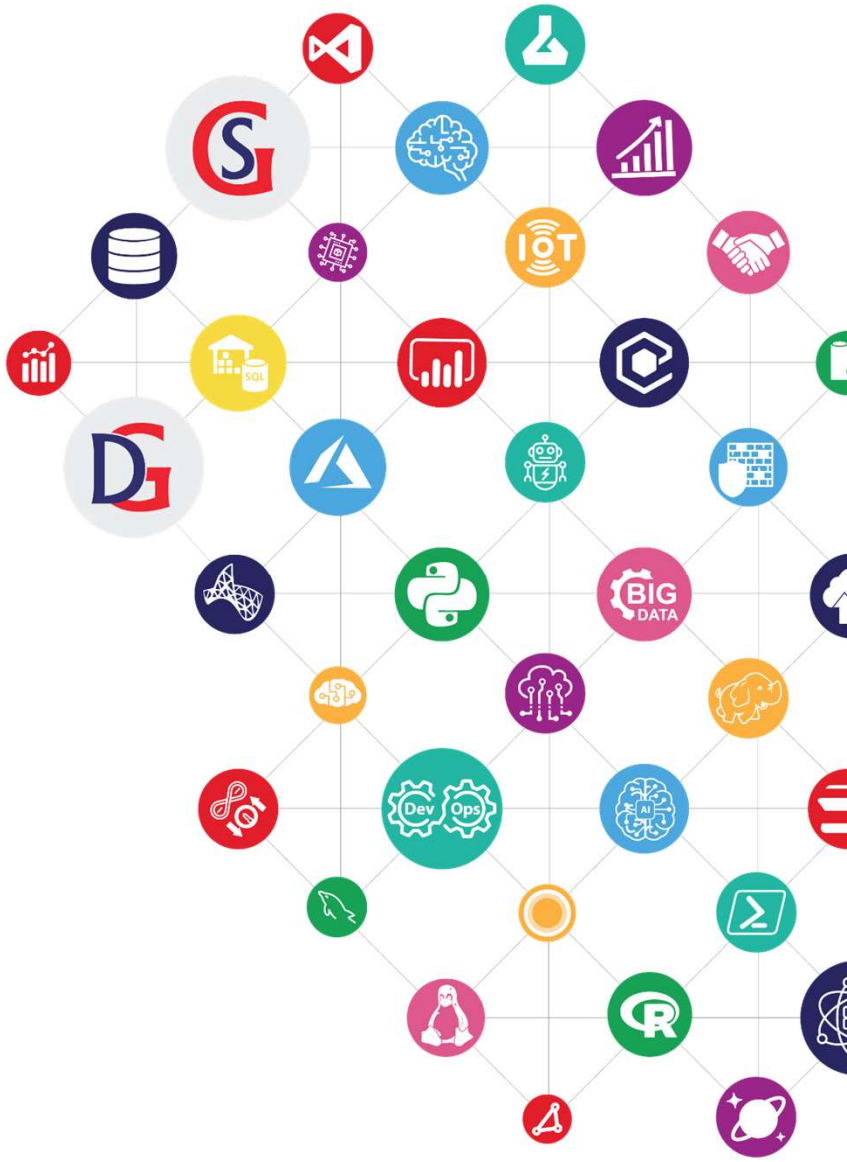
Power BI Universe

Mapping the hive

Thomas 'Tom' Martens

Solution Architect

Munich Re



DATA PLATFORM
VIRTUAL SUMMIT 2021

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

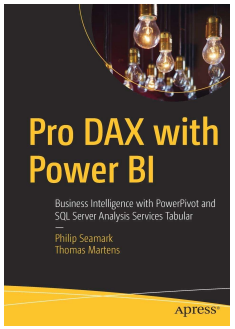
20+ years of experience designing analytical solutions using the MSFT data platform

Solution Architect
Munich Re (www.munichre.com)

tmartens@munichre.com

/tommartens68

@tommartens68



MVP Microsoft® Most Valuable Professional

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

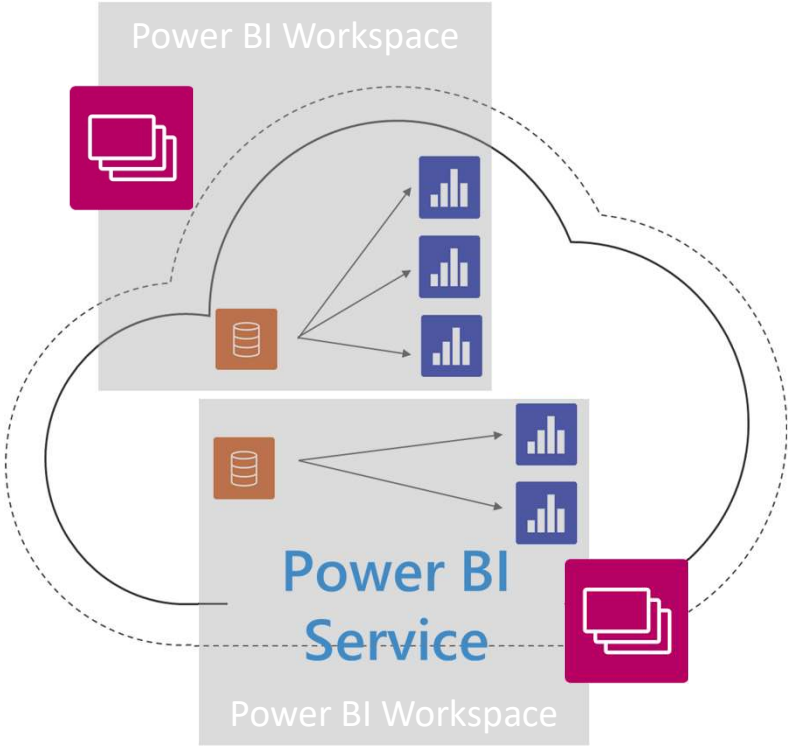
~50k workspaces

~40k employees

Why you should map your Power BI environment

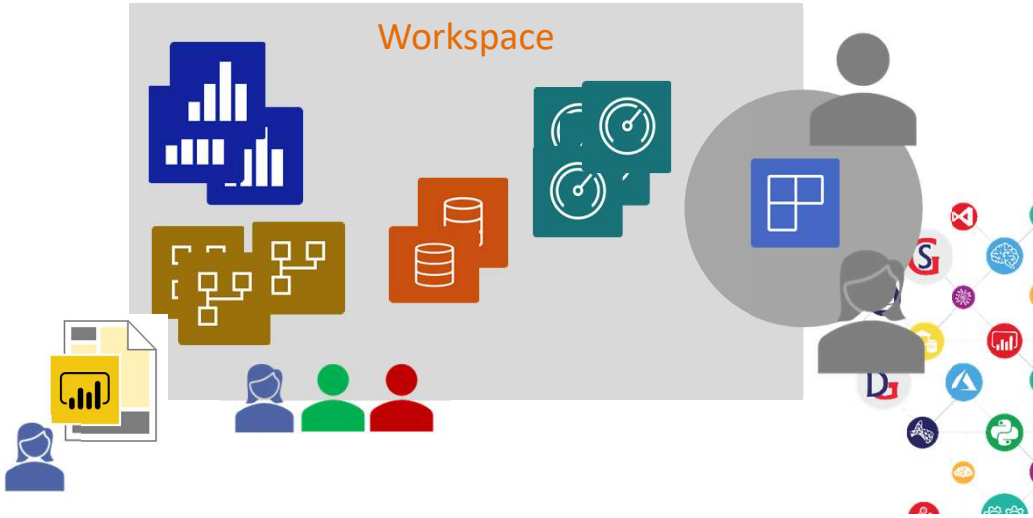
- Allow self-service but still govern data efficiently
- Help users discover available data
- Reduce data duplication
- … and there are many more reasons

The content of one workspace, …

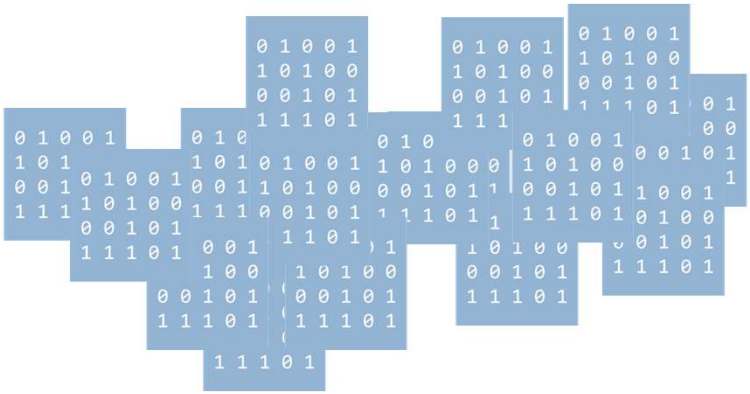


Power BI Workspace
Power BI artifacts:

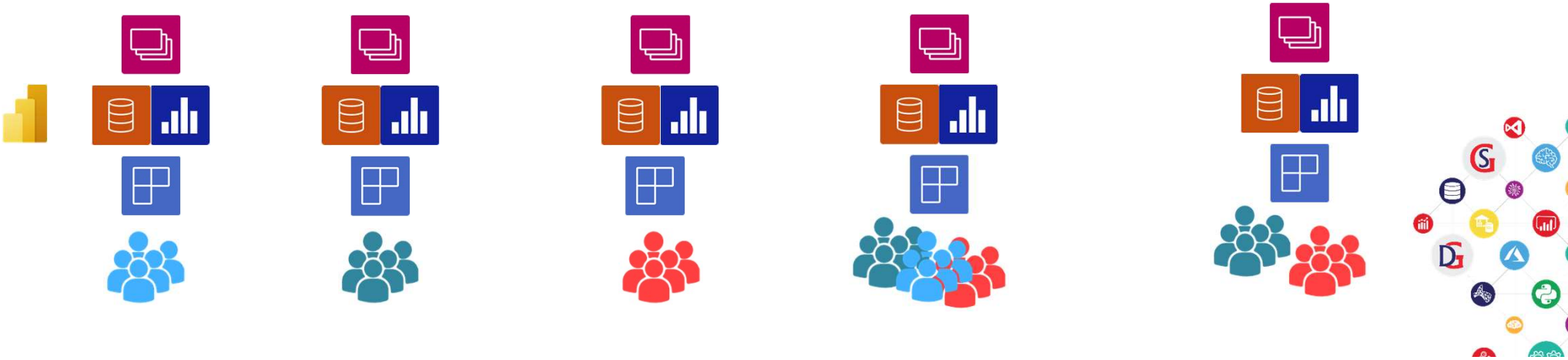
Dataflows
Datasets
Reports
Dashboards



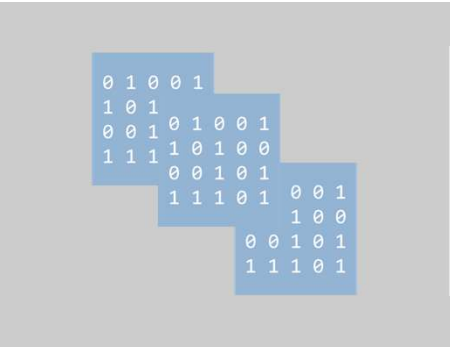
, but there will be many

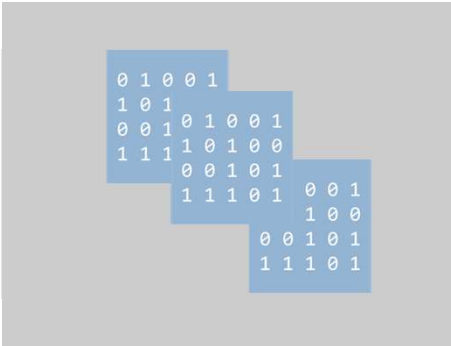


Don’t just throw Power BI at your users!



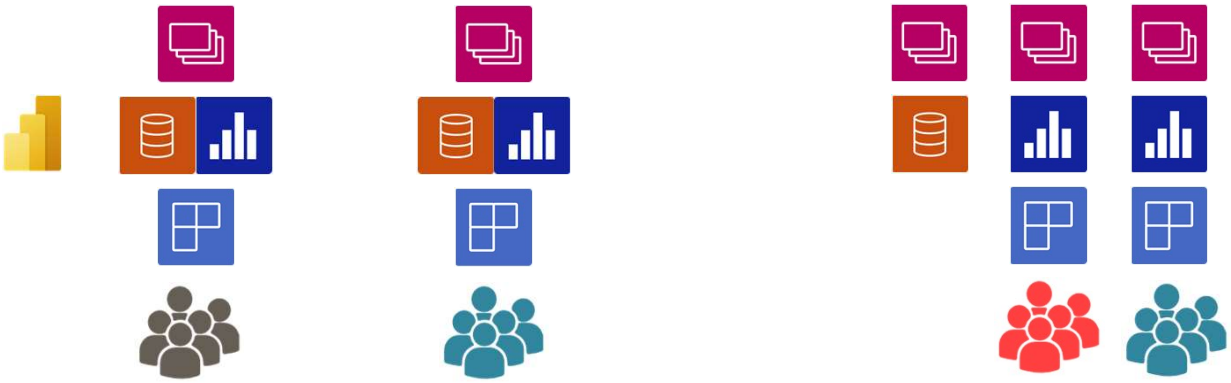
Mapping the environment, helps to stay organized





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

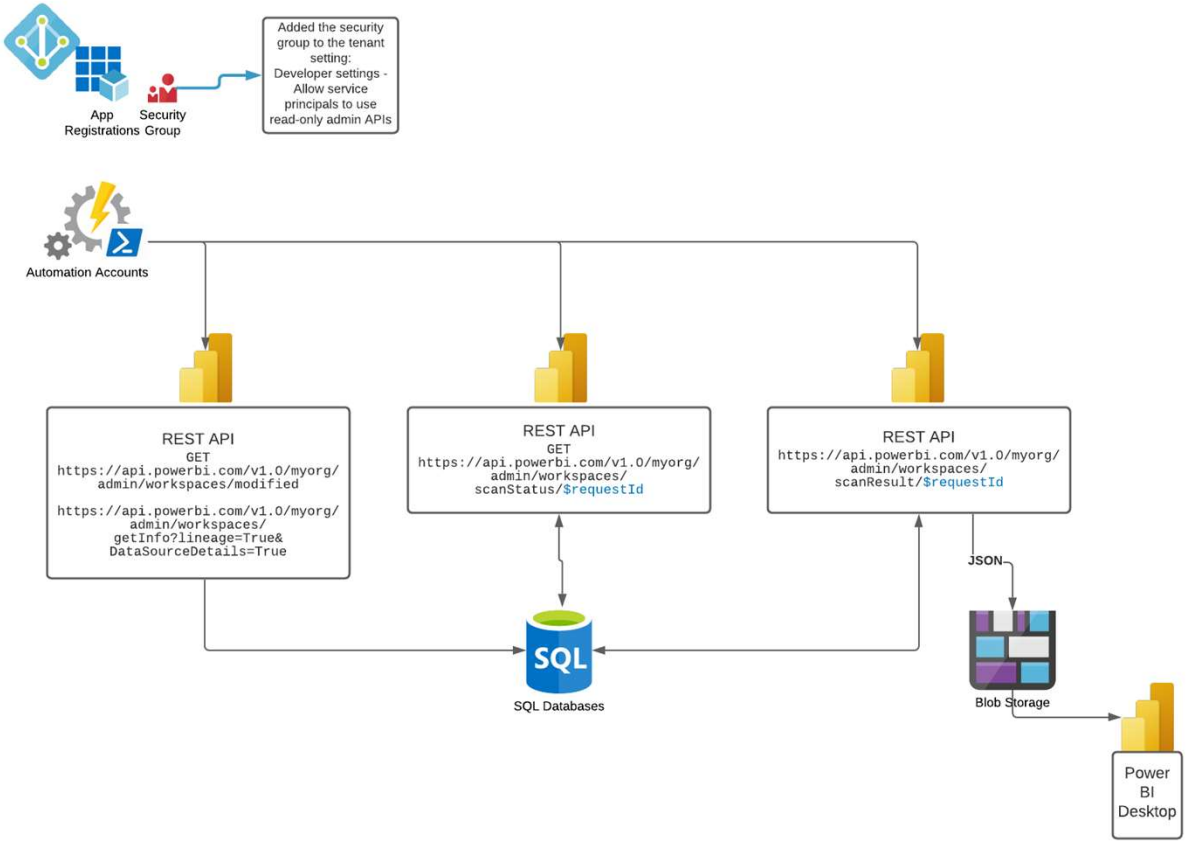
You will find the

- 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 ✕ Enter security groups

Apply Cancel

THE ARCHITECTURE

The Service Principal Application

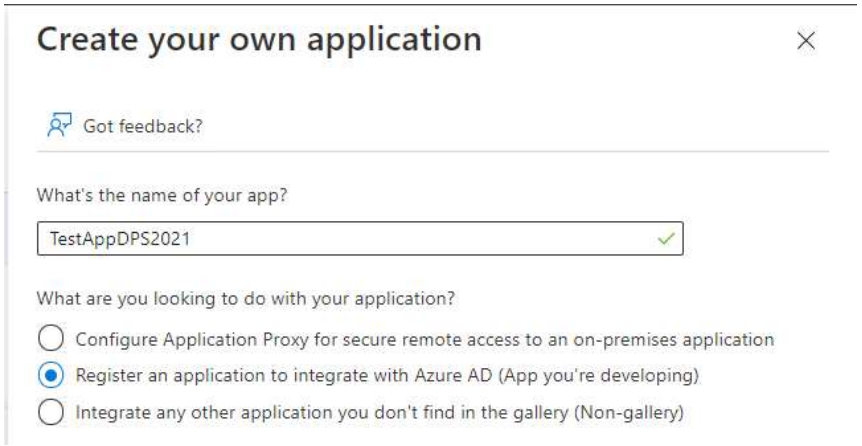
Demo: Creating a Service Principal

Find the files

https://github.com/tomatminceddata/thehive

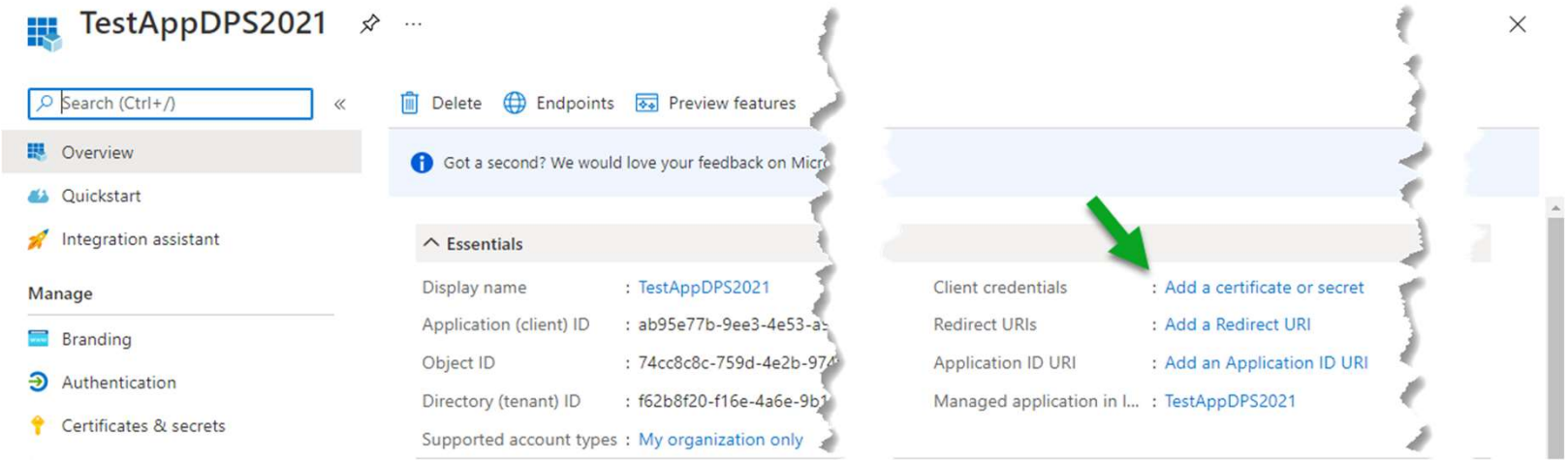
Create an App

AAD → Enterprise Applications → New Application



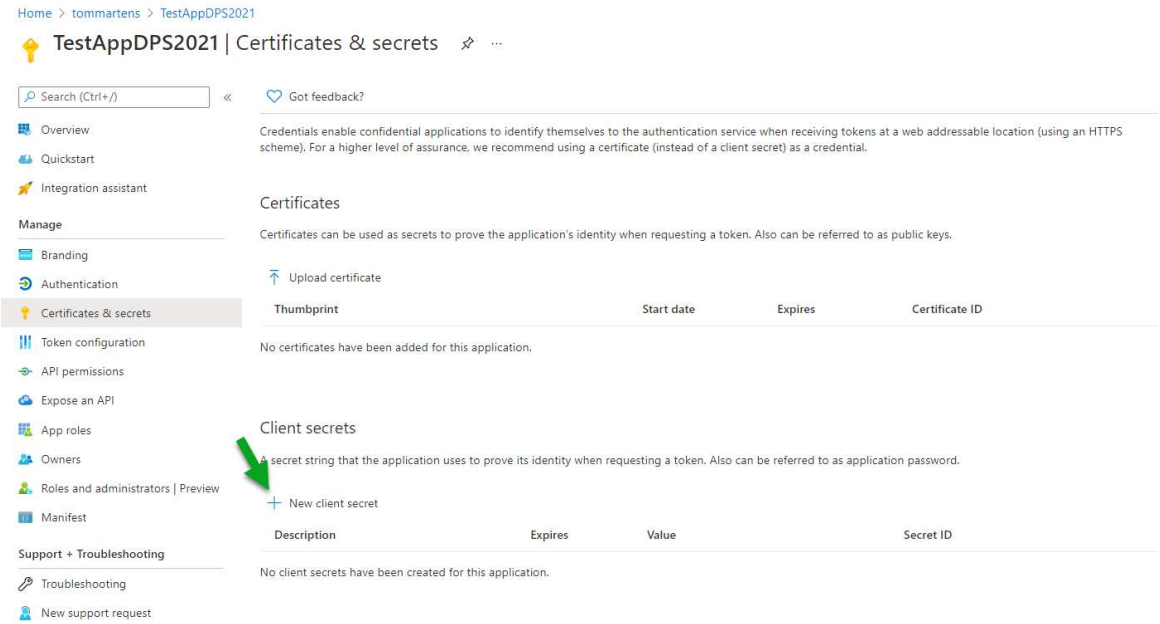
Create the App Secret

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



Create the App Secret

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

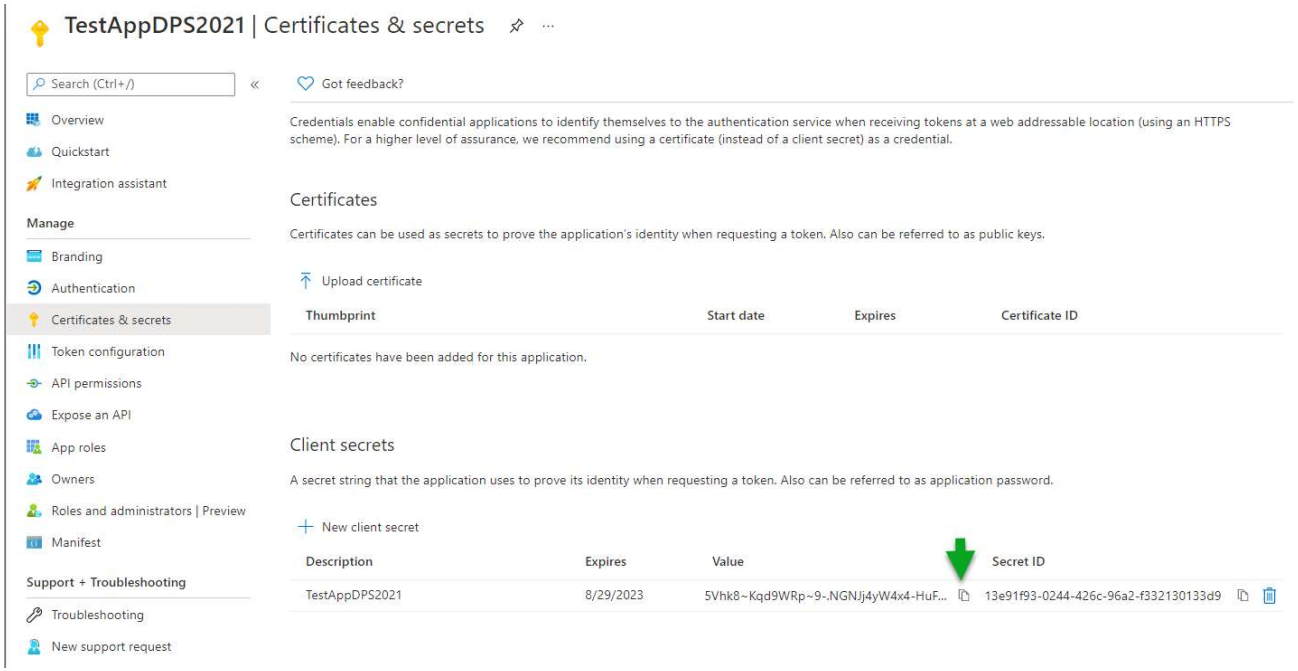


Create the App Secret

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

Don’t forget the secret!!!!

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



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

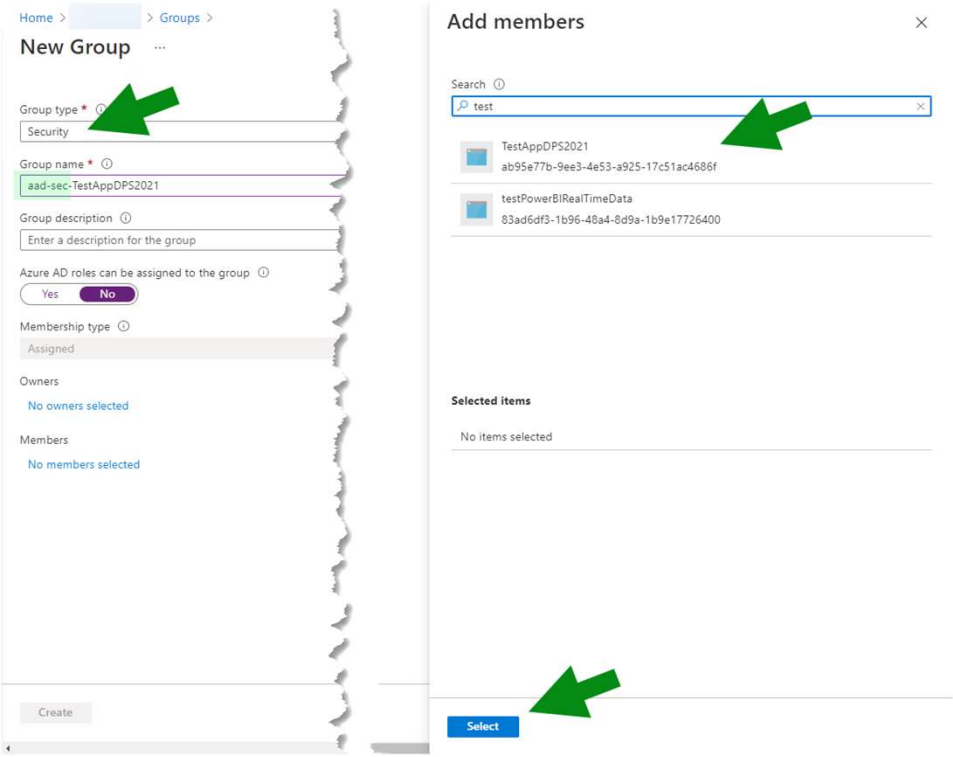
Demo: Creating a Security Group

Find the files

https://github.com/tomatminceddata/thehive

Create a Security Group

AAD → Groups → New Group



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 ✕ aad-sec

aad-sec-readPBImetadata

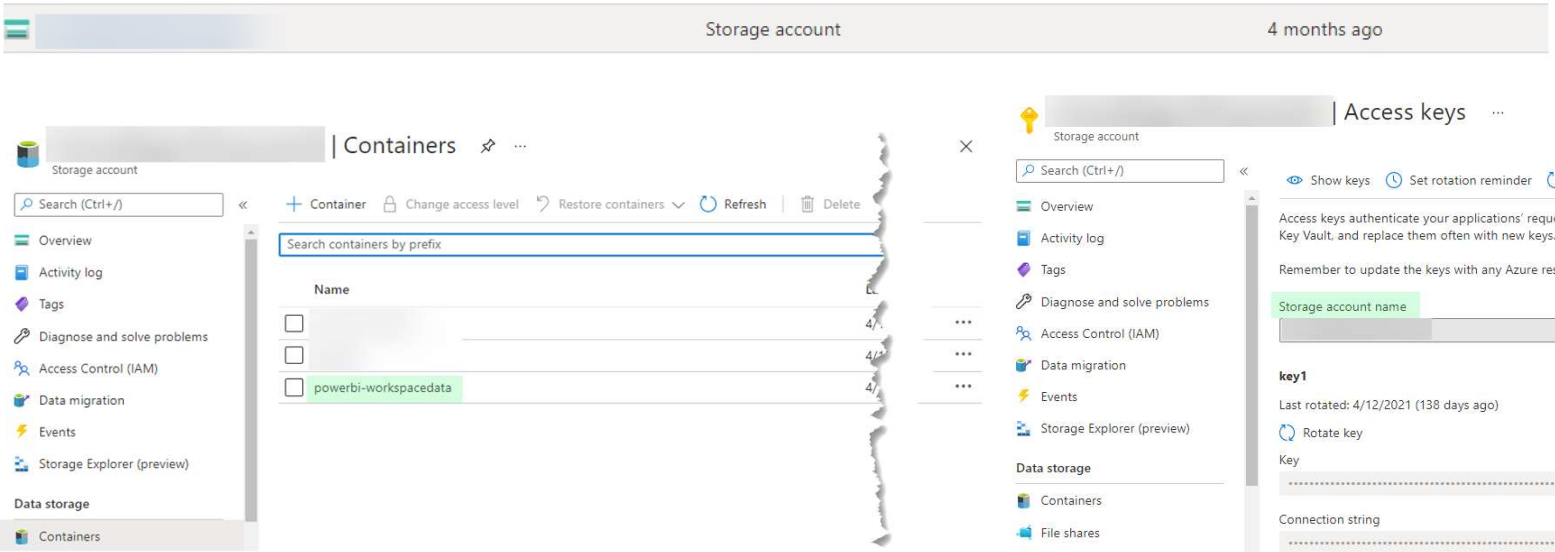
aad-sec-TestAppDPS2021

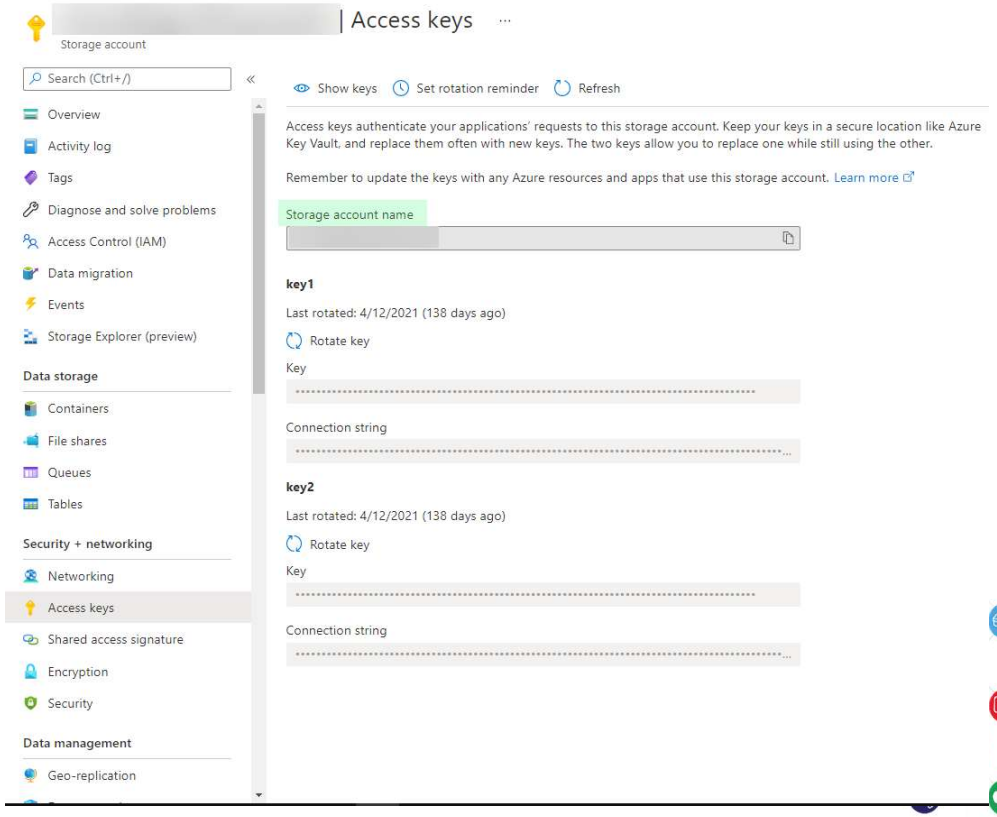
THE ARCHITECTURE

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





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 alle 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 alle 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

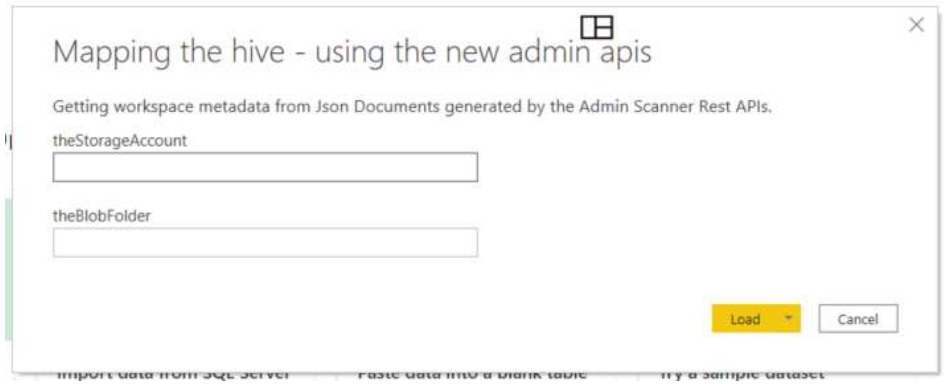
Find the files

https://github.com/tomatminceddata/thehive

Power BI – Import the template file

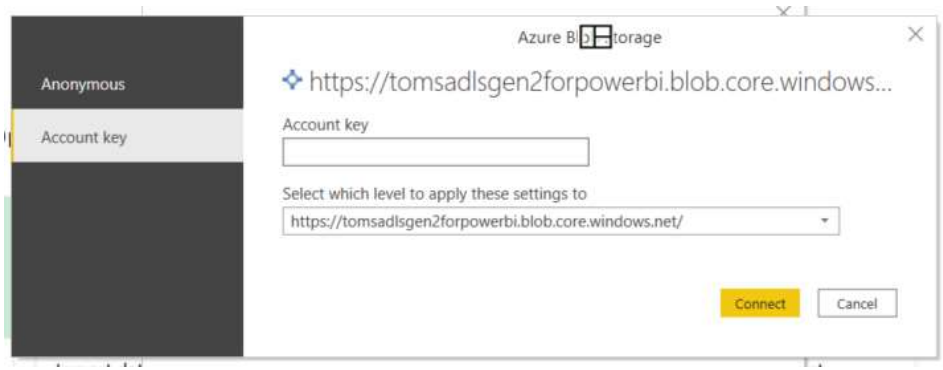
Connecting Power BI to the blob store requires two parameters

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



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

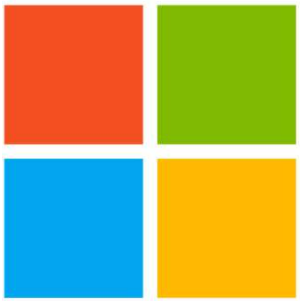
Recap

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**

DATA PLATFORM
VIRTUAL SUMMIT 2021

Accelerating Data-Driven Success