

# Power BI Universe

## Mapping the hive

**Thomas 'Tom' Martens**

Solution Architect

Munich Re



Accelerating Data-Driven Success



# Power BI Universe – Mapping the hive

- Session objectives
- Why is this important
- The architecture (and of course some Demos)



# About me



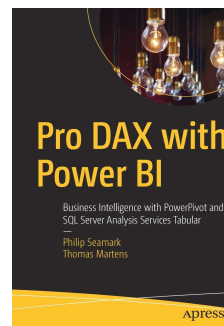
20+ years of  
experience designing  
analytical solutions  
using the MSFT data  
platform

Solution Architect  
Munich Re ([www.munichre.com](http://www.munichre.com))

✉ [tmartens@munichre.com](mailto:tmartens@munichre.com)

in [/tommartens68](#)

🐦 [@tommartens68](#)



# SESSION OBJECTIVES

## Find the files

<https://github.com/tomatminceddata/thehive>



# Session objectives

- Why it is important to know what's going on in your Power BI tenant
- Demonstration on how to setup all involved components to create a solution that maps your Power BI tenant
  - Service Principal / Security Group
  - Power BI Service tenant settings
  - Azure Data Lake Blob store
  - REST API
  - Power Shell



## WHY IS THIS IMPORTANT

## Find the files

<https://github.com/tomatminceddata/thehive>



# Some facts

~1000 on-premises data source connections

~1500  
Power BI  
Desktop  
installations

Power BI content is distributed across 3 Azure regions

~15k  
dataset  
refreshes  
per day

~50k  
workspaces

~40k  
employees



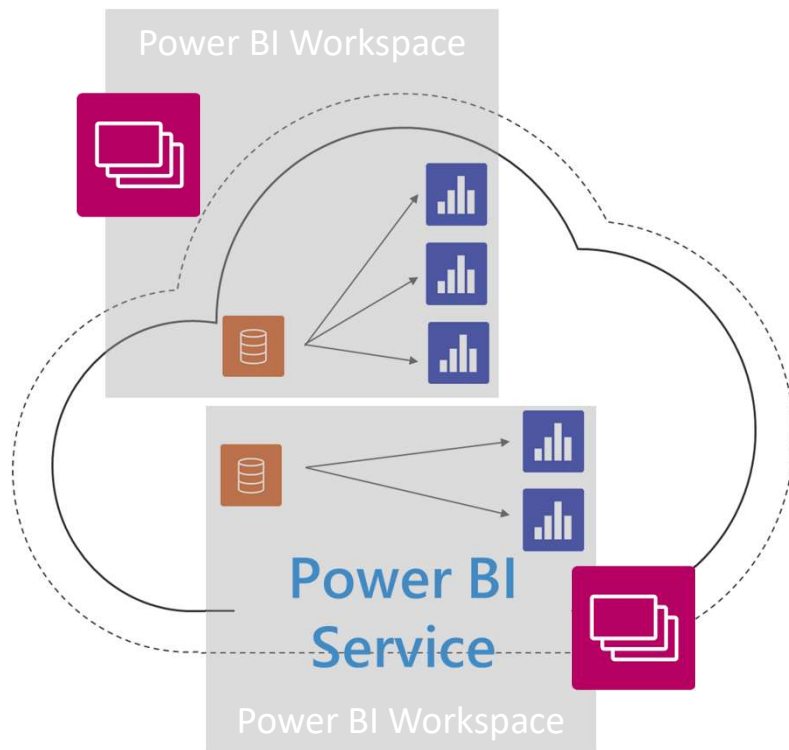
# Why you should map your Power BI environment

- Allow self-service but still govern data efficiently
- Help users discover available data
- Reduce data duplication
- ... and there are many more reasons





# The content of one workspace, ...



## Power BI Workspace

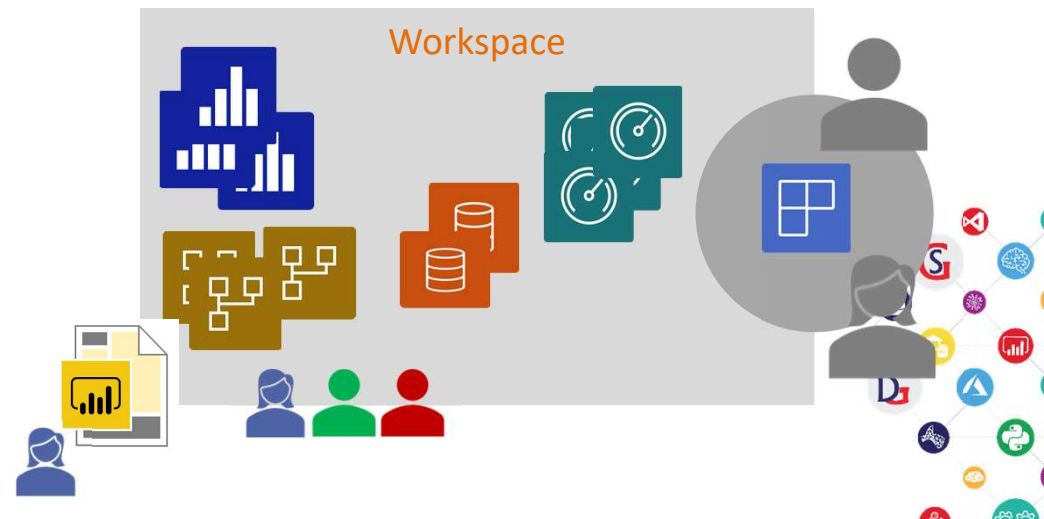
Power BI artifacts:

Dataflows

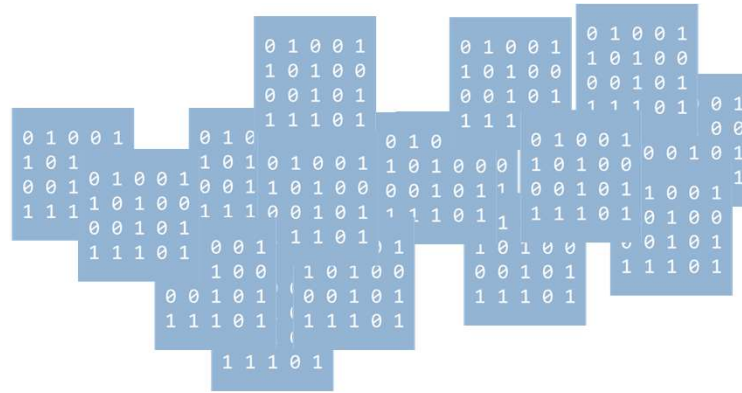
Datasets

Reports

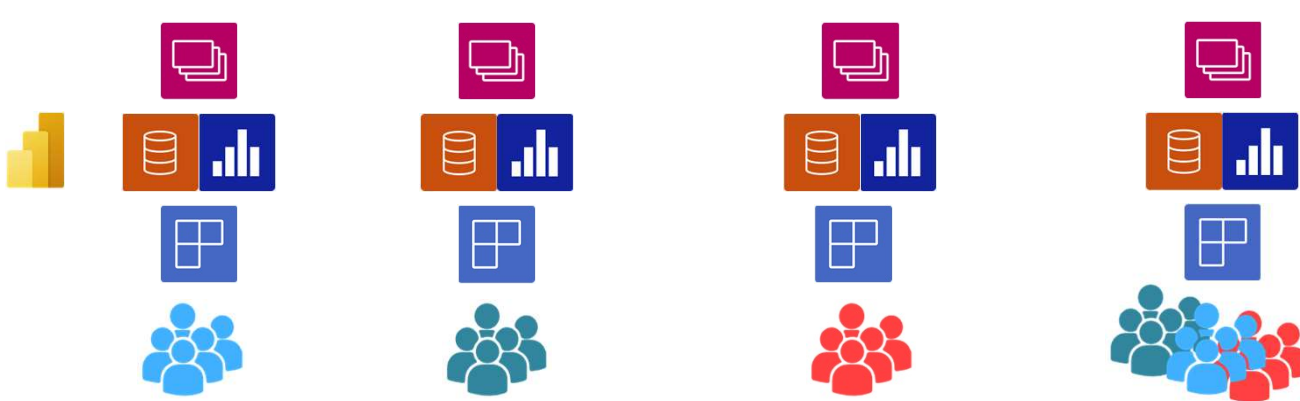
Dashboards



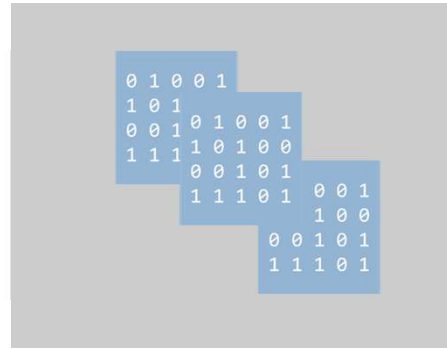
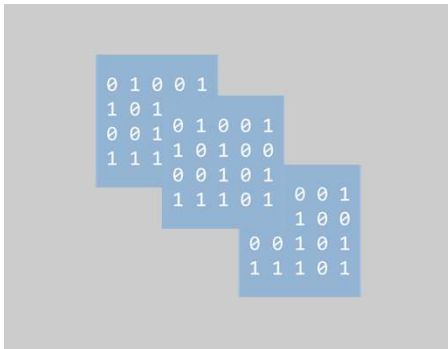
, but there will be many



# Don't just throw Power BI at your users!

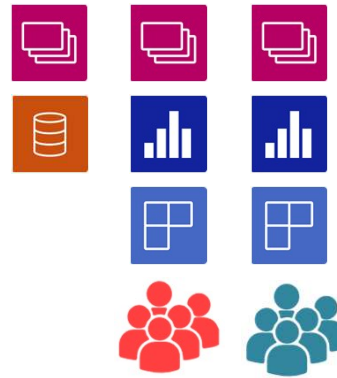


# Mapping the environment, helps to stay organized



Separate the data from the content!

- Create a data workspace
  - Content creators need Build Permission on the dataset
- Create 1 to n content workspaces
  - Content creators are **not** members of the data workspace
  - Content creators have at least the contributor role assigned in the content workspace
  - Content Creators are connecting to the Power BI data set
  - Row Level Security will be honored if data and content workspaces are separated



# THE ARCHITECTURE

Find the files

<https://github.com/tomatminceddata/thehive>



# One word in advance

You have to be aware that this is an ongoing “fun” project.

By fun, I mean I work on this project in my spare time.

I'm not done yet, but it works in the sense that metadata will be extracted, json documents containing the metadata will be stored in a blob store on Azure Data Lake, the json documents will be read by Power BI Desktop.



# Another word in advance

You will find the

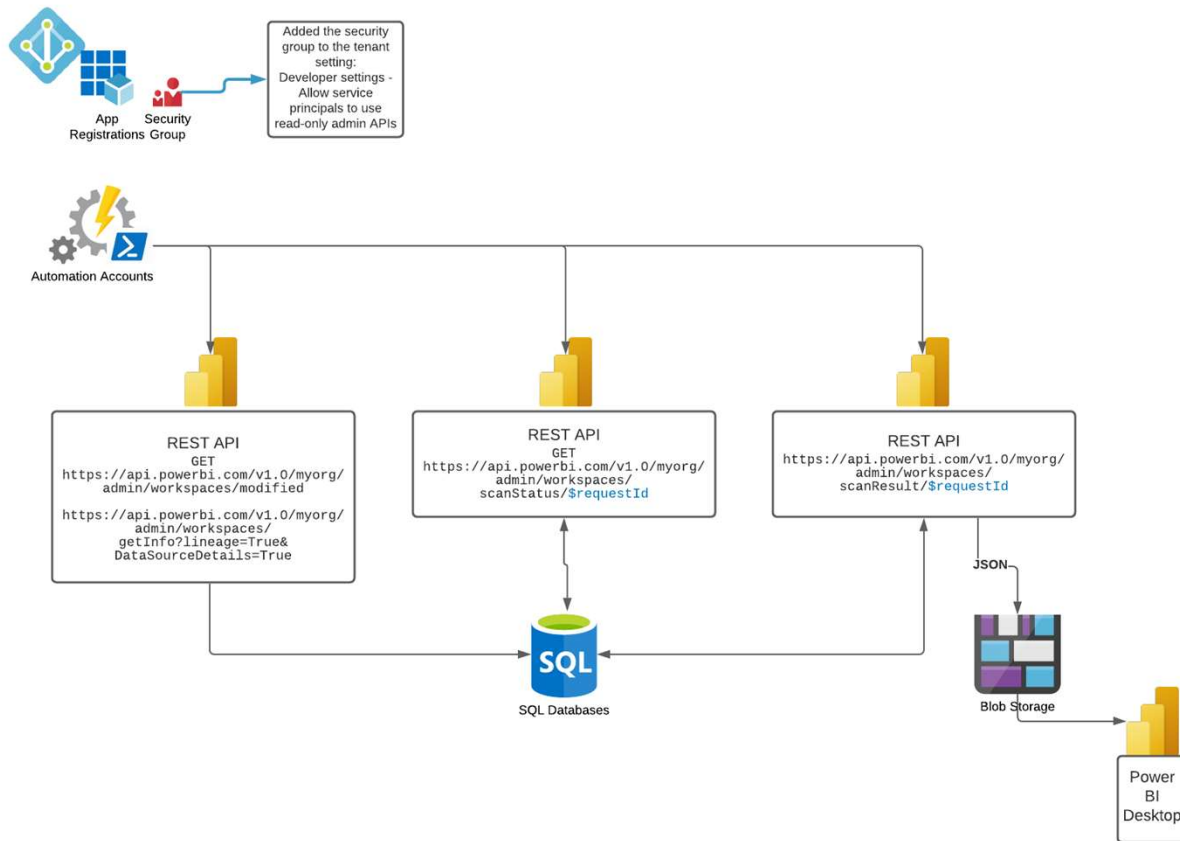
- PowerShell files
- the Power BI template file, and also
- a SQL statement that creates a SQL Server table

Next to the files, there is also a documentation that describes the solution in more detail and also describes the PowerShell scripts

here: <https://github.com/tomatminceddata/thehive>



# The Architecture



# The REST Admin APIs

- [Announcing new Admin APIs and Service Principal authentication to make for better tenant metadata scanning | Microsoft Power BI Blog | Microsoft Power BI](#)
- [Announcing Scanner API \(Admin REST APIs\) enhancements to include dataset tables, columns, measures, DAX expressions, and mashup queries | Microsoft Power BI Blog | Microsoft Power BI](#)
- [Announcing Admin APIs to Determine Access Rights \(Public Preview\) | Microsoft Power BI Blog | Microsoft Power BI](#)
- [Scanner API is now in GA | Microsoft Power BI Blog | Microsoft Power BI](#)





# THE ARCHITECTURE

The Service Principal Application

Find the files

<https://github.com/tomatminceddata/thehive>



# The Service Principal

A Service Principal is “a user” that is used to authenticate the application for the usage of resources, e.g., Power BI content and APIs.

This kind of “user” is necessary for automation.

[Enable service principal authentication for read-only admin APIs - Power BI | Microsoft Docs](#)



# Power BI Admin Portal – Admin API settings

---

## Developer settings

- ▶ Embed content in apps  
*Enabled for the entire organization*
- ▶ Allow service principals to use Power BI APIs  
*Disabled for the entire organization*
- ▶ Block ResourceKey Authentication  
*Disabled for the entire organization*

---

## Admin API settings

- ▶ Allow service principals to use read-only Power BI admin APIs  
*Enabled for a subset of the organization*
- ▶ Enhance admin APIs responses with detailed metadata (Preview)  
*Enabled for the entire organization*
- ▶ Enhance admin APIs responses with DAX and mashup expressions (Preview)  
*Enabled for the entire organization*



# Power BI Admin Portal – Admin API settings

## Admin API settings

### Allow service principals to use read-only Power BI admin APIs

*Enabled for a subset of the organization*

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

☒ Enabled

Apply to:

☐ The entire organization

☒ Specific security groups

aad-sec-pbi-readOnlyAdminAPI X Enter security groups

Apply

Cancel



# THE ARCHITECTURE

The Service Principal Application

Demo: Creating a Service Principal

**Find the files**

<https://github.com/tomatminceddata/thehive>




# Create an App

AAD → Enterprise Applications → New Application

Create your own application

×

 Got feedback?

What's the name of your app?

✓

What are you looking to do with your application?

☐ Configure Application Proxy for secure remote access to an on-premises application

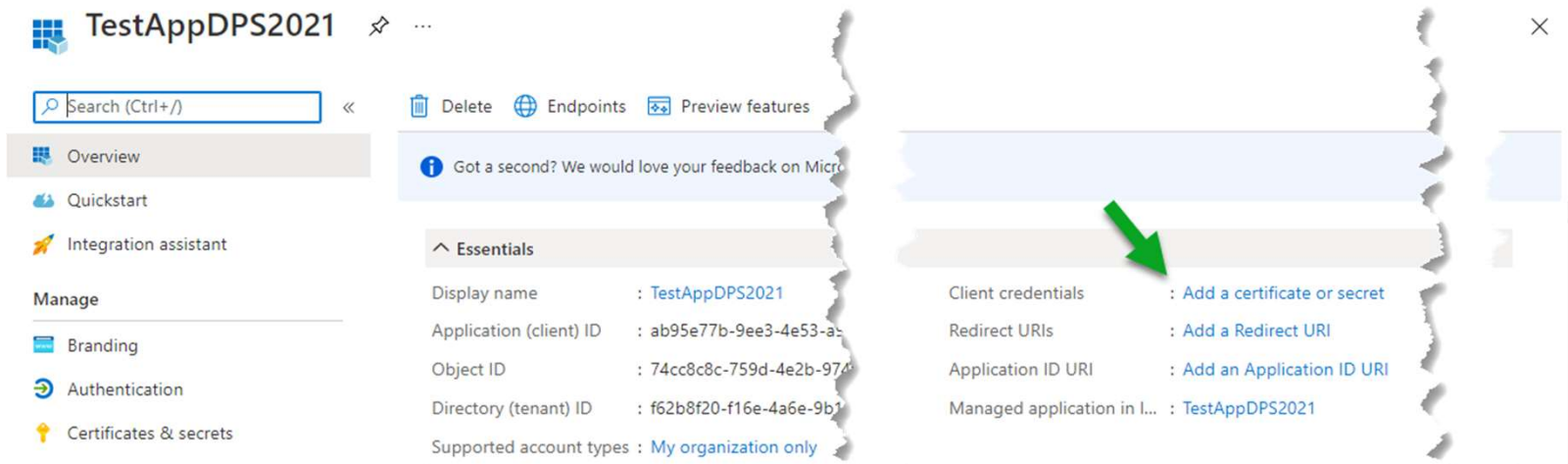
☒ Register an application to integrate with Azure AD (App you're developing)

☐ Integrate any other application you don't find in the gallery (Non-gallery)



# Create the App Secret

AAD → App Registrations → The App → Add a certificate or secret



The screenshot displays the Azure Portal interface for an application named 'TestAppDPS2021'. The left-hand navigation pane includes sections for 'Overview', 'Quickstart', 'Integration assistant', and 'Manage'. The 'Manage' section is expanded, showing options for 'Branding', 'Authentication', and 'Certificates & secrets'. The main content area is divided into two panels. The left panel, titled 'Essentials', lists key application details: Display name (TestAppDPS2021), Application (client) ID (ab95e77b-9ee3-4e53-a5-...), Object ID (74cc8c8c-759d-4e2b-97d-...), Directory (tenant) ID (f62b8f20-f16e-4a6e-9b1-...), and Supported account types (My organization only). The right panel, titled 'Client credentials', contains a table with the following information:

Client credentials	
Redirect URIs	<a href="#">Add a Redirect URI</a>
Application ID URI	<a href="#">Add an Application ID URI</a>
Managed application in I...	TestAppDPS2021

A green arrow points to the 'Add a certificate or secret' link in the 'Client credentials' section.



# Create the App Secret

... → The App → Add a certificate or secret → Certificates and secrets

Home > tommartens > TestAppDPS2021

TestAppDPS2021 | Certificates & secrets

Search (Ctrl+/) << Got feedback?

- Overview
- Quickstart
- Integration assistant
- Manage
  - Branding
  - Authentication
  - Certificates & secrets
  - Token configuration
  - API permissions
  - Expose an API
  - App roles
  - Owners
  - Roles and administrators | Preview
  - 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.

### Certificates

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	Start date	Expires	Certificate ID
No certificates have been added for this application.			

### Client secrets

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.			





# Create the App Secret

... → The App → Add a certificate or secret → Certificates and secrets

## Add a client secret



Description

Expires



Add

Cancel



# Don't forget the secret!!!!

Put the secret to a secure place or never forget the secret 😊

TestAppDPS2021 | Certificates & secrets

Search (Ctrl+/) Got feedback?

Overview Quickstart Integration assistant

Manage

- Branding
- Authentication
- Certificates & secrets**
- Token configuration
- API permissions
- Expose an API
- App roles
- Owners
- Roles and administrators | Preview
- 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.

### Certificates

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	Start date	Expires	Certificate ID
No certificates have been added for this application.			

### Client secrets

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
TestAppDPS2021	8/29/2023	5Vhk8~Kqd9WRp~9-.NGNJy4yW4x4-HuF...	13e91f93-0244-426c-96a2-f332130133d9



# Authentication using the App

To authenticate using the app, you need the App Key and the Secret.

The application key can be considered the user.

The secret can be considered the password for that user.



# THE ARCHITECTURE

The Security Group

Demo: Creating a Security Group

**Find the files**

<https://github.com/tomatminceddata/thehive>



# Create a Security Group

AAD → Groups → New Group

The screenshot shows the 'New Group' page in the Azure AD portal. On the left, the 'New Group' form is visible. A green arrow points to the 'Group type' dropdown, which is set to 'Security'. Another green arrow points to the 'Group name' field, which contains 'aad-sec-TestAppDPS2021'. A third green arrow points to the 'Add members' pane on the right. In this pane, a search bar contains the text 'test'. Below the search bar, two members are listed: 'TestAppDPS2021' and 'testPowerBIRRealTimeData'. A green arrow points to the 'Select' button at the bottom of the 'Add members' pane.

Home > Groups > New Group

**New Group**

Group type \*  
Security

Group name \*  
aad-sec-TestAppDPS2021

Group description  
Enter a description for the group

Azure AD roles can be assigned to the group  
Yes No

Membership type  
Assigned

Owners  
No owners selected

Members  
No members selected

Create

**Add members**

Search  
test

TestAppDPS2021  
ab95e77b-9ee3-4e53-a925-17c51ac4686f

testPowerBIRRealTimeData  
83ad6df3-1b96-48a4-8d9a-1b9e17726400

**Selected items**  
No items selected

Select



# Finally assign the new Security Group to the Power BI tenant settings

## Power BI Admin Portal → Admin API settings

### Admin API settings

#### Allow service principals to use read-only Power BI admin APIs

*Unapplied changes*

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

☒ Enabled

Apply to:

☐ The entire organization

☒ Specific security groups

aad-sec-pbi-readOnlyAdminAPI X aad-sec

**aad-sec-readPBImetadata**

**aad-sec-TestAppDPS2021**



# THE ARCHITECTURE

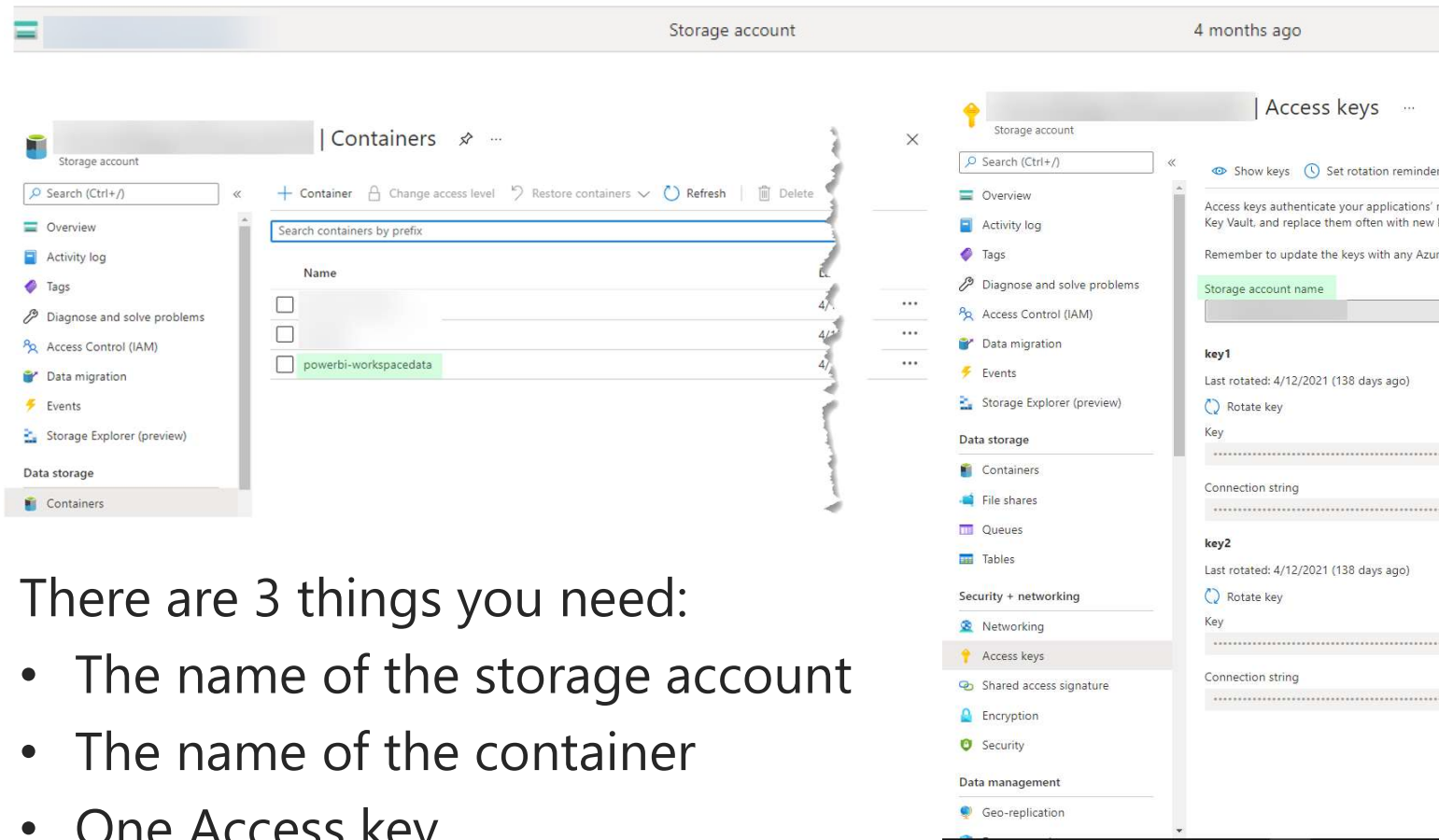
The Blob store (inside Azure Data Lake Gen 2)

**Find the files**

<https://github.com/tomatminceddata/thehive>



# Azure Data Lake Gen 2 – the blob store



The image shows two screenshots from the Azure portal. The left screenshot displays the 'Containers' view of a storage account, with a table listing containers. The right screenshot displays the 'Access keys' view, showing two keys (key1 and key2) with their rotation dates and connection strings.

Storage account

4 months ago

Containers

Search (Ctrl+/)

Container Change access level Restore containers Refresh Delete

Search containers by prefix

Name	4/12/2021
	4/12/2021
powerbi-workspacedata	4/12/2021

Overview

Activity log

Tags

Diagnose and solve problems

Access Control (IAM)

Data migration

Events

Storage Explorer (preview)

Data storage

Containers

File shares

Queues

Tables

Security + networking

Networking

Access keys

Shared access signature

Encryption

Security

Data management

Geo-replication

Access keys

Search (Ctrl+/)

Show keys Set rotation reminder Refresh

Access keys authenticate your applications' requests to this storage account. Keep your keys in a secure location like Azure Key Vault, and replace them often with new keys. The two keys allow you to replace one while still using the other.

Remember to update the keys with any Azure resources and apps that use this storage account. [Learn more](#)

Storage account name

key1

Last rotated: 4/12/2021 (138 days ago)

Rotate key

Key

Connection string

key2

Last rotated: 4/12/2021 (138 days ago)

Rotate key

Key

Connection string

There are 3 things you need:

- The name of the storage account
- The name of the container
- One Access key



# THE CODE

Find the files

<https://github.com/tomatminceddata/thehive>



# The Code - Basics

The Admin APIs are asynchronous, meaning you can't wait (or at least should not wait) for the result.

Basically, there are three steps:

- Get a list of workspaces
- Ask for the metadata of each single workspaces, this is a request, one request can only contain 100 workspaces. This means, the requests have to be batched
- Get the results for each request and store this result as a json document in the blob store



# The Code - Authentication

There is a little file that contains all the necessary information, the file looks like this, this file is used until all has been moved to Key Vault:

```
{
  "PowerBISP": {
    "user": "the Application key of the Power BI SPN",
    "pwd": "<the application secret>",
    "tenantid": "<your tenantid>"
  },
  "sqlinstance": {
    "user": "<the sqluser>",
    "pwd": "<the password of the sql user>",
    "instance": "<the sql instance>",
    "database": "<the sql database>"
  },
  "blob": {
    "storageAccount": "<the storage account>",
    "storageAccountKey": "<the storage account key>"
  }
}
```



# The Code - Authentication

There is a little file that contains all the necessary information, the file looks like this, this file is used until all has been moved to Key Vault:

```
# some parameters, these will be replaced by using runbook secrets
$someSecretThings = Get-Content "C:/@dev/GitHub/monitorthehive/some private information.Json"
$someSecretThings_Obj = $someSecretThings | ConvertFrom-Json

# Power BI Service Principal
$PBIAppId = ($someSecretThings_Obj.psobject.properties | Select name, value | where name -eq "PowerBISP").value.user
$PBISecret = ($someSecretThings_Obj.psobject.properties | Select name, value | where name -eq "PowerBISP").value.pwd
$PBITenantID = ($someSecretThings_Obj.psobject.properties | Select name, value | where name -eq "PowerBISP").value.tenanted

# Create credentials for the PBI Service Principal
$password = ConvertTo-SecureString $PBISecret -AsPlainText -Force
$Credentials = New-Object pscredential $PBIAppId, $password

# Connect using a Service Principal
Connect-PowerBIServiceAccount -ServicePrincipal -Credential $Credentials -Tenant $PBITenantID
```



# POWER BI

Extract the metadata and visualize your hive

Demo: Importing the Power BI template file and connecting to the blob store

## Find the files

<https://github.com/tomatminceddata/thehive>



# Power BI – Import the template file

Connecting Power BI to the blob store requires two parameters

- The name of the storage account
- The name of the container (the folder where the json documents will be stored)

Mapping the hive - using the new admin apis

Getting workspace metadata from Json Documents generated by the Admin Scanner Rest APIs.

theStorageAccount

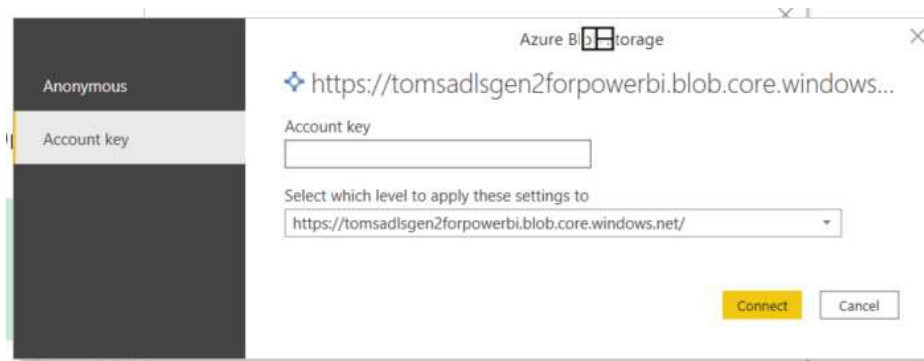
theBlobFolder

Load Cancel



# Power BI – Import the template file

Use one of the keys of the storage account to connect to the blob store that contains the json documents



# RECAP

Find the files

<https://github.com/tomatminceddata/thehive>





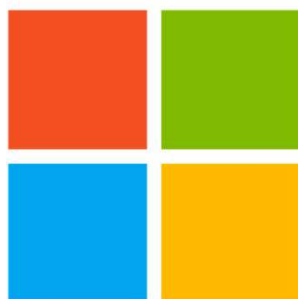
# Recap

Extracting and **analyzing metadata** from your Power BI environment can reveal tremendous **insights** on how **Power BI is used** in your organization. Some technologies are maybe unfamiliar like the Service Principal or Security groups, but if your environment is growing you will use them more frequently, e.g., **Security Groups** can be used for **Row Level Security**. As the REST APIs return json documents, **Azure blob store** is a “cheap” and flexible storage system.

**PowerShell**, helps a Power BI Service Administrator **answering many questions** or **automate simple tasks**, like creating a workspace on request.



**Special Thanks To**



**Microsoft**

**for supporting  
DataPlatformGeeks & SQLServerGeeks  
Community Initiatives**



# Thank You

## Three Ways to Win Prizes

Post your selfie with hash tag **#DPS2021**

Give Session & Conference Feedback

Visit our Sponsors & Exhibitors

Follow us on Twitter **@TheDataGeeks @DataAISummit**

