look for Log Analytics workspaces
  - Select your workspace pressing over the name
  - On the new blade under General category press Logs menu
  - Close the pop-up windows
  - Look for Custom Logs under Tables and press the arrow to show all the tables
  - If the script is running well and the Tenant have a normal use these 5 tables(based on the principal script) will be appear\* under Custom Logs:
    - **AuditAzureActiveDirectory\_CL**
    - **AuditExchange\_CL**
    - **AuditGeneral\_CL**
    - **AuditSharePoint\_CL**
    - **DLPALL\_CL**
  - Other tables are added from other scripts:
    - **AzureADUsers\_CL**
    - **AzureADRoles\_CL**
    - **AzureADDomains\_CL**
    - **Labels\_CL**
    - **MSProducts\_CL**
    - **RMSData\_CL**
    - **RMSDATADetails\_CL**

\*The first time that the script is running the tables can take between 5 to 15 minutes to appear under Custom Logs, if as an example no DLP rules are applied or Exchange is not used the tables will not appear until an operation is recorded.



## Export to use from Power BI

The next steps are required to do for each table (5):

- Double click in Table name, that action will be put the name of the table on query space
- Press Run button (is not require make changes over Time range)
- After pressing the query will be running and some results will be appeared at result area.
- The Export option will be available, press and select Export to Power BI (M query)
- The previous step will generate a txt file, save and rename the file in a know location using the table name as a filename.
- Under the same query, below Table name add the next query:

```
| where TimeGenerated > now(-730d)  
| summarize by  
    Year = datetime_part('Year',TimeGenerated),  
    Month = datetime_part('Month',TimeGenerated),  
    Day = datetime_part('Day',TimeGenerated)
```

- The previous will be used to resolve a download limit on Power BI, more information in this [link](#)
- Press Run button and Export to Power BI (M query), same as previous step, and add the word “date” as a prefix for filename.
- The results will be show years and months for the last 730 days of the data collected, in this case maybe only 1 year and one month.

## Steps

The screenshot shows the Azure portal search results for the query "log Analytics". The search bar at the top contains the text "log Analytics". Below the search bar, there are several filter tabs: All (selected), Services (17), Marketplace (7), Documentation (20), Resources (0), and Resource Groups (0). The main search results are categorized under "Services", "Marketplace", and "Documentation".

**Services**

- Log Analytics query packs
- Log Analytics workspaces (highlighted)
- Activity log
- Stream Analytics clusters

**Marketplace**

- Log Analytics Workspace
- Azure Log Analytics Agent Health
- FortiAnalyzer Centralized Log Analytics
- HPE OneView for Azure Log Analytics (v1.4.0)
- Logz.io - Cloud Monitoring and Observability
- Cloud-Native Observability with Logz.io (LEGACY)
- SEEPATH-managed-azure

**Documentation**

- Overview of Log Analytics in Azure Monitor - Azure Monitor
- Create Log Analytics workspaces - Azure Monitor | Microsoft Docs

## Steps

Home >

## Log Analytics workspaces

mscompliance

+ Create Open recycle bin Manage view Refresh Export to CSV Open query Assign tags

Sentinel Subscription equals all Resource group equals all Location equals all Add filter

No grouping List view

Name ↑	Resource group ↑↓	Location ↑↓	Subscription ↑↓
<input type="checkbox"/> Sentinel	LAB	East US	Azure subscription 1

...

# LOGS ANALYTICS WORKSPACE

## Steps

The screenshot shows the Azure Log Analytics workspace configuration interface. On the left, there's a navigation pane with options like 'Create', 'Open recycle bin', and a search bar. The main area displays the 'Sentinel' Log Analytics workspace details. It includes sections for 'Overview', 'Activity log', 'Access control (IAM)', 'Tags', 'Diagnose and solve problems', and 'Settings'. Under 'Settings', there are links for 'Locks', 'Agents management', 'Legacy agents management', 'Custom logs', 'Computer Groups', 'Data export', 'Linked storage accounts', 'Network isolation', 'Tables', 'General', 'Logs', and 'Solutions'. A yellow box highlights the 'Logs' section. The right side of the screen shows the workspace details with a note about migrating to Azure Monitor Agent. It lists the resource group (lab), status (Active), location (East US), subscription (Azure subscription 1), subscription ID (1c211111-1111-1111-1111-111111111111), and tags (Click here to add tags). Below this, there's a 'Get started with Log Analytics' section, a 'Connect a data source' section with options for Azure virtual machines (VMs), Windows and Linux Agents management, Storage account log, and System Center Operations Manager, and a 'Configure monitoring solutions' section with a 'View solutions' link. At the bottom, there's a 'Maximize your Log Analytics experience' section.

Home > Log Analytics workspaces >

**Log Analytics work...** mscompliance

+ Create Open recycle bin ...

**Sentinel**

Name ↑↓

**Sentinel** ...

SENTINEL

Search

Overview

Activity log

Access control (IAM)

Tags

Diagnose and solve problems

Settings

Locks

Agents management

Legacy agents management

Custom logs

Computer Groups

Data export

Linked storage accounts

Network isolation

Tables

**General**

Workspace summary

Workbooks

**Logs**

Solutions

**Sentinel** Log Analytics workspace

Search

Delete

The Log Analytics agents (MMA.OMS) used to collect logs from virtual machines and servers will no longer be supported after July 1, 2020. [migrating to Azure Monitor Agent](#)

**Essentials**

Resource group ([move](#)) : lab

Status : Active

Location : East US

Subscription ([move](#)) : [Azure subscription 1](#)

Subscription ID : 1c211111-1111-1111-1111-111111111111

Tags ([edit](#)) : [Click here to add tags](#)

**Get started with Log Analytics**

Log Analytics collects data from a variety of sources and uses a powerful query language to give you insights into the operation of your applications and resources. Use Azure Monitor to access the complete set of tools for monitoring all of your Azure resources.

**1 Connect a data source**

Select one or more data sources to connect to the workspace

Azure virtual machines (VMs)

Windows and Linux Agents management

Storage account log

System Center Operations Manager

**2 Configure monitoring solutions**

Add monitoring solutions that provide insights for applications and services in your environment

[View solutions](#)

Maximize your Log Analytics experience



# LOGS ANALYTICS WORKSPACE

## Steps

The screenshot shows the Microsoft Sentinel Log Analytics workspace interface. On the left, the navigation pane includes sections for Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Settings (Locks, Agents management, Legacy agents management, Custom logs, Computer Groups, Data export, Linked storage accounts, Network isolation), Tables, General (Workspace summary, Workbooks, Logs, Solutions, Usage and estimated costs, Properties, Service Map), and a New Query 1 tab. The 'Logs' item under General is highlighted with a yellow box. The main area displays a query editor titled 'New Query 1' with a search bar, filter options, and a list of tables: Sentinel, LogManagement, Microsoft Sentinel, and Custom Logs. The 'LogManagement' table is expanded, showing sub-tables like AuditAzureActiveDirectory\_CL, AuditExchange\_CL, AuditGeneral\_CL, AuditSharePoint\_CL, AzureADDomains\_CL, AzureADRoles\_CL, AzureADUsers\_CL, DLPAll\_CL, Labels\_CL, MSProducts\_CL, RMSData\_CL, and RMSDataDetails\_CL. Below the query editor is a 'Queries History' section listing recent queries with their execution details and 'Run' buttons:

- DLPAll\_CL | where TimeGenerated > now(-730d) | summarize dcount(Operation\_s) by Operation\_s  
1/30/2023, 1:23 PM | 4 results
- AuditSharePoint\_CL | where TimeGenerated > now(-730d) | summarize dcount(Operation\_s) by Operation\_s  
1/30/2023, 1:22 PM | 47 results
- AuditGeneral\_CL | where TimeGenerated > now(-730d) | summarize dcount(Operation\_s) by Operation\_s  
1/30/2023, 1:21 PM | 95 results
- AuditExchange\_CL | where TimeGenerated > now(-730d) | summarize dcount(Operation\_s) by Operation\_s  
1/30/2023, 1:20 PM | 33 results
- AuditAzureActiveDirectory\_CL | where TimeGenerated > now(-730d) | summarize dcount(Operation\_s) by Operation\_s  
1/30/2023, 1:14 PM | 38 results
- Labels\_CL  
1/27/2023, 3:38 PM | 422 results



## Steps

The screenshot shows the Microsoft Log Analytics workspace interface. At the top, there is a navigation bar with tabs for 'New Query 1\*', 'MCAR-Kazdemos', 'Select scope', 'Run' (highlighted in yellow), 'Time range : Last 24 hours', 'Save', 'Share', 'New alert rule', 'Export' (highlighted in yellow), 'Pin to', and more. Below the navigation bar, there are tabs for 'Tables', 'Queries', 'Functions', and '...'. A search bar and filter/group by options are also present. On the left, there is a 'Favorites' section with a note about adding favorites and a tree view of logs. The tree view includes 'LogManagement', 'Custom Logs' (expanded), and several log types like 'AuditAzureActiveDirectory\_CL' (highlighted in yellow), 'AuditExchange\_CL', 'AuditGeneral\_CL', 'AuditSharePoint\_CL', and 'DLPAII\_CL'. The main area displays a table titled 'Results' with columns: TimeGenerated [UTC], Id\_g, Operation\_s, OrganizationId\_g, and RecordType\_d. The table contains five rows of audit log data. An 'Export' dropdown menu is open on the right, showing options: 'Export to CSV - all columns', 'Export to CSV - displayed columns' (highlighted in yellow), 'Export to Power BI (M query)' (highlighted in yellow), and 'Open in Excel'.

TimeGenerated [UTC]	Id_g	Operation_s	OrganizationId_g	RecordType_d
8/22/2022, 5:55:11.000 PM	cb1c1e...	134f8200	UserLoginFailed	a...
8/22/2022, 8:52:32.000 PM	b168bc...	1831ba00	UserLoginFailed	a...
8/22/2022, 8:52:20.000 PM	c49e...	7203b400	UserLoginFailed	a...
8/22/2022, 5:55:15.000 PM	cb1c...	134f8200	UserLoggedIn	a...
8/22/2022, 8:52:20.000 PM	c49e...	03b400	UserLoginFailed	a...

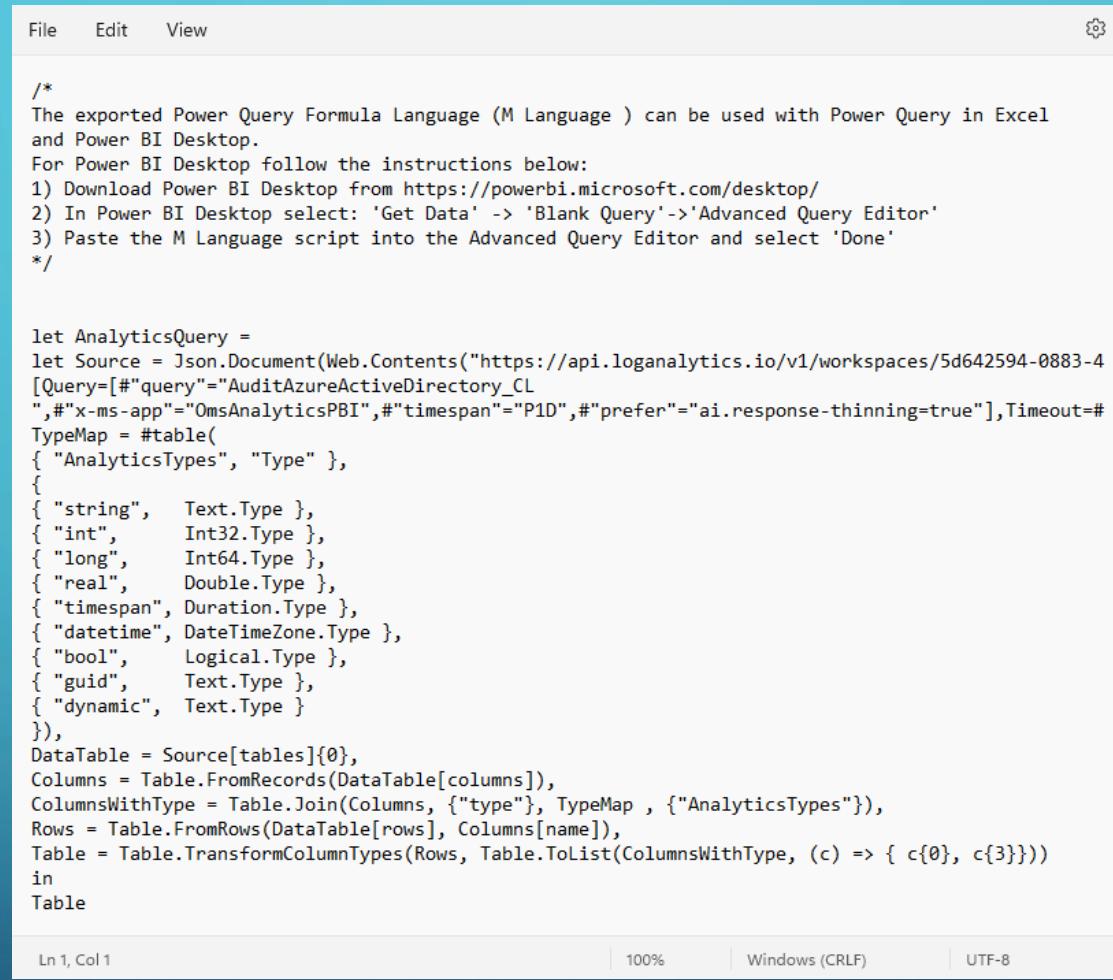
## Steps

The screenshot shows the Microsoft Log Analytics workspace interface. On the left, there's a sidebar with sections for Favorites, LogManagement, and Custom Logs, each containing a list of log types. The main area is a query editor titled "New Query 1\*". The query itself is:

```
1 AuditAzureActiveDirectory_CL  
2 | where TimeGenerated > now(-730d)  
3 | summarize by  
4 | Year = datetime_part('Year',TimeGenerated),  
5 | Month = datetime_part('Month',TimeGenerated)
```

The results pane shows a table with two columns: "Year" and "Month". The data is: Year: > 2,022, Month: 8. A dropdown menu under the "Export" button is open, showing options for exporting to CSV or Power BI. The "Export to Power BI (M query)" option is highlighted.

## Export example



The screenshot shows a code editor window with the following content:

```
/*
The exported Power Query Formula Language (M Language ) can be used with Power Query in Excel
and Power BI Desktop.
For Power BI Desktop follow the instructions below:
1) Download Power BI Desktop from https://powerbi.microsoft.com/desktop/
2) In Power BI Desktop select: 'Get Data' -> 'Blank Query'->'Advanced Query Editor'
3) Paste the M Language script into the Advanced Query Editor and select 'Done'
*/

let AnalyticsQuery =
let Source = Json.Document(Web.Contents("https://api.loganalytics.io/v1/workspaces/5d642594-0883-4
[Query=[#"query"="AuditAzureActiveDirectory_CL
",#"x-ms-app"="OmsAnalyticsPBI",#"timespan"="P1D",#"prefer"="ai.response-thinning=true"],Timeout=#
TypeMap = #table(
{ "AnalyticsTypes", "Type" },
{
{ "string", Text.Type },
{ "int", Int32.Type },
{ "long", Int64.Type },
{ "real", Double.Type },
{ "timespan", Duration.Type },
{ "datetime", DateTimeZone.Type },
{ "bool", Logical.Type },
{ "guid", Text.Type },
{ "dynamic", Text.Type }
}),
DataTable = Source[tables]{0},
Columns = Table.FromRecords(DataTable[columns]),
ColumnsWithType = Table.Join(Columns, {"type"}, TypeMap , {"AnalyticsTypes"}),
Rows = Table.FromRows(DataTable[rows], Columns[name]),
Table = Table.TransformColumnTypes(Rows, Table.ToList(ColumnsWithType, (c) => { c{0}, c{3}}))
in
Table

Ln 1, Col 1 | 100% | Windows (CRLF) | UTF-8
```

## TO OBTAIN THE LATEST INFORMATION ABOUT LABELS

- Logs Analytics works appending data, for that reason every time that the script is executed to update the information, that data is added, in that order of ideas is important to download the most recently information, to do that execute this query:

```
Labels_CL  
| where TimeGenerated > now(-730d)  
| project  
    DisplayName_s,  
    Guid_g,  
    Priority_d,  
    ParentLabelDisplayName_s,  
    TimeGenerated  
| summarize max(TimeGenerated) by Guid_g, DisplayName_s, Priority_d, ParentLabelDisplayName_s
```

- Then export as Power BI(M query) and add to the rest of queries under Power BI Desktop

## TO OBTAIN THE LATEST INFORMATION ABOUT SERVICES PLAN AND PRODUCT NAMES

- Logs Analytics works appending data, for that reason every time that the script is executed to update the information, that data is added, in that order of ideas is important to download the most recently information, to do that execute this query:

```
MSProducts_CL  
| where TimeGenerated > now(-730d)  
| project  
    Product_Display_Name_s,  
    GUID_g,  
    String_Id_s,  
    TimeGenerated  
|summarize max(TimeGenerated) by GUID_g, Product_Display_Name_s, String_Id_s
```

- Then export as Power BI(M query) and add to the rest of queries under Power BI Desktop

## TO OBTAIN THE LATEST INFORMATION ABOUT AZURE AD USERS AND LICENSING

Logs Analytics works appending data, for that reason every time that the script is executed to update the information, that data is added, in that order of ideas is important to download the most recently information, to do that execute this query:

```
AzureADInfo_CL  
| where TimeGenerated > now(-730d) and UserPrincipalName_s != ""  
| project  
    UserPrincipalName_s,  
    DisplayName_s,  
    AssignedLicenses_s,  
    Country_s,  
    City_s,  
    JobTitle_s,  
    Department_s,  
    Mail_s,  
    OfficeLocation_s,  
    TimeGenerated  
| summarize max(TimeGenerated) by UserPrincipalName_s, AssignedLicenses_s, City_s, Country_s, Department_s,  
    DisplayName_s, JobTitle_s, Mail_s, OfficeLocation_s
```

Then export as Power BI(M query) and add to the rest of queries under Power BI Desktop

## Connect to logs stored in your workspace\*

1. Open Power BI desktop
2. Close the initial welcome pop-up
3. Go to Get data and select Blank query
4. Under Power Query Editor window select Advanced Editor
5. Clear the example, open “your\_table\_name.txt” copy and paste all the information and press Done
6. Credentials can be request, open and select Organizational account, user with read permissions over Logs Analytics workspace are required
7. On the same window select New Source and press Blank Query again
8. Click on Advanced Editor and clear the sample
9. Open “date\_your\_table\_name.txt” copy and paste all the information and press Done
10. At top right under PROPERTIES rename the query with your Table name

\*Power BI templates are delivered with the script, these templates contains all the steps explained in this section, using the templates the next steps are not required



## Power BI limit resolved

- 11. At left right click over first query and select Create Function
- 12. A pop-up appears, press Create
- 13. Assign a name, like as Data\_YourTableName and press OK
- 14. Go to Advanced Editor and click it
- 15. A pop-up appears, press OK
- 16. Modify line 2 adding inside Source this additional data

Source = (Month as text, Year as text) => let AnalyticsQuery =

- 17. Looking on the same editor the variable # "timespan" = "P1D", and replace P1D with this:

AuditAzureActiveDirectory\_CL | where datetime\_part('Month', TimeGenerated) == "& Month &" and  
datetime\_part('Year', TimeGenerated) == "& Year &"

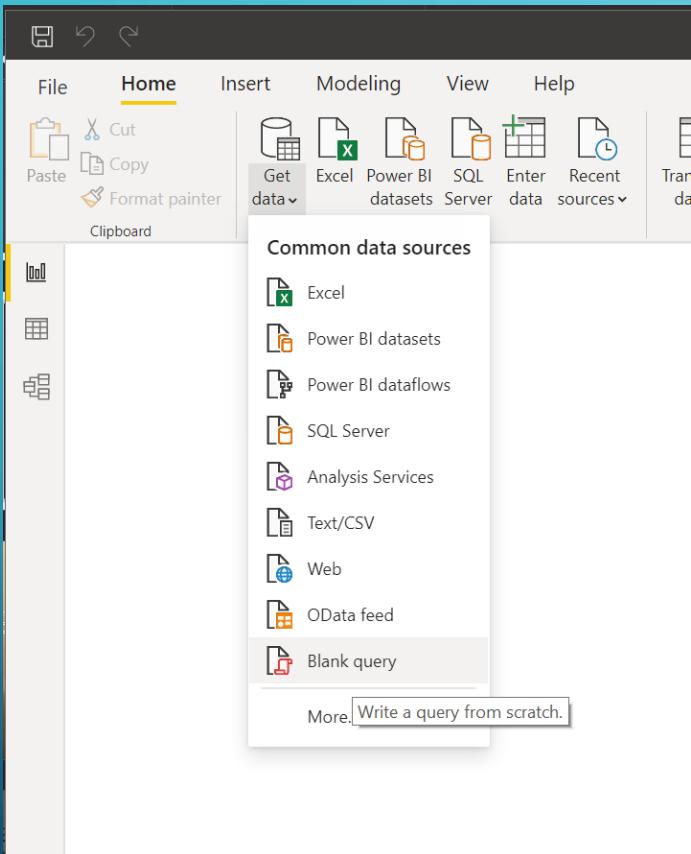
- 18. Take care on previous step to use the correct Table name and press Done
- 19. First query can be deleted, only the Date query and the function are required
- 20. Repeat from step 3 pressing New Source and Blank Query for each Table name, at the final you will have 5 queries and 5 functions.

## Power BI limit resolved

21. After finish the previous steps with the 5 tables is required merge the queries with the functions, to do that please go forward with the next steps taking care in each point
22. Select your query (the one with year and month as result), at the top select the tab Add Column and then press Invoke Custom Function
23. On the pop-up don't touch New column name under Function query select the function related to your Table name; after that 2 text areas will be appear fill Month with Month and Year with Year and press OK
24. A new column will be added with the capability to expand press the icon with 2 arrows one to right and the other to left located just right to column name
25. In new window all columns related to the original tab  will be appear (sometimes appear at bottom an option to load more columns, in that case please press that option) uncheck Use original name as prefix
26. Now all the columns will be added based on the date, that step resolve the 500k limit on Power BI, if the data is too many a field Day can be added to the Date query.
27. Is important go to TimeGenerated column and change the Data Type to Date/Time/Timezone this step is relevant to apply filters based on time

# POWER BI

## Steps



The screenshot shows the Power Query Editor window titled 'Untitled - Power Query Editor'. The 'Advanced Editor' button in the ribbon is circled in red. The 'Advanced Editor' dialog box is open, showing the M code for 'Query1':  
let  
 Source = ""  
in  
 Source

## Steps

Advanced Editor

## Query1

Display Options ?

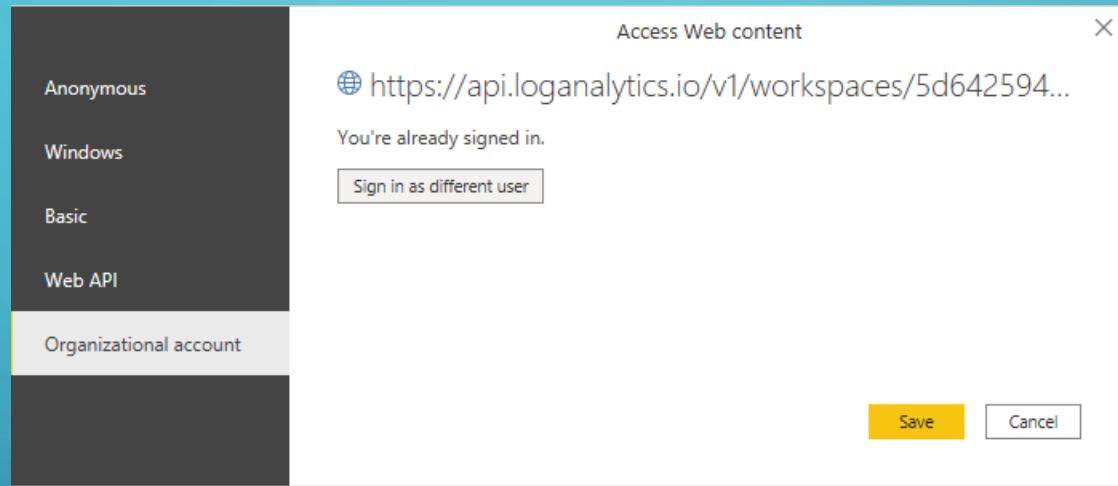
```
1  /*
2  The exported Power Query Formula Language (M Language ) can be used with Power Query in Excel
3  and Power BI Desktop.
4  For Power BI Desktop follow the instructions below:
5  1) Download Power BI Desktop from https://powerbi.microsoft.com/desktop/
6  2) In Power BI Desktop select: 'Get Data' -> 'Blank Query'->'Advanced Query Editor'
7  3) Paste the M Language script into the Advanced Query Editor and select 'Done'
8 */
9
10
11 let AnalyticsQuery =
12 let Source = Json.Document(Web.Contents("https://api.loganalytics.io/v1/workspaces/5d6425-...2d/query",
13 [Query=[#"query"="AuditAzureActiveDirectory_CL
14 ",#"x-ms-app"="OmsAnalyticsPBI",#"timespan"="P1D",#"prefer"="ai.response-thinning=true"],Timeout=#duration(0,0,4,0)])),
15 TypeMap = #table(
16 { "AnalyticsTypes", "Type" },
17 {
18 { "string", Text.Type },
19 { "int", Int32.Type },
20 { "long", Int64.Type },
21 { "real", Double.Type },
22 { "timespan", Duration.Type },
23 { "datetime", DateTimeZone.Type },
24 { "bool", Logical.Type },
25 { "guid", Text.Type },
26 { "dynamic", Text.Type }
27 }),
28 DataTable = Source[tables]{0},
29 Columns = Table.FromRecords(DataTable[columns]),
30 ColumnsWithType = Table.Join(Columns, { "type" }, TypeMap , {"AnalyticsTypes"}),
31 Rows = Table.FromRows(DataTable[rows], Columns[name]),
32 Table = Table.TransformColumnTypes(Rows, Table.ToList(ColumnsWithType, (c) => { c{0}, c{3} } ))
33 in
34 Table
35 in AnalyticsQuery
```

✓ No syntax errors have been detected.

Done Cancel

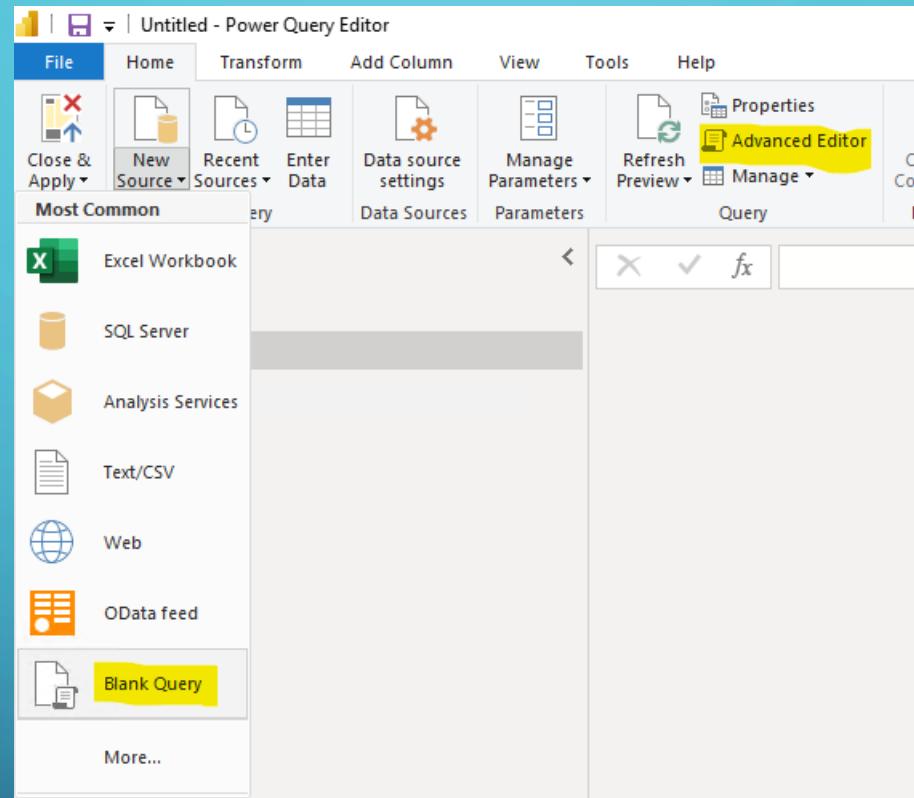
# POWER BI

## Steps



# POWER BI

## Steps



## Steps

Advanced Editor

## Query2

```
6 2) In Power BI Desktop select: 'Get Data' -> 'Blank Query'->'Advanced Query Editor'
7 3) Paste the M Language script into the Advanced Query Editor and select 'Done'
8 */
9
10
11 let AnalyticsQuery =
12 let Source = Json.Document(Web.Contents("https://api.loganalytics.io/v1/workspaces/5d64104c-xxxx-xxxx-xxxx-?d/query",
13 [Query=[#"query"]="AuditAzureActiveDirectory_CL
14 | where TimeGenerated > now(-730d)
15 | summarize by
16     Year = datetime_part('Year',TimeGenerated),
17     Month = datetime_part('Month',TimeGenerated)
18 ",#"x-ms-app"="OmsAnalyticsPBI",#"prefer"="ai.response-thinning=true"],Timeout=#duration(0,0,4,0))),
19 TypeMap = #table(
20 { "AnalyticsTypes", "Type" },
21 {
22 { "string",   Text.Type },
23 { "int",      Int32.Type },
24 { "long",     Int64.Type },
25 { "real",     Double.Type },
26 { "timespan", Duration.Type },
27 { "datetime", DateTimeZone.Type },
28 { "bool",     Logical.Type },
29 { "guid",     Text.Type },
30 { "dynamic",  Text.Type }
31 }),
32 DataTable = Source[tables]{0},
33 Columns = Table.FromRecords(DataTable[column]),
34 ColumnsWithType = Table.Join(Columns, {"type"}, TypeMap , {"AnalyticsTypes"}),
35 Rows = Table.FromRows(DataTable[rows], Columns[name]),
36 Table = Table.TransformColumnTypes(Rows, Table.ToList(ColumnsWithType, (c) => { c{0}, c{3}}))
37 in
38 Table
39 in AnalyticsQuery
```

Display Options ?

✓ No syntax errors have been detected.

Done Cancel

# POWER BI

## Steps

The screenshot shows the Microsoft Power Query Editor interface. The title bar reads "Untitled - Power Query Editor". The ribbon menu includes File, Home, Transform, Add Column, View, Tools, and Help. The Home tab is selected.

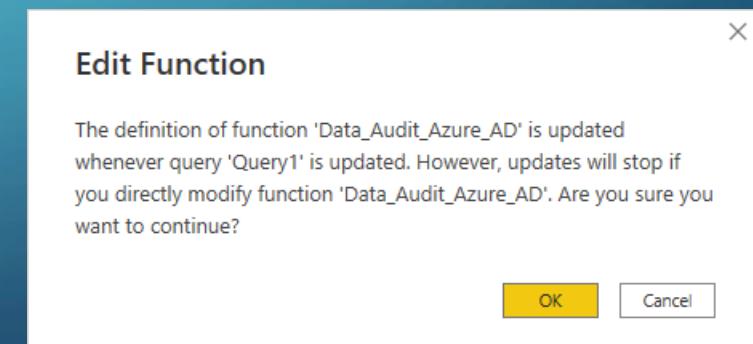
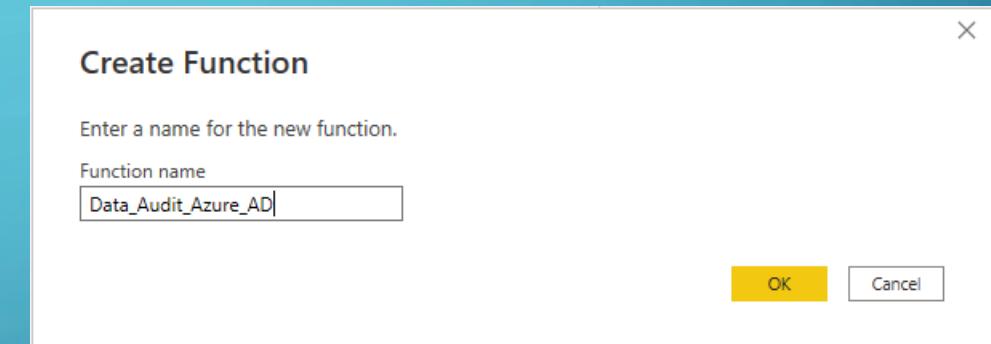
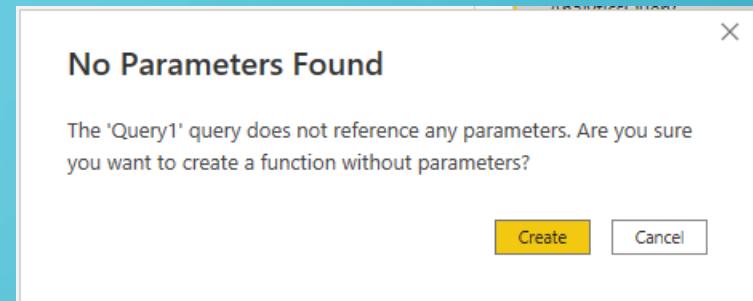
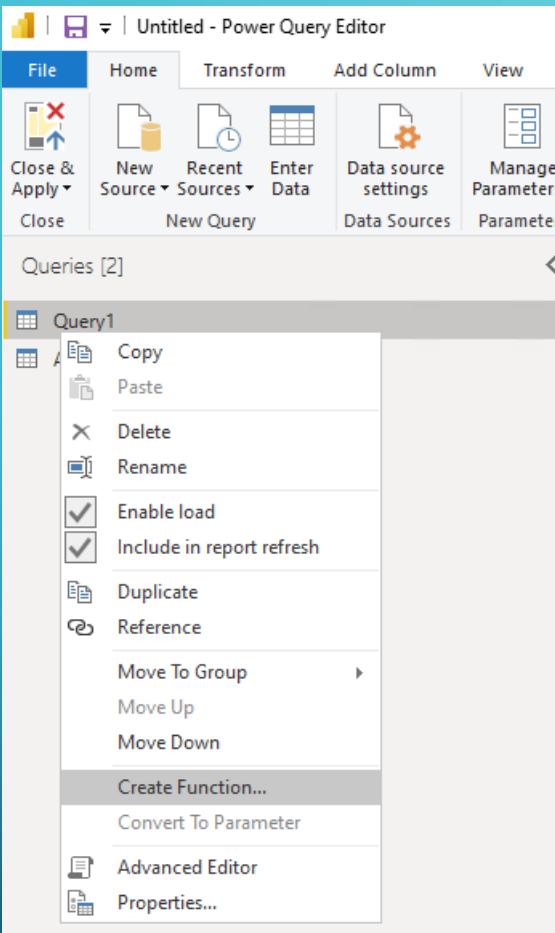
The main area displays a query titled "Audit Azure Active Directory". The formula bar shows the M code: `= let Source = Json.Document(Web.Contents("https://"))`. Below the formula bar, there is a preview pane showing a table with three columns: "Year" (containing "123"), "Month" (containing "123"), and "2022".

The ribbon toolbar contains various icons for managing queries, such as Close & Apply, New Source, Refresh, Manage Columns, Sort, and Transform.

The right side of the screen features the "Query Settings" pane, which includes sections for "PROPERTIES" (Name: Audit Azure Active Directory) and "APPLIED STEPS" (AnalyticsQuery).

# POWER BI

## Steps



## Steps

Advanced Editor

### Data\_Audit\_Azure\_AD

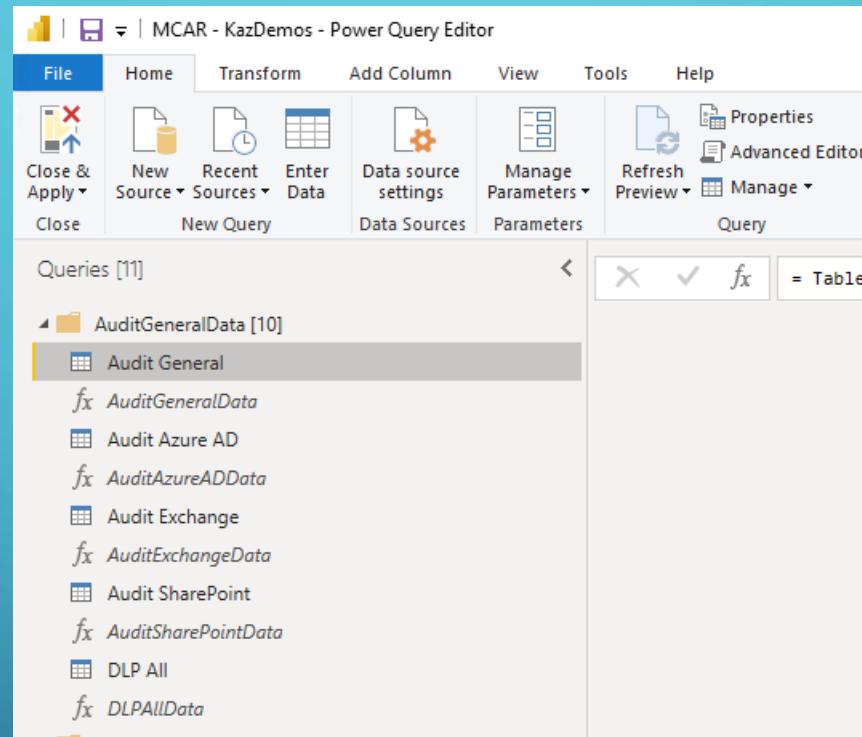
Display Options ?

```
1 let
2   Source = (Month as text, Year as text) => let AnalyticsQuery =
3     let Source = Json.Document(Web.Contents("https://api.loganalytics.io/v1/workspaces/5d64
4       [Query="#query"="AuditAzureActiveDirectory_CL
5       ",#x-ms-app="OmsAnalyticsPBI",#timespan="AuditAzureActiveDirectory_CL | where datetime_part('Month',TimeGenerated) == "& Month &" and datetime_part('Year'
6       TypeMap = #table(
7         { "AnalyticsTypes", "Type" },
8         {
9           { "string", Text.Type },
10          { "int", Int32.Type },
11          { "long", Int64.Type },
12          { "real", Double.Type },
13          { "timespan", Duration.Type },
14          { "datetime", DateTimeZone.Type },
15          { "bool", Logical.Type },
16          { "guid", Text.Type },
17          { "dynamic", Text.Type }
18        }),
19        DataTable = Source[tables]{0},
20        Columns = Table.FromRecords(DataTable[columns]),
21        ColumnsWithType = Table.Join(Columns, {"type"}, TypeMap , {"AnalyticsTypes"}),
22        Rows = Table.FromRows(DataTable[rows], Columns[name]),
23        Table = Table.TransformColumnTypes(Rows, Table.ToList(ColumnsWithType, (c) => { c{0}, c{3}}))
24      in
25      Table
26      in AnalyticsQuery
```

✓ No syntax errors have been detected.

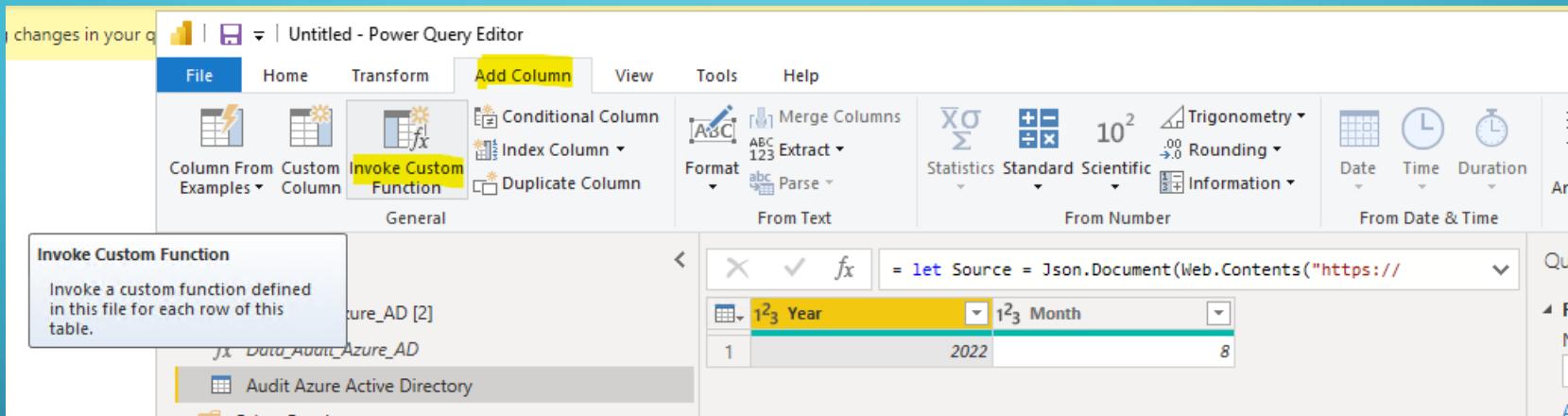
Done Cancel

## Steps

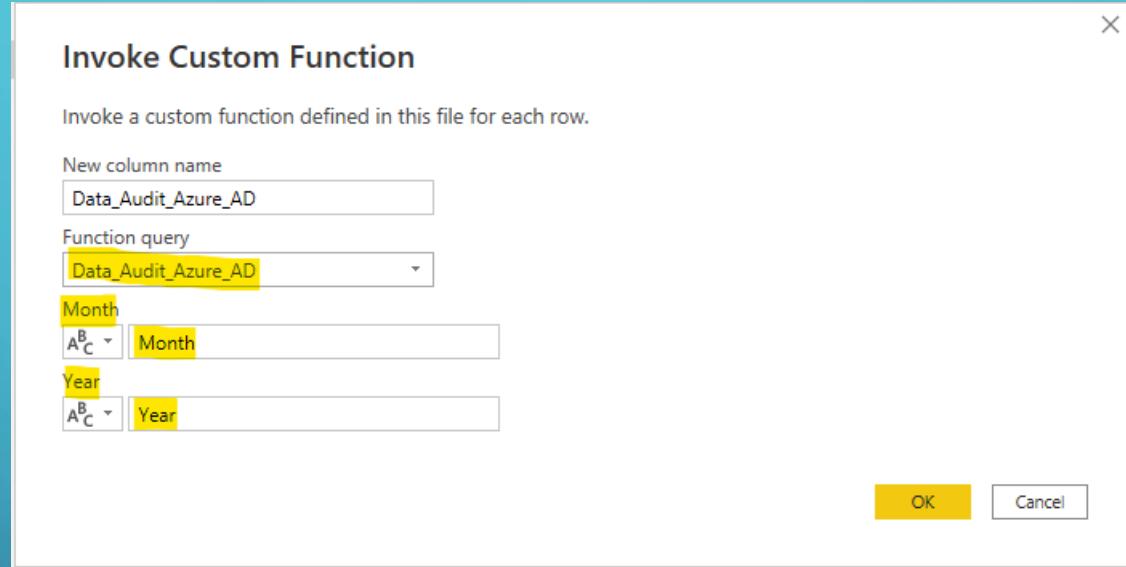


# POWER BI

## Steps

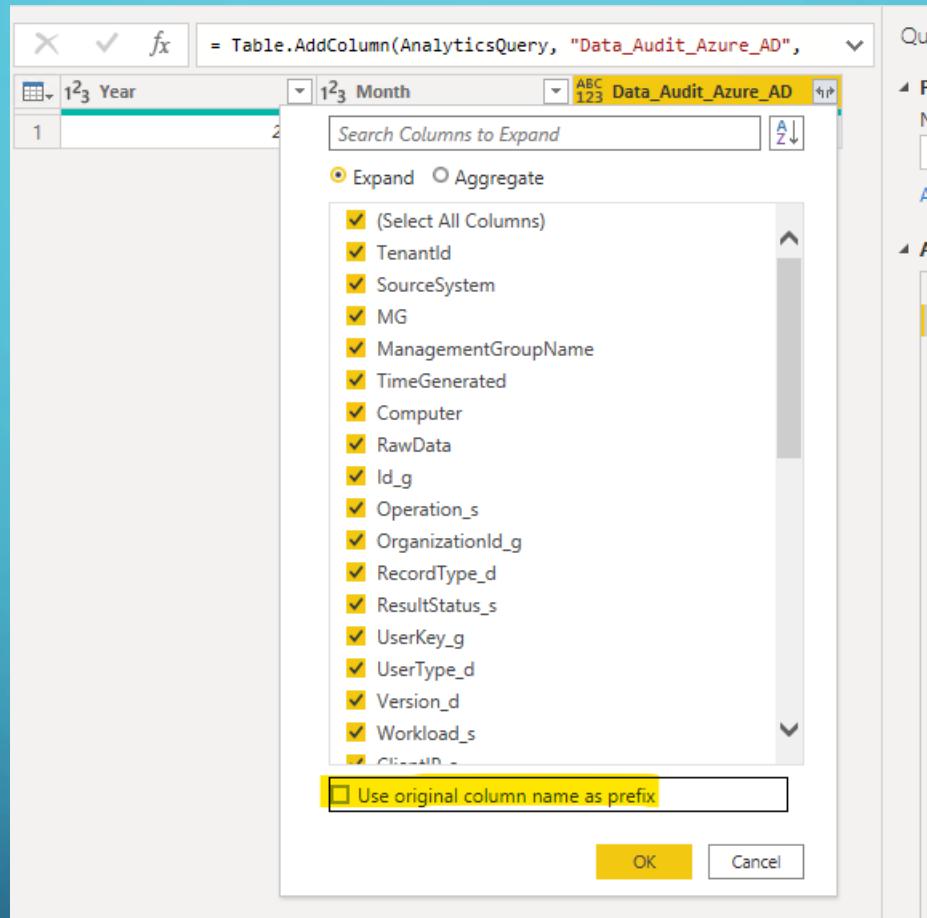


## Steps



# POWER BI

## Steps

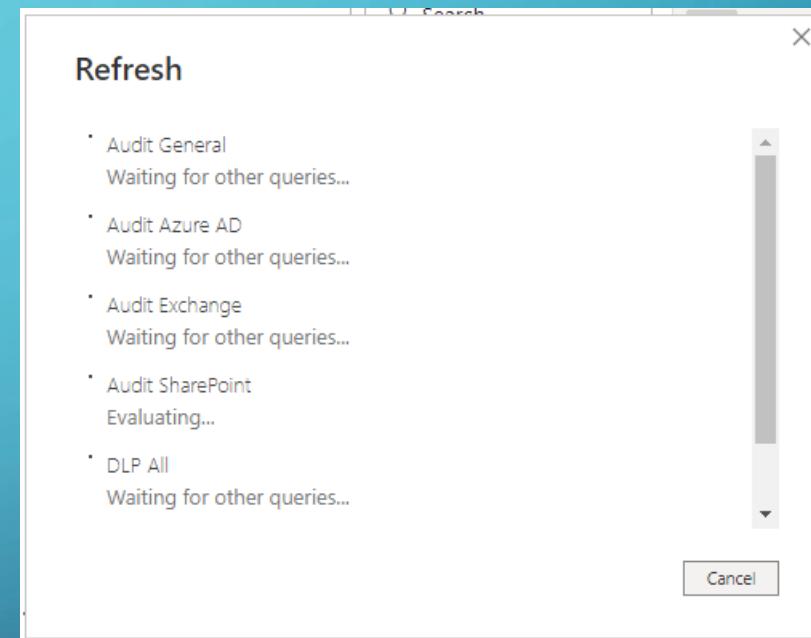
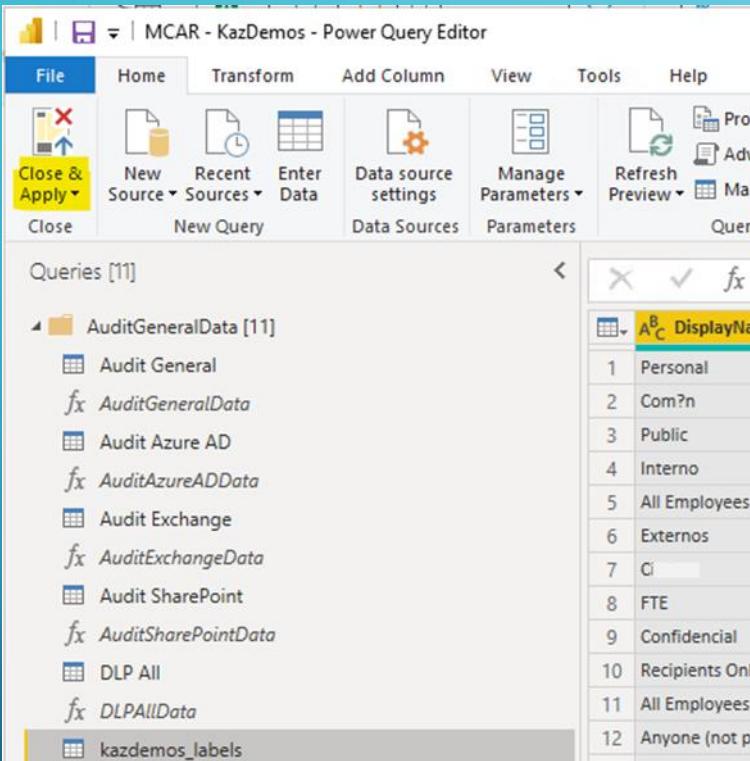


## Steps

The screenshot shows the Microsoft Power BI desktop interface. The ribbon at the top has the 'Home' tab selected. On the left, the 'Queries [2]' pane shows two queries: 'Data\_Audit\_Azure\_AD [2]' and 'Audit Azure Active Directory'. The main area displays a table with three rows and four columns. The first column contains row numbers 1, 2, and 3. The second column is empty. The third column is labeled 'ManagementGroupNa...' and contains the value 'ABC 123'. The fourth column is labeled 'TimeGenerated' and contains three timestamp values: '8/20/2022 7:45:23 AM +00:00', '8/20/2022 7:45:23 AM +00:00', and '8/20/2022 10:56:27 AM +00:00'. A context menu is open over the 'TimeGenerated' column, specifically over the third row's cell. The menu is titled 'Data Type: Any' and includes the following options: Decimal Number, Fixed decimal number, Whole Number, Percentage, Date/Time, Date, Time, Date/Time/Timezone (which is highlighted in yellow), Duration, Text, True/False, and Binary.

		ManagementGroupNa...	TimeGenerated
1		ABC 123	8/20/2022 7:45:23 AM +00:00
2		ABC 123	8/20/2022 7:45:23 AM +00:00
3		ABC 123	8/20/2022 10:56:27 AM +00:00

## Steps



# POWER BI

## Steps

The screenshot shows the Power BI Desktop interface with the following details:

- Top Bar:** MCAR - KazDemos - Power BI Desktop, Search, Sebastian Zamorano.
- Home Tab:** Selected.
- Data Section:** Get data, Excel, Data, SQL Server, Enter data, Data source types, Transform Refresh data, New visual, Text box, More visuals, New measure, Quick measure, Sensitivity, Publish.
- Visuals Section:** Clipboard, Count of Userid\_s by Userid\_s (a pie chart divided into 10 equal segments, each labeled "1 (10%)").
- Fields Section:** Fields pane showing a list of fields: Certificate, [...@support.on...](#), mike.wazowski@kazdemos.org, Not Available, randall.boggs@kazdemos.org, sebastian.zamorano.adm@kazdemos.org, ServicePrincipal 1b4d105e-6b01-4a..., ServicePrincipal 4eceacb2-6c0a-478..., ServicePrincipal 697b1d8d-1c87-4a..., ServicePrincipal ed05063e-6d20-44...).
- Filters Section:** A context menu is open over the pie chart, titled "Filters". It includes sections for "Filters on this page" (Add data fields here), "Filters on all pages" (Add data fields here), "Values" (Add data fields here), "Drill through" (Off), "Cross-report" (Off), "Keep all filters" (On), and "Add drill-through fields here".
- Page Navigation:** Page 1 of 1.
- Bottom Right:** Microsoft logo.

## To simplify, we can create templates

To easily export our Power BI effort, do the next steps:

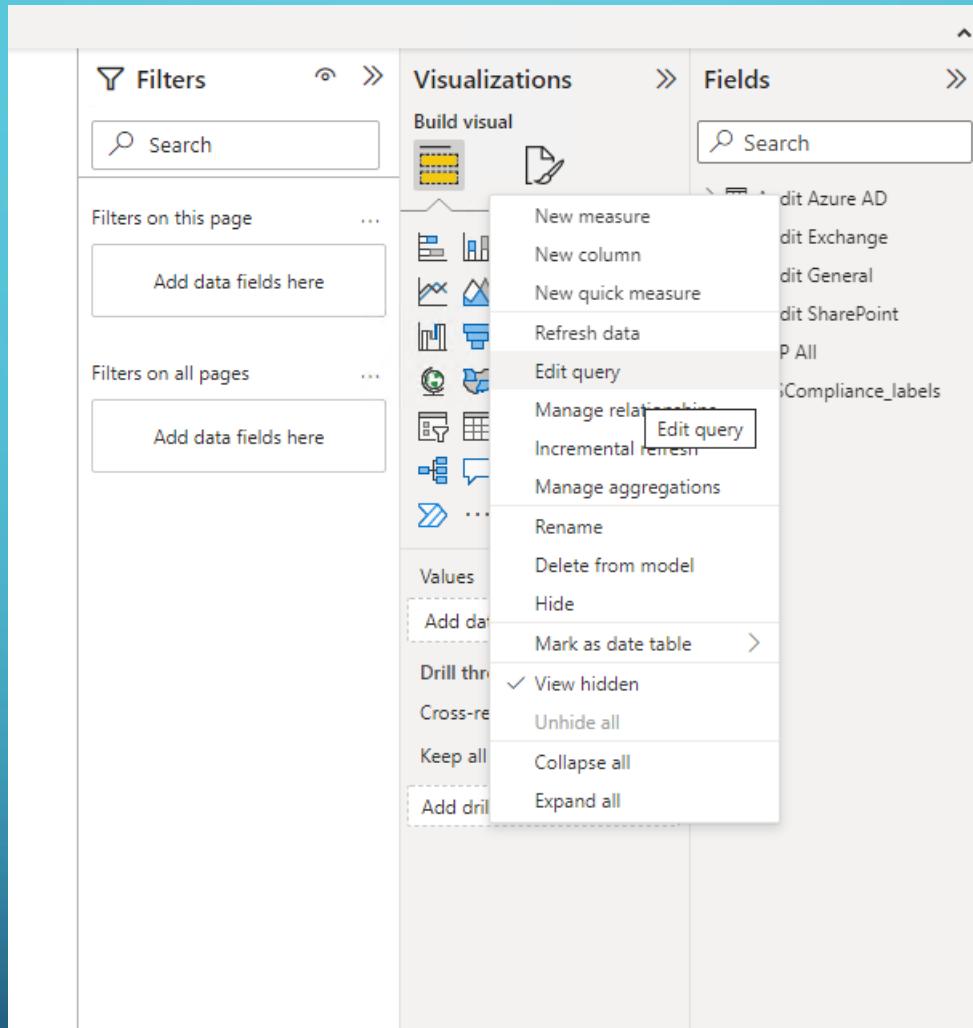
1. Right click at left over any of the table names and select Edit query
2. Under Power Query Editor go to Manage Parameters and select New Parameter
3. On the new pop-up window Add these:
  - Name: **Workspace\_ID**
  - Description: Workspace ID from Logs Analytics
  - Check Required
  - Type: Text
  - Suggested Values: Any value
  - Current Value: *{add your workspace id from Logs Analytics}*
4. After that, in each table(5) and function(5) is required change the Workspace ID with the new variable go through Advanced Editor

```
let Source = Json.Document(Web.Contents("https://api.loganalytics.io/v1/workspaces/" & Workspace_ID & "/query",
```

5. After finish, press Close & Apply
6. To export go to File and select Export and select Power BI template
7. Add a Template description
8. Set a File name and Location and Save

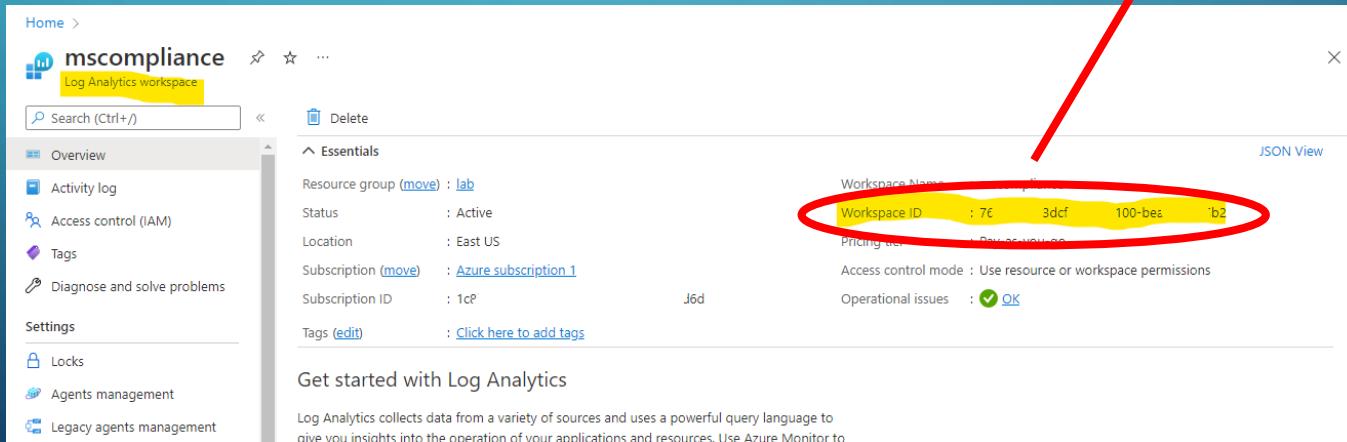
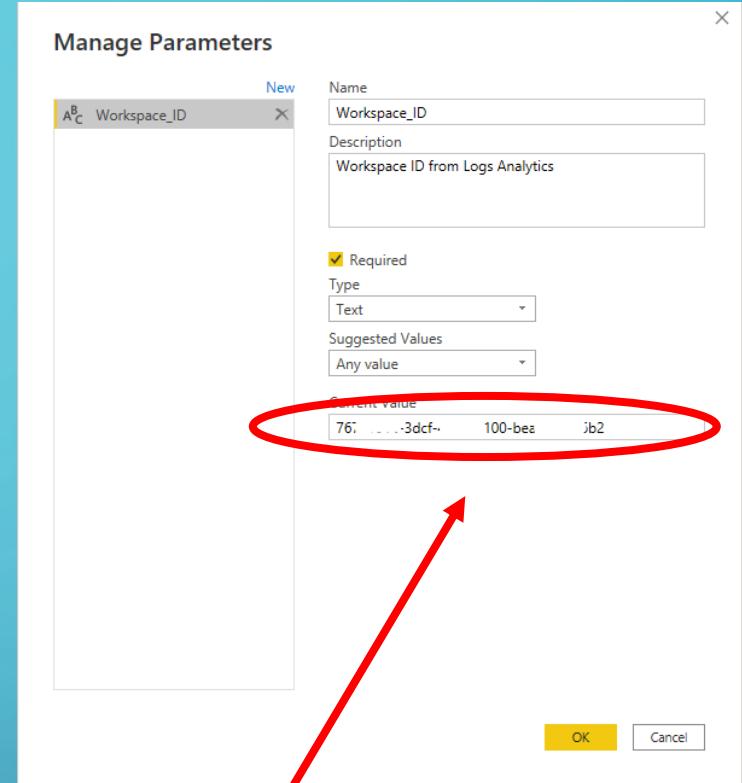
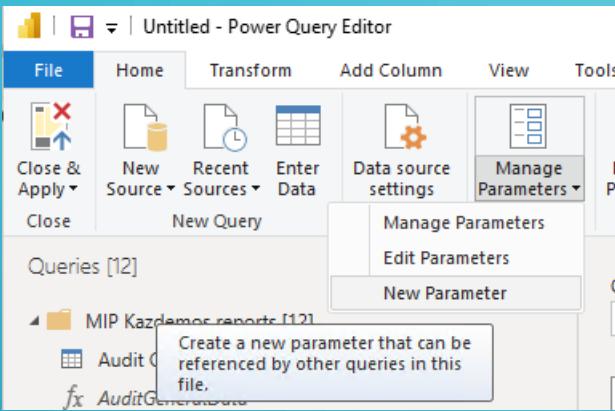
# POWER BI - TEMPLATES

## Steps



# POWER BI - TEMPLATES

## Steps



## Steps

The screenshot shows the Microsoft Power Query Editor window titled "Untitled - Power Query Editor". The ribbon menu includes File, Home, Transform, Add Column, View, Tools, and Help. The Home tab is selected. The ribbon also features Close & Apply, New, Recent, Sources, Data, Properties, Advanced Editor, and various data transformation icons. The status bar indicates "Data Type: Whole Number" and "Merge Queries", "Append Queries", and "Use First Row as Headers" options.

The left pane displays the "Queries [12]" list, which includes:

- MIP Kazdemos reports [12]
  - Audit General (selected)
  - AuditGeneralData
  - Audit Azure AD
  - AuditAzureADData
  - Audit Exchange
  - AuditExchangeData
  - Audit SharePoint
  - AuditSharePointData
  - DLP All
  - DLPAllData
  - MSCompliance\_labels
  - Workspace\_ID (76781014-3dc...)
- Other Queries

The main content area is titled "Audit General" and contains the following M Language script:

```
1 /*  
2 The exported Power Query Formula Language (M Language ) can be used with Power Query in Excel  
3 and Power BI Desktop.  
4 For Power BI Desktop follow the instructions below:  
5 1) Download Power BI Desktop from https://powerbi.microsoft.com/desktop/  
6 2) In Power BI Desktop select: 'Get Data' -> 'Blank Query'->'Advanced Query Editor'  
7 3) Paste the M Language script into the Advanced Query Editor and select 'Done'  
8 */  
9  
10  
11 let AnalyticsQuery =  
12 let Source = Json.Document(Web.Contents("https://api.loganalytics.io/v1/workspaces/"& Workspace_ID &"/query",  
13 [Query="#query=AuditGeneral_CL  
14 | where TimeGenerated > now(-730d)  
15 | summarize by  
16 Year = datetime_part('Year',TimeGenerated),  
17 Month = datetime_part('Month',TimeGenerated)  
18 ",#"x-ms-app="OmsAnalyticsPBI",#"prefer="ai.response-thinning=true"],Timeout=#duration(0,0,4,0)])),  
19 TypeMap = #table(  
20 { "AnalyticsTypes", "Type" },  
21 {  
22 { "string", Text.Type },  
23 { "int", Int32.Type },  
24 { "long", Int64.Type },  
25 { "real", Double.Type },  
26 { "timespan", Duration.Type },  
27 }  
28 )  
29 in TypeMap
```

A green checkmark icon and the text "No syntax errors have been detected." are displayed at the bottom left. At the bottom right are "Done" and "Cancel" buttons.

## Steps

The screenshot shows the Microsoft Power Query Editor interface. The title bar reads "Untitled - Power Query Editor". The ribbon menu includes File, Home, Transform, Add Column, View, Tools, and Help. The Home tab is selected. The ribbon also features Close & Apply, New, Recent, Sources, Data, Properties, Advanced Editor, and various data transformation icons like Sort, Filter, and Group. A status bar at the bottom indicates "Data Type: Any" and "Use First Row as Headers".

The left pane displays a tree view of "Queries [12]" under "MIP Kazdemos reports [12]". The "Audit General" folder is expanded, showing several queries: Audit General (selected), Audit Azure AD, AuditAzureADData, Audit Exchange, AuditExchangeData, Audit SharePoint, AuditSharePointData, DLP All, DLPAllData, MSCompliance\_Labels, and Workspace\_ID (76781014-3dc). An "Other Queries" folder is also present.

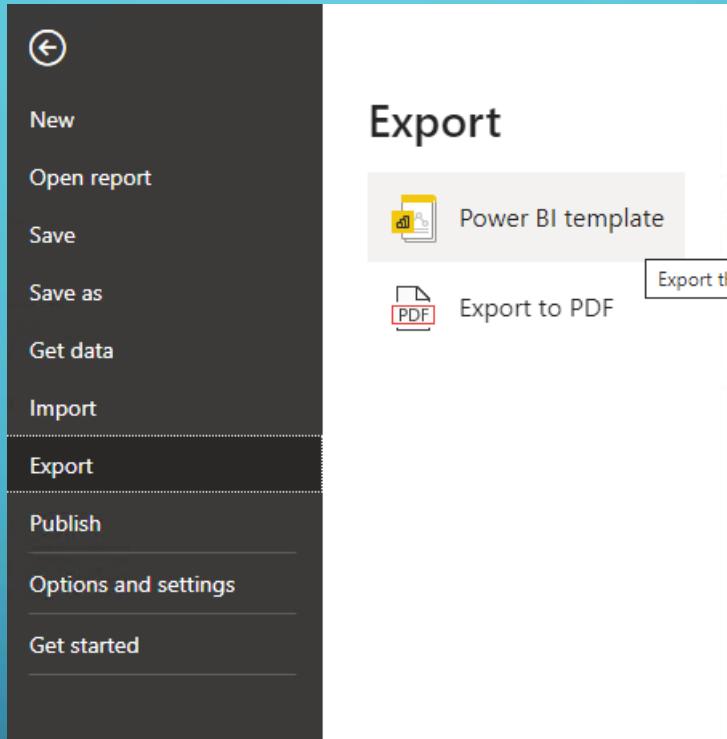
The main pane shows the MRE (Mashup Run Environment) code for the "AuditGeneralData" query:

```
1 let
2     Source = (Month as text, Year as text) => let AnalyticsQuery =
3         let Source = Json.Document(Web.Contents("https://api.loganalytics.io/v1/workspaces/& Workspace_ID &/query",
4             [Query=#"query"="AuditGeneral_CL
5             ,#"x-ms-app"="OmsAnalyticsPBI","#timespan"="AuditGeneral_CL | where datetime_part('Month',TimeGenerated) == "& Month &" and datetime_
6             TypeMap = #table(
7                 { "AnalyticsTypes", "Type" },
8                 [
9                     { "string", Text.Type },
10                    { "int", Int32.Type },
11                    { "long", Int64.Type },
12                    { "real", Double.Type },
13                    { "timespan", Duration.Type },
14                    { "datetime", DateTimeZone.Type },
15                    { "bool", Logical.Type },
16                    { "guid", Text.Type },
17                    { "dynamic", Text.Type }
18                }),
19                DataTable = Source[tables]{0},
20                Columns = Table.FromRecords(DataTable[columns]),
21                ColumnsWithType = Table.Join(Columns, {"type"}, TypeMap, {"AnalyticsTypes"}),
22                Rows = Table.FromRows(DataTable[rows], Columns[name]),
23                Table = Table.TransformColumnTypes(Rows, Table.ToList(ColumnsWithType, (c) => { c{0}, c{3}}))
24            in
25            Table
26            in AnalyticsQuery
```

A green checkmark icon and the text "No syntax errors have been detected." are displayed below the code. At the bottom right are "Done" and "Cancel" buttons.

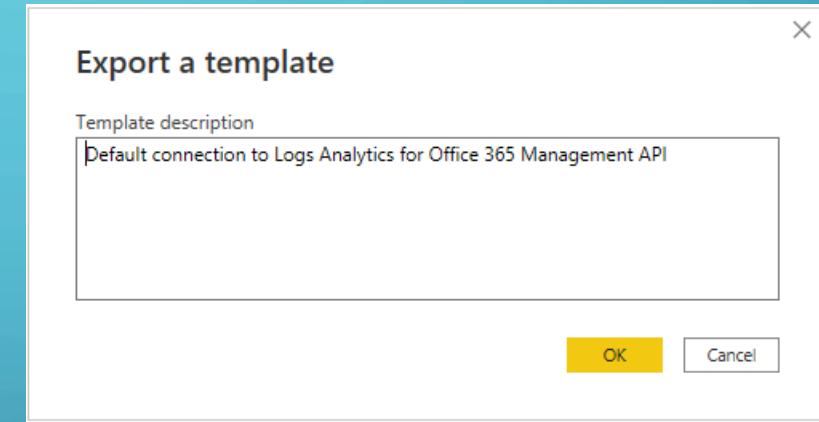
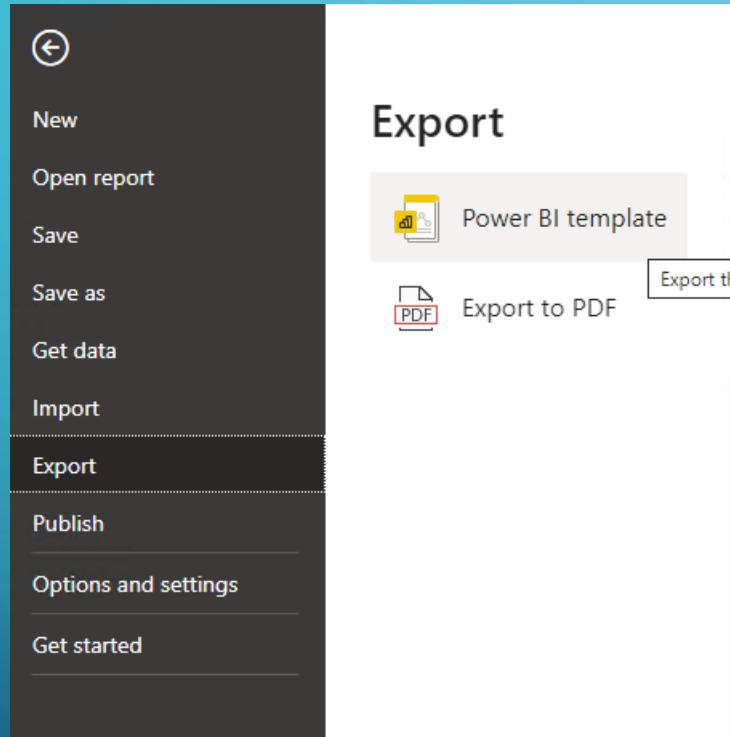
# POWER BI - TEMPLATES

## Steps



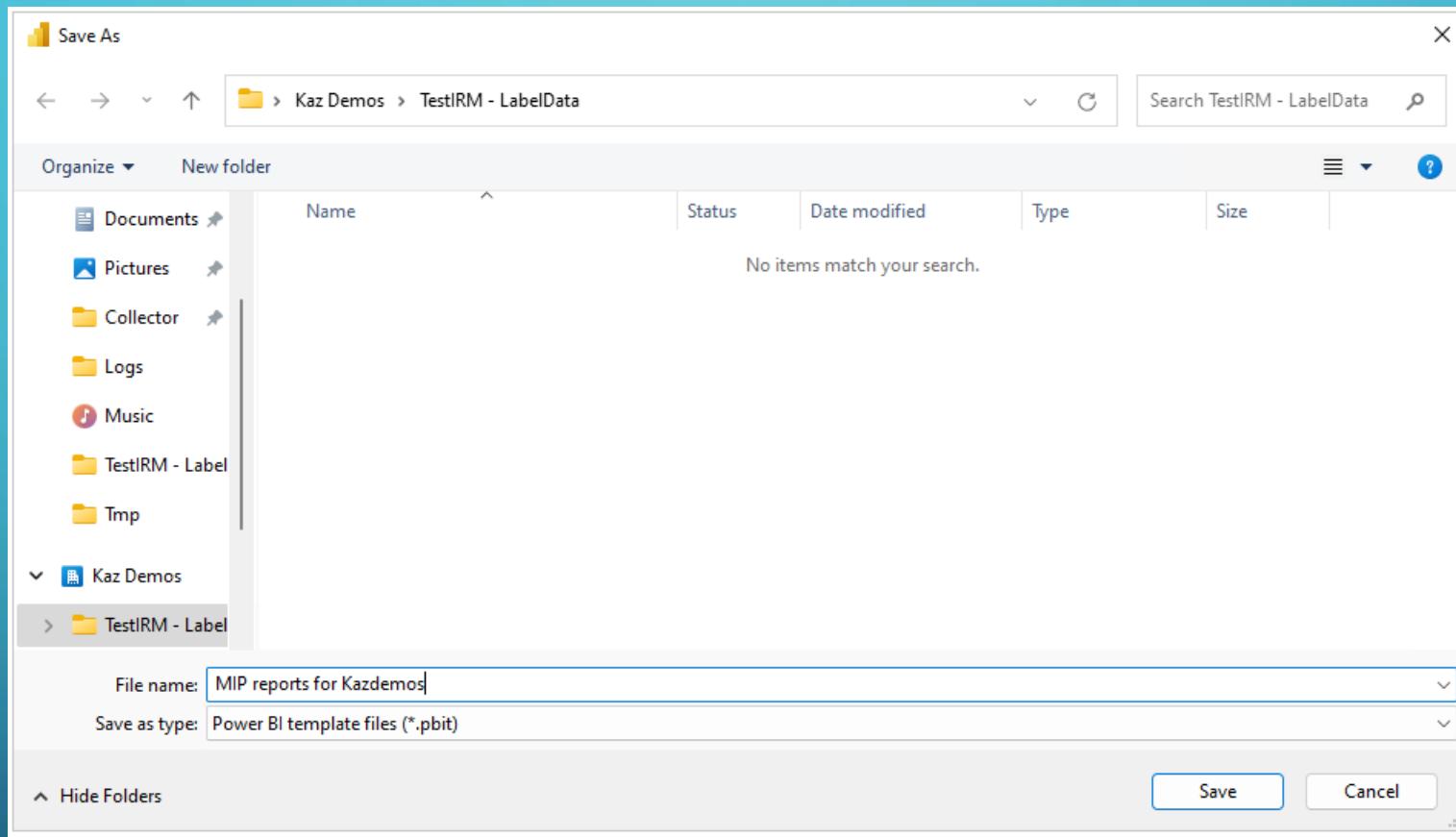
# POWER BI - TEMPLATES

## Steps



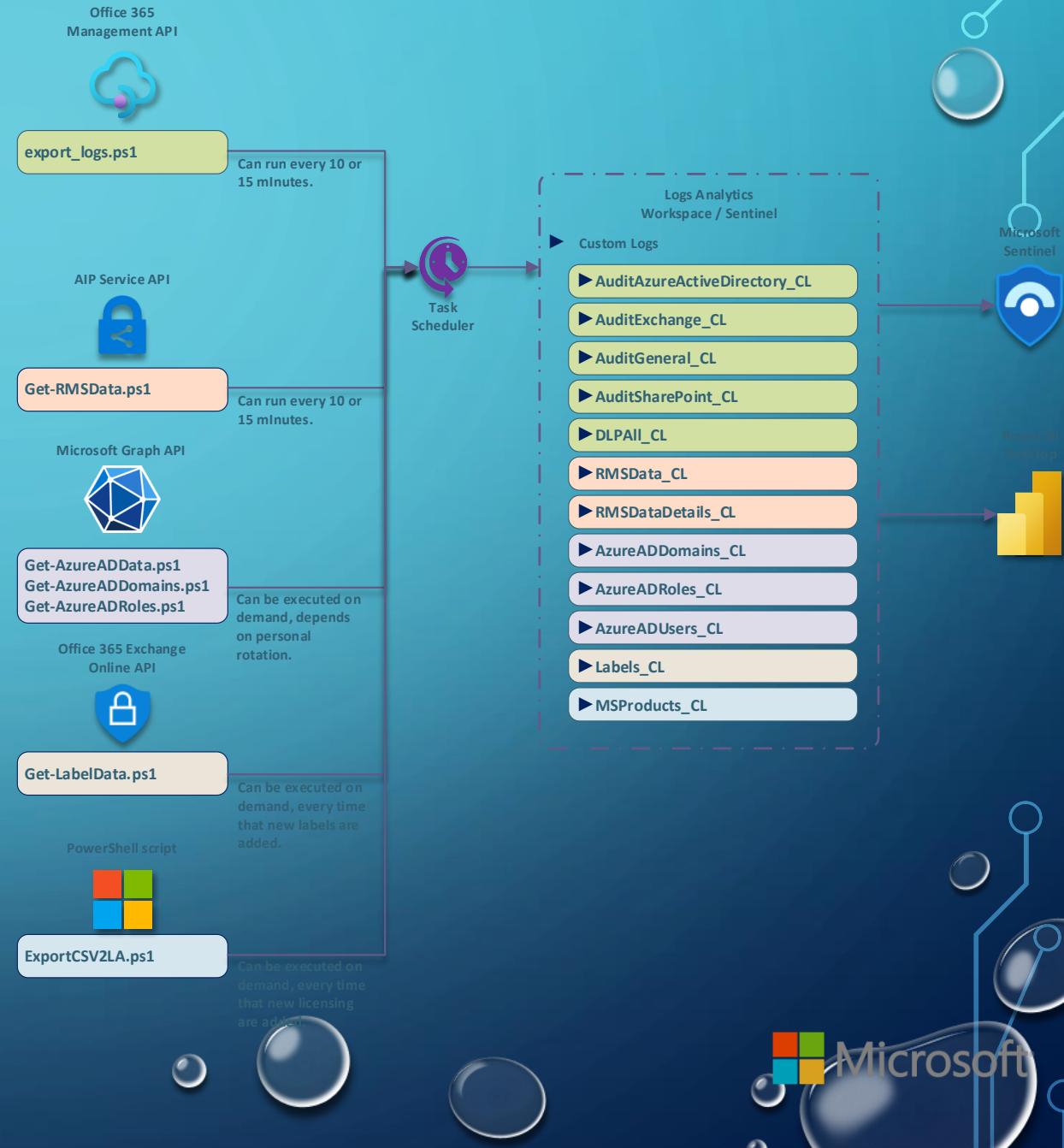
# POWER BI - TEMPLATES

## Steps



## TO IMPORT ADVANCED POWER BI TEMPLATE

- Validate these scripts was executed first:
  - Export\_logs.ps1 and is running under task scheduler
  - .\ExportCSV2LA.ps1 -FileName '.\Support Data\Product names and service plan identifiers for licensing.csv' - TableName MSProducts (was executed and the Tablename is created with the same name shown)
  - .\Get-AzureADData.ps1 (run on demand)
  - .\Get-AzureADRoles.ps1 (run on demand)
  - .\Get-AzureADDomains.ps1 (run on demand)
  - .\Get-LabelData.ps1 (run on demand)
  - .\Get-RMSData.ps1 and is running under task scheduler
- If it's the first time wait around 15 minutes until the tables are populated and appear under Custom Logs on Logs menu in your Logs Analytics workspace

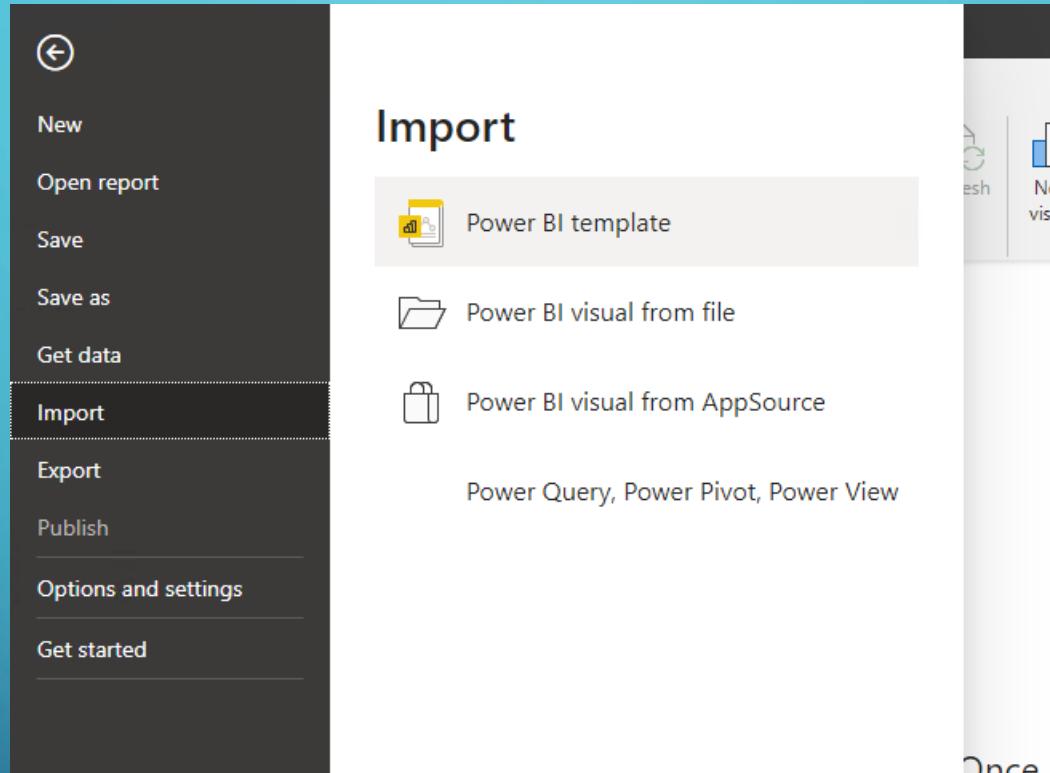


## To import a Template

- With this steps we will reduce a lot the previous effort to connect each data source and all the previous configuration
- Open Power BI Desktop
- Close the pop-up window
- Go to File then Import and select Power BI template
- Locate your template file, select them and press Open
- Workspace\_ID will be required, enter your Workspace ID and press Load
- Depending the account used as sign-in on Power BI, new credentials will be required
- To validate the right settings, open Power Query Editor doing a right click over any of the table names located at left under Fields and selecting Edit query
- Check Workspace\_ID variable that match with your Workspace ID
- Validate that you are using the right matrix for Label names and GUIDs, in case that not match with the current workspace ID information, remove and create a new one based on previous steps explanation.
- Close & Apply

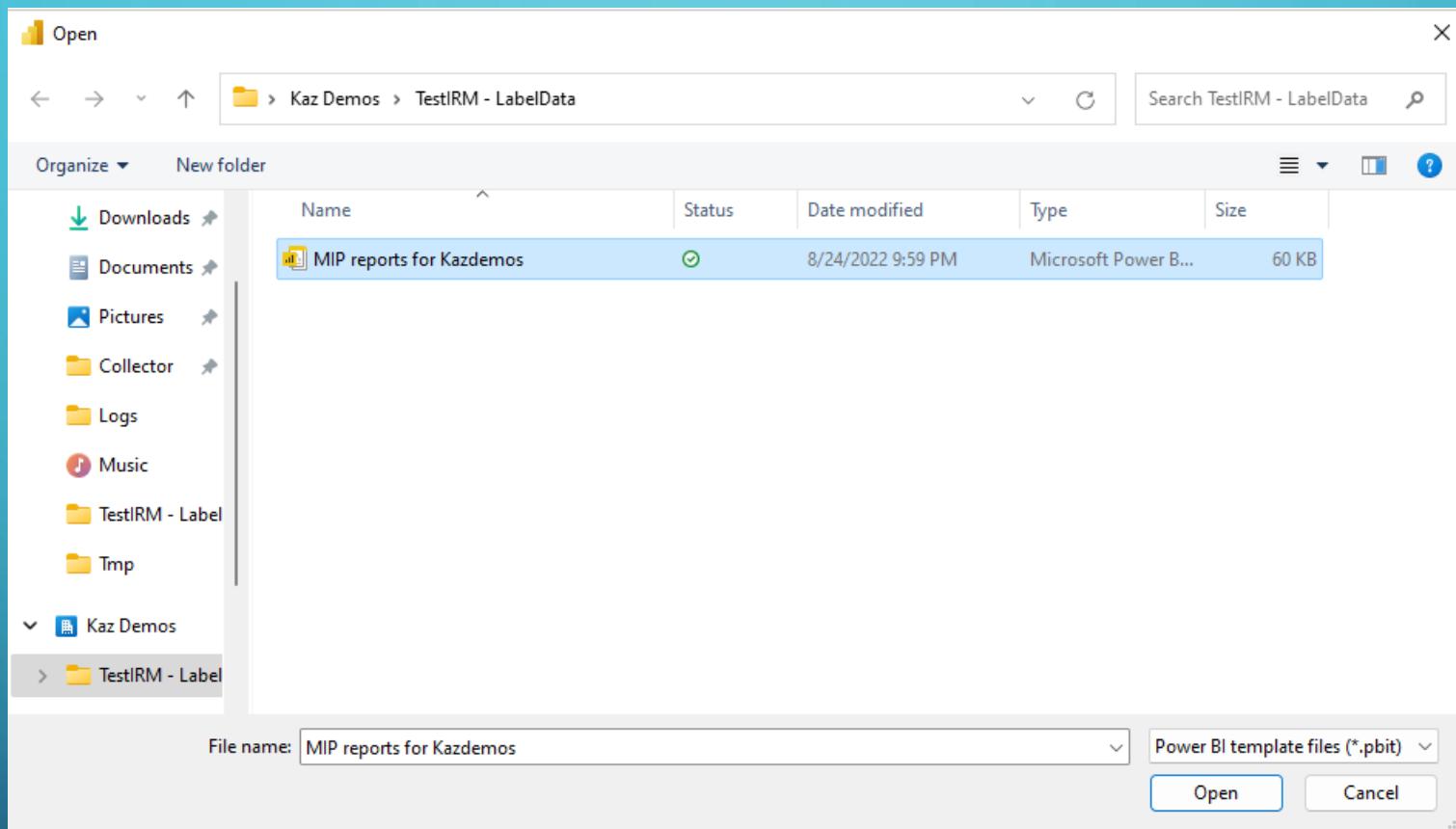
# POWER BI - TEMPLATES

## Steps



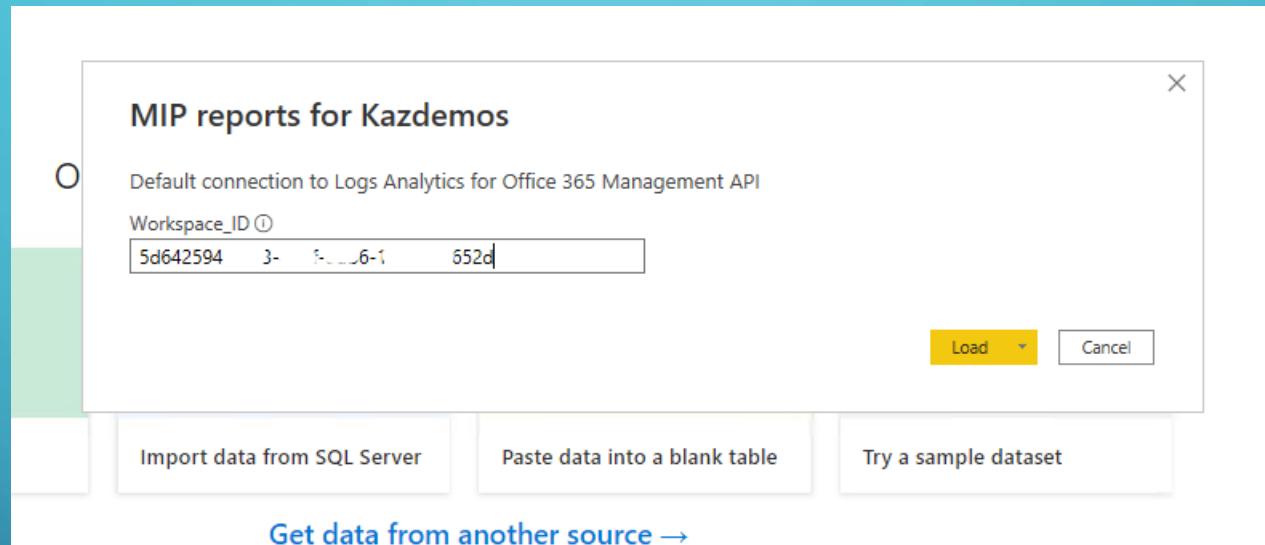
# POWER BI - TEMPLATES

## Steps

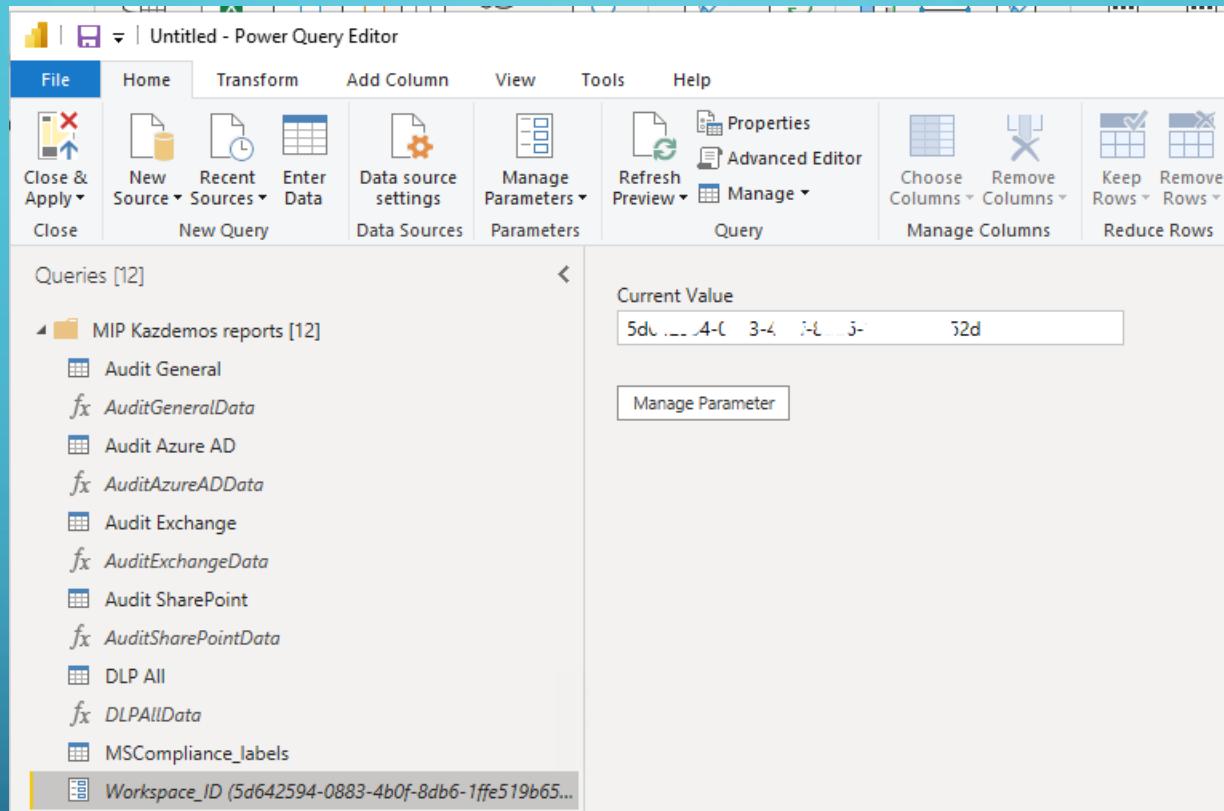


# POWER BI - TEMPLATES

## Steps



## Steps



## TO CREATE A FUNCTION ON POWER BI FOR GEO LOCATION BASED ON IP ADDRESSES

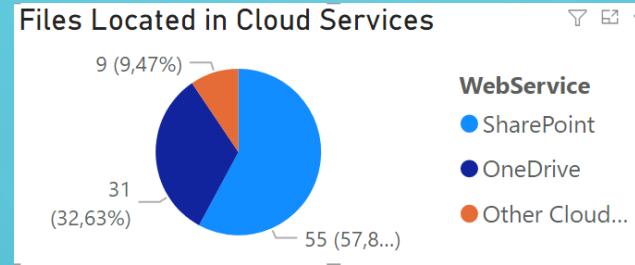
- Open Power BI, press “Get Data” and select “Blank query” in the new window open select “Advanced Editor” and replace the content with this Function.

```
let
    Source = (#"IP Address" as text) => let
        Source = Json.Document(Web.Contents("http://ipwhois.app/json/" & "#IP Address")),
        #"Converted to Table" = Record.ToTable(Source),
        #"Transposed Table" = Table.Transpose(#"Converted to Table"),
        #"Promoted Headers" = Table.PromoteHeaders(#"Transposed Table")
    in
        #"Promoted Headers"
    in
        Source
```

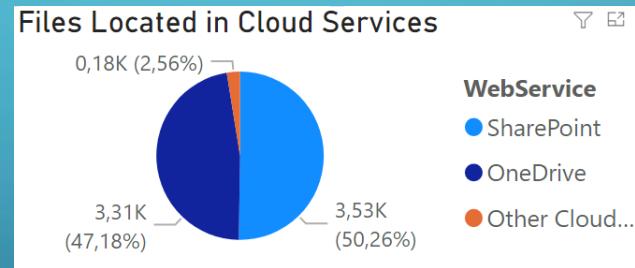
- At left menu change the name to something like as “GeoLocationFunction”

## Avoid some bad reports

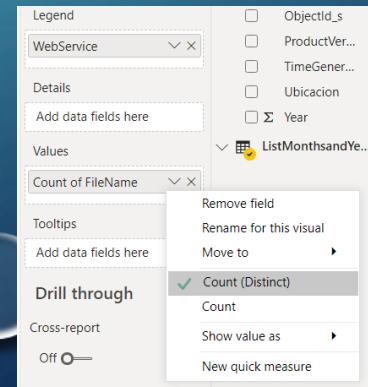
- Sometimes we need take care with the values and filters used when we create our reports to show wrong information, here an example:



- In the image shown we add a filter to count distinct FileName, but the Legend is related to Files located in Cloud Services, if we can not do that, we obtain this result:



- You can make the change in the same Power Bi on filters



# Links

## Links of Interest

- [Incremental refresh in Power BI](#)
- [Sensitivity labels in Power BI](#)
- [Exceed the 500,000 row limit in Application Insights and Log Analytics with Power BI](#)
- [Kusto](#)

#### URL Used ####

- [Power BI Query Editor - Getting IP Address Details from IP Address - Reporting/Analytics Made easy with FourMoo and Power BI](#)

#### Geo Location services by IP ####

- [14 Best IP Geolocation API to Offer Personalized Content \(geekflare.com\)](#)

#### Geo location APIs tested ####

- [IP Lookup API and IP Geolocation Documentation \(ipwhois.io\)](#)
- [IP-API.com - Geolocation API - Documentation - JSON \(ip-api.com\)](#)
- [Pricing | IP Geolocation API \(ipify.org\)](#)
- [IP Geolocation API with country information](#)
- [Free IP Geolocation API and Accurate IP Geolocation Database](#)



# THANK YOU

Sebastián Zamorano



+56 9 5207 3058



sebastian.zamorano@microsoft.com



<https://aka.ms/MPARR-GitHub>



[myprofile.kazblog.me](http://myprofile.kazblog.me)



Some additional effort to release soon

- II Kusto queries for the new schema, the idea is download only columns Required for this reports
  - ✓ Some examples and templates for Power BI dashboards
  - ✓ Geo location on Power BI based on IP address
  - ✓ Add Azure AD schema to generate reports based on areAs or departments
  - ✓ Add Azure RMS logs to show access from external users to protected Information
- II Incremental refresh on Power BI to reduce Logs Analytics cost
  - ✓ Publish different reports for separate audience on Power BI Online
- II Fields dictionary
- II Relationship map between tables