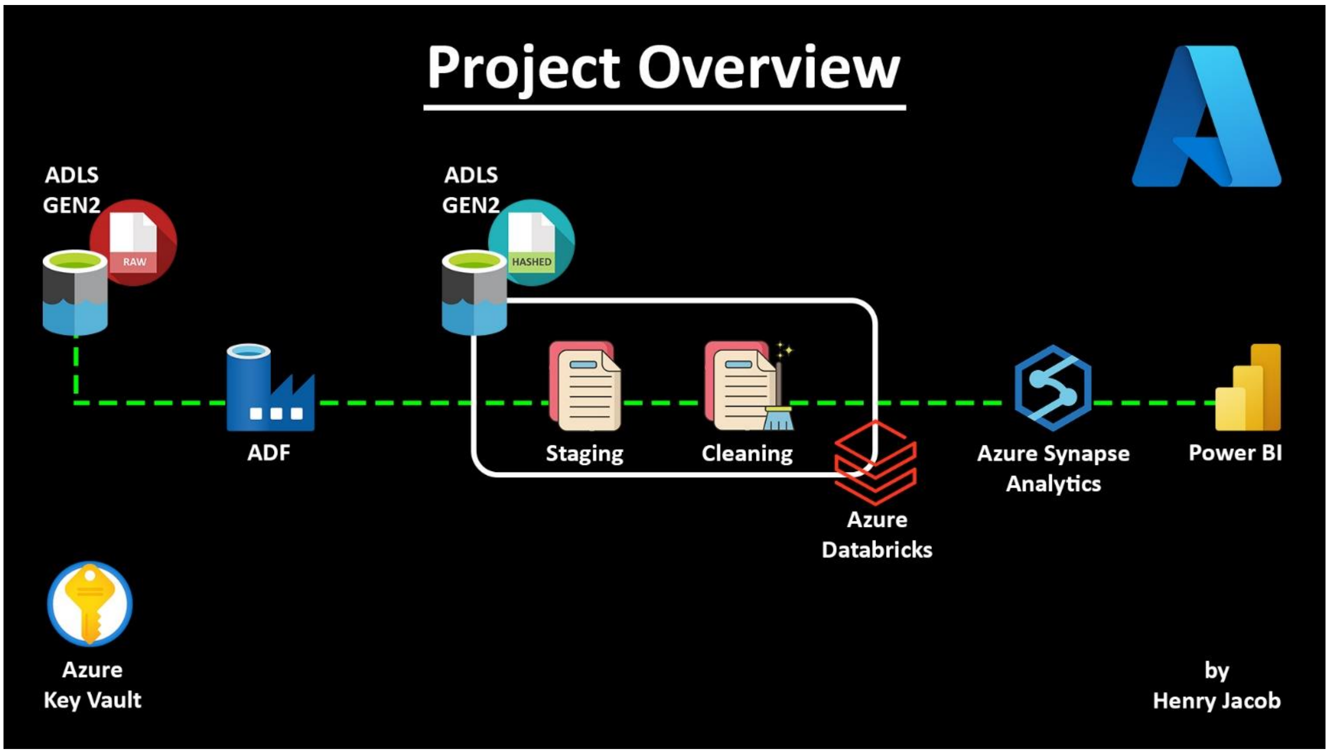
# Azure end to end project – Healthcare



**Scenario:**

- Sensitive patient information should only be stored in the raw storage.
- Access to sensitive patient data should be restricted after ingestion, and no one should have visibility to this data beyond the ingestion process.

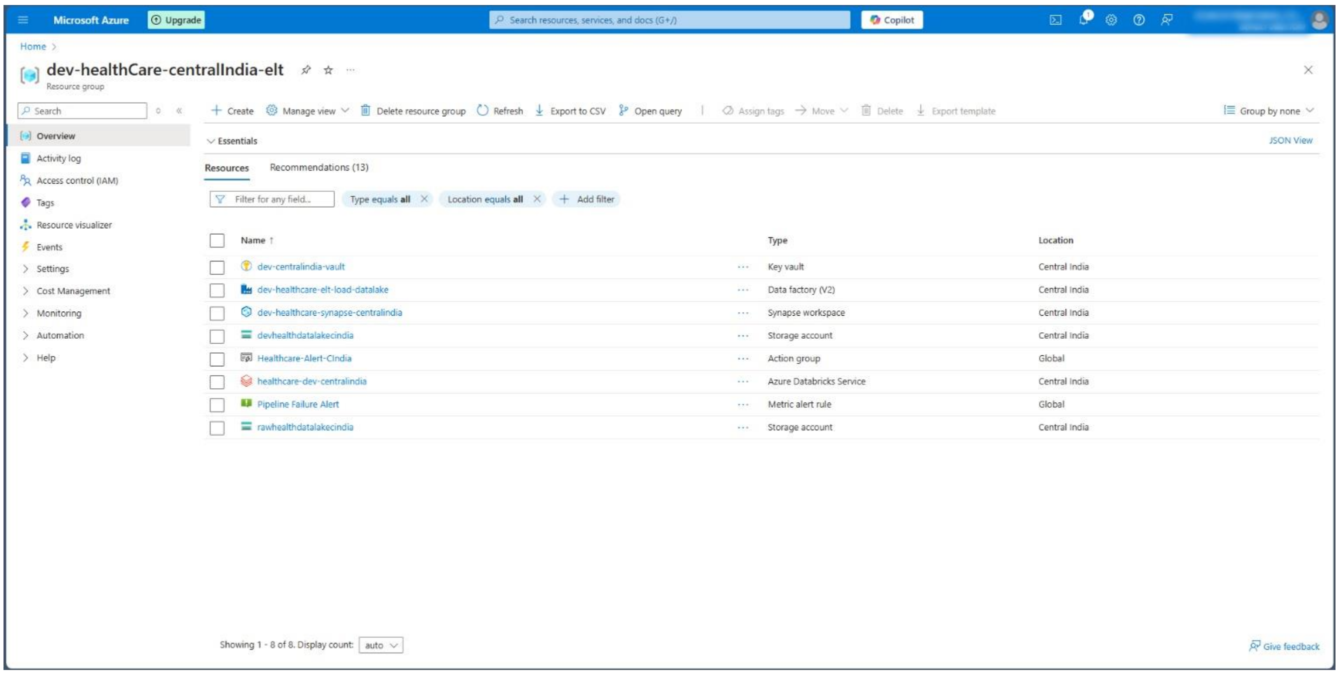
**Assumption:**

Let's assume that the ADLS Gen2 raw data is stored outside of Azure.

**Services used:**

- Azure Data Lake Gen2 (Raw & Staging and Cleaning)
- Azure Data Factory
- Azure Databricks
- Azure Synapse Analytics
- Power BI
- Azure Key Vault

## Resource Group:



dev-healthCare-centralIndia-elt
Resource group

| Name ↑ | Type | Location |
|---|---|---|
| dev-centralindia-vault | Key vault | Central India |
| dev-healthcare-elt-load-datalake | Data factory (V2) | Central India |
| dev-healthcare-synapse-centralindia | Synapse workspace | Central India |
| devhealthdatalakecindia | Storage account | Central India |
| Healthcare-Alert-CIndia | Action group | Global |
| healthcare-dev-centralindia | Azure Databricks Service | Central India |
| Pipeline Failure Alert | Metric alert rule | Global |
| rawhealthdatalakecindia | Storage account | Central India |

Showing 1 - 8 of 8. Display count: auto

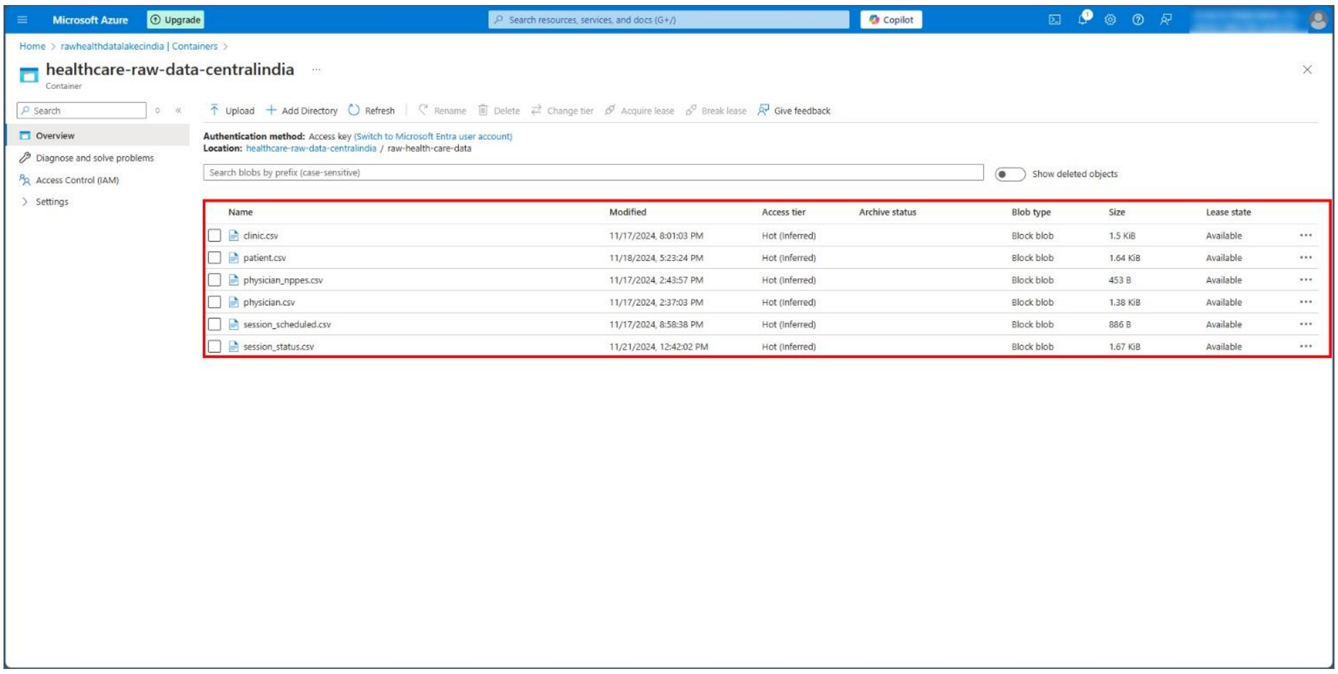
- **dev-centralindia-vault** – To store and use credentials.
- **dev-healthcare-elt-load-datalake** – To retrieve data from the raw storage account, hash, join, and load it into the staging container in the dev storage account.
- **dev-healthcare-synapse-centralindia** – To retrieve data from the cleaned container in the dev storage account and derive insights using SQL queries.
- **devhealthdatalakecindia** – To store the staging and cleaned data in two different containers.
- **Healthcare-Alert-CIndia** (auto-created by ADF) – To group notification channels.
- **healthcare-dev-centralindia** – To retrieve data from the staging container in the dev storage account, clean it, and load it into the cleaned container in the dev storage account.
- **Pipeline Failure Alert** (auto-created by ADF) – To send an email alert if the pipeline fails to run.
- **rawhealthdatalakecindia** – To store raw data only.

As we assumed the raw data is stored outside of Azure, two different storage accounts have been created:

- **rawhealthdatalakecindia** – to store raw data.
- **devhealthdatalakecindia** – to store staged and cleaned data.

## Azure Data Lake Gen2 – Raw Data

The raw files can be accessed here.

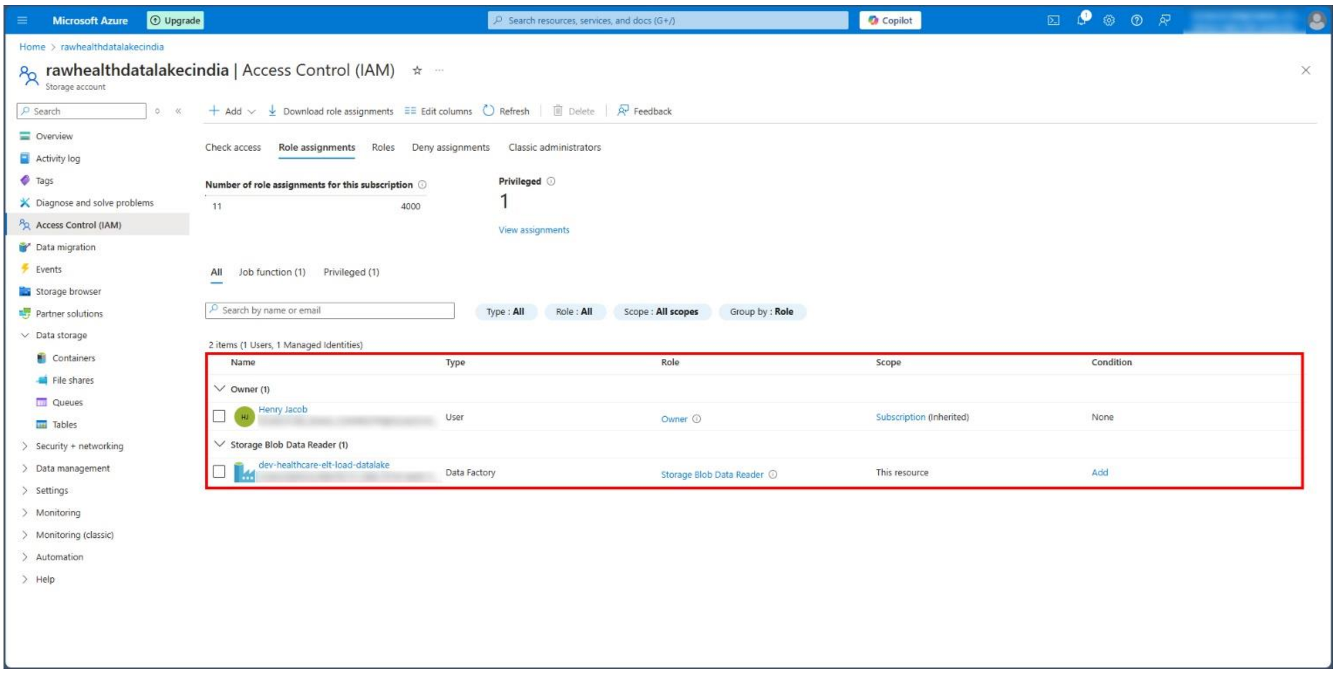


**Authentication method:** Access key (Switch to Microsoft Entra user account)
**Location:** healthcare-raw-data-centralindia / raw-health-care-data

| Name | Modified | Access tier | Archive status | Blob type | Size | Lease state |
|---|---|---|---|---|---|---|
| clinic.csv | 11/17/2024, 8:01:03 PM | Hot (Inferred) | | Block blob | 1.5 KiB | Available |
| patient.csv | 11/18/2024, 5:23:24 PM | Hot (Inferred) | | Block blob | 1.64 KiB | Available |
| physician_nppes.csv | 11/17/2024, 2:43:57 PM | Hot (Inferred) | | Block blob | 453 B | Available |
| physician.csv | 11/17/2024, 2:37:03 PM | Hot (Inferred) | | Block blob | 1.38 KiB | Available |
| session_scheduled.csv | 11/17/2024, 8:58:38 PM | Hot (Inferred) | | Block blob | 886 B | Available |
| session_status.csv | 11/21/2024, 12:42:02 PM | Hot (Inferred) | | Block blob | 1.67 KiB | Available |

## Azure Data Lake Gen2 – Raw Data – IAM

- **Owner** – Since I am the owner of the account.
- **Storage Blob Data Reader** – Assigned to Azure Data Factory (since the files need to be read by Azure Data Factory for further processing).



Number of role assignments for this subscription
11 4000

Privileged
1
View assignments

2 items (1 Users, 1 Managed Identities)

| Name | Type | Role | Scope | Condition |
|---|---|---|---|---|
| Owner (1) | | | | |
| Henry Jacob [redacted] | User | Owner | Subscription (Inherited) | None |
| Storage Blob Data Reader (1) | | | | |
| dev-healthcare-elt-load-datalake [redacted] | Data Factory | Storage Blob Data Reader | This resource | Add |

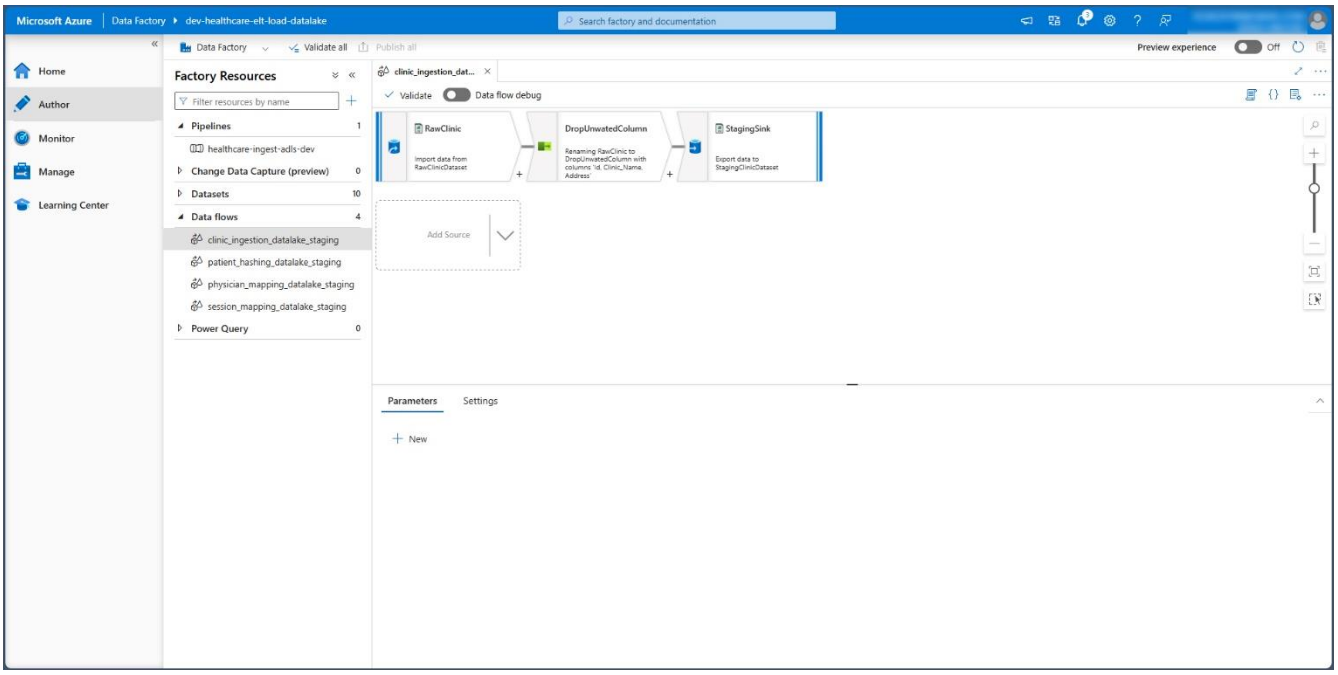
## Azure Data Factory

Azure Data Factory is used only to perform data hashing, joining, and dropping unwanted columns.

## Data Flows

### clinic_ingestion_datalake_staging

Unwanted columns were removed from the clinic data and ingested into the staging layer.



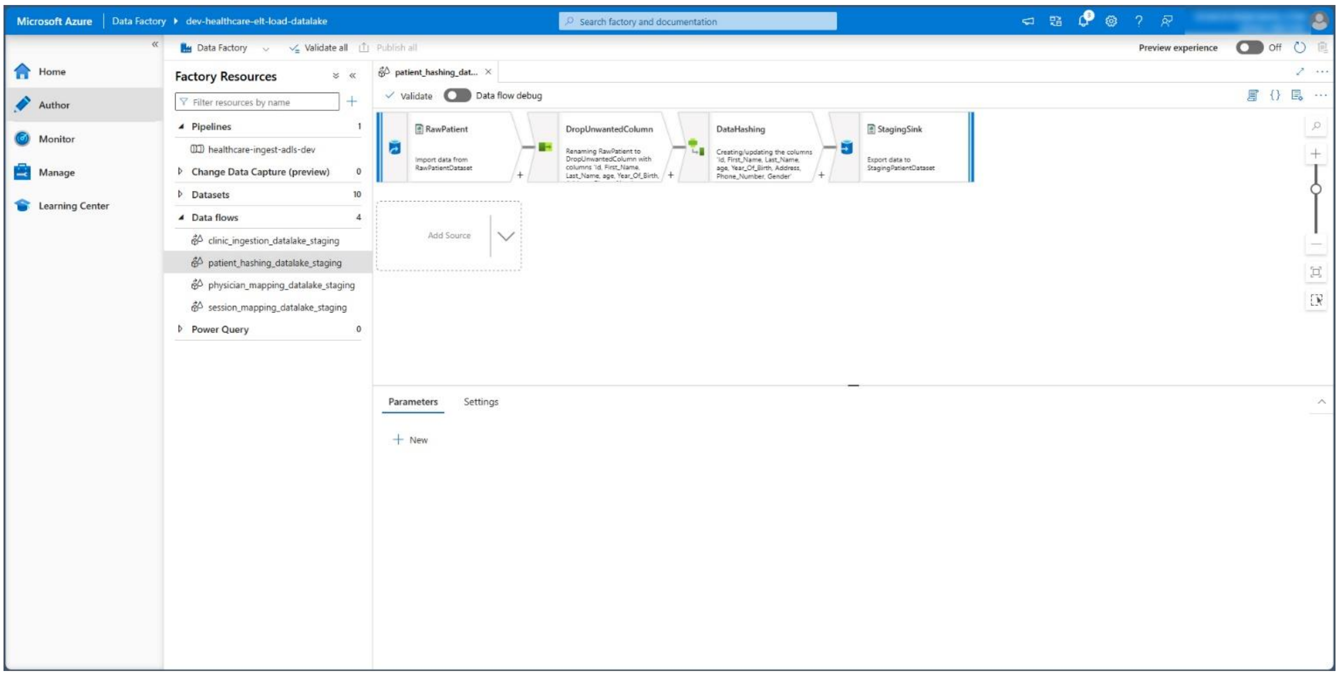
RawClinic: Import data from RawClinicDataset

DropUnwatedColumn: Renaming RawClinic to DropUnwatedColumn with columns 'Id, Clinic_Name, Address'

StagingSink: Export data to StagingClinicDataset

### patient_hashing_datalake_staging

The patients' data was hashed, and unwanted columns were removed.



RawPatient: Import data from RawPatientDataset

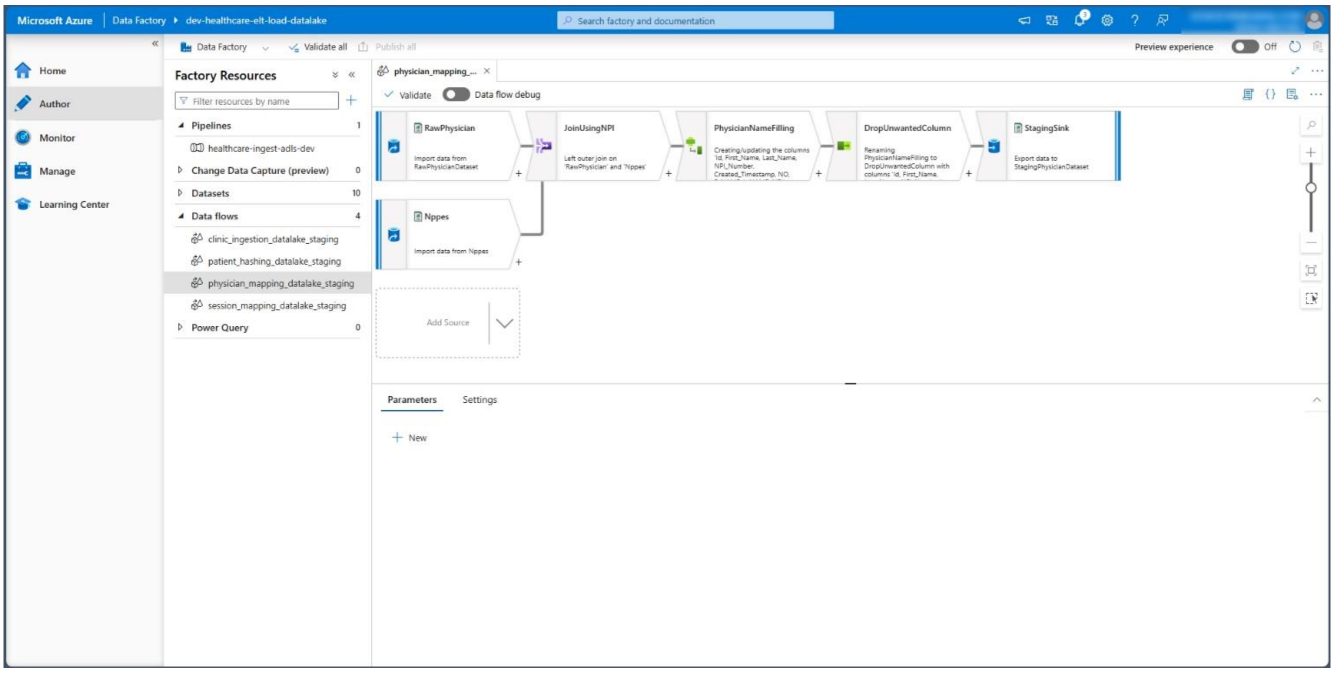
DropUnwantedColumn: Renaming RawPatient to DropUnwantedColumn with columns 'Id, First_Name, Last_Name, age, Year_Of_Birth,

DataHashing: Creating/updating the columns 'Id, First_Name, Last_Name, age, Year_Of_Birth, Address, Phone_Number, Gender'

StagingSink: Export data to StagingPatientDataset

## physician_mapping_datalake_staging

The physician data was joined with NPPES, names were filled in if missing, and unwanted columns were removed.



RawPhysician
Import data from RawPhysicianDataset

JoinUsingNPI
Left outer join on 'RawPhysician' and 'Nppes'

PhysicianNameFilling
Creating/updating the columns 'Id, First_Name, Last_Name, NPI_Number, Created_Timestamp, NO,

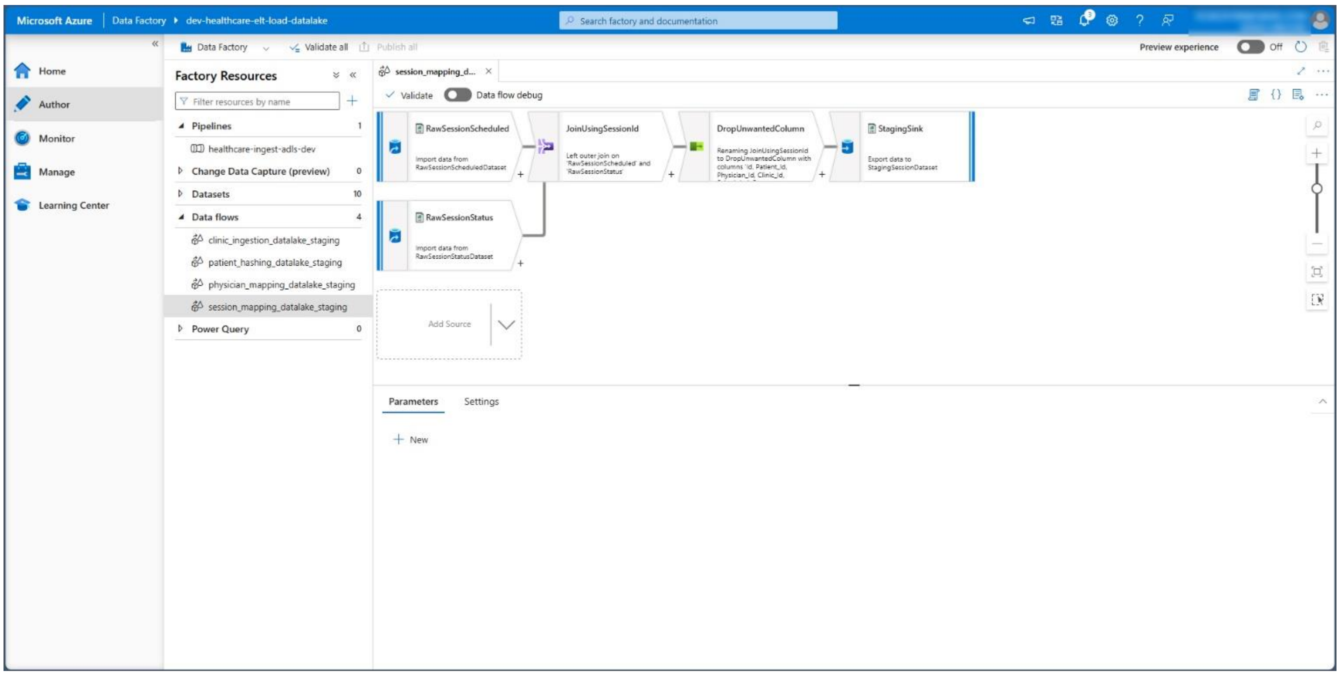
DropUnwantedColumn
Renaming PhysicianNameFilling to DropUnwantedColumn with columns 'Id, First_Name,

StagingSink
Export data to StagingPhysicianDataset

Nppes
Import data from Nppes

## session_mapping_datalake_staging

The scheduled sessions were joined with session status, and unwanted columns were dropped.



RawSessionScheduled
Import data from RawSessionScheduledDataset

JoinUsingSessionId
Left outer join on 'RawSessionScheduled' and 'RawSessionStatus'

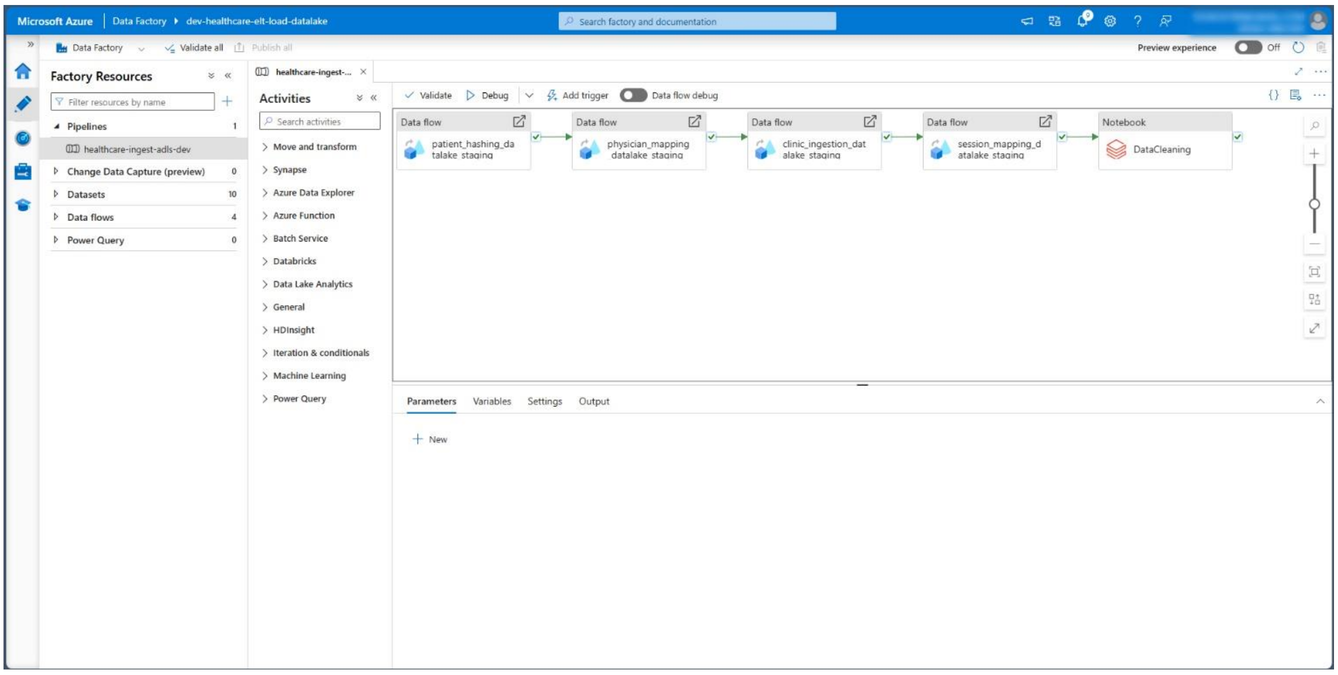
DropUnwantedColumn
Renaming JoinUsingSessionId to DropUnwantedColumn with columns 'Id, Patient_Id, Physician_Id, Clinic_Id,

StagingSink
Export data to StagingSessionDataset

RawSessionStatus
Import data from RawSessionStatusDataset

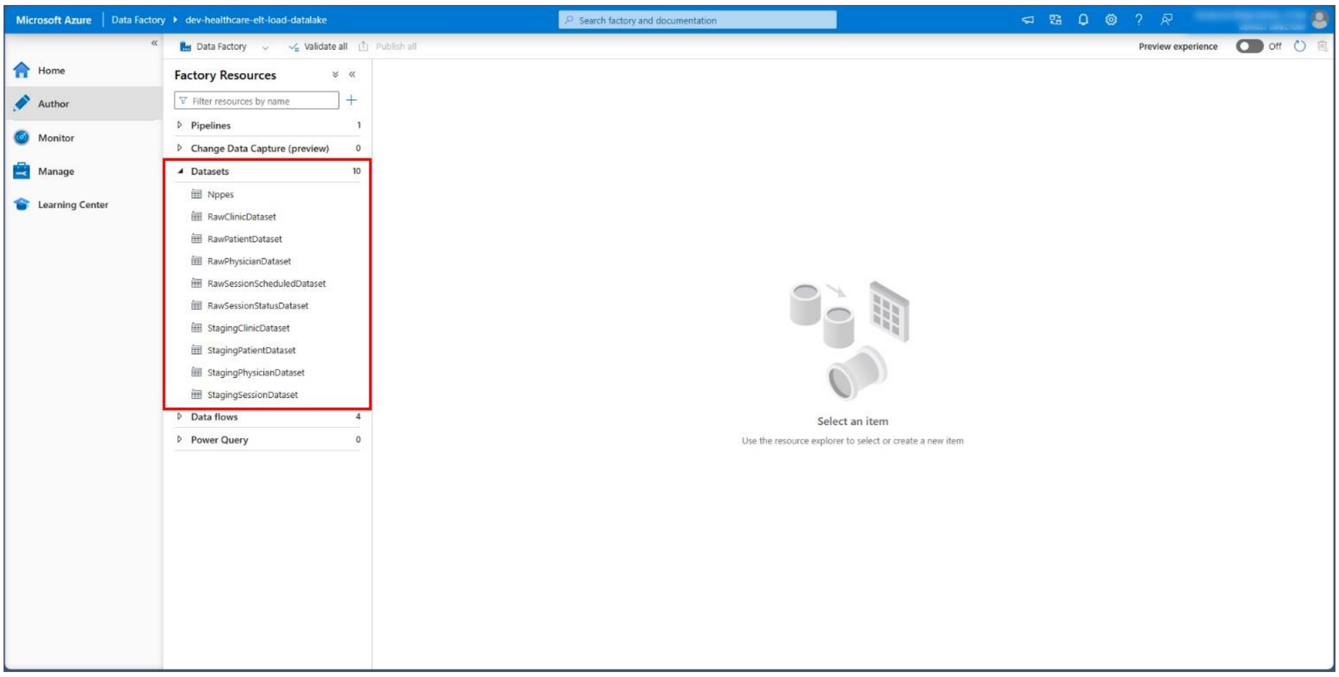
## Pipeline

All the data flow activities were added sequentially, and a notebook was included for the data cleaning process.



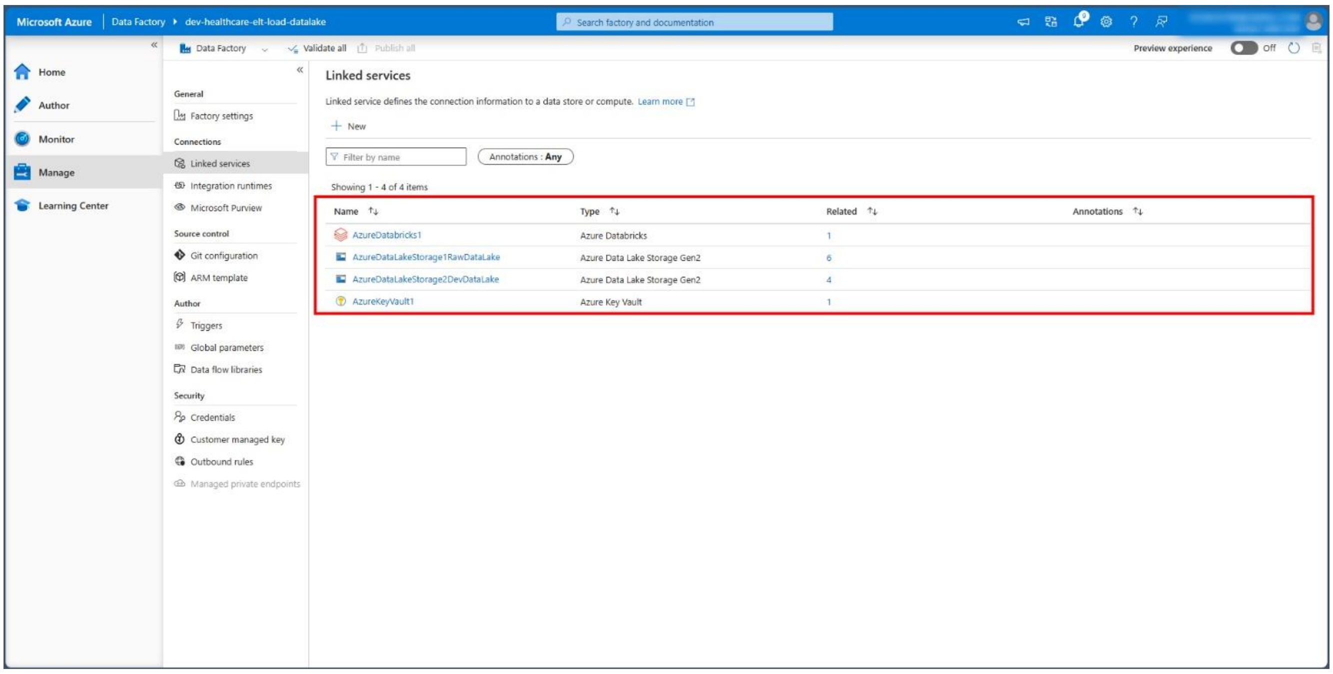
## Datasets

Source and sink datasets in Azure Data Factory.



## Linked Services

- **AzureDatabricks1** – to connect with Azure Databricks.
- **AzureDataLakeStorage1RawDataLake** – to connect with the source (storage account containing raw data).
- **AzureDataLakeStorage2DevDataLake** – to connect with the sink (storage account which is ready to be loaded with staged and cleaned data).
- **AzureKeyVault1** – to securely access credentials (Databricks Access Token).



Linked services

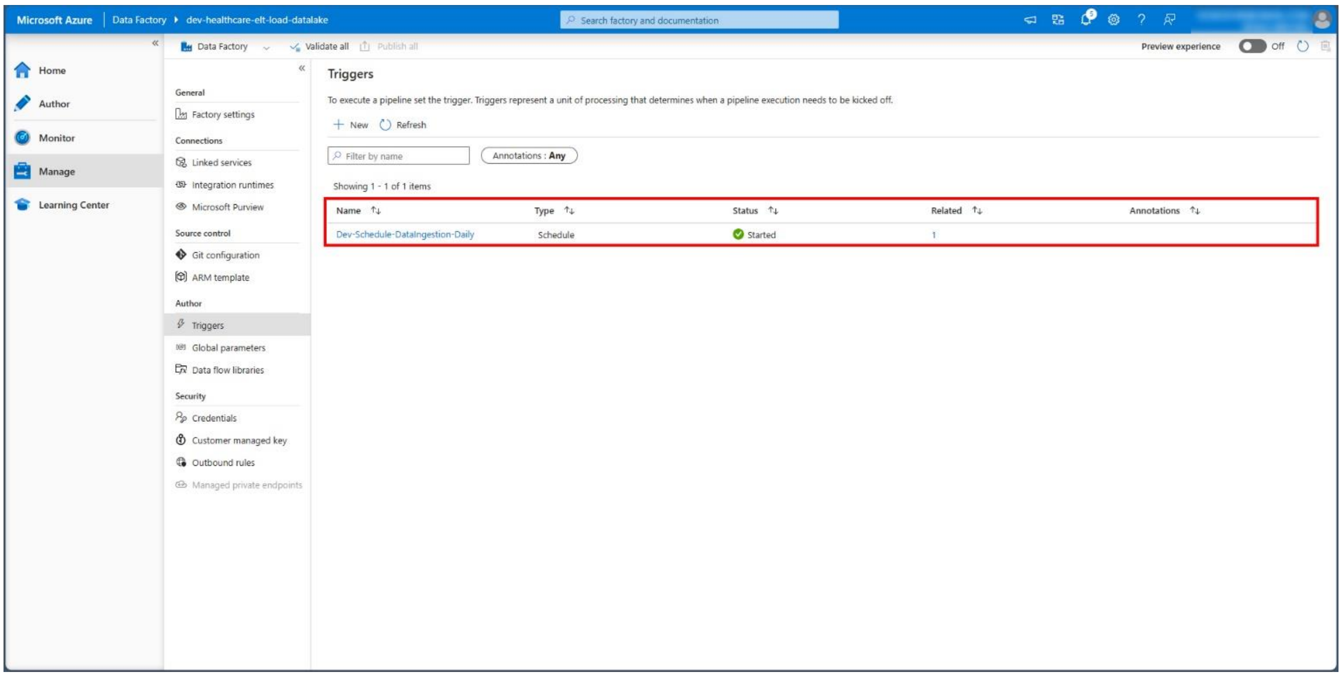
Linked service defines the connection information to a data store or compute. Learn more

Showing 1 - 4 of 4 items

| Name | Type | Related | Annotations |
|---|---|---|---|
| AzureDatabricks1 | Azure Databricks | 1 | |
| AzureDataLakeStorage1RawDataLake | Azure Data Lake Storage Gen2 | 6 | |
| AzureDataLakeStorage2DevDataLake | Azure Data Lake Storage Gen2 | 4 | |
| AzureKeyVault1 | Azure Key Vault | 1 | |

## Trigger – Daily Schedule

A schedule-type trigger was added to run the pipeline on a daily basis.



Triggers

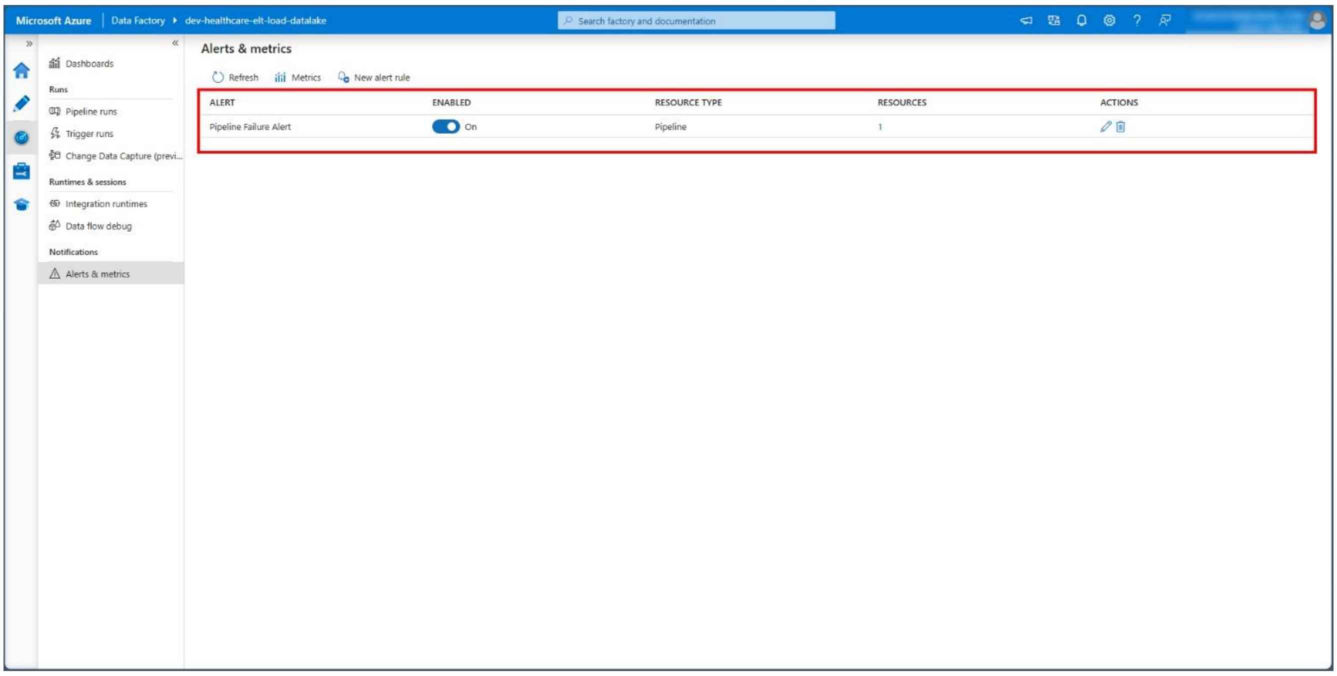
To execute a pipeline set the trigger. Triggers represent a unit of processing that determines when a pipeline execution needs to be kicked off.

Showing 1 - 1 of 1 items

| Name | Type | Status | Related | Annotations |
|---|---|---|---|---|
| Dev-Schedule-DataIngestion-Daily | Schedule | ✓ Started | 1 | |

**Alert – Pipeline Failure**

Email notification if the pipeline failed to run.



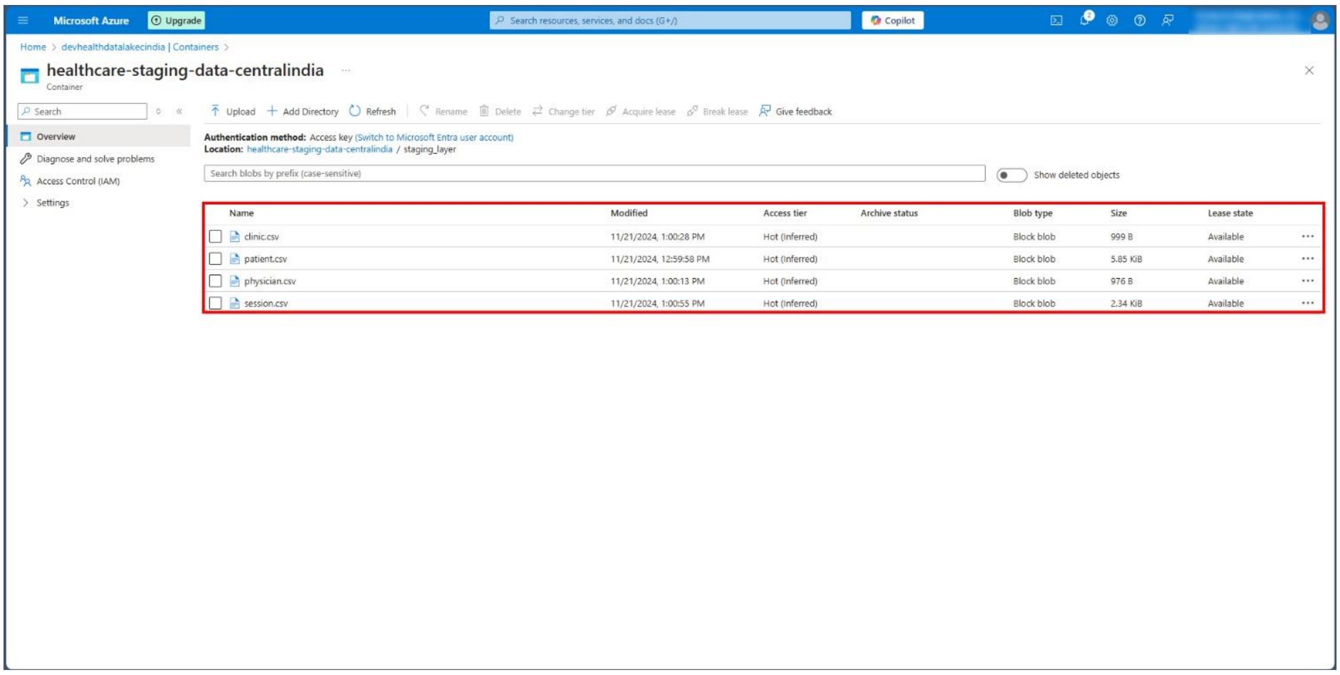
Alerts & metrics

| ALERT | ENABLED | RESOURCE TYPE | RESOURCES | ACTIONS |
|---|---|---|---|---|
| Pipeline Failure Alert | On | Pipeline | 1 | |

## Azure Data Lake Gen2 – Staged and Cleaned Data

### Staging Data Container

Staged data loaded from Azure Data Factory can be accessed here.



healthcare-staging-data-centralindia

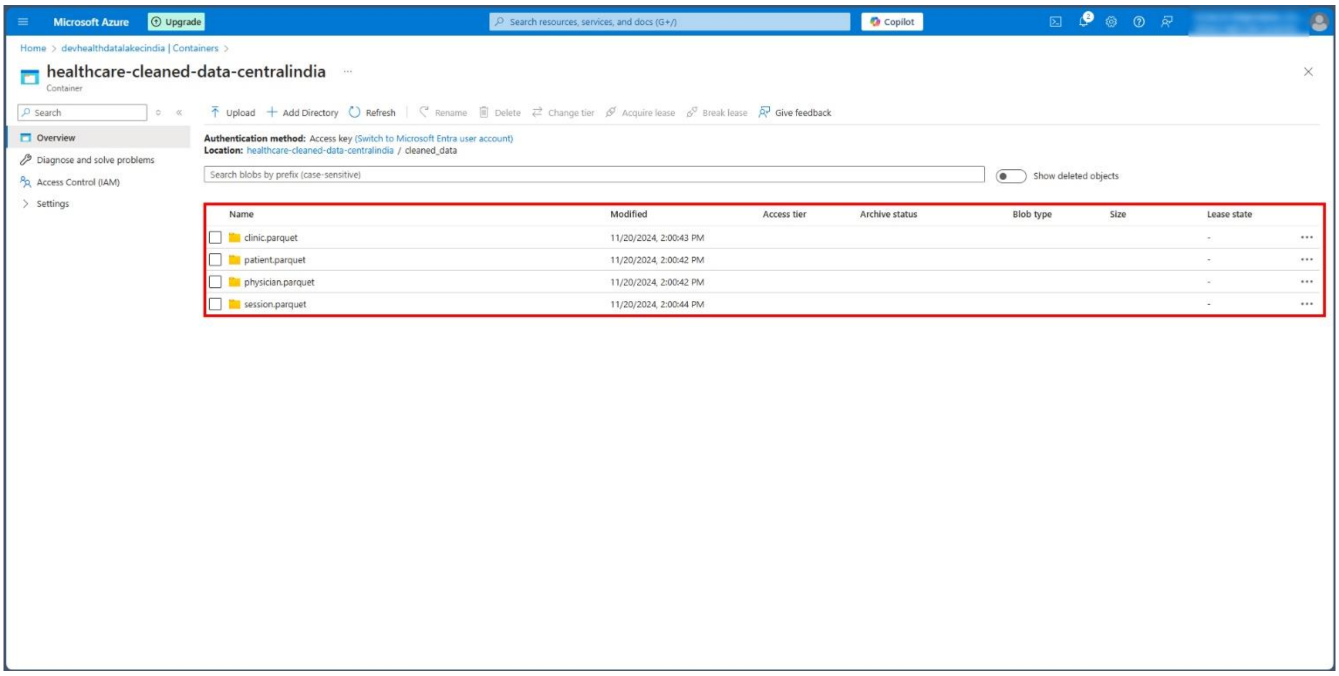
Container

**Authentication method:** Access key (Switch to Microsoft Entra user account)
**Location:** healthcare-staging-data-centralindia / staging_layer

| Name | Modified | Access tier | Archive status | Blob type | Size | Lease state |
|---|---|---|---|---|---|---|
| clinic.csv | 11/21/2024, 1:00:28 PM | Hot (Inferred) | | Block blob | 999 B | Available |
| patient.csv | 11/21/2024, 12:59:58 PM | Hot (Inferred) | | Block blob | 5.85 KiB | Available |
| physician.csv | 11/21/2024, 1:00:13 PM | Hot (Inferred) | | Block blob | 976 B | Available |
| session.csv | 11/21/2024, 1:00:55 PM | Hot (Inferred) | | Block blob | 2.34 KiB | Available |

### Cleaned Data Container

Cleaned data loaded from Azure Databricks can be accessed here.



healthcare-cleaned-data-centralindia

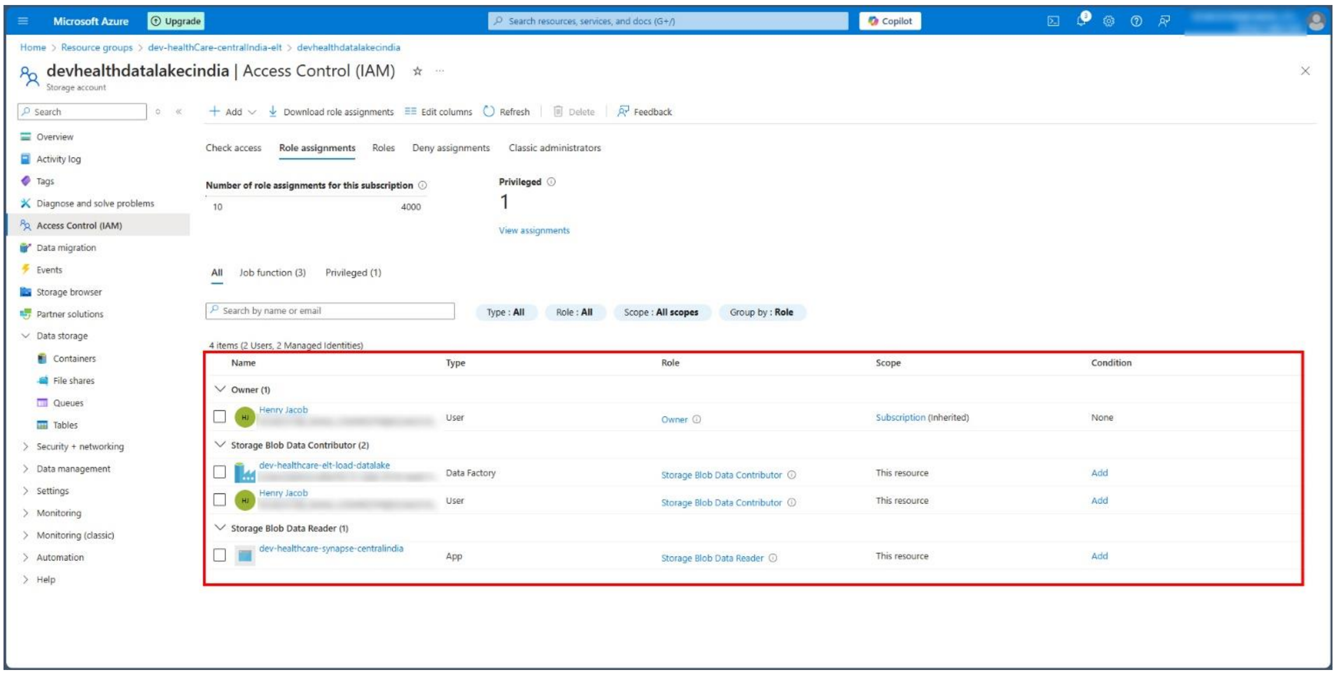
Container

**Authentication method:** Access key (Switch to Microsoft Entra user account)
**Location:** healthcare-cleaned-data-centralindia / cleaned_data

| Name | Modified | Access tier | Archive status | Blob type | Size | Lease state |
|---|---|---|---|---|---|---|
| clinic.parquet | 11/20/2024, 2:00:43 PM | | | | | - |
| patient.parquet | 11/20/2024, 2:00:42 PM | | | | | - |
| physician.parquet | 11/20/2024, 2:00:42 PM | | | | | - |
| session.parquet | 11/20/2024, 2:00:44 PM | | | | | - |

## Azure Data Lake Gen2 –Staged and Cleaned Data – IAM

- **Owner** – Since I am the owner of the account.
- **Storage Blob Data Contributor** – Assigned to Azure Data Factory (since the files need to be written by Azure Data Factory).
- **Storage Blob Data Contributor** – Assigned to myself for Azure Databricks (As I enabled credential pass-through to read and write).
- **Storage Blob Data Reader** – Assigned to Azure Synapse Analytics (since the files need to be read by Azure Synapse Analytics).



devhealthdatalakecindia | Access Control (IAM)
Storage account

Number of role assignments for this subscription: 10 / 4000

Privileged: 1

Type : **All** · Role : **All** · Scope : **All scopes** · Group by : **Role**

4 items (2 Users, 2 Managed Identities)

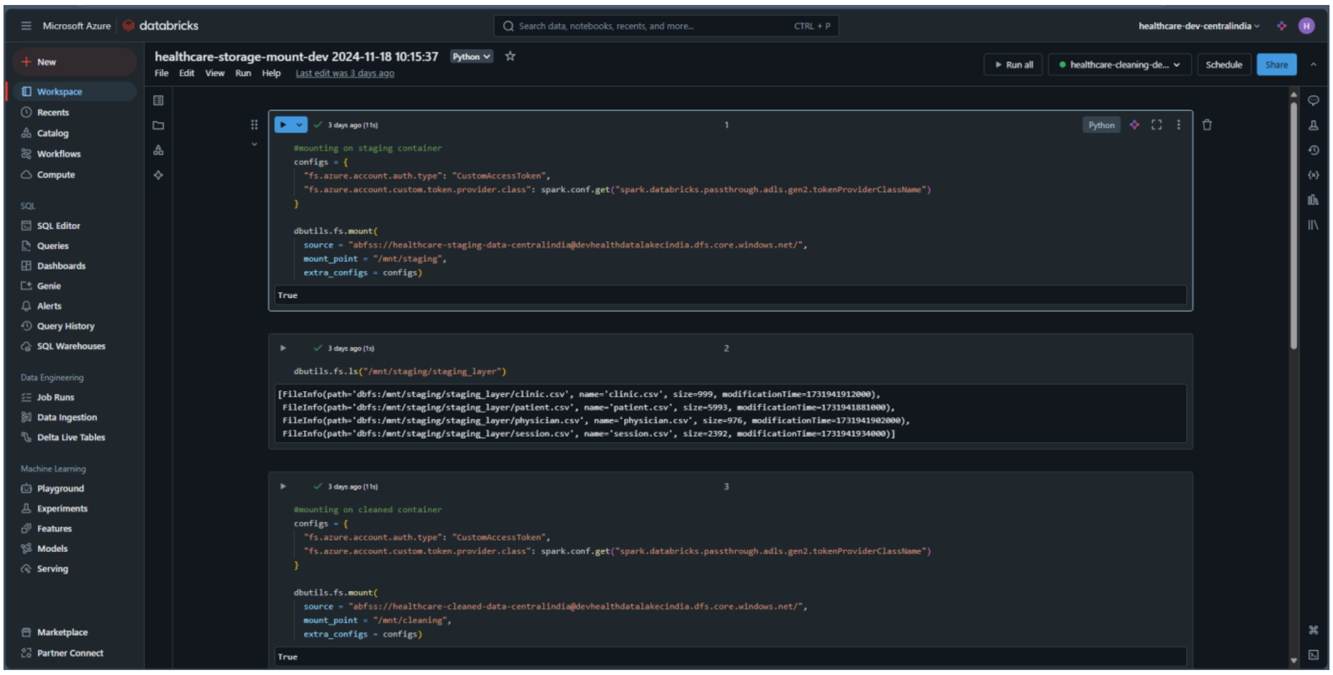
| Name | Type | Role | Scope | Condition |
|---|---|---|---|---|
| **Owner (1)** | | | | |
| Henry Jacob [redacted] | User | Owner | Subscription (Inherited) | None |
| **Storage Blob Data Contributor (2)** | | | | |
| dev-healthcare-elt-load-datalake [redacted] | Data Factory | Storage Blob Data Contributor | This resource | Add |
| Henry Jacob [redacted] | User | Storage Blob Data Contributor | This resource | Add |
| **Storage Blob Data Reader (1)** | | | | |
| dev-healthcare-synapse-centralindia | App | Storage Blob Data Reader | This resource | Add |

## Azure Databricks

The notebooks can be accessed here.

## For mounting

A separate notebook was created for the mounting purpose.


Microsoft Azure
databricks
healthcare-storage-mount-dev 2024-11-18 10:15:37
Python
Last edit was 3 days ago
Run all
Schedule
Share
healthcare-dev-centralindia

```
#mounting on staging container
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://healthcare-staging-data-centralindia@devhealthdatalakecindia.dfs.core.windows.net/",
  mount_point = "/mnt/staging",
  extra_configs = configs)
```

```
True
```

```
dbutils.fs.ls("/mnt/staging/staging_layer")
```

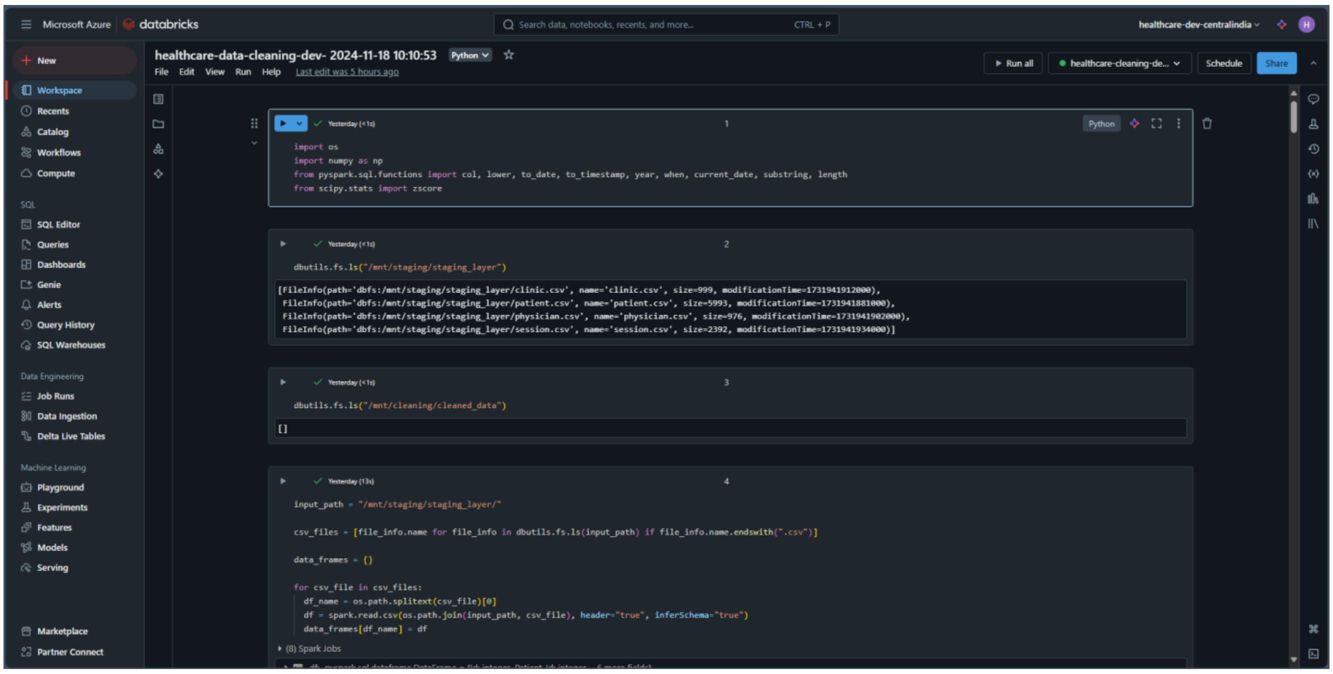
```
[FileInfo(path='dbfs:/mnt/staging/staging_layer/clinic.csv', name='clinic.csv', size=999, modificationTime=1731941912000),
 FileInfo(path='dbfs:/mnt/staging/staging_layer/patient.csv', name='patient.csv', size=5993, modificationTime=1731941881000),
 FileInfo(path='dbfs:/mnt/staging/staging_layer/physician.csv', name='physician.csv', size=976, modificationTime=1731941902000),
 FileInfo(path='dbfs:/mnt/staging/staging_layer/session.csv', name='session.csv', size=2392, modificationTime=1731941934000)]
```

```
#mounting on cleaned container
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://healthcare-cleaned-data-centralindia@devhealthdatalakecindia.dfs.core.windows.net/",
  mount_point = "/mnt/cleaning",
  extra_configs = configs)
```

```
True
```

## For cleaning

Another notebook was created for the cleaning purpose.


Microsoft Azure
databricks
healthcare-data-cleaning-dev- 2024-11-18 10:10:53
Python
Last edit was 5 hours ago
Run all
Schedule
Share
healthcare-dev-centralindia

```
import os
import numpy as np
from pyspark.sql.functions import col, lower, to_date, to_timestamp, year, when, current_date, substring, length
from scipy.stats import zscore
```

```
dbutils.fs.ls("/mnt/staging/staging_layer")
```

```
[FileInfo(path='dbfs:/mnt/staging/staging_layer/clinic.csv', name='clinic.csv', size=999, modificationTime=1731941912000),
 FileInfo(path='dbfs:/mnt/staging/staging_layer/patient.csv', name='patient.csv', size=5993, modificationTime=1731941881000),
 FileInfo(path='dbfs:/mnt/staging/staging_layer/physician.csv', name='physician.csv', size=976, modificationTime=1731941902000),
 FileInfo(path='dbfs:/mnt/staging/staging_layer/session.csv', name='session.csv', size=2392, modificationTime=1731941934000)]
```

```
dbutils.fs.ls("/mnt/cleaning/cleaned_data")
```

```
[]
```

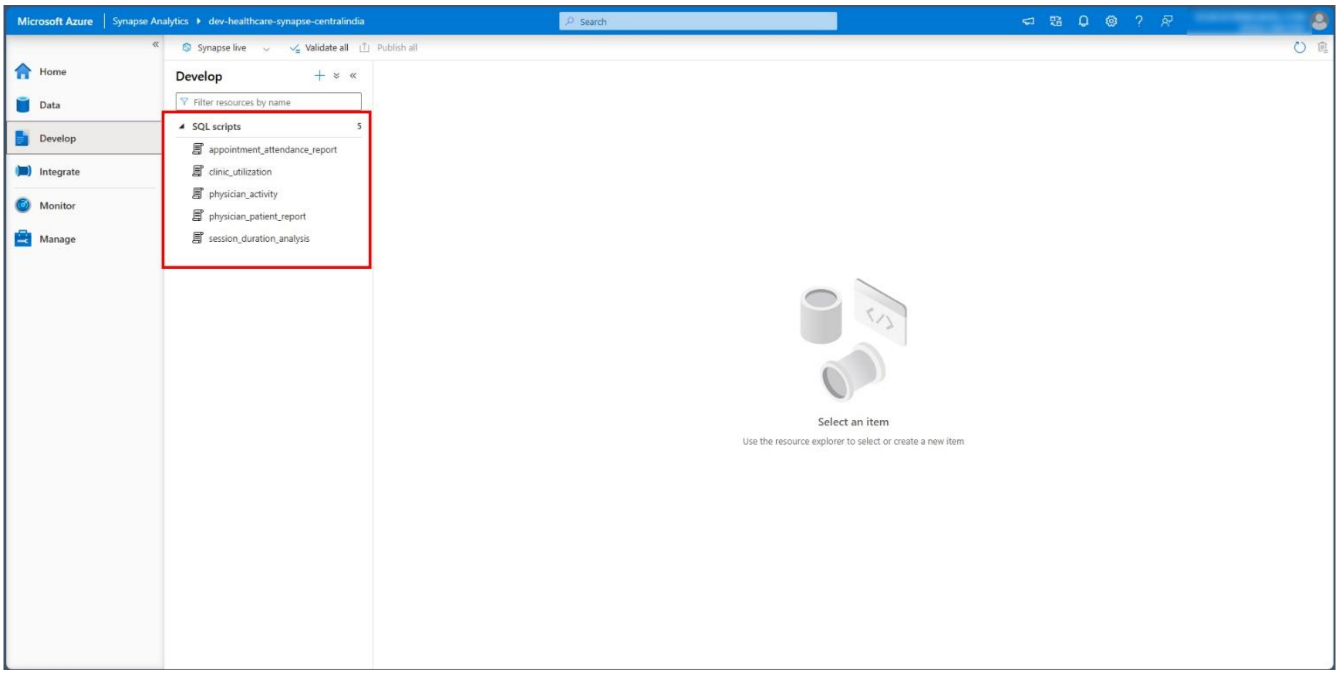
```
input_path = "/mnt/staging/staging_layer/"

csv_files = [file_info.name for file_info in dbutils.fs.ls(input_path) if file_info.name.endswith(".csv")]

data_frames = {}

for csv_file in csv_files:
  df_name = os.path.splitext(csv_file)[0]
  df = spark.read.csv(os.path.join(input_path, csv_file), header="true", inferSchema="true")
  data_frames[df_name] = df
```

```
(8) Spark Jobs
```

## Azure Synapse Analytics

### Scripts

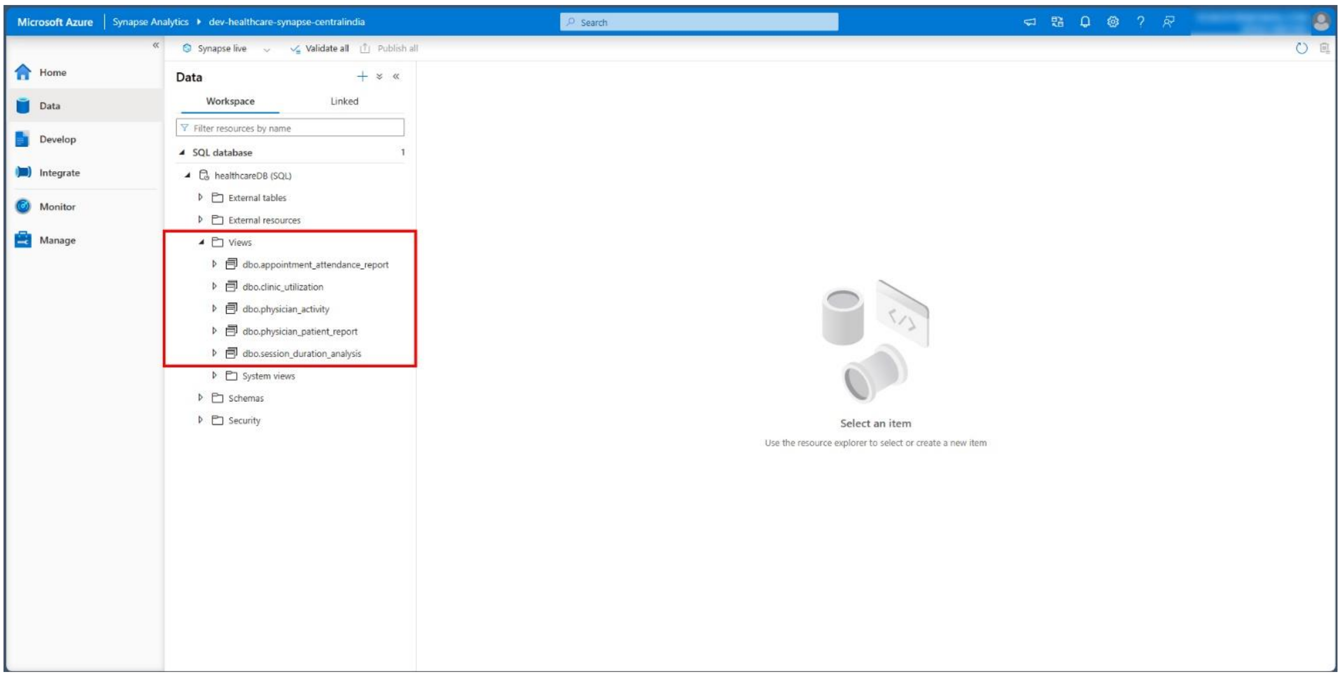
The scripts used in synapse analytics can be accessed here.



SQL scripts 5
- appointment_attendance_report
- clinic_utilization
- physician_activity
- physician_patient_report
- session_duration_analysis

### Views

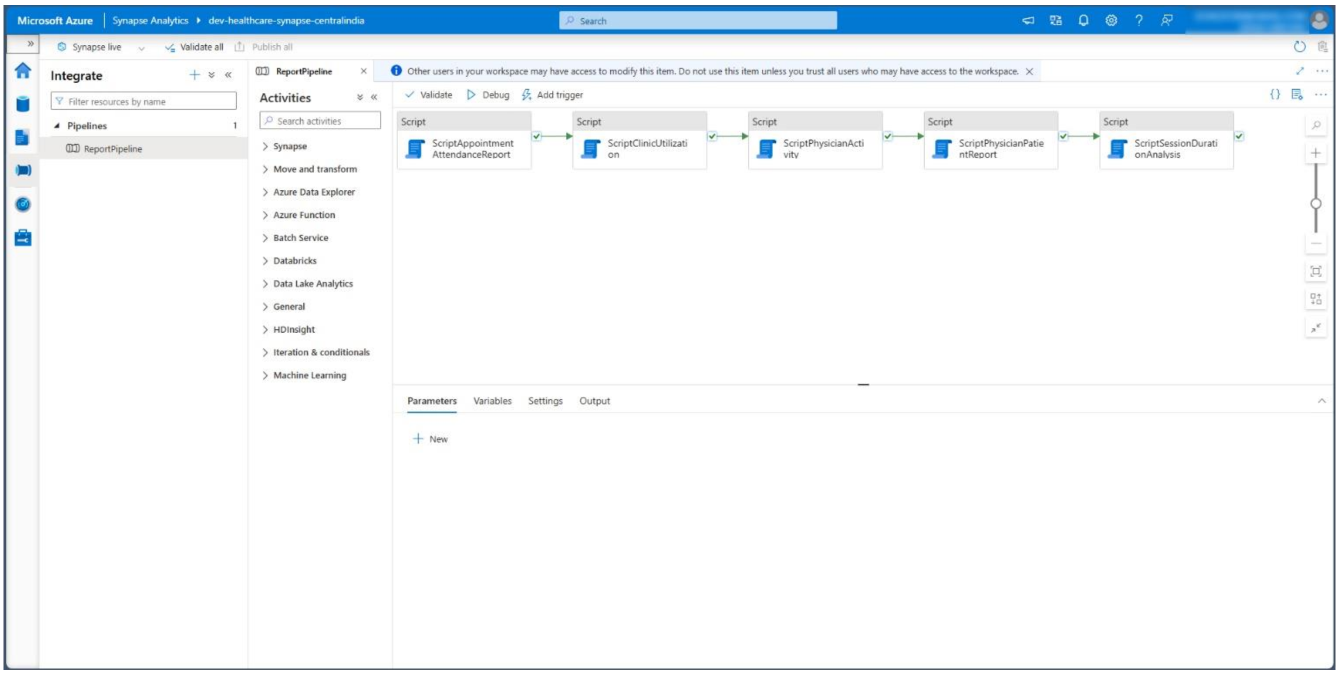
The above scripts are used to create these views.



SQL database 1
- healthcareDB (SQL)
  - External tables
  - External resources
  - Views
    - dbo.appointment_attendance_report
    - dbo.clinic_utilization
    - dbo.physician_activity
    - dbo.physician_patient_report
    - dbo.session_duration_analysis
    - System views
  - Schemas
  - Security

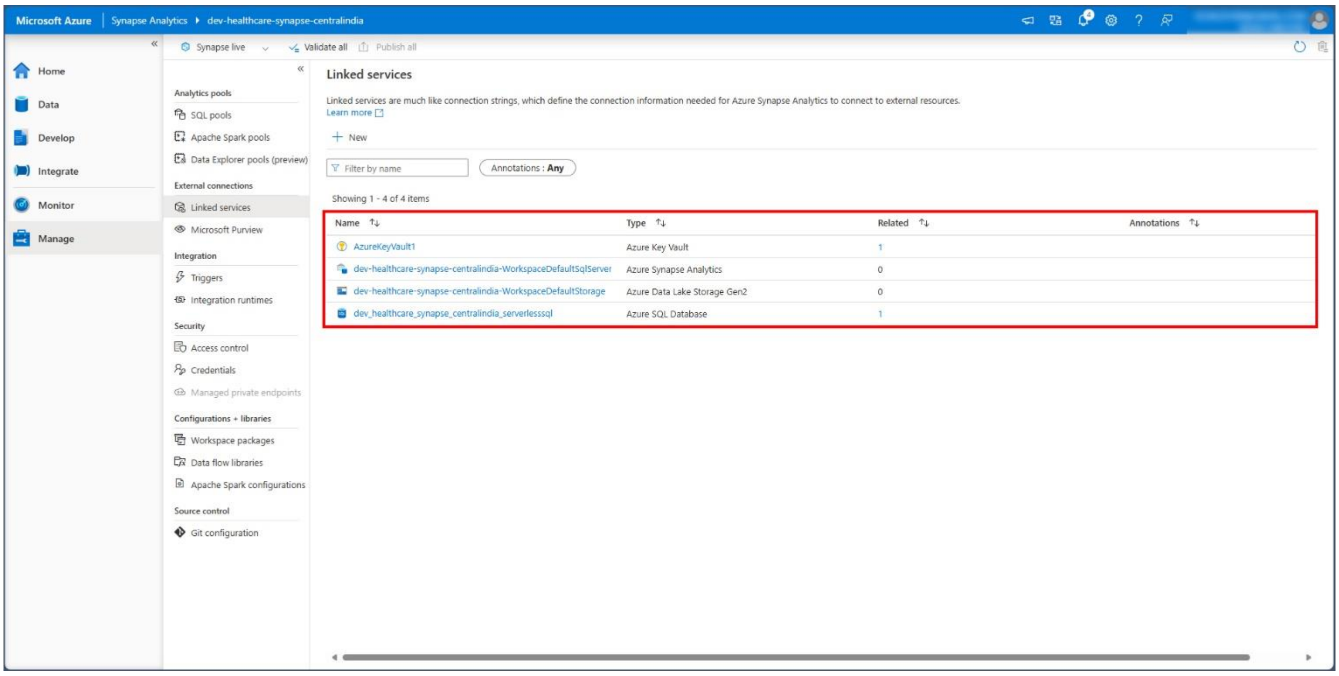
## Pipeline

All the scripts activities were added sequentially for reporting layer.


Microsoft Azure
Synapse Analytics
dev-healthcare-synapse-centralindia
Integrate
Pipelines
ReportPipeline
Activities
Validate
Debug
Add trigger
Other users in your workspace may have access to modify this item. Do not use this item unless you trust all users who may have access to the workspace.
Script
ScriptAppointmentAttendanceReport
Script
ScriptClinicUtilization
Script
ScriptPhysicianActivity
Script
ScriptPhysicianPatientReport
Script
ScriptSessionDurationAnalysis
Parameters
Variables
Settings
Output

## Linked Services

- **AzureKeyVault1** – to securely access credentials (SQL password).
- **dev-healthcare-synapse-centralindia-WorkspaceDefaultSqlServer** – default linked service for Synapse Analytics SQL Server.
- **dev-healthcare-synapse-centralindia-WorkspaceDefaultStorage** – default linked service for Synapse Analytics Storage.
- **dev_healthcare_synapse_centralindia_serverlesssql** – to connect with the SQL database.



Linked services are much like connection strings, which define the connection information needed for Azure Synapse Analytics to connect to external resources.

Showing 1 - 4 of 4 items

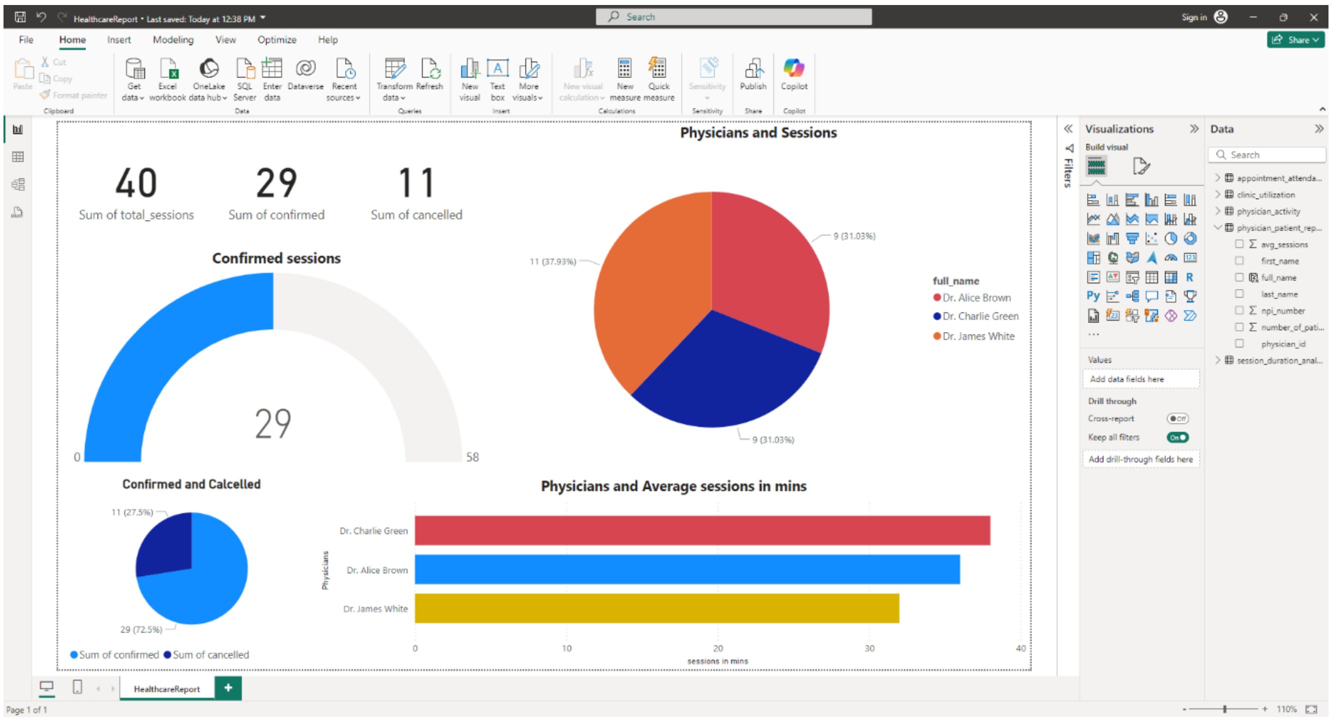
| Name | Type | Related | Annotations |
|---|---|---|---|
| AzureKeyVault1 | Azure Key Vault | 1 | |
| dev-healthcare-synapse-centralindia-WorkspaceDefaultSqlServer | Azure Synapse Analytics | 0 | |
| dev-healthcare-synapse-centralindia-WorkspaceDefaultStorage | Azure Data Lake Storage Gen2 | 0 | |
| dev_healthcare_synapse_centralindia_serverlesssql | Azure SQL Database | 1 | |

### Power BI

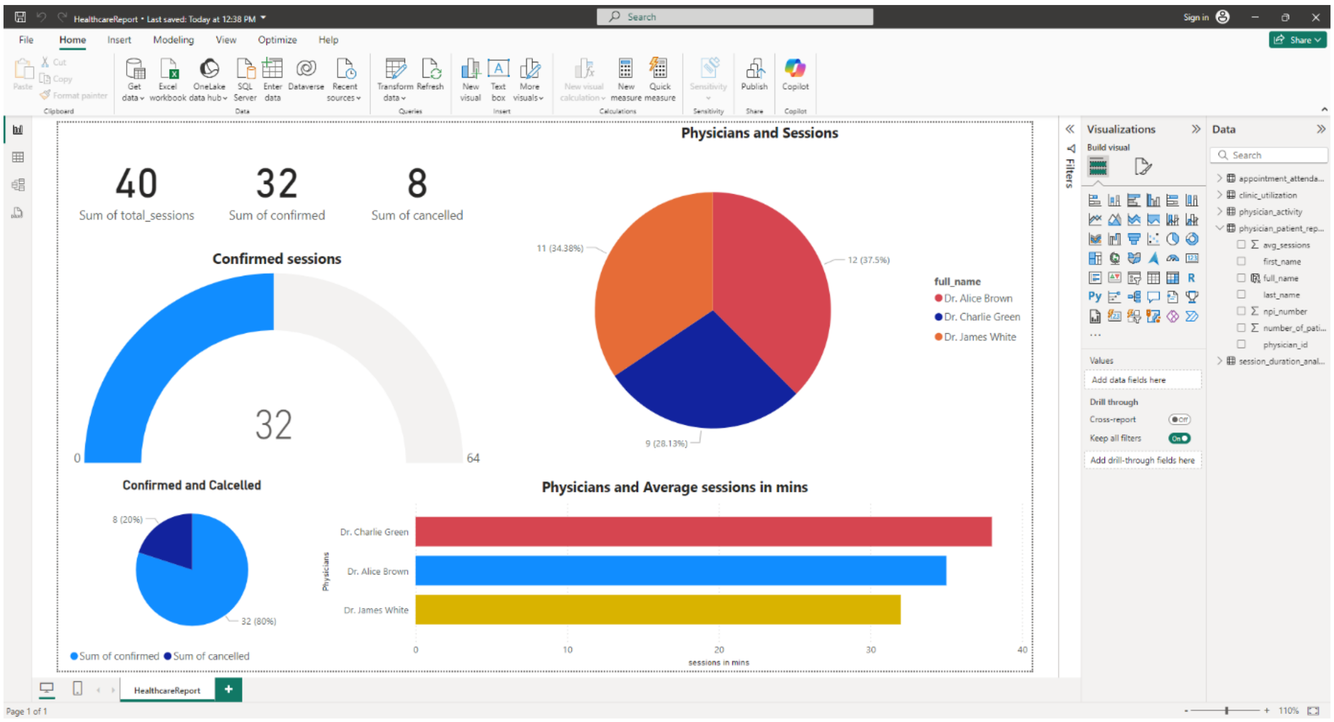
Power BI Desktop is used for visualizations.

The visualizations were checked by updating the raw session status. The visualizations are attached before and after updating the raw data.

### Before session status update

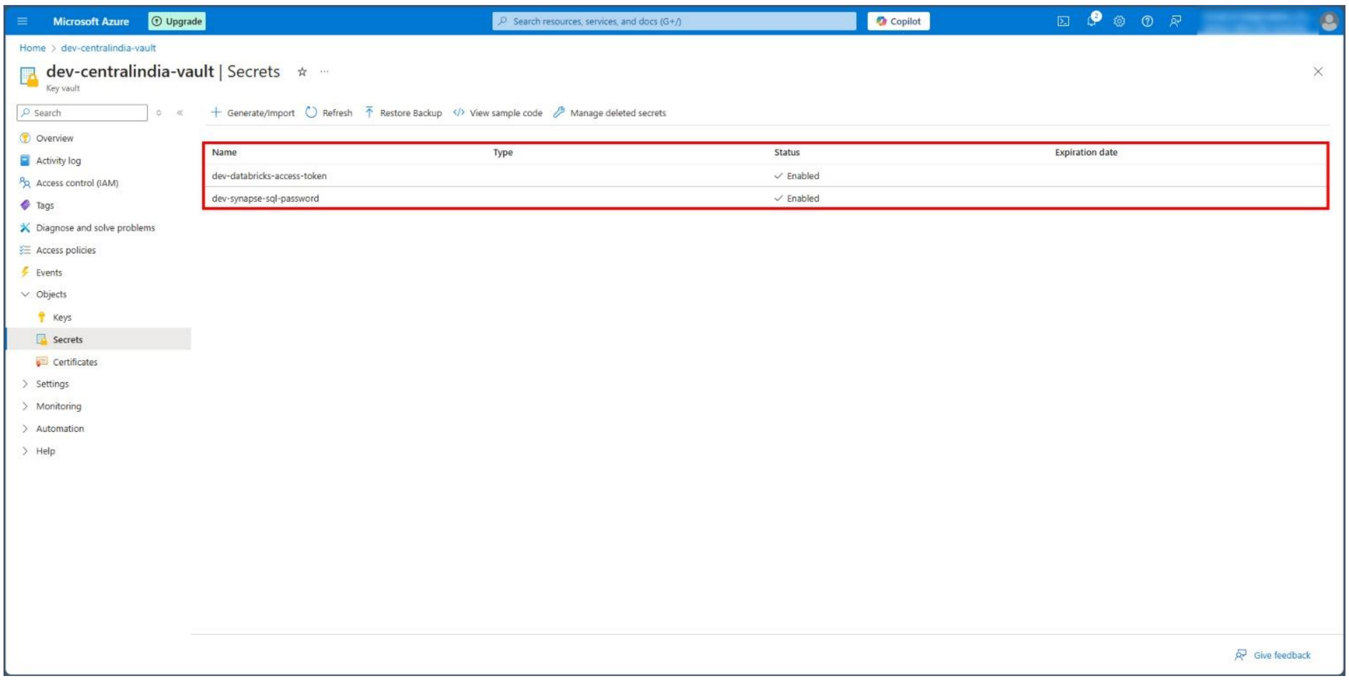


### After session status update



## Azure Key Vault

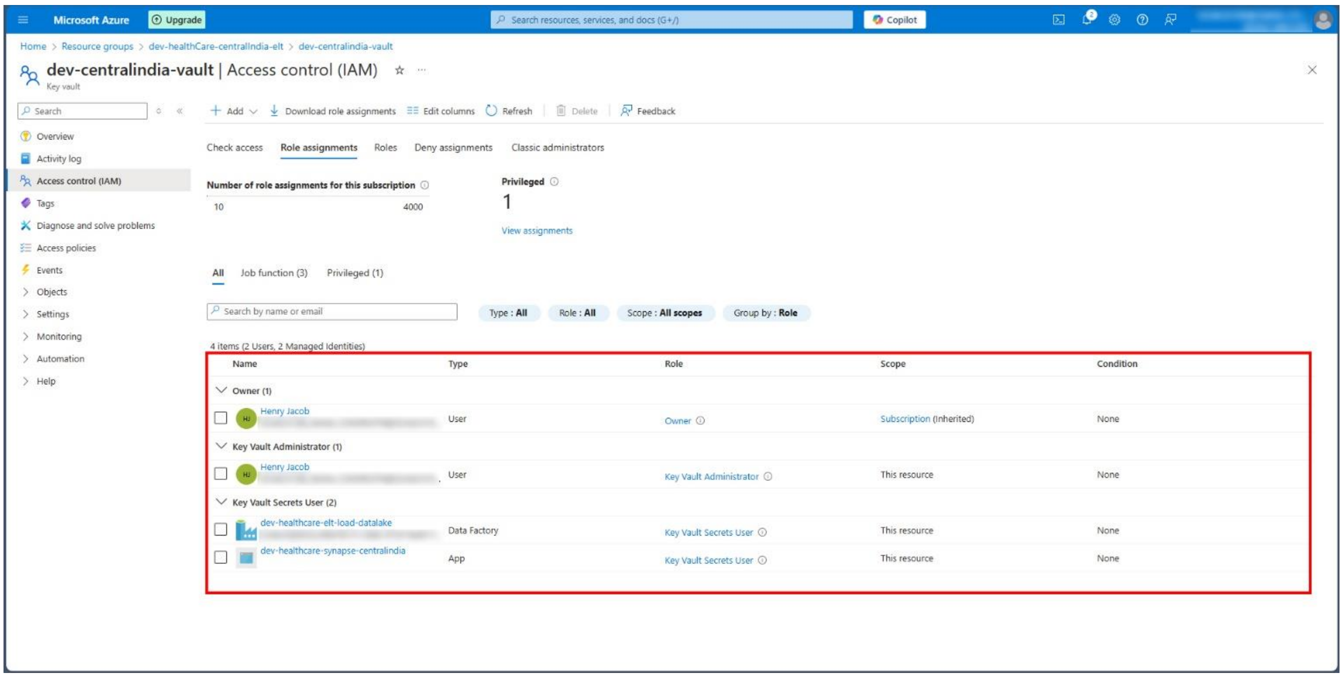
Credentials are stored inside the **Key Vault**.



| Name | Type | Status | Expiration date |
|---|---|---|---|
| dev-databricks-access-token | | ✓ Enabled | |
| dev-synapse-sql-password | | ✓ Enabled | |

## Key Vault – IAM

- **Owner** – Since I am the owner of the account.
- **Key Vault Administrator** – Assigned to myself (I need to be an administrator to assign permissions to others).
- **Key Vault Secret User** – Assigned to Azure Data Factory (since the credentials [Databricks Access Token] needs to be read by Azure Data Factory).
- **Key Vault Secret User** – Assigned to Azure Synapse Analytics (since the credentials [SQL Password] needs to be read by Azure Synapse Analytics).



4 items (2 Users, 2 Managed Identities)

| Name | Type | Role | Scope | Condition |
|---|---|---|---|---|
| Owner (1) | | | | |
| Henry Jacob [redacted] | User | Owner | Subscription (Inherited) | None |
| Key Vault Administrator (1) | | | | |
| Henry Jacob [redacted] | User | Key Vault Administrator | This resource | None |
| Key Vault Secrets User (2) | | | | |
| dev-healthcare-elt-load-datalake [redacted] | Data Factory | Key Vault Secrets User | This resource | None |
| dev-healthcare-synapse-centralindia | App | Key Vault Secrets User | This resource | None |

***Don't forget to delete unused resources*** after the project is completed

to avoid unnecessary charges.

---

It is **not recommended** to **share** your **storage account name**, **access keys**, or other sensitive information **in public**.

However, I am sharing the screenshot here to demonstrate the standard naming convention I maintained.

I deleted all my resources before publishing this document, and I do not plan to reuse or recover the resources.