Lab 3 - Linked Services

Requirements

To start the lab, it's essential that Lab2 has been completed.

Objective

To allow data to flow over the recently created IRs, connections to the respective services must be made. During the lab, you will establish multiple connections, with e.g.:

- a SQL database (e.g. a source system or Data Warehouse)
- a Storage account (e.g. like a Data Lake)
- a File system (e.g. a share)

Some of these sources can be accessed with the help of *managed identity*: in this case, AAD grants rights to the Data Factory. Other sources you will have to access with a *secret*, such as a certificate or a username/password. These *secrets* are centrally stored in Azure in the Key Vault. From there, you can easily determine which services can view which *secrets*.

Task 1 - Azure Key Vault

Azure Data Factory can be easily linked with Azure Key Vault, where we store passwords and connection strings. We can have a connection to a source filled by a *secret* from the *Key Vault*. At the moment that ADF makes a connection with that source, ADF will first retrieve the *secret* from the Key Vault.

Before we can access *secrets* from the Key Vault, we will have to attach the Key Vault as a *Linked Service* first.

1. Go back to the **unlinked** ADF. Then click on Manage again. Go to **Linked Services**.
2. Click on **New**, and search for **Key vault**. Click on the **Azure Key vault**.
3. Give the Linked services a clear name. The recommended format is to start with LS_, the name of the service in your resource group and ending with _environment.
    - Practical example: `LS_KV_Dataplatform_PRD`
    - Training example: `LS_KV_rcc4bh5724jim_Training`
      In the naming, a dash (-) is not allowed. An *underscore* (_) is possible.
4. Choose the **Azure Subscription** that you are using in the training.
5. At **Azure Key vault Name** choose the key vault from your Key Vault (this starts with `kv_`).
6. Click on the **Test Connection** button to validate that the connection can be established. If this goes wrong, let the trainer know.
7. When the test is complete and a **Green dot** appears, the Linked Service can be created by clicking on **Create**.
8. The Linked Service to the Azure Key Vault has now been created, but it has not been published yet. Click on the **Blue button** with the text **Publish all** and then on the **Publish** button. By publishing, the changes go live, and the Key Vault can be used.

Task 2 - Databases

With the Key Vault connected, it is possible to retrieve passwords to set up a secure connection with, for example, the databases.

1. Click on **New**, and search for **SQL**. Double-click the **Azure SQL Databases**.
2. Give the Linked services a clear name, for example `LS_sqldb_source`
3. Choose at **Connect via integration runtime** your own made **Azure IR**.
4. Choose at **Server Name** the Server name as it appears in your resource group.
5. Choose at **Database Name** the source Database name as it appears in your resource group. The source database starts with **sqldb-source-** as a name.
6. Fill in the **User Name** with the SQL admin account named: **sqladmin**.
7. For the option between **Password** and **Azure Key Vault**, choose the Key vault.
8. Choose at **AKV linked service** the previously created Key Vault Linked Service.
9. Choose at **Secret Name** the option **sqladmin**
10. Click on the **Test Connection** button to validate that the connection can be established. If this goes wrong, let the trainer know.
11. When the test is complete and a **Green dot** appears, the Linked Service can be created by clicking on **Create**.
12. Repeat Task 2, but now for the **sqldb-target** Database.

You have now created two Linked Services. This enables ADF to connect to the two databases.

Task 3 - Storage Account

The second source we add is a Storage Account. We can use this, for example, as a *landing zone* for the data, or as a Data Lake.

1. Click on **New**, and search for **storage**. Click on the **Azure Blob Storage**.
2. Give the Linked services a clear name.
3. Choose at **Connect via integration runtime** your own made **Azure IR**.
4. Choose at **Storage account name** the storage account as it appears in your resource group.
5. Click on the **Test Connection** button to validate that the connection can be established. If this goes wrong, let the trainer know.
6. When the test is complete and a **Green dot** appears, the Linked Service can be created by clicking on **Create**.

The rights on the Storage Account are distributed via Azure AD. So you didn't have to use a *secret* for this.

Task 4 - File system

The third source we add is an on-premises filesystem. Because the filesystem is on-premises, we need to use the correct Integration Runtime! Also, this VM is not in our domain, so we need to indicate which username/password we will log in with.

1. Click on **New**, and search for **file**. Click on the **File system**.
2. Give the Linked services a clear name.
3. Choose at **Connect via integration runtime** the **Self-Hosted IR**.
4. Fill in the **Host** with the following **D:**
5. Fill in the **User Name** with the SQL admin account named: **sqladmin**.
6. For the option between **Password** and **Azure Key Vault**, choose the Key vault.

7. Choose at **AKV linked service** the previously created Key Vault Linked Service.
8. Choose at **Secret Name** the option **sqladmin**
9. Click on the **Test Connection** button to validate that the connection can be established. This will fail, but we'll fix that in a moment!
10. When the test is complete and a **Green dot** appears, the Linked Service can be created by clicking on **Create**.
11. Click on the **Blue button** with the text **Publish all** and then on the **Publish** button.

> ## Why does the connection with the File System fail
>
> Since version 5.22, the Self-Hosted Integration Runtime has stricter security measures. One of these measures is that a self-hosted IR cannot just access local files.
>
> In the training, however, we want to access these files to simulate that an on-premises source is present. Therefore, we will have to turn off this security measure. Here's how:
>
> 1. Connect to the VM
> 2. Open the start menu and type **Powershell**
> 3. Choose **Run as administrator**
>   In Powershell, you now enter the following:
> 4. `cd 'C:\Program Files\Microsoft Integration Runtime\5.0\Shared\'`
> 5. `.\dmgcmd -DisableLocalFolderPathValidation`
>
> The setting is now adjusted, and the IR restarts. The Linked Service should now work **Test Connection** to `D:\` should work.

Table of Contents