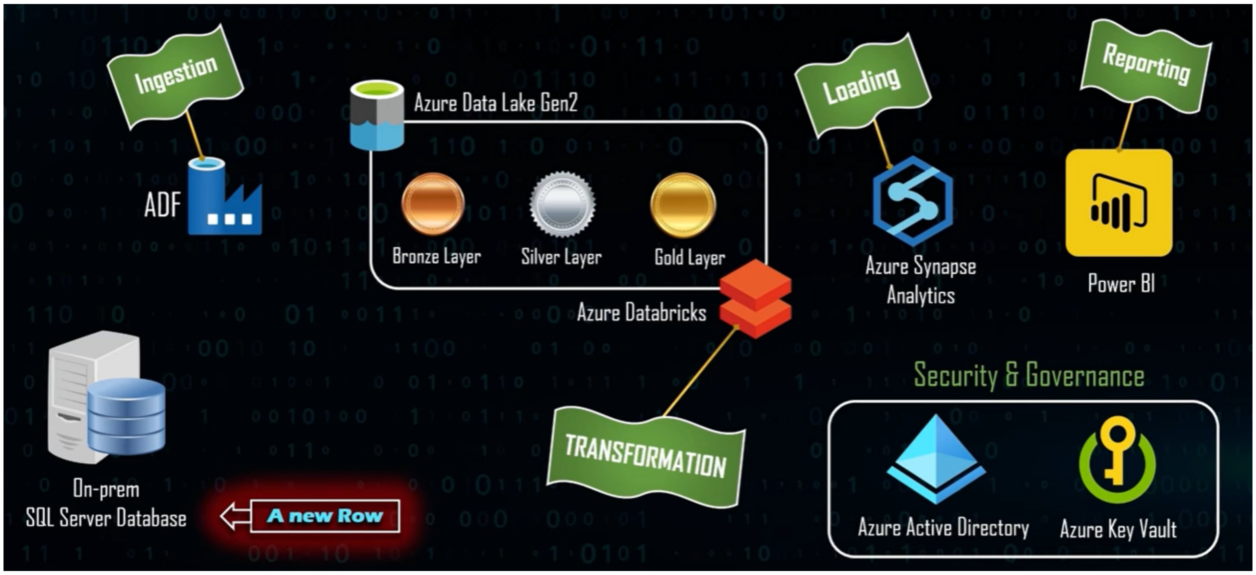
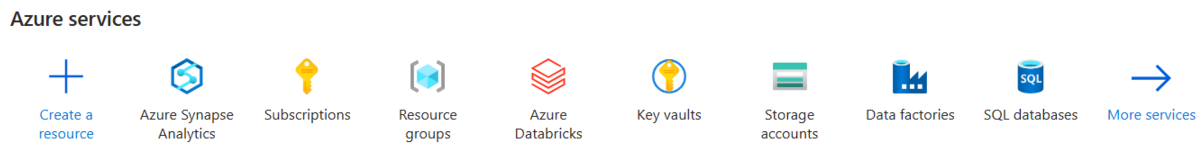
# End to End Data Engineering Project

| | |
|---|---|
| Created by | Ⓓ Devansh Gupta |
| Created time | @September 7, 2023 10:34 AM |



## STEP 1: Creation of Resource Group

Create a resource group by selecting the Resource Groups option in Azure Services. Name it according to convention.



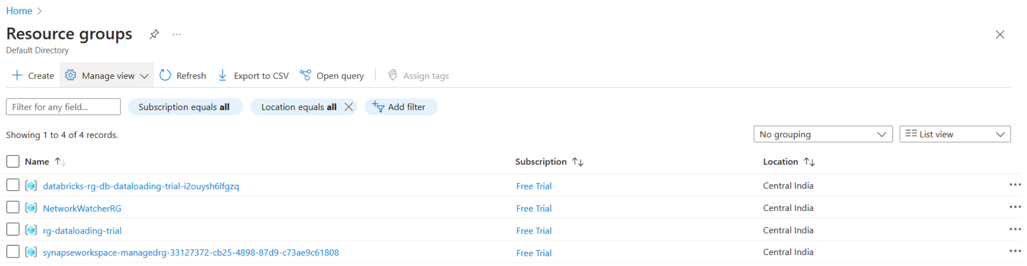
Then add all the below listed resources in the resource group -

1. Azure Data Factory Resource
2. Azure Databricks
3. Azure Data Lake Storage Gen 2
4. Azure Key Vault
5. Azure Synapse Analytics (it is better to add this later)

Many of these services are chargeable and thus, should be used carefully.

## I .Creating resource groups

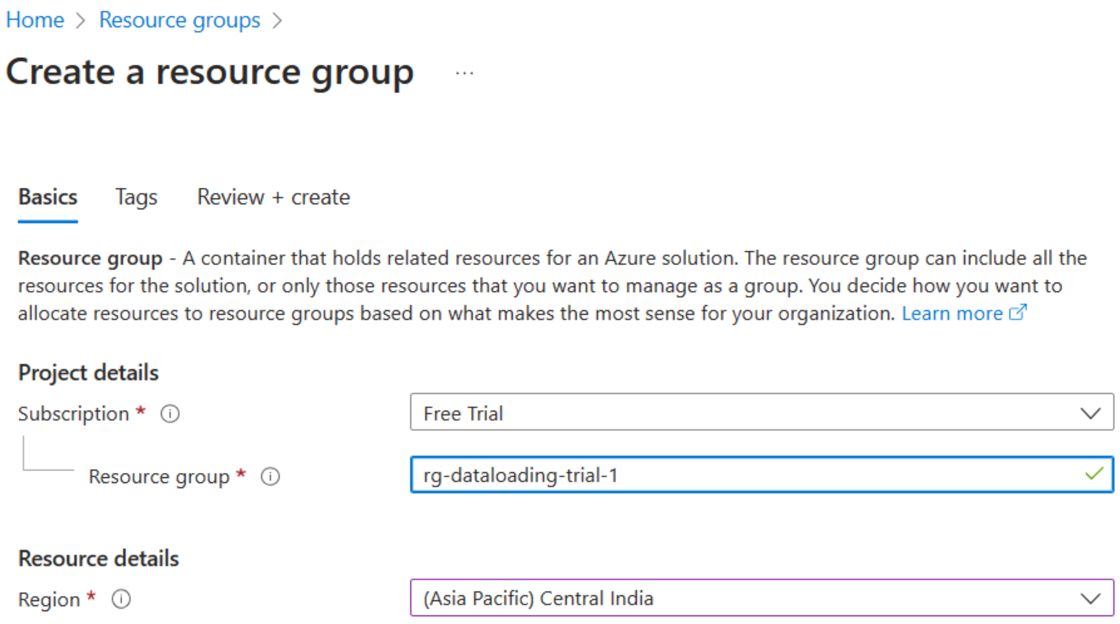
1. Sign in to the Azure portal.
2. Select **Resource groups**.
3. Select **Create**



Showing 1 to 4 of 4 records.

| Name | Subscription | Location |
|---|---|---|
| databricks-rg-db-dataloading-trial-i2ouysh6lfgzq | Free Trial | Central India |
| NetworkWatcherRG | Free Trial | Central India |
| rg-dataloading-trial | Free Trial | Central India |
| synapseworkspace-managedrg-33127372-cb25-4898-87d9-c73ae9c61808 | Free Trial | Central India |

4. Enter the following values:
    - **Subscription**: Select your Azure subscription.
    - **Resource group**: Enter a new resource group name.
    - **Region**: Select an Azure location, such as **Central India**



Create a resource group ...

**Resource group** - A container that holds related resources for an Azure solution. The resource group can include all the resources for the solution, or only those resources that you want to manage as a group. You decide how you want to allocate resources to resource groups based on what makes the most sense for your organization. Learn more

**Project details**

Subscription * : Free Trial

Resource group * : rg-dataloading-trial-1

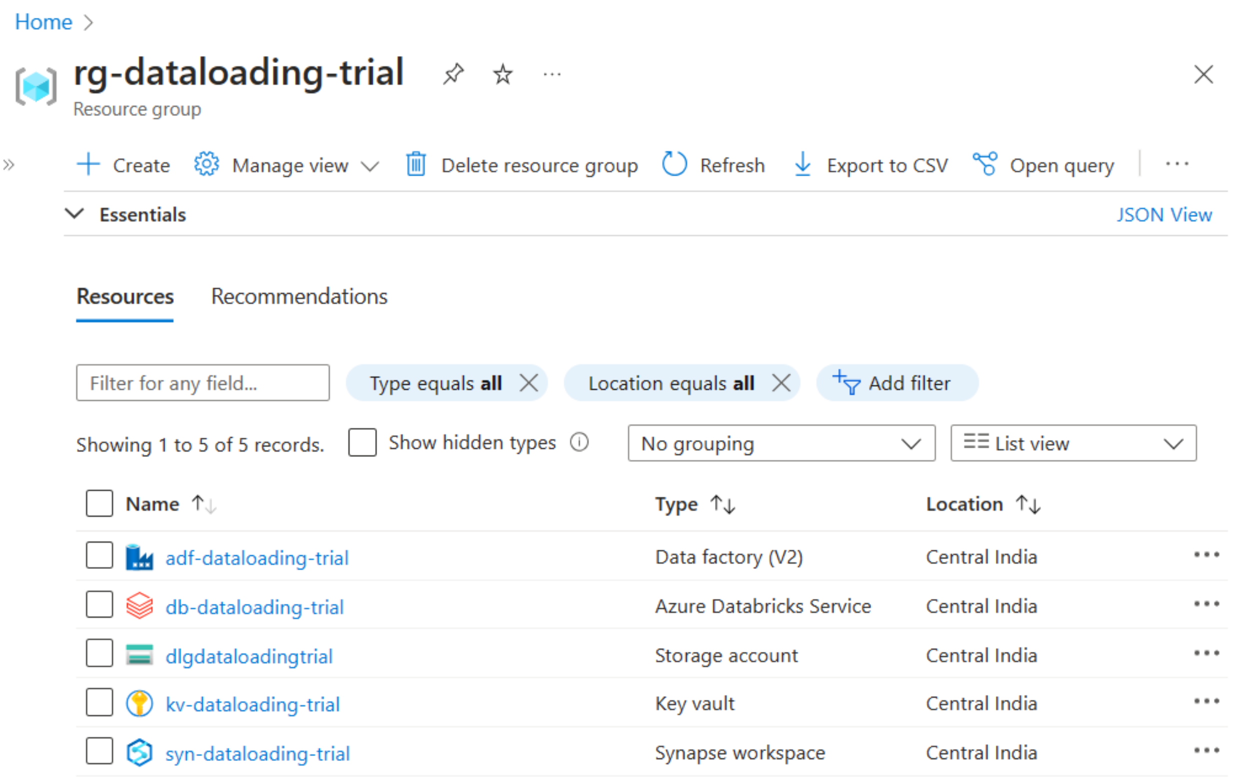
**Resource details**

Region * : (Asia Pacific) Central India

5. Select **Review + Create**
6. Select **Create**. It takes a few seconds to create a resource group.

## II.Adding Resouces to Resource Groups

Add all resources as shown below -



Home >

rg-dataloading-trial

Resource group

Essentials

Resources | Recommendations

Filter for any field... | Type equals **all** | Location equals **all** | Add filter

Showing 1 to 5 of 5 records. □ Show hidden types | No grouping | List view

| Name | Type | Location |
|---|---|---|
| adf-dataloading-trial | Data factory (V2) | Central India |
| db-dataloading-trial | Azure Databricks Service | Central India |
| dlgdataloadingtrial | Storage account | Central India |
| kv-dataloading-trial | Key vault | Central India |
| syn-dataloading-trial | Synapse workspace | Central India |

The settings that should be applied while creating each resource are plenty straightforward as should be applied correctly. Videos on YouTube can be referred to see the correct configuration of each resource.

To create a Azure Key Vault -

Quickstart - Create an Azure Key Vault with the Azure portal

Quickstart showing how to create an Azure Key Vault using the Azure portal

https://learn.microsoft.com/en-us/azure/key-vault/general/quick-create-portal

Microsoft Learn

To create a Azure DataLake Gen2 -

Create a storage account for Azure Data Lake Storage Gen2 - Azure Storage

Learn how to create a storage account for use with Azure Data Lake Storage Gen2.

https://learn.microsoft.com/en-us/azure/storage/blobs/create-data-lake-storage-account

Microsoft Learn

To create a Azure Datafactory Resource -

Create an Azure Data Factory - Azure Data Factory

Learn how to create a data factory using UI from the Azure portal.

https://learn.microsoft.com/en-us/azure/data-factory/quickstart-create-data-factory

Microsoft Learn

To create a Azure Databricks Service -

Get started: Account and workspace setup - Azure Databricks

Learn how to set up a Databricks free trial and a cloud provider account with Azure.

https://learn.microsoft.com/en-us/azure/databricks/getting-started/

Microsoft Learn

To create a Azure Synapse Analytics Workspace -

Quickstart: create a Synapse workspace - Azure Synapse Analytics

Create an  Synapse workspace by following the steps in this guide.

https://learn.microsoft.com/en-us/azure/synapse-analytics/quickstart-create-workspace

Microsoft Learn

# STEP 2: Setting up the environment

## I .Adding database to SQL Server Management Studio

1. Setup SQL Server Management Studio by first installing SQL Server and then SQL Server Management Studio from the internet. Appropriate videos can be referred to from YouTube and the internet.

This section provides direct link to download `AdventureWorks` sample databases, and instructions for restoring them to SQL Server, Azure SQL Database, and Azure SQL Managed Instance.

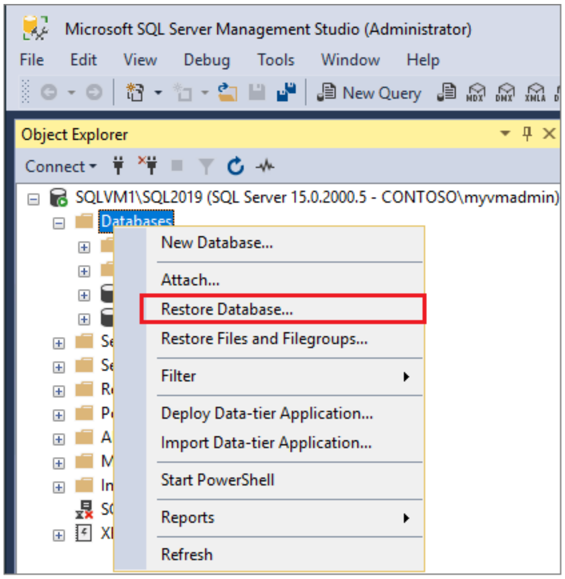
https://github.com/Microsoft/sql-server-samples/releases/download/adventureworks/AdventureWorksLT2017.bak

You can use the `.bak` file to restore your sample database to your SQL Server instance.

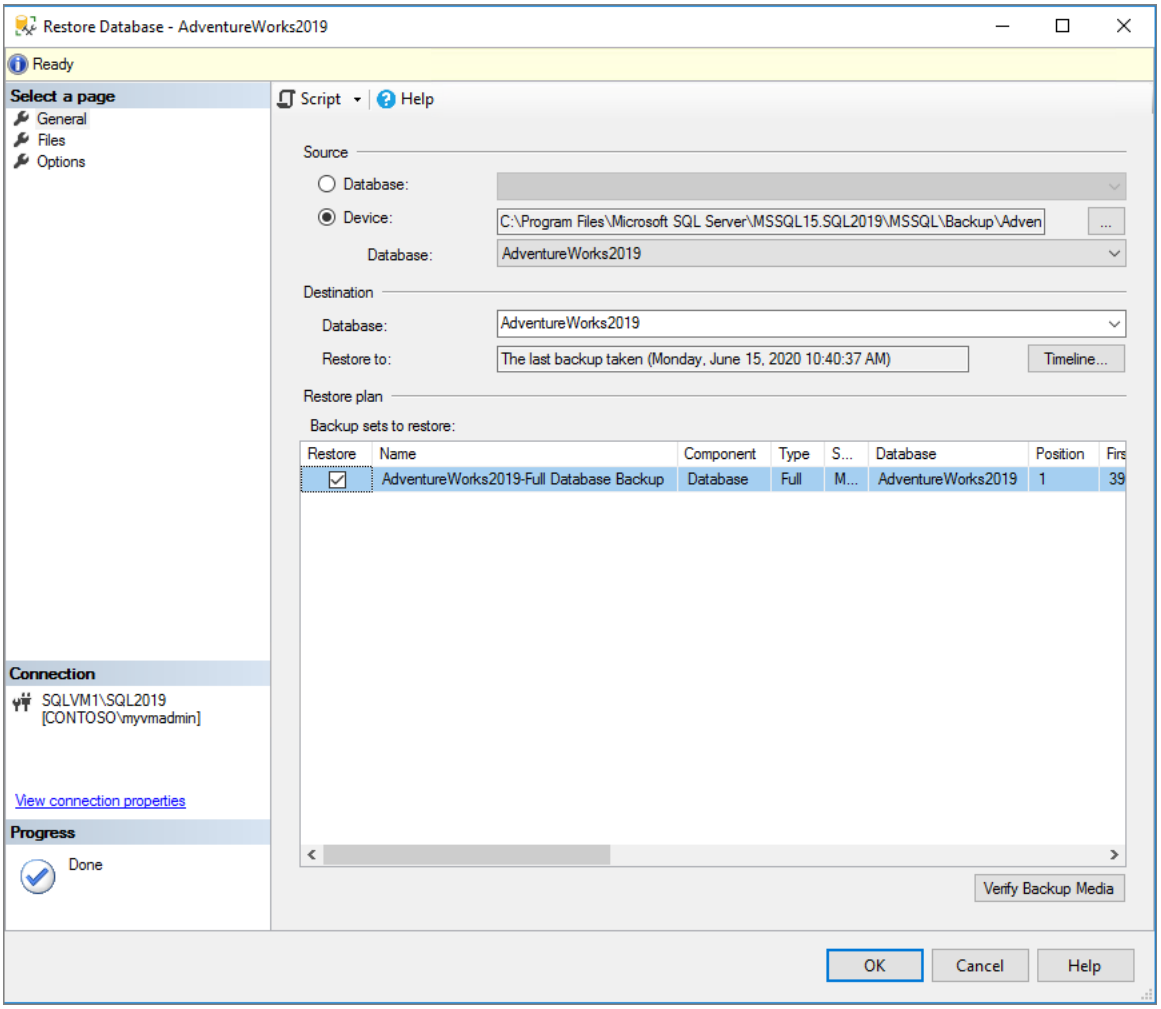
To restore your database in SSMS, follow these steps:

1. Download the appropriate `.bak` file from the link provided above.
2. Move the `.bak` file to your SQL Server backup location. This varies depending on your installation location, instance name and version of SQL Server. For example, the default location for a default instance of SQL Server 2019 (15.x) is:

   `C:\Program Files\Microsoft SQL Server\MSSQL15.MSSQLSERVER\MSSQL\Backup`.
3. Open SSMS and connect to your SQL Server instance.
4. Right-click **Databases** in **Object Explorer** > **Restore Database...** to launch the **Restore Database** wizard.


Microsoft SQL Server Management Studio (Administrator)
File
Edit
View
Debug
Tools
Window
Help
New Query
Object Explorer
Connect
SQLVM1\SQL2019 (SQL Server 15.0.2000.5 - CONTOSO\myvmadmin)
Databases
New Database...
Attach...
Restore Database...
Restore Files and Filegroups...
Filter
Deploy Data-tier Application...
Import Data-tier Application...
Start PowerShell
Reports
Refresh

5. Select **Device** and then select the ellipses **(...)** to choose a device.
6. Select **Add** and then choose the `.bak` file you recently moved to the backup location. If you moved your file to this location but you're not able to see it in the wizard, this typically indicates a permissions issue - SQL Server or the user signed into SQL Server doesn't have permission to this file in this folder.
7. Select **OK** to confirm your database backup selection and close the **Select backup devices** window.
8. Check the **Files** tab to confirm the **Restore as** location and file names match your intended location and file names in the **Restore Database** wizard.
9. Select **OK** to restore your database.



Source

○ Database:

● Device: C:\Program Files\Microsoft SQL Server\MSSQL15.SQL2019\MSSQL\Backup\Adven

Database: AdventureWorks2019

Destination

Database: AdventureWorks2019

Restore to: The last backup taken (Monday, June 15, 2020 10:40:37 AM)

Restore plan

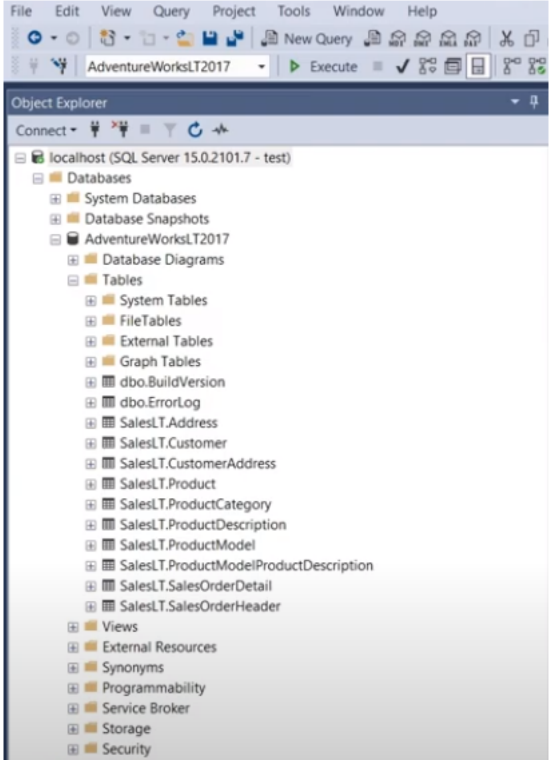
Backup sets to restore:

| Restore | Name | Component | Type | S... | Database | Position | Firs |
|---|---|---|---|---|---|---|---|
| ☑ | AdventureWorks2019-Full Database Backup | Database | Full | M... | AdventureWorks2019 | 1 | 39 |

SQLVM1\SQL2019
[CONTOSO\myvmadmin]

Done

Doing so will import a sample database by the name - AdventureWorksLT2017 to your SQL Server Management Studio.



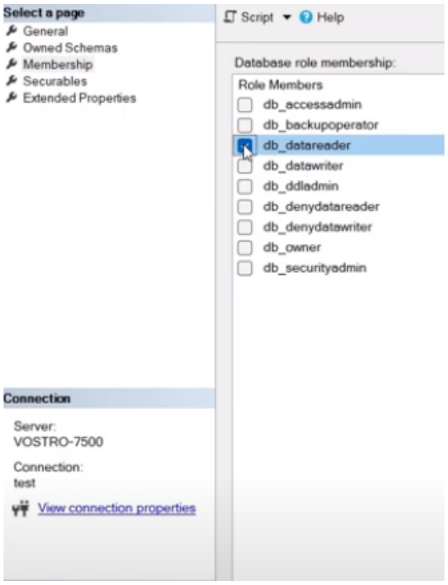
## II.Creating Login and Password for the Database

1. Create a SQL Notebook file with the following script -

```
CREATE LOGIN <username> WITH PASSWORD = <'password'>
create user devansh for login devansh
```

   Open this in the SQL Server Management Studio and run this to create a username and password for accessing the database.

2. To check if the above user with the username and password have been assigned to the database, open the Security → Users → Username. Write click on the username and then click on Properties and under Membership tab, assign db_datareader to the user.



Database role membership:

Role Members

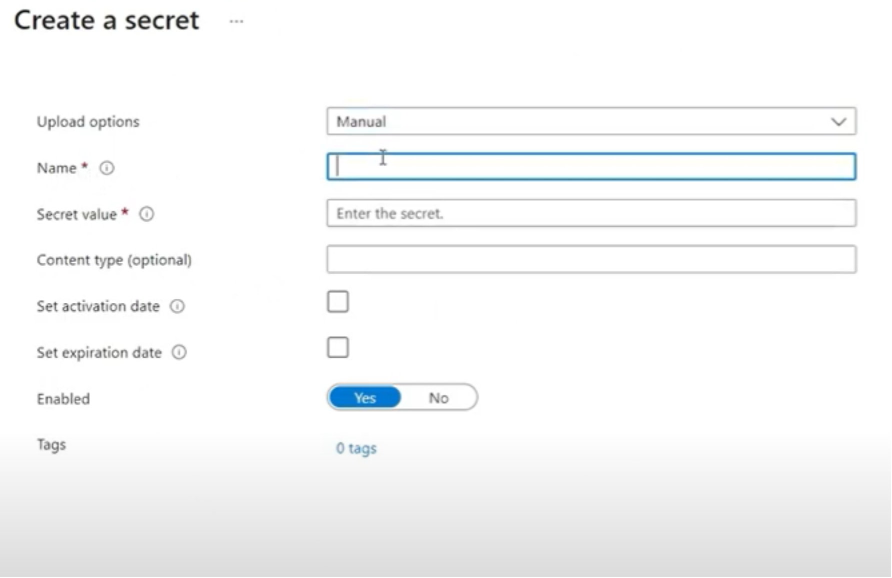
- □ db_accessadmin
- □ db_backupoperator
- ☑ db_datareader
- □ db_datawriter
- □ db_ddladmin
- □ db_denydatareader
- □ db_denydatawriter
- □ db_owner
- □ db_securityadmin

Connection

Server:
VOSTRO-7500

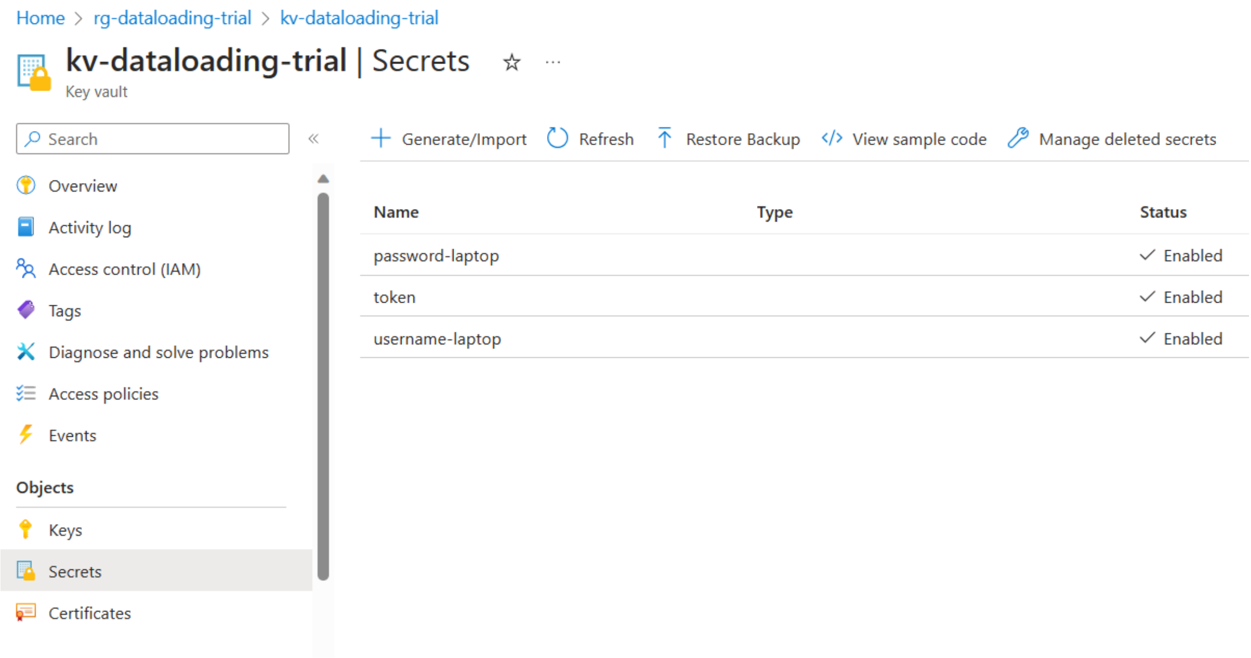
Connection:
test

3. Also now add the username and password as secrets in the Azure Key Vault that you have created above in the resource group.



Create a secret

| | |
|---|---|
| Upload options | Manual |
| Name * | |
| Secret value * | Enter the secret. |
| Content type (optional) | |
| Set activation date | □ |
| Set expiration date | □ |
| Enabled | [✓ Yes] No |
| Tags | 0 tags |



kv-dataloading-trial | Secrets

| Name | Type | Status |
|---|---|---|
| password-laptop | | ✓ Enabled |
| token | | ✓ Enabled |
| username-laptop | | ✓ Enabled |

## STEP 3: Data Ingestion

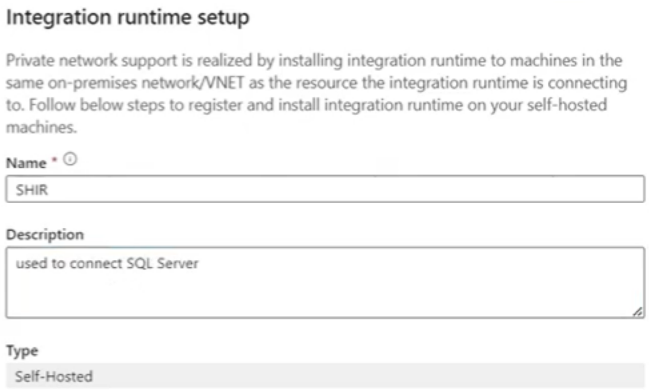
1. Launch Azure Data Factory Workspace from the Data Factory that you have created in the resource group. Here go to Manage tab → Integration Runtimes → And then click on New.

   The already existing Integration Runtime which is AutoResolveIntegrationRuntime is used to connect cloud components to each other.

2. But in order to connect the on-premises SQL Database (AdventureWorksLT2017), we need to create a new Integration Runtime.

   On clicking New → Azure, Self-Hosted → Self-Hosted → Set the name as “SHIR” → Create.

   This creates the new Integration Runtime.



Integration runtime setup

Private network support is realized by installing integration runtime to machines in the same on-premises network/VNET as the resource the integration runtime is connecting to. Follow below steps to register and install integration runtime on your self-hosted machines.

Name *: SHIR

Description: used to connect SQL Server

Type: Self-Hosted

3. We get two links - one for Express Setup and one for Manual Setup.

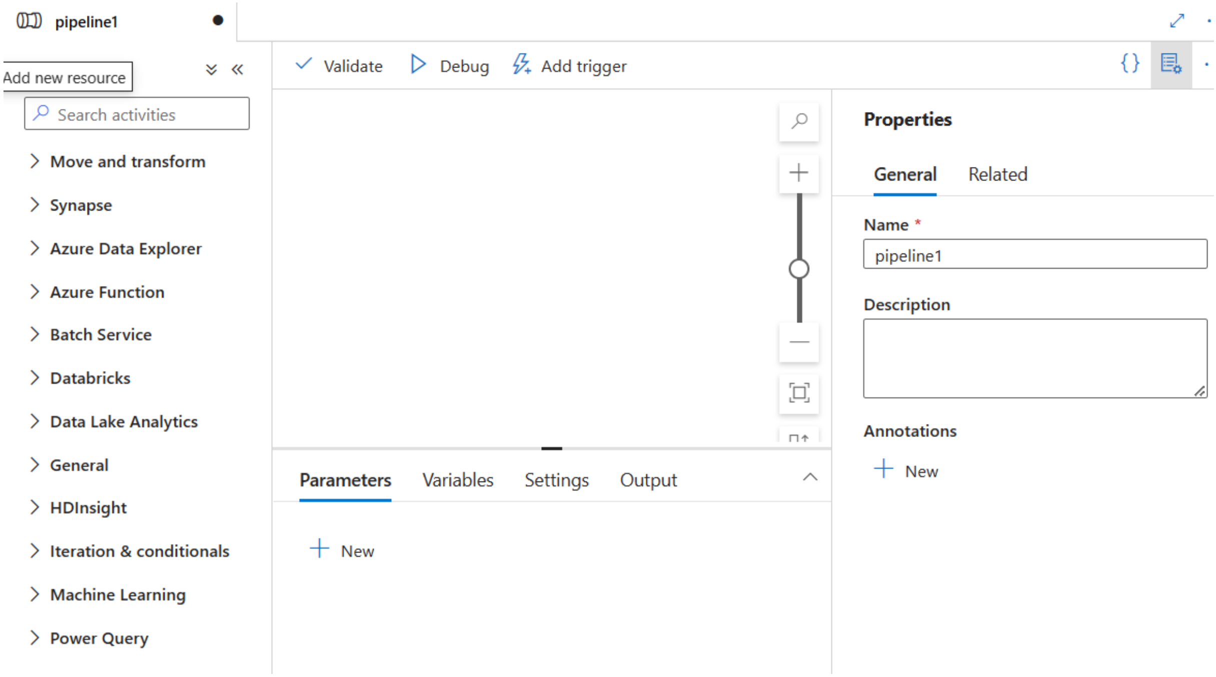
Download the Express Setup as it is more convenient, and it will finish downloading in some time. This is the set-up for a new Integration Runtime that connects cloud to the machine that the on-premises database is present on.

## Ⅰ.Creating Pipeline to Copy only one table.

1. To create a new pipeline -

   Author tab → Click on "+" icon → Select Pipeline → This creates a new pipeline.

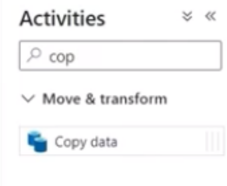
   You can rename it if you want.



Name * pipeline1

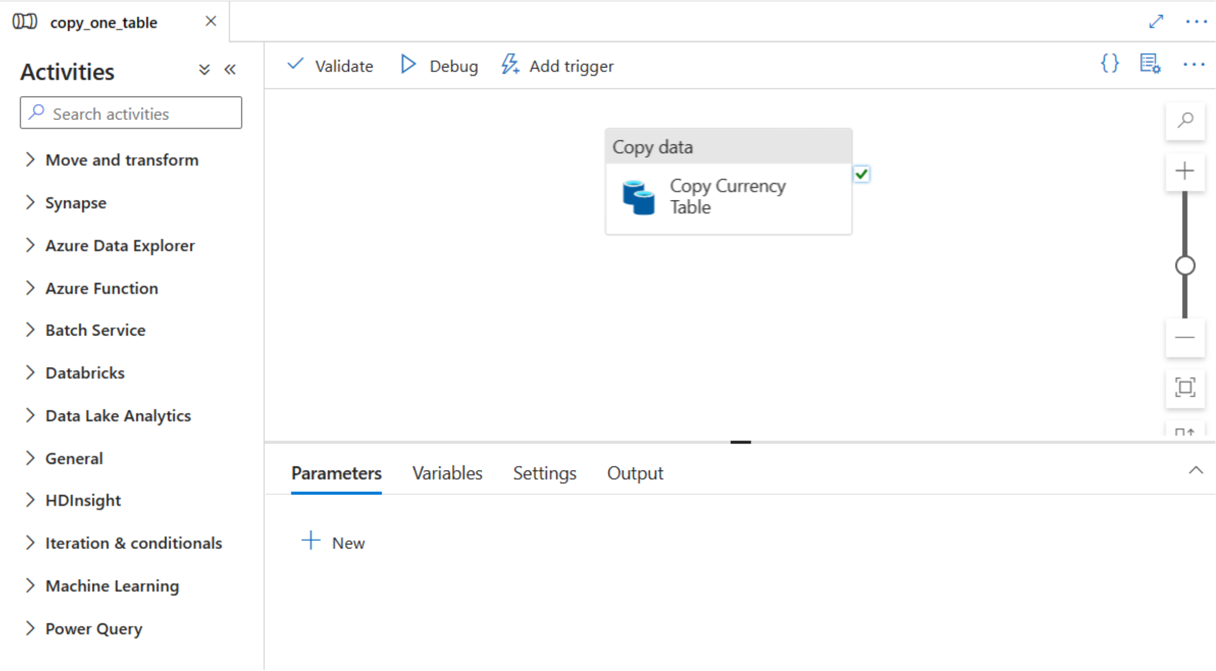
Description

2. Now go to Activities and search for Copy data -



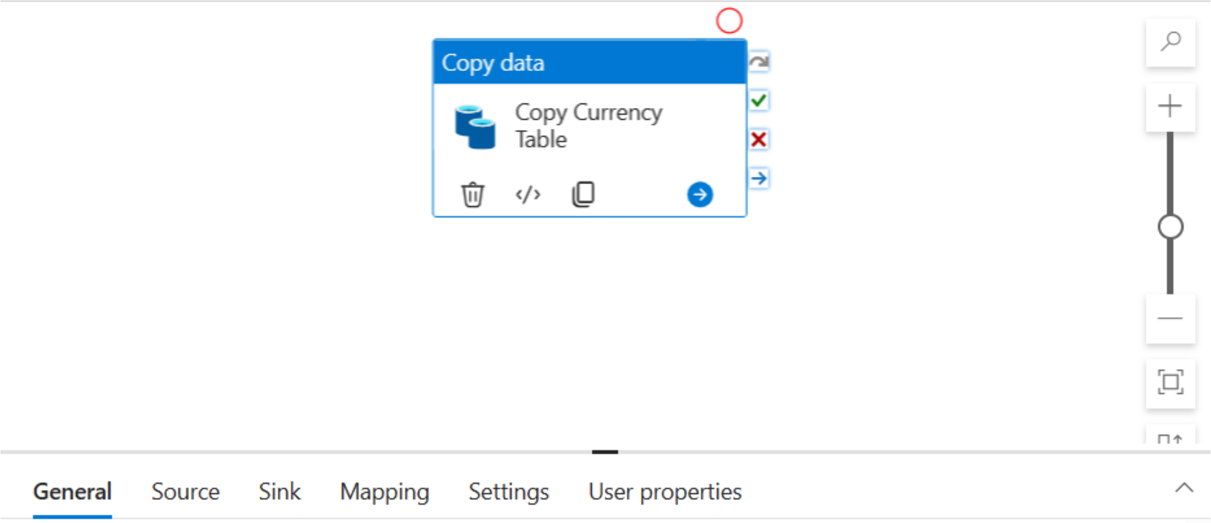
cop

Now drag and drop this Activity in the workspace that you have been provided with.

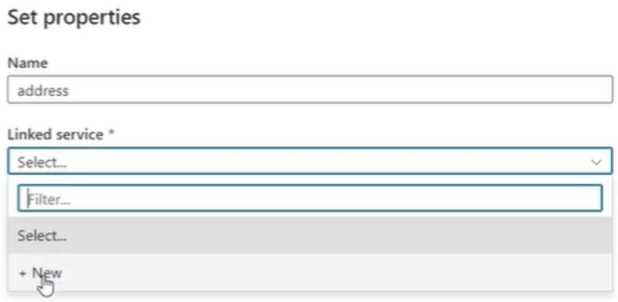


Also change the name of the Activity to make it more convenient to understand what it is doing.

3. To configure the activity, you have to configure the Source and Sink tabs respectively.



For Source dataset, click on +New → Then search for SQL Server → And name it as per your convenience. But the option below it - Linked Services have to be configured as done below -



Set properties

Name

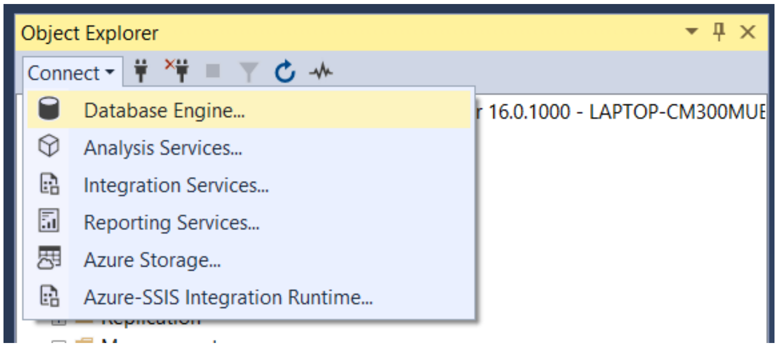
address

Linked service *

Select...

## Configuring Linked Service

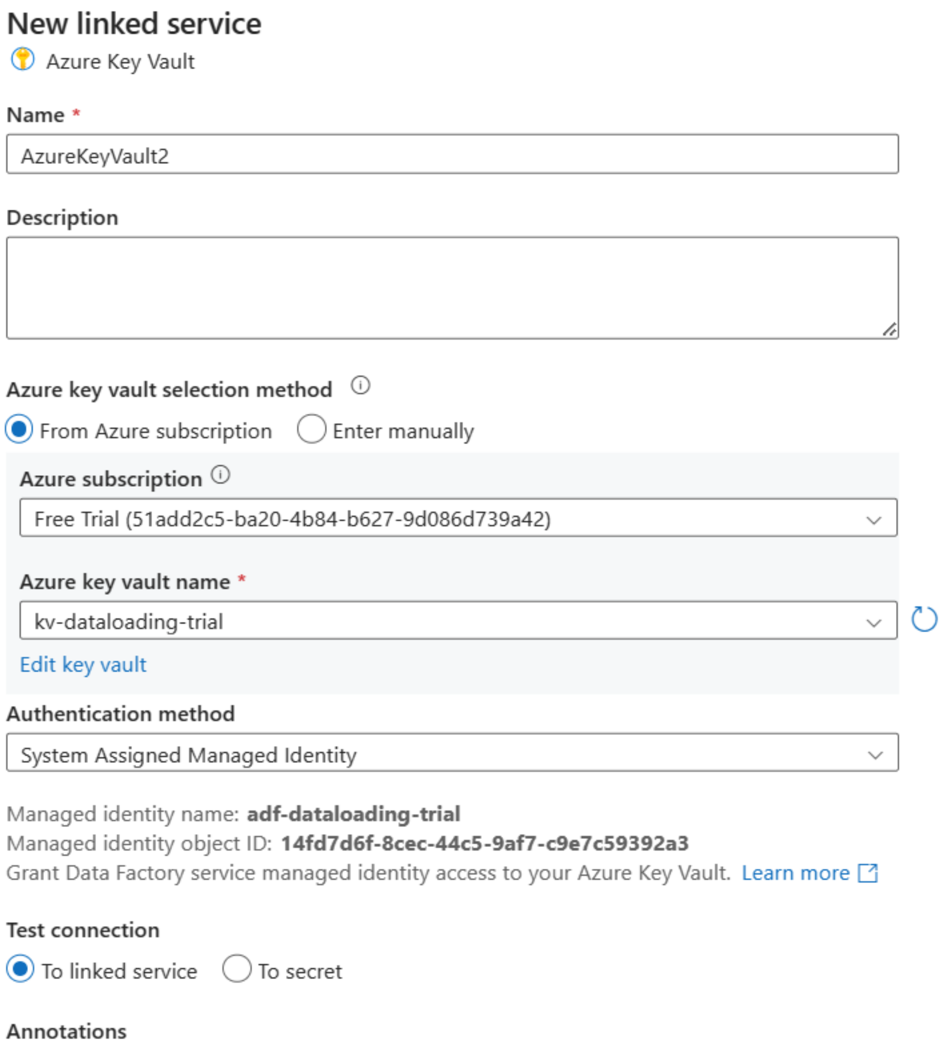
1. Linked Service can be named as per convenience. This ensures connectivity of cloud and the on-premises database.
2. For the option Connect via integration runtime, Select SHIR as it is used to connect cloud to on-premises machines.
3. Now for Server Name, this can be obtained from your SQL Server Management Studio from the Connect option -



4. Also add the Database name as - AdventureWorksLT2017.
5. Now select the Authentication type as SQL authentication:

   Also, enter the username as the one set when we added the database to our SQL Server Management Studio, this username is also stored in the Azure Key Vault along with the password.
6. For the password, use the Azure Key Vault to retrieve the password (as this ensures more safety).
   a. Here, you will have to create a AKV linked service.

      To do this fill all the fields while referring to below image-



New linked service

Azure Key Vault

Name *

AzureKeyVault2

Description

Azure key vault selection method

● From Azure subscription ○ Enter manually

Azure subscription

Free Trial (51add2c5-ba20-4b84-b627-9d086d739a42)

Azure key vault name *

kv-dataloading-trial

Edit key vault

Authentication method

System Assigned Managed Identity

Managed identity name: **adf-dataloading-trial**

Managed identity object ID: **14fd7d6f-8cec-44c5-9af7-c9e7c59392a3**

Grant Data Factory service managed identity access to your Azure Key Vault. Learn more

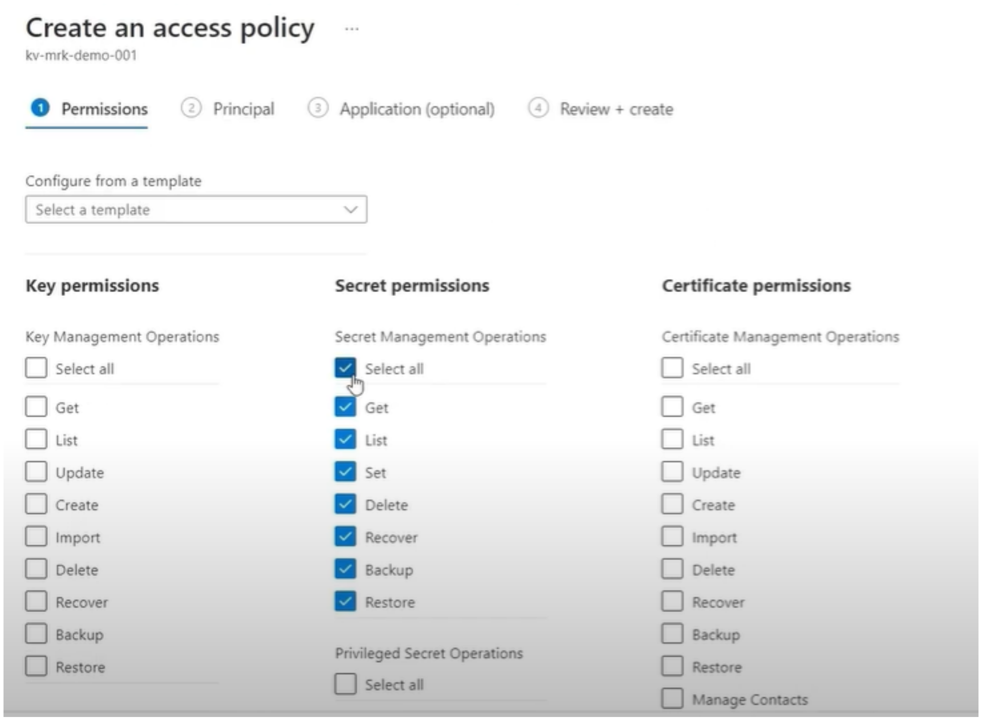
Test connection

● To linked service ○ To secret

Annotations

b. Now, after configuring the AKV linked state, Secret Name can be retrieved by clicking on the dropdown.

NOTE- Sometimes, you may not be able to retrieve the Secret Name due to accessibility issues. To resolve these, create an access policy in the Azure Key Vault and give your account all these permissions -



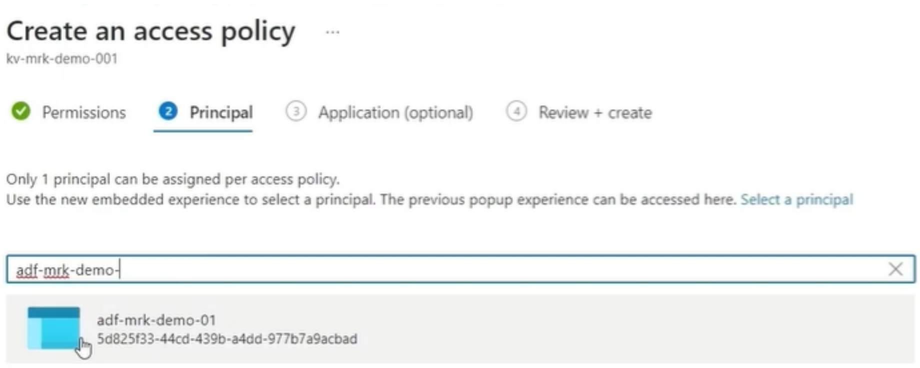
Create an access policy ...

kv-mrk-demo-001

Configure from a template

Select a template

| Key permissions | Secret permissions | Certificate permissions |
|---|---|---|
| Key Management Operations | Secret Management Operations | Certificate Management Operations |
| □ Select all | ☑ Select all | □ Select all |
| □ Get | ☑ Get | □ Get |
| □ List | ☑ List | □ List |
| □ Update | ☑ Set | □ Update |
| □ Create | ☑ Delete | □ Create |
| □ Import | ☑ Recover | □ Import |
| □ Delete | ☑ Backup | □ Delete |
| □ Recover | ☑ Restore | □ Recover |
| □ Backup | Privileged Secret Operations | □ Backup |
| □ Restore | □ Select all | □ Restore |
| | | □ Manage Contacts |



Create an access policy ...

kv-mrk-demo-001

Only 1 principal can be assigned per access policy.
Use the new embedded experience to select a principal. The previous popup experience can be accessed here. Select a principal

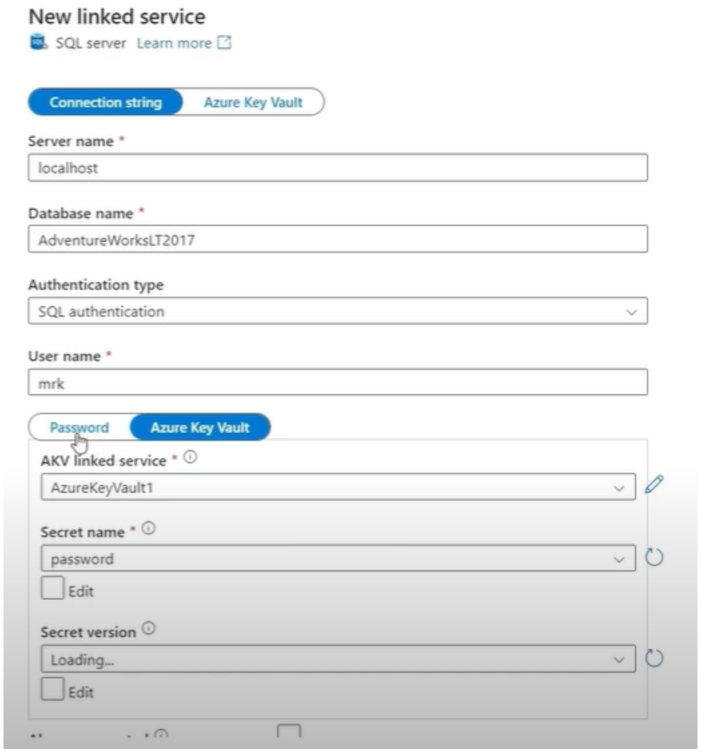
adf-mrk-demo-

adf-mrk-demo-01
5d825f33-44cd-439b-a4dd-977b7a9acbad

Select the name of your Azure Data Factory.

Doing so will let you access the Secret Name.

c. Final Linked service should look like this -



New linked service

SQL server Learn more

Connection string | Azure Key Vault

Server name *
localhost

Database name *
AdventureWorksLT2017

Authentication type
SQL authentication

User name *
mrk

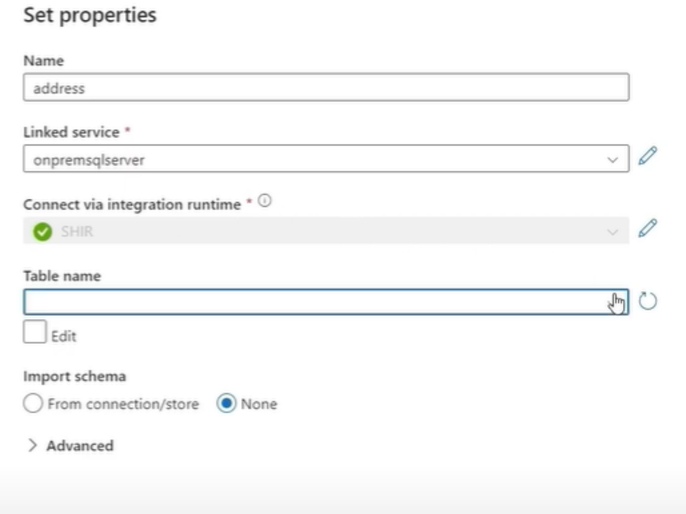
Password | Azure Key Vault

AKV linked service *
AzureKeyVault1

Secret name *
password
□ Edit

Secret version
Loading...
□ Edit

4. After configuring the linked service, the screen becomes something like this -



Set properties

Name
address

Linked service *
onpremsqlserver

Connect via integration runtime *
SHIR

Table name

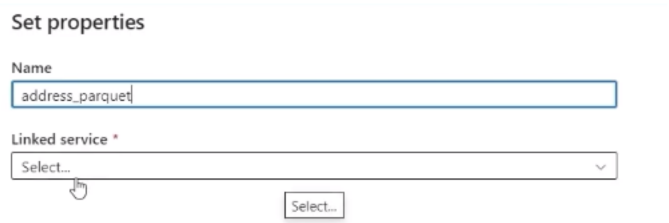
□ Edit

Import schema
○ From connection/store ● None

> Advanced

5. Now select the table name of the table that you want to import - like SalesLT.Address.

6. Similarly, configure the Sink dataset. For sink, Sink dataset → +New → Search for Azure Data Lake Storage Gen 2 → Select Parquet Format → Takes you to this page -



Set properties

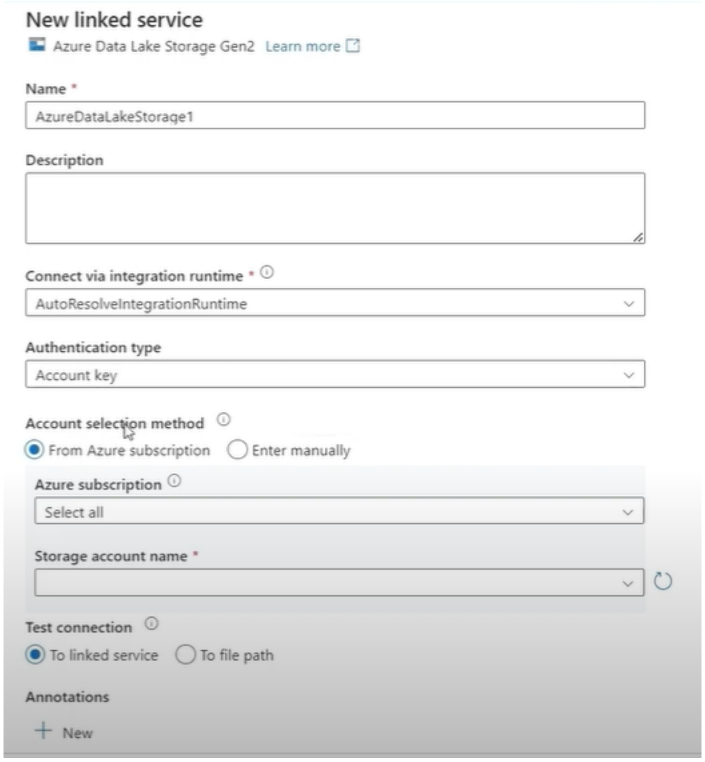
Name

address_parquet

Linked service *

Select...

Now similar to when we created Linked service for Source dataset, we do the same for Sink dataset.



New linked service

Azure Data Lake Storage Gen2

Name *

AzureDataLakeStorage1

Description

Connect via integration runtime *

AutoResolveIntegrationRuntime

Authentication type

Account key

Account selection method

● From Azure subscription ○ Enter manually

Azure subscription

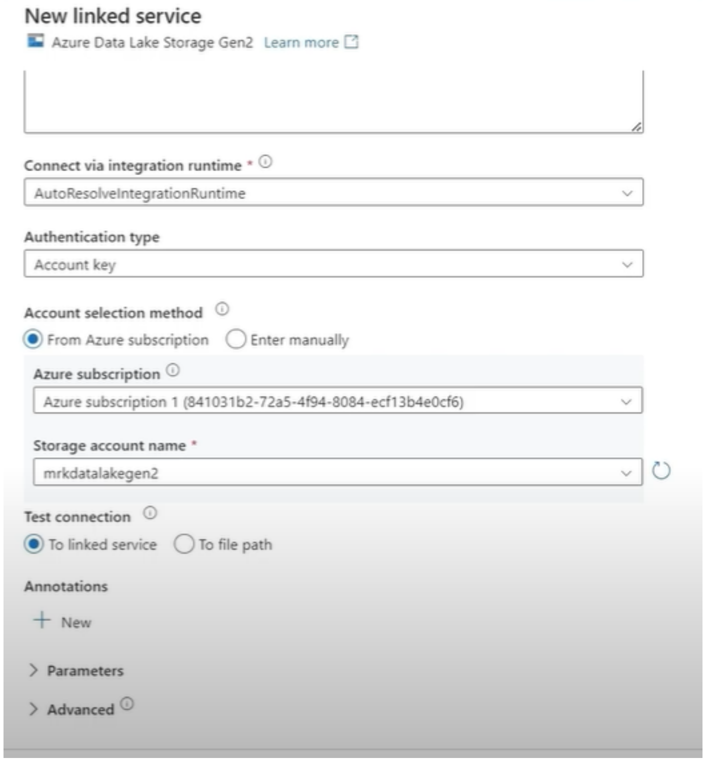
Select all

Storage account name *

Test connection

● To linked service ○ To file path

Annotations



New linked service

Azure Data Lake Storage Gen2

Connect via integration runtime *: AutoResolveIntegrationRuntime

Authentication type: Account key

Account selection method: ● From Azure subscription ○ Enter manually

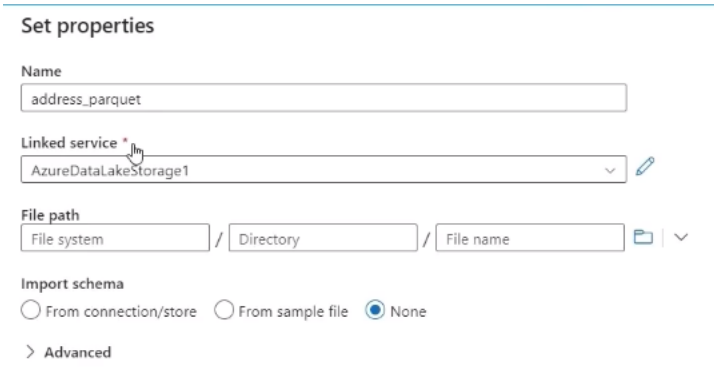
Azure subscription: Azure subscription 1 (841031b2-72a5-4f94-8084-ecf13b4e0cf6)

Storage account name *: mrkdatalakegen2

Test connection: ● To linked service ○ To file path

Annotations

7. Now as soon as you configure Linked services,



Set properties

Name: address_parquet

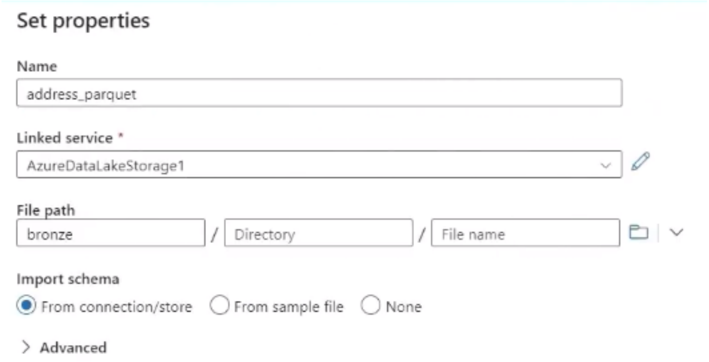
Linked service *: AzureDataLakeStorage1

File path: File system / Directory / File name

Import schema: ○ From connection/store ○ From sample file ● None

You get a screen like this. Now select the folder or file (in the Data Lake) that you want to store all the databases that we are extracting from the on-premises database.

Create a folder named “bronze” in the Azure Data Lake Gen 2 and the select the path to it in the option.



Set properties

Name: address_parquet

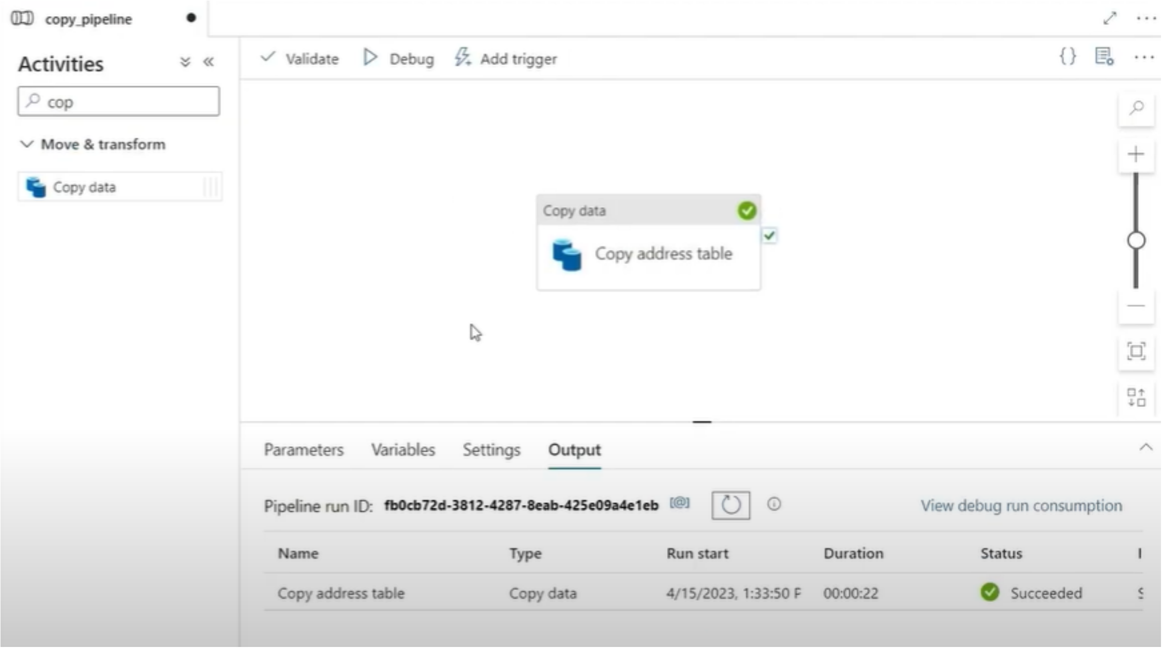
Linked service *: AzureDataLakeStorage1

File path: bronze / Directory / File name

Import schema: ● From connection/store ○ From sample file ○ None

> Advanced

8. Finally, the Pipeline is configured. Now test the pipeline using the Debug or Add trigger option above the pipeline.



Pipeline run ID: fb0cb72d-3812-4287-8eab-425e09a4e1eb

| Name | Type | Run start | Duration | Status |
|---|---|---|---|---|
| Copy address table | Copy data | 4/15/2023, 1:33:50 F | 00:00:22 | ✓ Succeeded |

## II .Creating Pipeline to Copy all tables.

1. To create a new pipeline -

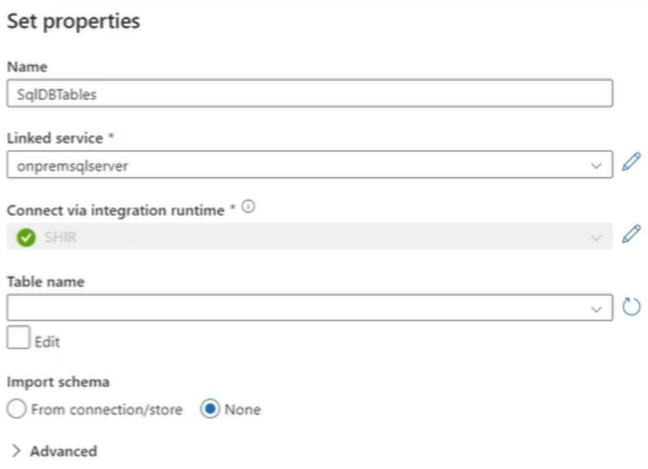
   Author tab → Click on “+” icon → Select Pipeline → This creates a new pipeline.

   You can rename it if you want.

2. Now go to Activities and search for Lookup-

   Change the name to your convenience.

   Now under Settings tab → Source dataset +New → SQL Server → We get a tab like this -



Set properties

Name
SqlDBTables

Linked service *
onpremsqlserver

Connect via integration runtime *
SHIR

Table name

□ Edit

Import schema
○ From connection/store ● None

> Advanced

As Linked Service is already created, just select the created one. Integration Runtime that has to be selected is SHIR as we are connecting on-premises to cloud.

3. Now, as Source dataset is configured, we have to configure below options too.
   a. Uncheck the box - First row only
   b. Use query - Select Query option and paste the following query-

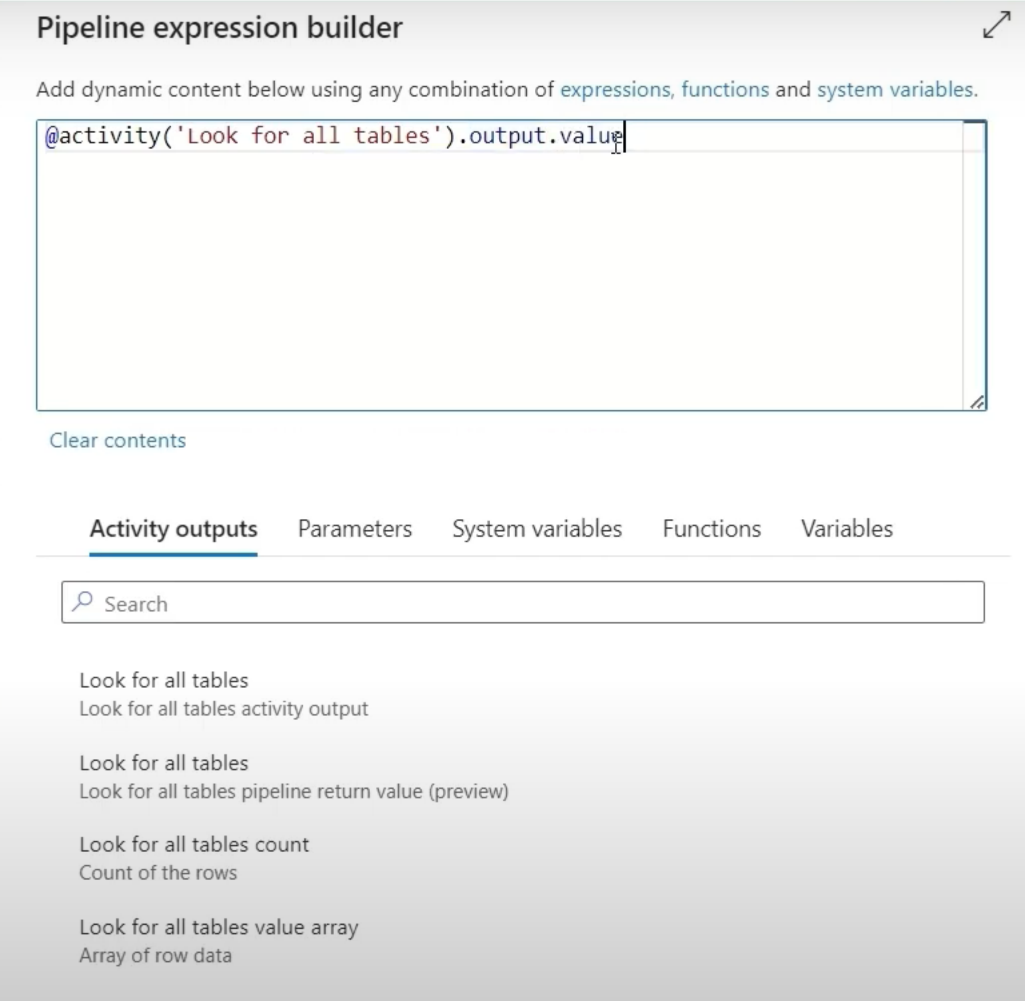
```
SELECT
s.name as SchemaName,
t.name as TableName
FROM sys.tables t
INNER JOIN sys.schemas s
ON t.schema_id = s.schema_id
WHERE s.name = 'SalesLT'
```

   c. This is done and Lookup is configured.

4. Now go to Activities and search for ForEach-
5. Connect end of Lookup Activity to ForEach activity.

   Change the name to your convenience.

   Now under Settings tab → Go to Items field → Add dynamic content → Select look all tables and due to this, a query will be visible in the top message box. Just add .value to the end of it.



Pipeline expression builder

Add dynamic content below using any combination of expressions, functions and system variables.

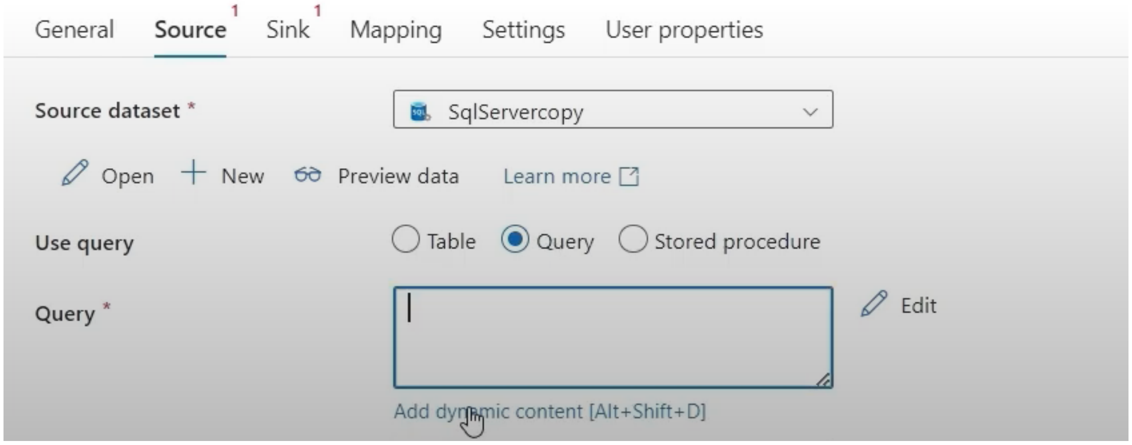
```
@activity('Look for all tables').output.value
```

6. Now under ForEach, go under Activities tab → Click on the pencil icon → This takes us inside the ForEach Activity.

   Now under here, add a new Copy data Activity as we want to **copy all tables for each look up value.**

   But we have to configure the Copy data Activity -

   1. Configure for Source and Sink dataset using the above same method. After configuring Source dataset and then, using the same Linked service that we created earlier.
   2. Next, for source dataset, for the Use query option, select Query and then click on Add dynamic content.



Source dataset * SqlServercopy

Use query ○ Table ● Query ○ Stored procedure

Query *

3. Now copy paste the below script in the whitespace -

```
@{concat('SELECT * FROM ', item().SchemaName,'.',item().TableName)}
```

4. For sink dataset, +New → Azure Data Lake Storage Gen 2 → Parquet →And follow same steps as done in some above steps.

5. Now we need the folder structure of all files to be -bronze/Schema/Tablename/Tablename..parquet

   To do this, we open (pencil icon) the Sink dataset and here we go the Parameters tab → +New and then Create two parameters schemaname and tablename of type String and set the value of these parameters as @item.SchemaName and @item.TableName respectively.

   Now go to Connection tab in Sink dataset, and here add file path (by adding dynamic content) and paste this in the whitespace -

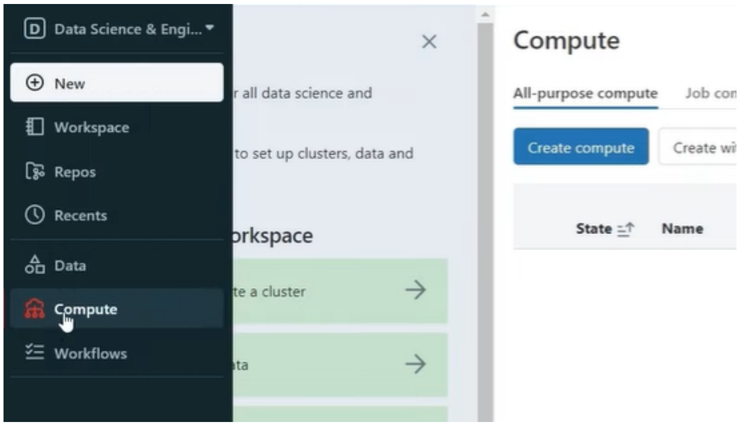
```
@{concat(dataset().schemaName, '/',dataset().tableName)}
```

```
@{concat(dataset().tableName,'.parquet')}
```

7. We are done and the pipeline is created. Now visit the bronze folder in the Azure Data Lake Storage Gen 2 and check to see all tables as .parquet files.

## STEP 4: Data Transformation

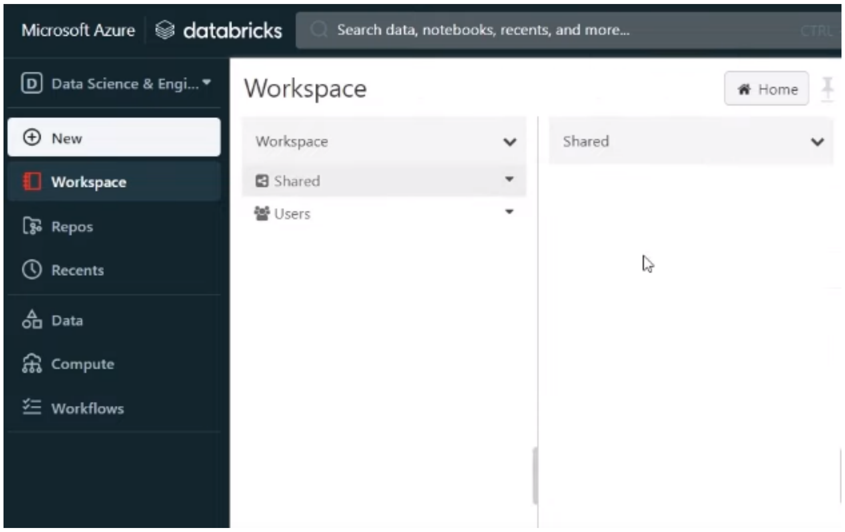
1. Sign in to the Azure Databricks portal. Here, create a cluster/compute by going to the Compute tab.



1. Policy - Unrestricted
2. Single Node
3. Else everything should be default as suggested by Azure.
4. Just enable credential passthrough for user-level data access.

   By enabling this, any user that has access to the Azure Data Lake Storage Gen 2 can also access it through the Databricks.

2. Create 2 more folders in Azure Data Lake Storage Gen 2 and name these Silver and Gold to represent the level of transformations.
2. Now open workspace tab on the left of the screen. Then go the Shared directory inside which create 3 notebooks to record transformations - 1. bronze to silver 2. silver to gold and 3. storagemount which is used to mount the 3 directories.



```
configs = {
"fs.azure.account.auth.type": "CustomAccessToken",
"fs.azure.account.custom.token.provider.class": spark.conf.get("spark.databricks.passthrough.adls.gen2.tokenProviderClassName")
}
```

```
dbutils.fs.mount(
source = "abfss://<container-name>@<storage-account-name>.dfs.core.windows.net/",
mount_point = "/mnt/<mount-name>",
extra_configs = configs)
```

Here replace container-name and mount-name with bronze, silver and gold respectively to mount all the three directories.

```
dbutils.fs.ls(”/mnt/bronze/SalesLT/”)
```

This code can be used to access all the files in the bronze directory.

4. Now in the bronze to silver directory,

```
table_name=[]
for i in dbutils.fs.ls('mnt/bronze/Sales/'):
    table_name.append(i.name.split('/')[0])

from pyspark.sql.functions import from_utc_timestamp, date_format
from pyspark.sql.types import TimestampType

for i in table_name:
    path = '/mnt/bronze/Sales/'+i+'/'+i+'.parquet'
    df=spark.read.format('parquet').load(path)
    column=df.columns

    for col in column:
        if "Date" in col or "date" in col:
            df = df.withColumn(col,date_format(from_utc_timestamp(df[col].cast(TimestampType()),"UTC"),"yyyy-MM-dd"))


    output_path = '/mnt/silver/' +i +'/'
    df.write.format('delta').mode("overwrite").save(output_path)
```

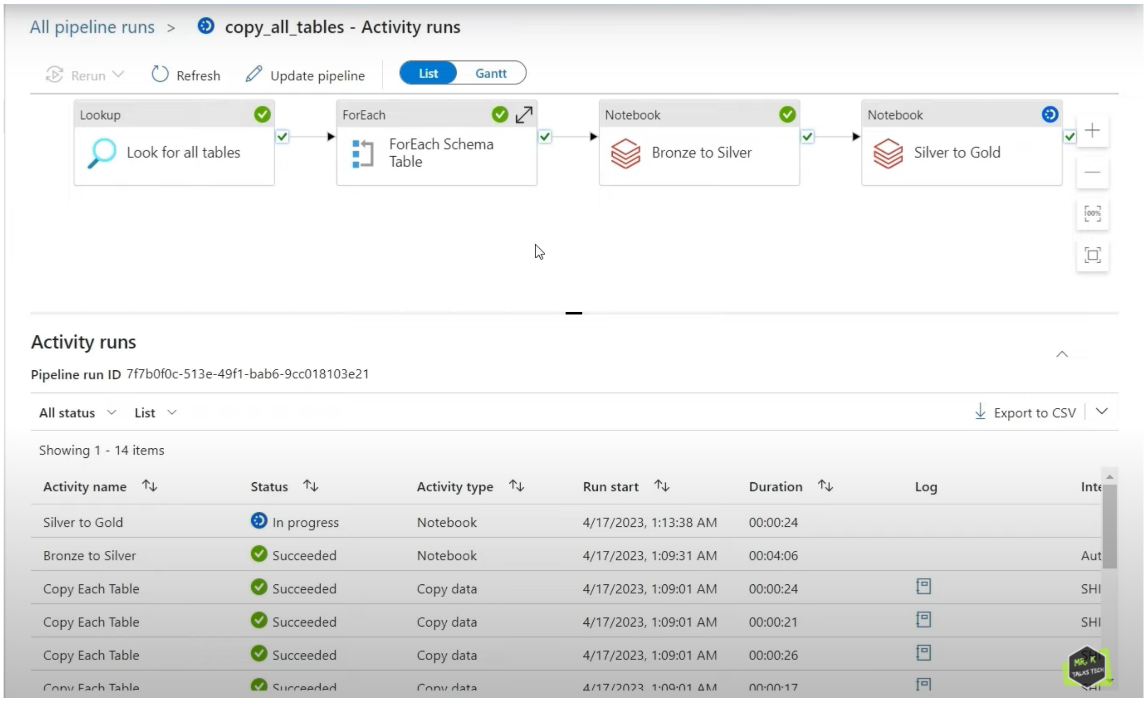
Write this code to apply the first set of changes (converts all columns that contain date to standard date format, removes all timestamps). All new files use the .delta format.

5. Now in the silver to gold directory,

```
table_name=[]
for i in dbutils.fs.ls('mnt/silver/'):
    table_name.append(i.name.split('/')[0])


from pyspark.sql.functions import col,regexp_replace
from pyspark.sql import SparkSession

for i in table_name:
    path = '/mnt/silver/'+i
    df=spark.read.format('delta').load(path)
    column=df.columns

    for colo in column:
        new_col="".join(["_"+char if char.isupper() and not colo[j-1].isupper() else char for j,char in enumerate(colo)]).lstrip("_
        df=df.withColumnRenamed(colo,new_col)


    output_path = '/mnt/gold/' +i +'/'
    df.write.format('delta').mode("overwrite").save(output_path)
```

Write this code to apply the first set of changes (converts all column names like FirstColumn to First_Column).

6. To automate the process of running these notebooks every time a change happens, we can create a pipeline on top of the previous one which then runs all these processes in an order and regularly. To do this,
    a. Open Azure Data Factory first create a Linked service between Azure Data Factory and Azure Databricks. Manage → Linked Services → +New.
    b. Fill in all the information as shown below -
        i. Name - Any name of your choice
        ii. Integration Runtime - AutoResolveIntegrationRuntime
        iii. Azure subscription - Your subscription
        iv. Databricks workspace - Pops up with linked subscription
        v. Authentication type - Access Token
            1. To get this access token, go to User Settings in the Databricks workspace and then generate new token. Copy this token and save it in Azure Key Vault which stores the value of this token.
            2. Now coming back to Data Factory, select AKV Linked Service as the one which you have previously made.
            3. Secret name should be the name of the token that you have stored as a secret in the key vault.
        vi. Choose from existing clusters - Choose the cluster that you made in Azure Databricks.
    c. Now in the pipeline workspace, we already have Lookup and ForEach Activities. Now select Databricks Notebook Activity from the left menu and add 2 such activities (one for each notebook - bronze to silver and silver to gold).
    d. To configure both Activities, choose one of them and then go to the Azure Databricks menu. Select Databricks Linked service as the Linked service that you just created.
    e. Then go the Settings tab (next to the Azure Databricks tab) and select the Notebook Path which is /Shared/bronzetosliver or /Shared/silvertogold in this case.
    f. Now publish all changes.
    g. Run this pipeline (using Debug or Trigger now) to check if the connections are working.



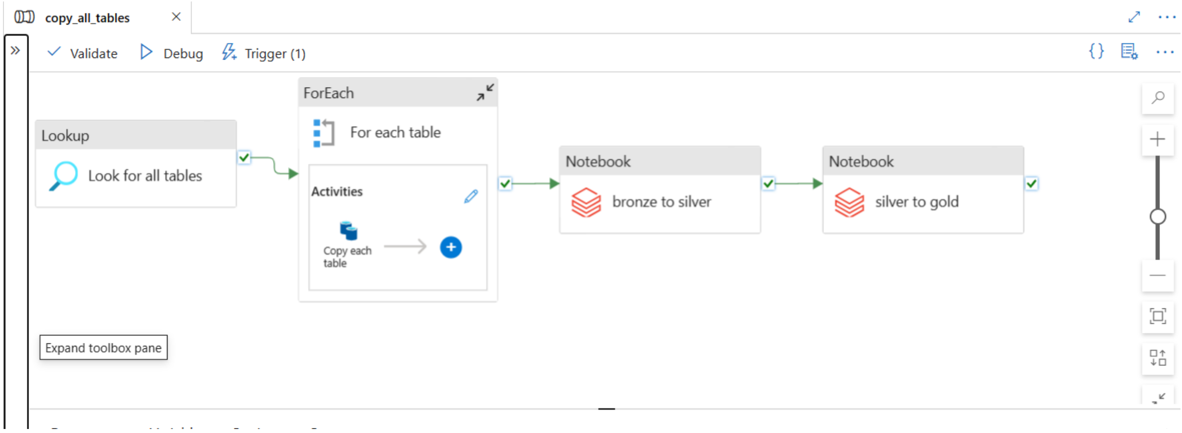
Activity runs

Pipeline run ID 7f7b0f0c-513e-49f1-bab6-9cc018103e21

Showing 1 - 14 items

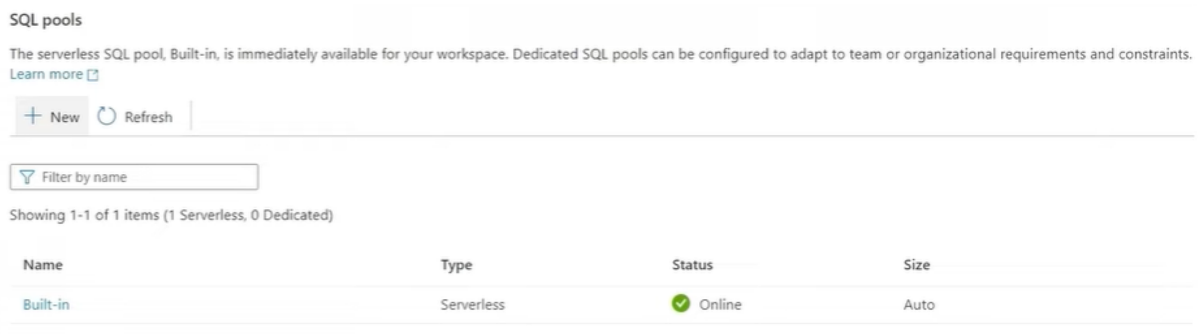
| Activity name | Status | Activity type | Run start | Duration | Log | Inte |
|---|---|---|---|---|---|---|
| Silver to Gold | In progress | Notebook | 4/17/2023, 1:13:38 AM | 00:00:24 | | |
| Bronze to Silver | Succeeded | Notebook | 4/17/2023, 1:09:31 AM | 00:04:06 | | Aut |
| Copy Each Table | Succeeded | Copy data | 4/17/2023, 1:09:01 AM | 00:00:24 | 🗎 | SHI |
| Copy Each Table | Succeeded | Copy data | 4/17/2023, 1:09:01 AM | 00:00:21 | 🗎 | SHI |
| Copy Each Table | Succeeded | Copy data | 4/17/2023, 1:09:01 AM | 00:00:26 | 🗎 | |
| Copy Each Table | Succeeded | Copy data | 4/17/2023, 1:09:01 AM | 00:00:17 | 🗎 | SHI |

Thus, you have now created a pipeline which ingests and transforms data.



# STEP 5: Data Loading

1. Open Azure Synapse Analytics studio. The Azure Synapse Analytics Resource is already linked to Azure Data Lake Storage Gen 2 as when creating this Synapse resource, we give the name and primary folder of the corresponding Azure Data Lake Storage Gen 2.
2. Now, under the Data tab in Synapse Workspace, create a SQL Database workspace.
   a. Create a Serverless SQL pool type Database and name it conventionally (say gold_db).
   b. There is a already Built-In Serverless Database pre-existing.



SQL pools

The serverless SQL pool, Built-in, is immediately available for your workspace. Dedicated SQL pools can be configured to adapt to team or organizational requirements and constraints.

Showing 1-1 of 1 items (1 Serverless, 0 Dedicated)

| Name | Type | Status | Size |
|---|---|---|---|
| Built-in | Serverless | ✓ Online | Auto |

3. Now in Data → Linked → We can access our Storage Container.
4. Now in the Develop tab, import this SQL Notebook file by the name of sp_CreateSQLServerlessView_gold: (execute this)

```
USE gold_db_serverless
GO

CREATE OR ALTER PROC CreateSQLServerlessView_gold @ViewName nvarchar(100)
AS
BEGIN

DECLARE @statement varchar(MAX)
  SET @statement = N'CREATE OR ALTER VIEW ' + @ViewName+' AS
    SELECT *
    FROM
      OPENROWSET(
      BULK ''https://dlgdataloadingtrial.dfs.core.windows.net/gold/' + @ViewName+'/'',
      FORMAT = ''DELTA''
    )as [result]
  '

EXEC(@statement)
END
GO
```

5. Now go Manage tab and under Manage tab, Create a new Linked Service → Search for Azure SQL Database.

   Now the following values are to be configured -

   Name - Any name of your choice

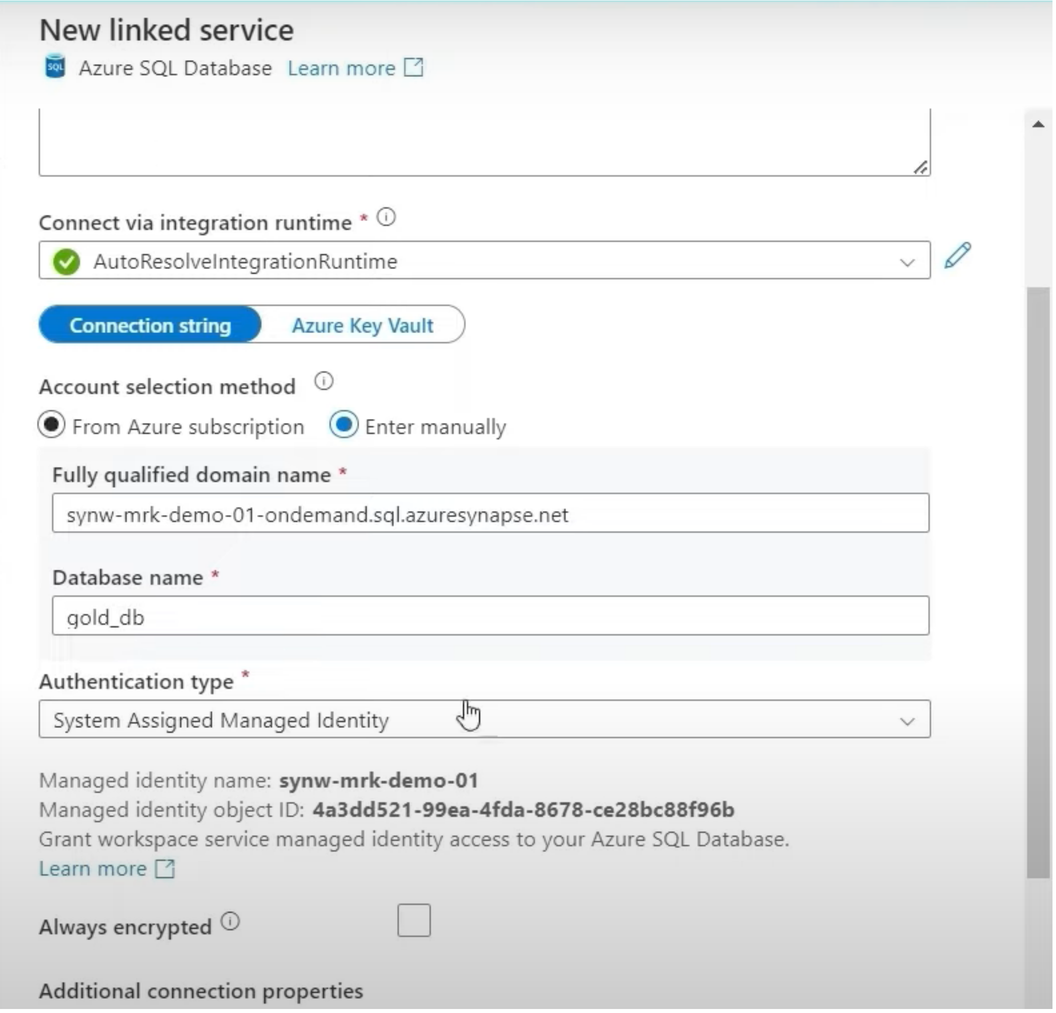
   Integration Runtime - AutoResolveIntegrationRuntime

   Account Selection Method - Enter manually

   Fully qualified domain name - Go to Resource Group → Synapse workspace → Properties → Select serverless SQL endpoint and input it here.

   Database name - Created earlier (gold_db)

   Authentication type - System Assigned Managed Identity



New linked service

Azure SQL Database Learn more

Connect via integration runtime *

AutoResolveIntegrationRuntime

Connection string | Azure Key Vault

Account selection method

○ From Azure subscription ● Enter manually

Fully qualified domain name *

synw-mrk-demo-01-ondemand.sql.azuresynapse.net

Database name *

gold_db

Authentication type *

System Assigned Managed Identity

Managed identity name: **synw-mrk-demo-01**
Managed identity object ID: **4a3dd521-99ea-4fda-8678-ce28bc88f96b**
Grant workspace service managed identity access to your Azure SQL Database.
Learn more

Always encrypted □

Additional connection properties

6. Finally go to Integrate tab and create a new pipeline which connects to our previous pipeline and is triggered when the pipeline in data factory is triggered.

7. For activities in this pipeline -

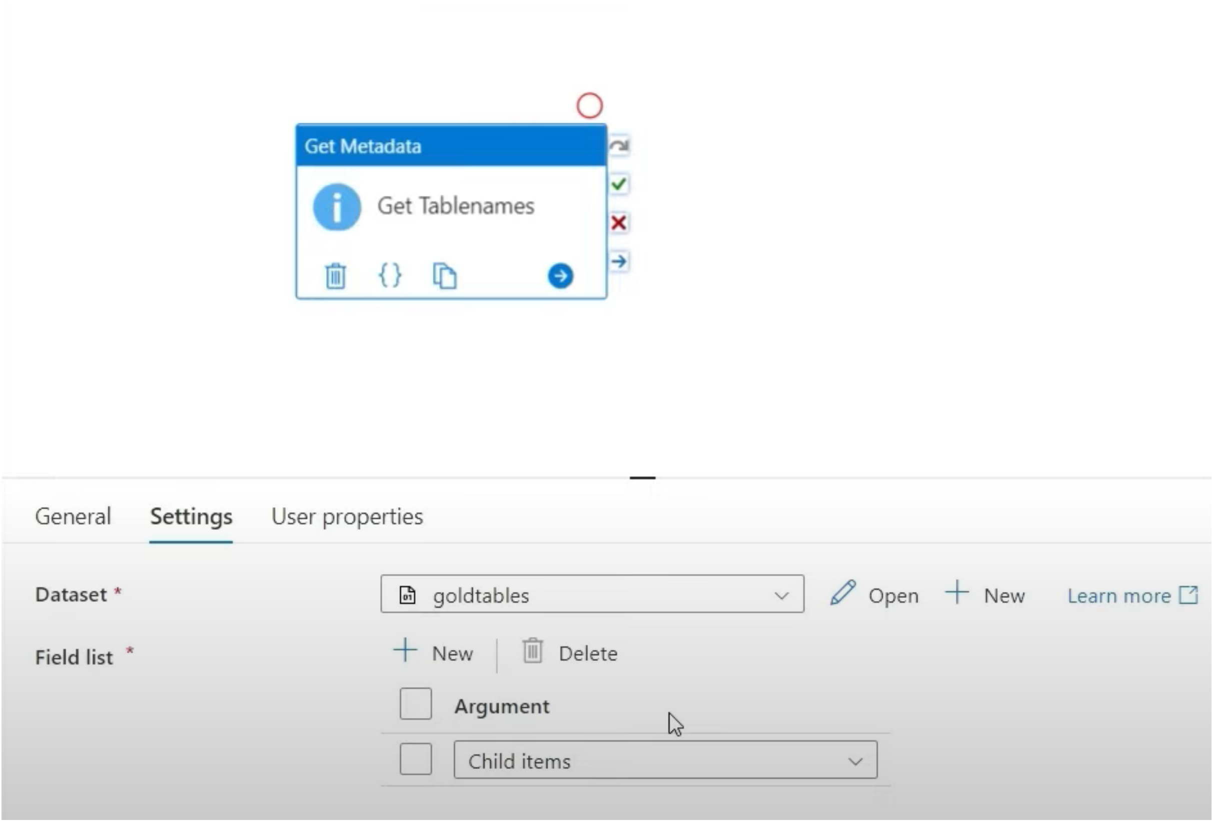
   a. First take Get Metadata Activity and change it's name to something appropriate like Get tablenames.

   b. Under the Settings tab, +New Dataset and then select Azure Data Lake Storage Gen 2 and then select format as .binary as we only need to get the tablenames.

   c. After selecting all these, select the Linked service which we created above.

   d. After this, set File path to the Gold directory of the Storage container by browsing through the container.

   e. Now again under the Settings tab, Select +New in Field List and then select Child Items.



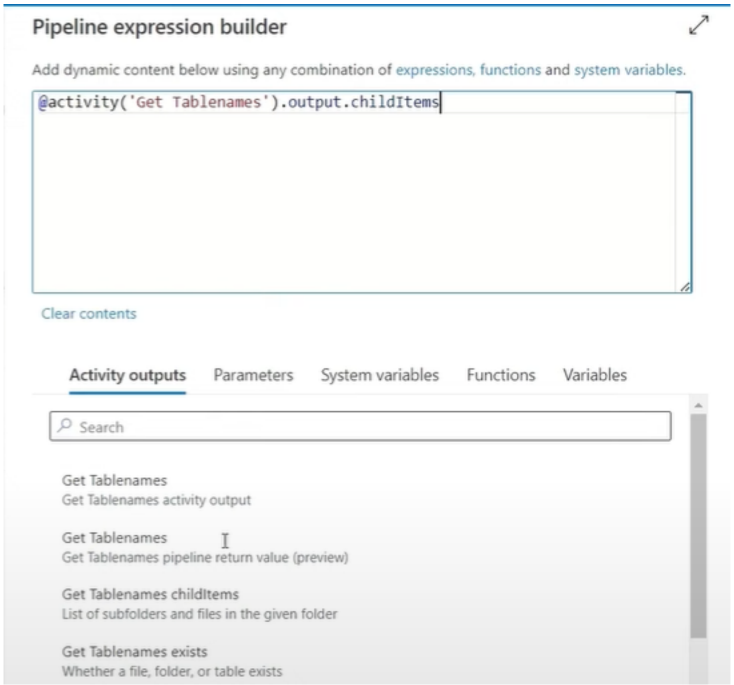
Dataset * goldtables

Field list *

| □ | Argument |
|---|---|
| □ | Child items |

8. Now add a new Activity. ForEach activity is added and configured accordingly. Name it as For Each table name.
    a. In this, under Settings tab → Items → Click on whitespace then on Add Dynamic Content. Then, on the new whitespace that opens, add this code -

```
@activity(’Get Tablenames’).output.ChildItems
```



Pipeline expression builder

Add dynamic content below using any combination of expressions, functions and system variables.

```
@activity('Get Tablenames').output.childItems
```

Clear contents
Activity outputs
Parameters
System variables
Functions
Variables
Search
Get Tablenames
Get Tablenames activity output
Get Tablenames
Get Tablenames pipeline return value (preview)
Get Tablenames childItems
List of subfolders and files in the given folder
Get Tablenames exists
Whether a file, folder, or table exists

b. Now, add new Activites under ForEach activity. Add a Stored Procedure activity.

   i. Under Settings tab, for Linked services select the new Linked Service that we created just earlier.

   ii. Under the Stored Procedure Name, [dbo].[CreateSQLServerlessView_gold] comes up as this is the stored procedure that we executed earlier. Select this.

   iii. Also under Stored Procedure Parameters,

      1. Name - ViewName

      2. Type - String

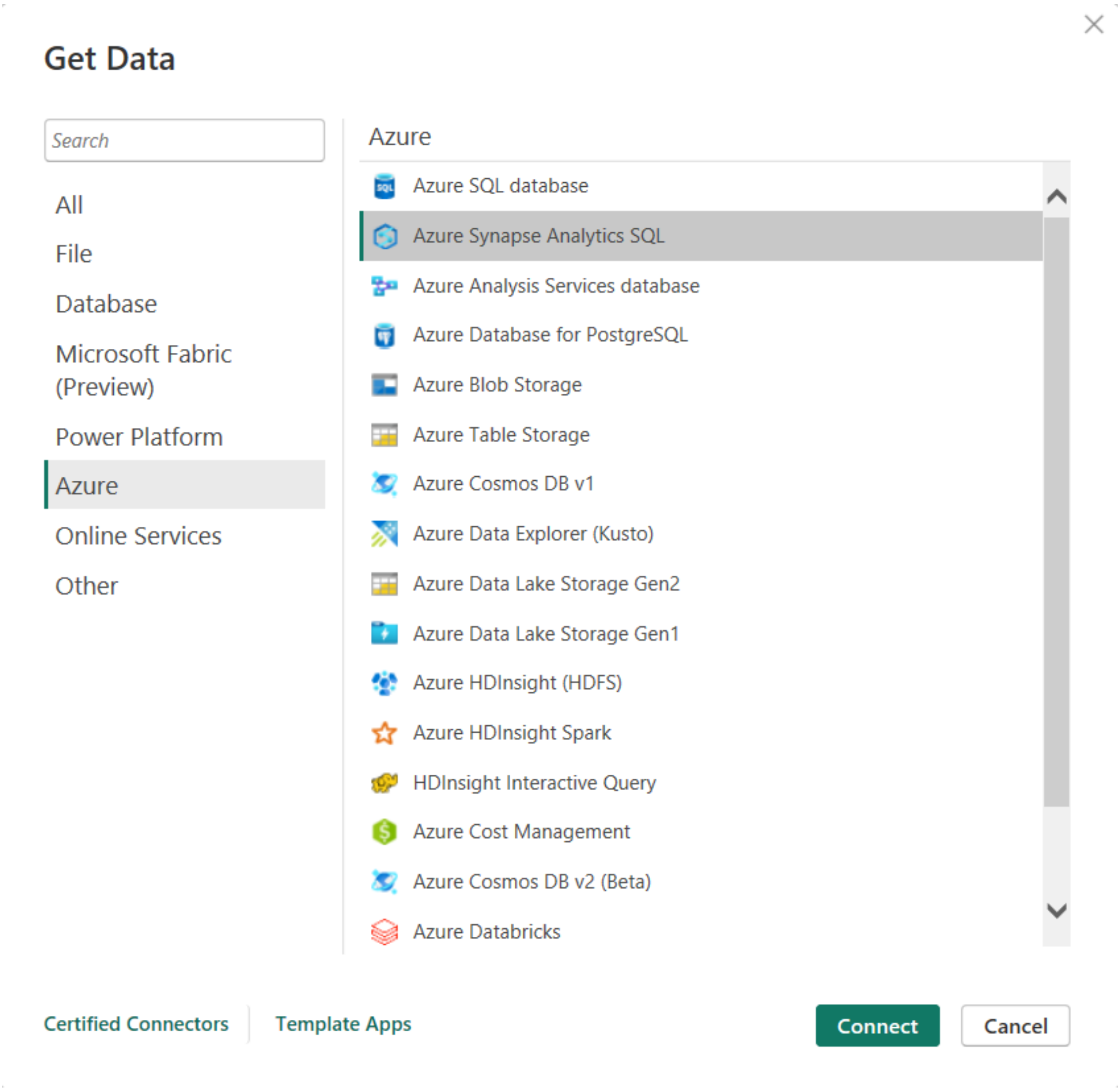
      3. Value - Add Dynamic content → @item().name

9. Now we are done. Publish all pending updates and run the new Pipeline using the Debug or Trigger Now options.

# STEP 6: Data Reporting

To export all created views, and to prepare reports on the same database, we use POWER BI.

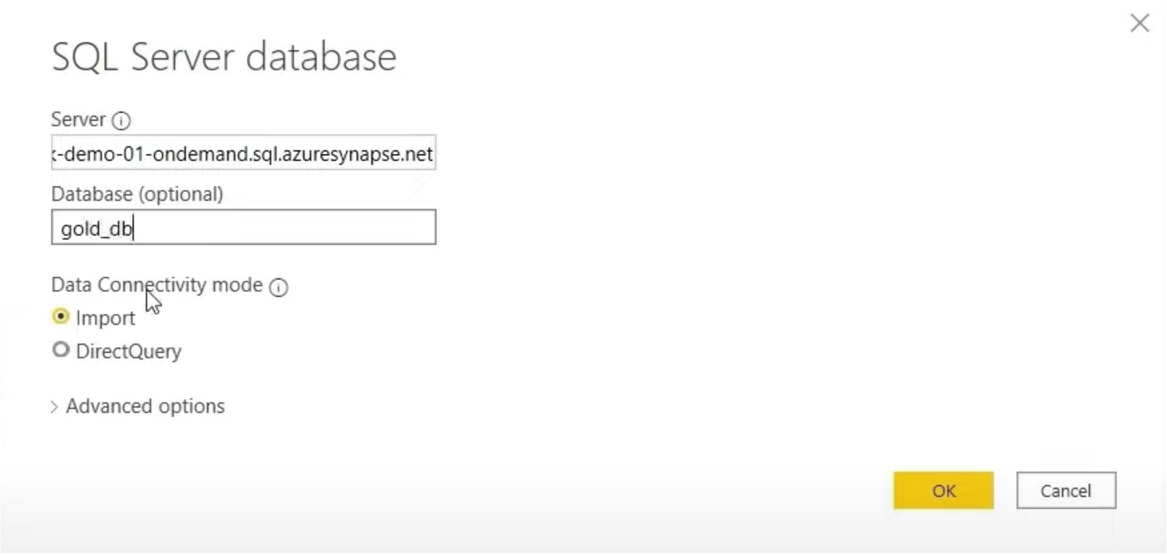
Use the Get Data tab → More…. → Azure → Azure Synapse Analytics SQL, as shown below:



After clicking on Connect option, it will ask you the name of the Server and Database.

For Server → Copy the Serverless SQL Endpoint as we did before

For Database → Write the name of the database which in this case is: gold_database (we created this database in Azure Synapse Analytics which contains the views of all original database tables).



SQL Server database

Server

-demo-01-ondemand.sql.azuresynapse.net

Database (optional)

gold_db

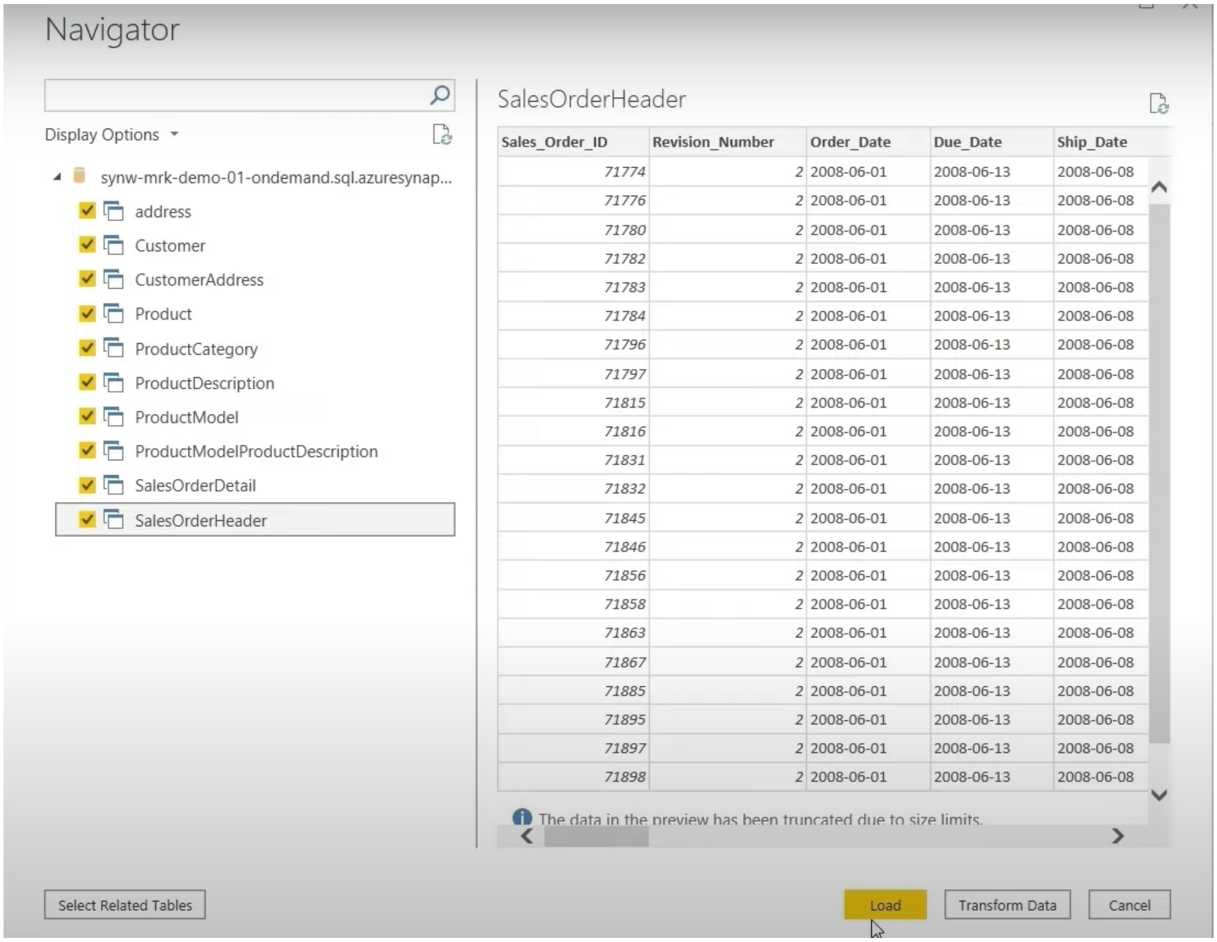
Data Connectivity mode

● Import

○ DirectQuery

Now on continuing, there will be 3 methods to authenticate and connect to the SQL Server Database. Select the Microsoft Account option and then sign in with your Microsoft Account with which you use Microsoft azure.

Now a Navigator window will open and here select all tables and load them →



- synw-mrk-demo-01-ondemand.sql.azuresynap...
  - ☑ address
  - ☑ Customer
  - ☑ CustomerAddress
  - ☑ Product
  - ☑ ProductCategory
  - ☑ ProductDescription
  - ☑ ProductModel
  - ☑ ProductModelProductDescription
  - ☑ SalesOrderDetail
  - ☑ SalesOrderHeader

SalesOrderHeader

| Sales_Order_ID | Revision_Number | Order_Date | Due_Date | Ship_Date |
|---|---|---|---|---|
| 71774 | 2 | 2008-06-01 | 2008-06-13 | 2008-06-08 |
| 71776 | 2 | 2008-06-01 | 2008-06-13 | 2008-06-08 |
| 71780 | 2 | 2008-06-01 | 2008-06-13 | 2008-06-08 |
| 71782 | 2 | 2008-06-01 | 2008-06-13 | 2008-06-08 |
| 71783 | 2 | 2008-06-01 | 2008-06-13 | 2008-06-08 |
| 71784 | 2 | 2008-06-01 | 2008-06-13 | 2008-06-08 |
| 71796 | 2 | 2008-06-01 | 2008-06-13 | 2008-06-08 |
| 71797 | 2 | 2008-06-01 | 2008-06-13 | 2008-06-08 |
| 71815 | 2 | 2008-06-01 | 2008-06-13 | 2008-06-08 |
| 71816 | 2 | 2008-06-01 | 2008-06-13 | 2008-06-08 |
| 71831 | 2 | 2008-06-01 | 2008-06-13 | 2008-06-08 |
| 71832 | 2 | 2008-06-01 | 2008-06-13 | 2008-06-08 |
| 71845 | 2 | 2008-06-01 | 2008-06-13 | 2008-06-08 |
| 71846 | 2 | 2008-06-01 | 2008-06-13 | 2008-06-08 |
| 71856 | 2 | 2008-06-01 | 2008-06-13 | 2008-06-08 |
| 71858 | 2 | 2008-06-01 | 2008-06-13 | 2008-06-08 |
| 71863 | 2 | 2008-06-01 | 2008-06-13 | 2008-06-08 |
| 71867 | 2 | 2008-06-01 | 2008-06-13 | 2008-06-08 |
| 71885 | 2 | 2008-06-01 | 2008-06-13 | 2008-06-08 |
| 71895 | 2 | 2008-06-01 | 2008-06-13 | 2008-06-08 |
| 71897 | 2 | 2008-06-01 | 2008-06-13 | 2008-06-08 |
| 71898 | 2 | 2008-06-01 | 2008-06-13 | 2008-06-08 |

The data in the preview has been truncated due to size limits.

Now use all POWER BI tools to build an informative report.