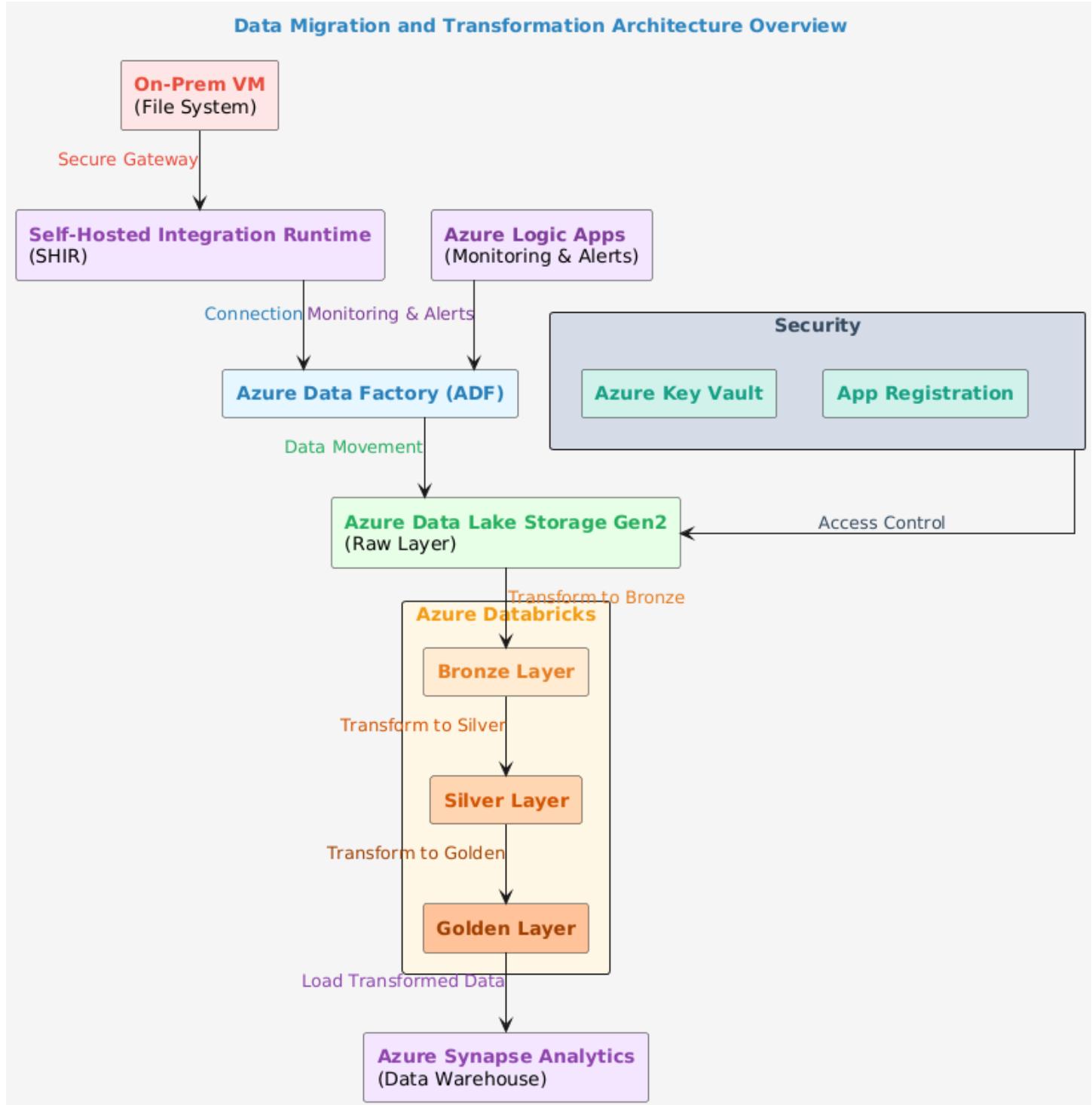


# On-Prem to Azure Data Migration Architecture

This architecture outlines a comprehensive approach for migrating data from an on-premises environment to Azure using various Azure services. The required resources include:



Follow on linkedin  
@Shivakiran kotur

1. **On-Prem VM and File System:** The on-prem VM will host the file system with data in formats such as TXT, CSV, and Parquet.
2. **Azure Data Factory (ADF):** ADF will serve as the primary tool for orchestrating and automating the data migration process.
3. **Azure Data Lake Storage Gen2 (ADLS Gen2):** Used to store raw, preprocessed, and processed data. The raw data will be stored in the landing layer, while cleaned and transformed data will be moved to the preprocessed and processed layers.
4. **Azure Synapse Analytics:** The destination data warehouse where the final processed data will be loaded for analytics and reporting.
5. **Azure Databricks with PySpark:** Used to create the bronze, silver, and golden layers. This involves cleaning the data by removing nulls and duplicates, transforming data by joining tables, and applying business logic to create structured and ready-for-reporting datasets.
6. **Azure App Registration:** Facilitates secure connections to ADLS Gen2 via mount points for data access.
7. **Azure Logic Apps:** Used to set up alert mechanisms for monitoring the migration and data processing workflows.
8. **Azure Key Vault:** Stores and manages secrets, keys, and credentials for secure access to Azure resources.

## Architecture Overview:

1. **On-Prem Data Storage:** We start by establishing an on-prem VM with a file system containing data in formats such as TXT, CSV, and Parquet.
2. **Connecting On-Prem to Azure:** Once Azure Data Factory (ADF) is set up, we use the Self-Hosted Integration Runtime (SHIR) to create a secure gateway between the on-prem VM and Azure. The SHIR allows ADF to connect to on-prem data sources.
3. **Data Movement to ADLS Gen2:** Using ADF, we perform various activities like Lookup, Metadata, Copy, and Stored Procedure activities to move the data from on-premises to Azure Data Lake Storage Gen2 (ADLS Gen2), specifically into the raw or landing layer.
4. **Data Transformation in Azure Databricks:** In Azure Databricks, we create the bronze, silver, and golden layers:



- **Bronze Layer:** Raw data is cleaned by removing nulls, duplicates, and irrelevant records.
- **Silver Layer:** Preprocessed data is transformed with additional logic, such as applying joins or other business rules.
- **Golden Layer:** The final, cleaned, and transformed data is ready for reporting and analytics.

**5. Data Loading into Synapse:** The processed data is then loaded into **Azure Synapse Analytics** as a data warehouse, where it will be used for analytics and reporting.

## 6. Security and Connectivity:

- **App Registration:** Provides secure access to ADLS Gen2 by using mount points.
- **Azure Key Vault:** Ensures the safe storage of secrets and credentials required for accessing resources.

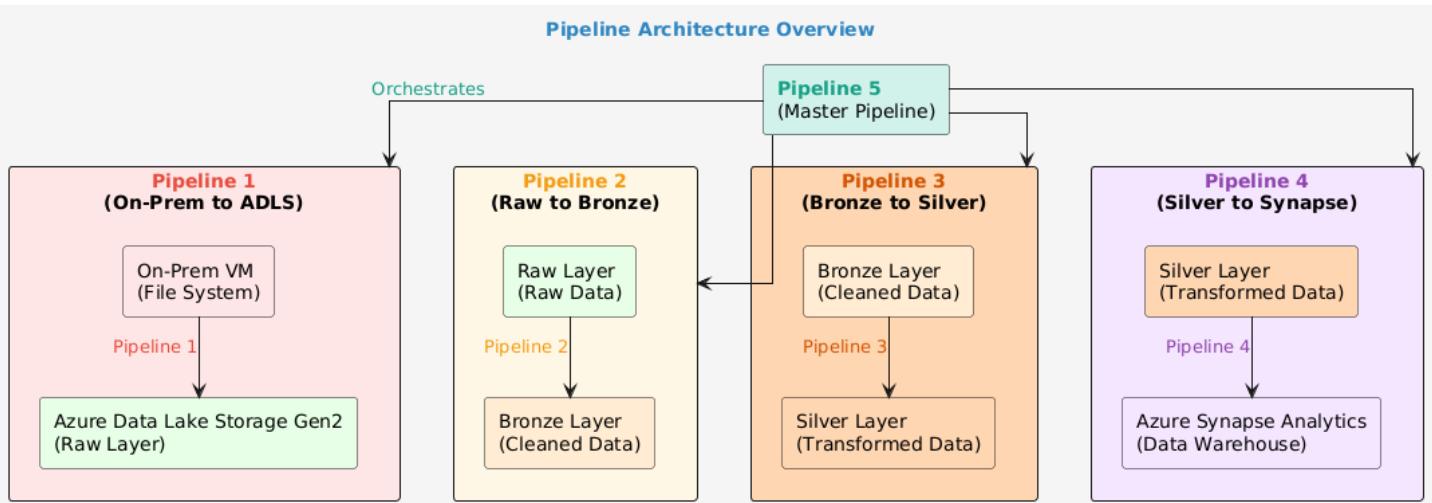
**7. Monitoring and Alerts:** **Azure Logic Apps** are configured to send alerts on various activities, such as task failures or successful migrations, ensuring smooth monitoring of the pipeline.

This architecture ensures a secure, automated, and scalable solution for migrating on-prem data to Azure, transforming it for analytics, and storing it in a centralized data warehouse.

## Here are the pipeline headings:

1. **Pipeline 1: On-Prem to ADLS**
2. **Pipeline 2: Raw ADLS to Bronze ADLS**
3. **Pipeline 3: Bronze ADLS to Silver ADLS**
4. **Pipeline 4: Silver ADLS to SQL Data Warehouse (Synapse)**
5. **Pipeline 5: Master Pipeline**





**Step → create on-prem VM**

Microsoft Azure

All services > Virtual machines >

Create a virtual machine ...

Basics Disks Networking Management Monitoring Advanced SQL Server settings Tags Review + create

Create a virtual machine that runs Linux or Windows. Select an image from Azure marketplace or use your own customized image. Complete the Basics tab then Review + create to provision a virtual machine with default parameters or review each tab for full customization. [Learn more](#)

This subscription may not be eligible to deploy VMs of certain sizes in certain regions.

**Project details**

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription \*  Azure for Students

Resource group \*  svkdev

**Instance details**

Virtual machine name \*

Region \*  (US) East US

Availability options  Availability zone

Availability zone \*  Zones 1

You can now select multiple zones. Selecting multiple zones will create one VM per zone. [Learn more](#)

< Previous Next : Disks >



Follow on linkedin  
@Shivakiran kotur

**Microsoft Azure** Search resources, services, and docs (G+)

All services > Virtual machines >

### Create a virtual machine

Security type: Standard

Image: SQL Server 2019 Enterprise on Windows Server 2019 - x64 Gen1

VM architecture: x64

Run with Azure Spot discount:

Size: Standard\_D2s\_v3 - 2 vcpus, 8 GiB memory (₹10,778.07/month)

Administrator account:

- Username: shivvk
- Password:
- Confirm password:

Inbound port rules: Select which virtual machine network ports are accessible from the public internet. You can specify more limited or granular.

**Microsoft Azure** Search resources, services, and docs (G+)

All services > Virtual machines >

### Create a virtual machine

Basics Disks Networking Management Monitoring Advanced **SQL Server settings** Tags Review + create

**Security & Networking**

SQL connectivity: Private (within Virtual Network)

Port: 1433

**SQL Authentication**

SQL Authentication:  Enable

Login name: shivvk

Password:

Azure Key Vault integration:  Disable  Enable

**Storage configuration**

Customize performance, size, and workload type to optimize storage for this virtual machine. For optimal performance, separate drives will be created for data and log storage by default. [Learn more about SQL Server best performance practices.](#)

Storage: SQL Data: 1024 GiB, 5000 IOPS, 200 MB/s, Premium SSD  
SQL Log: 1024 GiB, 5000 IOPS, 200 MB/s, Premium SSD  
SQL TempDb: Use local SSD drive [Change configuration](#)

**SQL instance settings**

Customize additional SQL instance settings including collation, MAXDOP, server memory limit and optimize for ad-hoc

**Review + create** < Previous Next : Tags >

Under imagine select your sql server

Next directly go to sql server by giving next → next → enable the sql auth



Follow on linkedin  
@Shivakiran kotur

## Step2 → create adf

All services > onpremeadls\_1698390608681 | Overview > onpremeadls | Containers >

global Container

Search

Upload Add Directory Refresh Rename Delete Change tier

Overview Diagnose and solve problems Access Control (IAM)

Authentication method: Access key (Switch to Microsoft Entra user account)  
Location: global

Search blobs by prefix (case-sensitive)

Name	Modified
bronze	
raw	
silver	

Settings Shared access tokens Manage ACL Properties Metadata

## Step 3 → create adls gen 2 storage account

create a container → global

under global → create 3 directory as raw, bronze, silver

step 4 → create key vault

Home > Key vaults >

Create a key vault ...

Basics Access configuration Networking Tags Review + create

Configure data plane access for this key vault

To access a key vault in data plane, all callers (users or applications) must have proper authentication and authorization. Authentication establishes the identity of the caller. Authorization determines which operations the caller can execute.

Permission model

Grant data plane access by using a [Azure RBAC](#) or [Key Vault access policy](#)

Azure role-based access control (recommended)  Vault access policy

Resource access

Azure Virtual Machines for deployment   
 Azure Resource Manager for template deployment   
 Azure Disk Encryption for volume encryption

Access policies

Access policies enable you to have fine grained control over access to vault items. [Learn more](#)

Name ↑↓	Email ↑↓	Key Permissions	Secret Permissions	Certifica
USER	NAMITHA.2020ECE043@presidencyuniversity.in	Get, List, Update, Create, Import, Delete, Recover, Bac... Get, List, Set, Delete, Recover, Backup, Resto	Get, List	

1 Permissions 2 Principal 3 Application (optional) 4 Review + create

Configure from a template

Key, Secret, & Certificate Management

Key permissions	Secret permissions	Certificate permissions
Key Management Operations	Secret Management Operations	Certificate Management Operations
<input checked="" type="checkbox"/> Select all	<input checked="" type="checkbox"/> Select all	<input checked="" type="checkbox"/> Select all
<input checked="" type="checkbox"/> Get	<input checked="" type="checkbox"/> Get	<input checked="" type="checkbox"/> Get
<input checked="" type="checkbox"/> List	<input checked="" type="checkbox"/> List	<input checked="" type="checkbox"/> List
<input checked="" type="checkbox"/> Update	<input checked="" type="checkbox"/> Set	<input checked="" type="checkbox"/> Update
<input checked="" type="checkbox"/> Create	<input checked="" type="checkbox"/> Delete	<input checked="" type="checkbox"/> Create
<input checked="" type="checkbox"/> Import	<input checked="" type="checkbox"/> Recover	<input checked="" type="checkbox"/> Import
<input checked="" type="checkbox"/> Delete	<input checked="" type="checkbox"/> Backup	<input checked="" type="checkbox"/> Delete
<input checked="" type="checkbox"/> Recover	<input checked="" type="checkbox"/> Restore	<input checked="" type="checkbox"/> Recover
<input checked="" type="checkbox"/> Backup		<input checked="" type="checkbox"/> Backup
<input checked="" type="checkbox"/> Restore		<input checked="" type="checkbox"/> Restore
Cryptographic Operations	Privileged Secret Operations	
<input type="checkbox"/> Select all	<input type="checkbox"/> Select all	
<input type="checkbox"/> Decrypt	<input type="checkbox"/> Purge	
<input type="checkbox"/> Encrypt		

Previous Next

## step 5 → create a dedicated sql pool(i.e sql dw)

Microsoft Azure Search resources, services, and docs (G+)

Home > Dedicated SQL pools (formerly SQL DW) > Create dedicated SQL pool (formerly SQL DW)

Configure performance

Configure your performance level that best fits your needs.

**Gen2** Offers the highest performance and storage scalability options for intensive workloads.  
Starting at 118.59 INR / hour

**Gen1** Offers the lowest compute scale options for less demanding workloads.  
Not available  
Starting at -- / hour

Learn more about performance level ricing  
Scale your system DW100c DW100c 118.59 INR / hour 100 cDWU

**Project details**  
Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription \* Azure for Students  
Resource group \* svkdev Create new

**SQL pool details**  
Enter required settings for this SQL pool, including picking a logical server and configuring the performance level.

SQL pool name \* testpool  
Server \* svkloaddata (East US) Create new

Performance level \* Gen2 DW1000c Select performance level

Review + create Next : Networking > Apply

## Create an access policy

1 Permissions 2 Principal 3 Application (optional) 4 Review + create

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](#)

adfdev

adfdevsvk a3fc6c6a-6914-4b89-81eb-6c617c99e2f6

Under principal give access to your ADF so that adf can access the key vault

After giving access review and create key vault.



Follow on linkedin  
@Shivakiran\_kotur

## Step 6 → go to VM and connect to VM

The screenshot shows the Microsoft Azure portal interface. On the left, the navigation pane is open with the 'Connect' option selected under 'Settings'. In the center, the details for a virtual machine named 'onpremenvm' are displayed, including its public IP address (20.119.100.139). To the right, a 'Remote Desktop Connection' window is open, prompting for a computer name (set to 'Example: computer.fabrikam.com') and credentials (User name: None specified). Below these windows, a 'Native RDP' connection method is highlighted with a 'Select' button.

In your local open RDP and paste the ip address and give credentials and login in.

Minimize the RDP

## Step 7 → go to ADF and create Self hosted run time

The screenshot shows the 'Integration runtime setup' page within the Azure Data Factory service. The left sidebar lists various factory settings like General, Connections, and Integration runtimes. The main area displays the 'Integration runtimes' section, which lists one item: 'AutoResolveIntegrationRuntime' of type 'Azure' and status 'Running'. To the right, there's a list of available integration runtime types: 'Azure Self-Hosted', 'Azure-SSIS', and 'Airflow (Preview)'. Each entry includes a brief description of its function.

## Integration runtime setup

### Network environment:

Choose the network environment of the data source / destination or external compute to which the integration runtime will connect to for data flows, data movement or dispatch activities:



#### Azure

Use this for running data flows, data movement, external and pipeline activities in a fully managed, serverless compute in Azure.



#### Self-Hosted

Use this for running activities in an on-premises / private network

[View more](#) ▾

### External Resources:

You can use an existing self-hosted integration runtime that exists in another resource. This way you can reuse your existing infrastructure where self-hosted integration runtime is setup.



#### Linked Self-Hosted

[Learn more](#) ↗

[Continue](#)

[Back](#)

[Cancel](#)



Follow on linkedin  
@Shivakiran kotur

**Integration runtime setup**

Settings Nodes Auto update Sharing Links

Install integration runtime on Windows machine or add further nodes using the Authentication Key.

Name

**Option 1: Express setup**  
Click here to launch the express setup for this computer

**Option 2: Manual setup**  
Step 1: Download and install integration runtime  
Step 2: Use this key to register your integration runtime

Name   
Key1     
Key2   

**Close**

Here download the integration runtime in your VM (remember to copy the keys)  
Step → go to VM → local server → off the enhanced security

Server Manager

Server Manager • Local Server

Dashboard Local Server All Servers File and Storage Services

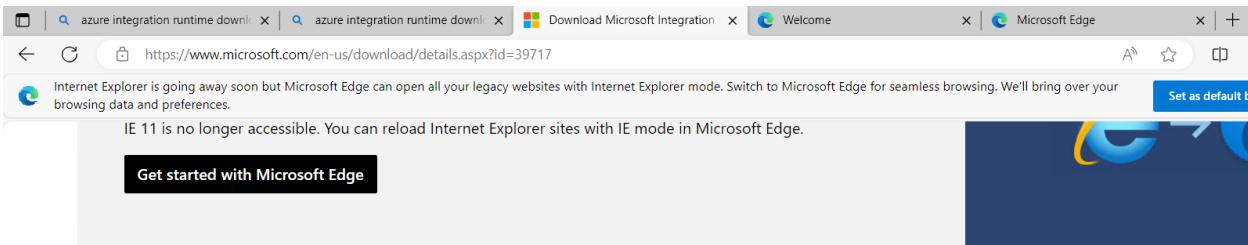
**PROPERTIES** For onpremenvm

Computer name	onpremenvm	Last installed updates	10/17/2023 2:05 AM
Workgroup	WORKGROUP	Windows Update	Download updates only, using Microsoft Update
		Last checked for updates	Today at 7:14 AM
Windows Defender Firewall	Public: On	Windows Defender Antivirus	Real-Time Protection: On
Remote management	Enabled	Feedback & Diagnostics	Settings
Remote Desktop	Enabled	IE Enhanced Security Configuration	On
NIC Teaming	Disabled	Time zone	(UTC) Coordinated Universal Time
Ethernet 2	IPv4 address assigned by DHCP, IPv6 enabled	Product ID	00430-00000-00000-AA481 (activated)
Operating system version	Microsoft Windows Server 2019 Datacenter	Processors	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz
Hardware information	Microsoft Corporation Virtual Machine	Installed memory (RAM)	8 GB
		Total disk space	2188.48 GB

**EVENTS** All events | 17 total

Filter                                     <img alt="Sort icon" data-bbox

Open the explorer → download the integration run time in your RDP



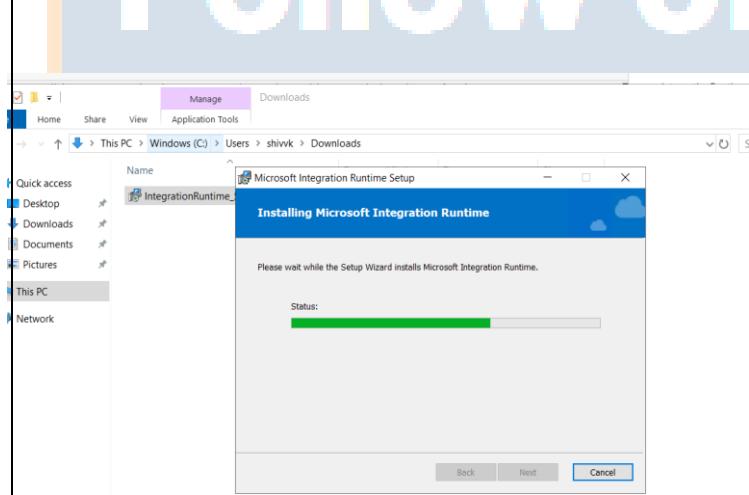
## Microsoft Integration Runtime

The Microsoft Integration Runtime is a customer managed data integration infrastructure used by Azure Data Factory and Azure Synapse Analytics to provide data integration capabilities across different network environments.

Important! Selecting a language below will dynamically change the complete page content to that language.

Select language

[Download](#)

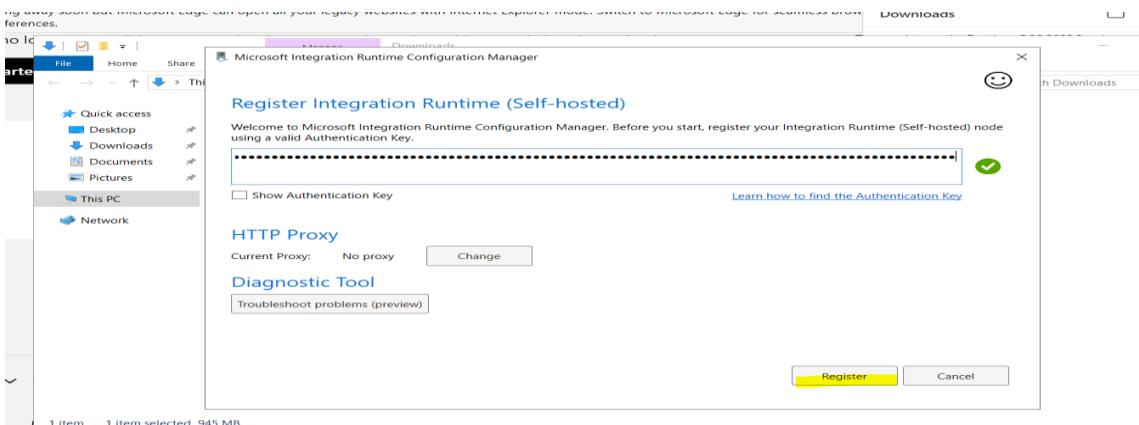


once downloaded install in RDP

once downloaded copy the key from ADF and paste it and register in RPD



Follow on linkedin  
@Shivakiran kotur



## Step 8 → mean while connect your Sql pool to your SSMS

**testpool (svkloaddata/testpool)**

Your dedicated SQL pools (formerly DW) can now be accessed from a Synapse workspace. Create a workspace here.

**Essentials**

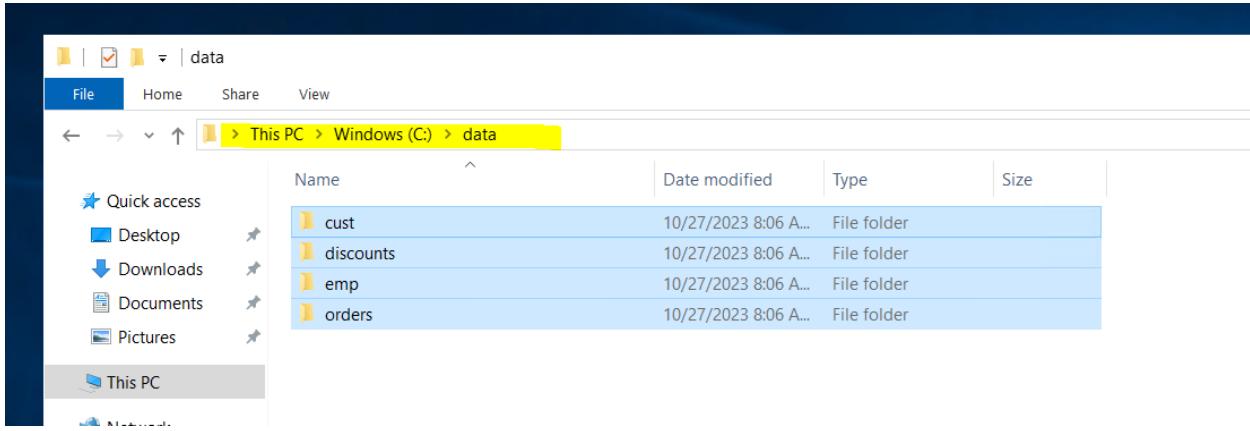
Resource group (move) : svkdev	Server name : <a href="#">svkloaddata.database.windows.net</a> <span style="border: 1px solid #ccc; padding: 2px;">Copied</span>
Status : Online	Connection strings : <a href="#">Show database connection strings</a>
Location : East US	Performance level : Gen2: DW100c
Subscription (move) : Azure for Students	Maintenance schedule : Sun 12:00 UTC (8h) / Wed 12:00 UTC (8h)
Subscription ID : c16facce-26d0-4d53-89de-f460cfda370	Geo-backup policy : Enabled
Tags (edit) : Add tags	

**Features (4)**

- Transparent data encryption**: Encrypt your database, backups, and logs. NOT CONFIGURED
- Auditing**: Track your SQL pool events and write them to an audit log in an Azure storage account. NOT CONFIGURED
- Microsoft Defender for SQL**: Detects anomalous activity for potential security threats using Advanced Threat Protection and Vulnerability Assessment. NOT CONFIGURED
- Geo-backup**: Protect your data by enabling a geo-backup once per day to a paired data center. ENABLED

## Step 9 → put the csv files in c folder of VM to create the file system data

(make sure they are in txt format as in VM there is no Excel to support)



## Step 10 → we need to disable dmcmd.exe in integration runtime folder in VM

which is c folder

C:\Program Files\Microsoft Integration Runtime\5.0\Shared

Open power shell in RDP and disable the dmcmd.exe file from integration runtime folder

Open PowerShell. Navigate to the directory where dmcmd.exe is located using the cd (Change Directory) command. For example:

powershell

cd "C:\Program Files\Microsoft Integration Runtime\5.0\Shared"

After you have navigated to the correct directory, you can run the executable as follows:

powershell

.\dmcmd.exe -DisableLocalFolderPathValidation

```
Administrator: Windows PowerShell 172.17.3.243:25
PS C:\Program Files\Microsoft Integration Runtime\5.0\Shared> .\dmcmd.exe -DisableLocalFolderPathValidation
Failed to parse the command line options (Each command line option(DisableLocalFolderPathValidation) should be prefixed by '-' or '/')! Please check your input.

Usage:
  dmcmd [ -EnableRemoteAccess "<port>" ["<thumbprint>"] -EnableRemoteAccessInContainer "<port>" ["<thumbprint>"] -DisableRemoteAccess -Key "<AuthenticationKey>" -GenerateBackupFile "<filePath>" "<password>" -ImportBackupFile "<filePath>" "<password>" -SwitchServiceAccount "<domain\user>" ["<password>"] -LogLevel <logLevel> -EventLogVerboseSetting <On/Off> -EnableLocalMachineAccess -DisableLocalMachineAccess -EnableLocalFolderPathValidation -DisableLocalFolderPathValidation -EnableExecuteSsisPackage -DisableExecuteSsisPackage -GetExecuteSsisPackage ]
```

Detail:

```
-EnableRemoteAccess "<port>" ["<thumbprint>"] Enable remote access to current node from High Availability nodes and/or Credential Manager
-EnableRemoteAccessInContainer "<port>" ["<thumbprint>"] Enable remote access to current node when the node is running in Container
-DisableRemoteAccess Disable remote access to current node
-Key "<AuthenticationKey>" Renew the Authentication Key
-GenerateBackupFile "<filePath>" "<password>" Generate a backup file for current node, including the node authentication key and data store credentials
-ImportBackupFile "<filePath>" "<password>" Restore the node from a backup file
-SwitchServiceAccount "<domain\user>" ["<password>"] Set DIAHostService to run as a new account. Use empty password ("") for system account or virtual account.
-LogLevel <logLevel> Set ETW log level (Off, Error, Verbose or All)
-EventLogVerboseSetting <On/Off> Set Event Log Verbose level (off, On)
-EnableLocalMachineAccess Enable access to localhost and private IP of the machine.
-DisableLocalMachineAccess Disable access to localhost and private IP of the machine.
-EnableLocalFolderPathValidation Enable validation against local folder paths of the machine.
-DisableLocalFolderPathValidation Disable validation against local folder paths of the machine.
-EnableExecuteSsisPackage Enable SSIS package execution on local Self-hosted Integration Runtime machine.
-DisableExecuteSsisPackage Disable SSIS package execution on local Self-hosted Integration Runtime machine.
-GetExecuteSsisPackage Get the value if ExecuteSsisPackage is enabled on Self-hosted Integration Runtime machine.
```

```
PS C:\Program Files\Microsoft Integration Runtime\5.0\Shared> .\dmcmd.exe -DisableLocalFolderPathValidation
```



## Step 11 → create the linked services in ADF for

Linked service → File system → self hosted

The screenshot shows the 'New linked service' dialog for 'File system'. It includes fields for 'Connect via integration runtime' (set to 'SelfhostedIR'), 'Host' (set to 'C:\Data'), 'User name' (empty), 'Password' (empty), and 'Annotations' (empty). At the bottom are 'Create', 'Back', 'Test connection', and 'Cancel' buttons.

in Host → give path where we have copied the file system data in VM

username → VM / RDP username

password → place the password in key vault and use it as the password

later check for test connection → connection should be established.(if dgmcmd is disabled it will get connected)

Similarly create for ADLS gen 2 using key vault →

url → <https://<adlscontainername>.dfs.core.windows.net>

copy the connection string of the container from adls--> storage → adls container→access key→connection string copy→ key vault→ secret→ create the key

create the linked service for key vault if not created to access the secret



Follow on linkedin  
@Shivakiran kotur

## New linked service

 Azure Data Lake Storage Gen2 [Learn more](#)

From Azure subscription  Enter manually

URL \*

`https://onprmdatoadlsgen.dfs.core.windows.net`

[Storage account key](#)

[Azure Key Vault](#)

AKV linked service \* ⓘ

AzureKeyVault1

Secret name \* ⓘ

adlskey

Edit

Add dynamic content [Alt+Shift+D]



Secret version

Latest version

Edit

Test connection ⓘ

To linked service  To file path

Create

Back

 Test connection

Canc



Validate all Publish all

### Linked services

Linked service defines the connection information to a data store or compute. [Learn more](#)

+ New

Filter by name Annotations : Any

Showing 1 - 3 of 3 items

Name ↑↓	Type ↑↓	Related ↑↓
AzureKeyVault1	Azure Key Vault	2
LS_ADLSGEN2_sink	Azure Data Lake Storage Gen2	0
LS_OnpremSource	File system	0

Similary create the linked service for Sql pool (linked service → snapse analytics)

Here we use the key vault secret connection using dot.net connection string



Follow on linkedin  
@Shivakiran kotur

testpool (destinserver/testpool) | Connection strings

ADO.NET JDBC ODBC PHP Go

ADO.NET (Active Directory passwordless authentication)

Microsoft.Data.SqlClient Quickstart Entity Framework Core Quickstart

Server=tcp:destinserver.database.windows.net,1433;Initial Catalog=testpool;Encrypt=True;TrustServerCertificate=False;Connection Timeout=30;Authentication='Active Directory Default';

ADO.NET (SQL authentication)

Server=tcp:destinserver.database.windows.net,1433;Initial Catalog=testpool;Persist Security Info=False;User ID=sravan;Password={your\_password};MultipleActiveResultSets=False;Encrypt=True;TrustServerCertificate=False;Connection Timeout=30;

Copy Copied

replace the password → create the key vault secret

→ Configure the firewall for sql pool

testpool (destinserver/testpool) | Open in Visual Studio

Common Tasks

Query editor (preview)

Build dashboards + reports

Model and cache data

Open in Visual Studio

Monitoring

Query activity

Alerts

Metrics

Diagnostic settings

Logs

Automation

Configure Firewall

Configure your firewall settings to ensure this machine can access the database.

Get Visual Studio

Download and install SQL Server Data Tools on Visual Studio 2015 Update 2, Visual Studio 2013 Update 4 or a later version.

Download Visual Studio

Download SQL Server Data Tools

Open in Visual Studio

Firewall settings

destinserver (SQL server)

Save Discard Add client IP

Rule name	Start IP	End IP
ClientIPAddress_2023-8...	49.204.8.9	49.204.8.9
ClientIPAddress_2023-08-28...	49.204.8.9	49.204.8.9

Virtual networks

+ Add existing virtual network + Create new virtual network

there a option to enable the azure services, have to enable it

Default Proxy Redirect

Allow Azure services and resources to access this server

Yes No

Client IP address

New linked service

Azure Synapse Analytics [Learn more](#)

SelfhostedIR

Connection string [Azure Key Vault](#)

AKV linked service \* [Edit](#)

AzureKeyVault1

Secret name \* [Edit](#)

synapsepassword

Secret version [Edit](#)

Latest version

Authentication type \* [Edit](#)

Sql Authentication or Managed Identity

Authentication reference method [Edit](#)

Inline  Credential

Create Back [Test connection](#) Cancel

use selfhosted IR which is created for runtime

## Step 12 → execute the following scripts in testpool database in SSMS

Table syntax:

```
-- Create the table 'metadata'  
CREATE TABLE metadata (  
    sourcefilename VARCHAR(50),  
    storagepath VARCHAR(50),  
    isactive INT,  
    status VARCHAR(50)  
);
```

-- Insert data into the 'metadata' table

```
INSERT INTO metadata (sourcefilename, storagepath, isactive, status)  
VALUES  
    ('cust', 'cust', 0, 'ready'),  
    ('orders', 'orders', 0, 'ready'),  
    ('emp', 'emp', 0, 'ready'),  
    ('discounts', 'discounts', 0, 'ready');
```

-- Create the 'metadata\_usp' stored procedure

```
CREATE PROCEDURE metadata_usp (@status VARCHAR(50), @sourcefilename  
VARCHAR(50)  
AS  
BEGIN  
    UPDATE metadata
```



Follow on linkedin  
@Shivakiran kotur

```
SET status = @status  
WHERE sourcefilename = @sourcefilename;  
END;
```

-- Create the 'reset\_status\_usp' stored procedure

```
CREATE PROCEDURE reset_status_usp  
AS  
BEGIN  
    UPDATE metadata  
    SET status = 'ready';  
END;
```

## Step 13 → Create the pipeline in ADF

First activity is lookup → to lookup metadata table from azure synapse

The screenshot shows the 'Settings' tab for a 'Lookup' activity in the ADF pipeline editor. The activity is named 'Lookup1'. The 'Query' field contains the following T-SQL:

```
select * from metadata where status <> 'succeeded'
```

A yellow box highlights the query text. To the left, there is a 'Set properties' dialog for the 'DS\_Synapse\_Lookup' dataset, showing the linked service 'LS\_Synapse\_destination' and table name 'dbo.metadata'. Another yellow box highlights the dataset name.

**Take lookup activity → setting → create dataset → create dataset as below**



Follow on linkedin  
@Shivakiran kotur

## Preview data

Linked service: LS\_Synapse\_destination

Object:

#	sourcefoldername	storagepath	isactive	status
1	orders	orders	0	ready
2	cust	cust	0	ready
3	discounts	discounts	0	ready
4	emp	emp	0	ready

Step → take for each activity next to lookup, take the output of lookup as input to for each loop

## Pipeline expression builder

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

```
@activity('Lookup1').output.value  
[@] value value: any[]
```

Inside for each → take copy activity → copy activity

for copy activity source is file system → create the dataset →

file system → csv file

New dataset

In pipeline activities and data flows, reference a dataset to specify the location of data within a data store. Learn more [\[?\]](#)

Select a data store

file

All Azure Database File Generic protocol NoSQL S

Azure File Storage File system

**Set properties**

Name: Ds\_Filesystem\_Sourcedataset

Linked service: LS\_OnpremSource

Connect via integration runtime: SelfhostedIR

File path: C:\Data / [Directory] / File name

First row as header:

Import schema: From connection/store, From sample file, None

> Advanced

for sink use adls gen 2 → create dataset for the same



Follow on linkedin  
@Shivakiran kotur

Set properties

Name: Ds\_Adlsgen2\_sinkdataset

Linked service: LS\_ADLSGEN2\_sink

Connect via integration runtime: SelfhostedIR

File path: global / raw / File name

First row as header: checked

Import schema: From connection/store

Advanced

Cancel

Here file path should be in sink raw, as we are going to put raw data in raw folder

Parameterize the directory → for file system dataset source

And in sink dataset also parameterize

DelimitedText Ds\_Adlsgen2\_sinkdataset

Connection Schema Parameters

Linked service: LS\_ADLSGEN2\_sink

Integration runtime: SelfhostedIR

File path: global / raw/@{dataset().dataset\_param}/@{f...

Compression type: Select...

Column delimiter: Comma (,)

Add → directory in form of  
Raw/filename/yyyy/mm/dd  
It should be created as folder.

Microsoft Azure | Data Factory | onprmdtafactory | Search factory and documentation

Pipeline expression builder

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

```
raw@{dataset().dataset_param}@{formatDateTime(utcnow(), 'yyyy')}/{formatDateTime(utcnow(), 'MM')}/{formatDateTime(utcnow(), 'dd')}
```

Clear contents

Parameters Functions

utcnow

Collapse all

Date Functions

utcnow Returns the current timestamp as a string.

Ok Cancel

General    **Source**    Sink    Mapping    Settings    User properties

Source dataset \* Ds\_FileSystem\_Sourcedataset

File path type: File path in dataset

Take output from lookup activity

```

    "count": 4,
    "value": [
      {
        "sourcefoldername": "orders",
        "storagepath": "orders",
        "isactive": 0,
        "status": "ready"
      },
      {
        "sourcefoldername": "cust",
        "storagepath": "cust",
        "isactive": 0,
        "status": "ready"
      }
    ]
  
```

Source →

sourcefoldername

General    **Sink**    Source    Mapping    Settings    User properties

Sink dataset \* Ds\_AdlsGen2\_sinkdataset

Value: @item().storagepath

Sink → storage path

Pipeline expression builder

Add dynamic content below using any combination of expressio

@item().storagepath

Following the copy activity take store proc activity

General    **Settings**    User properties

Stored procedure parameters

Import    New    Delete

Name	Type	Value
sourcefoldername	String	@item().sourcefoldername
status	String	succeeded

Use → metadata\_ups as store proc,

And import parameter,

Add sourcefoldername as dynamically.

Status hard code succeeded indicating the copy activity finished

PL\_Onprem\_Cloud > ForEach1

General Settings User properties

Stored procedure name \* [dbo].[metadata\_usp]

Stored procedure parameters

Name	Type	Value
sourcefilename	String	@item().sourcefilename
status	String	Failed

Add dynamic content [Alt+Shift+D]

Take another store procedure activity for failure

Use the same store procedure and hardcode status as failed

@Shivakiran kotur

Next take the store procedure outside the for each activity

Here we reset the sp on success

```

CREATE PROCEDURE reset_status_usp
AS
BEGIN
    UPDATE metadata
    SET status = 'ready';
END;

```

END;

The screenshot shows the Logic Apps Designer interface. At the top, there are navigation tabs: Save, Discard, Run Trigger, Designer, Code view, Parameters, Templates, Connectors, Help, Info, and Try Preview Designer. Below the tabs is a search bar labeled 'Search connectors and triggers'. Under the search bar, there are two tabs: 'Triggers' and 'Actions'. The 'Triggers' tab is selected, showing a list of available triggers. One trigger, 'When a HTTP request is received', is highlighted with a yellow background. Other triggers listed include 'When an Azure Security Center alert is manually triggered (Obsolete - see description)' and 'When an Event Grid resource event occurs'. Below the triggers, there is a section titled 'Don't see what you need?' with a link to 'Help us decide which connectors and triggers to add next with UserVoice'.

## Step → Create a logic app

Create logic app → go to resource → create blank new → search for http → select request → add new parameter → method → GET → add next step → gmail → send email → name = gmail → sign in → add the to email → subject → save → url will be generated in http → copy url

This screenshot shows a logic app workflow. It starts with a trigger 'When a HTTP request is received'. An arrow points down to an action 'Send email (V2)'. The 'Send email (V2)' action has the following settings: To: cloudfreakotechnology@gmail.com; satheeshreddy085@gmail.com; gurijalasanthoshreddy@gmail.com; vsrisubha13@gmail.com; vsrisubha13@gmail.com. Subject: pipeline failed. Body: contains a rich text editor with a message 'cust table not available'. There is also a 'Add new parameter' button.



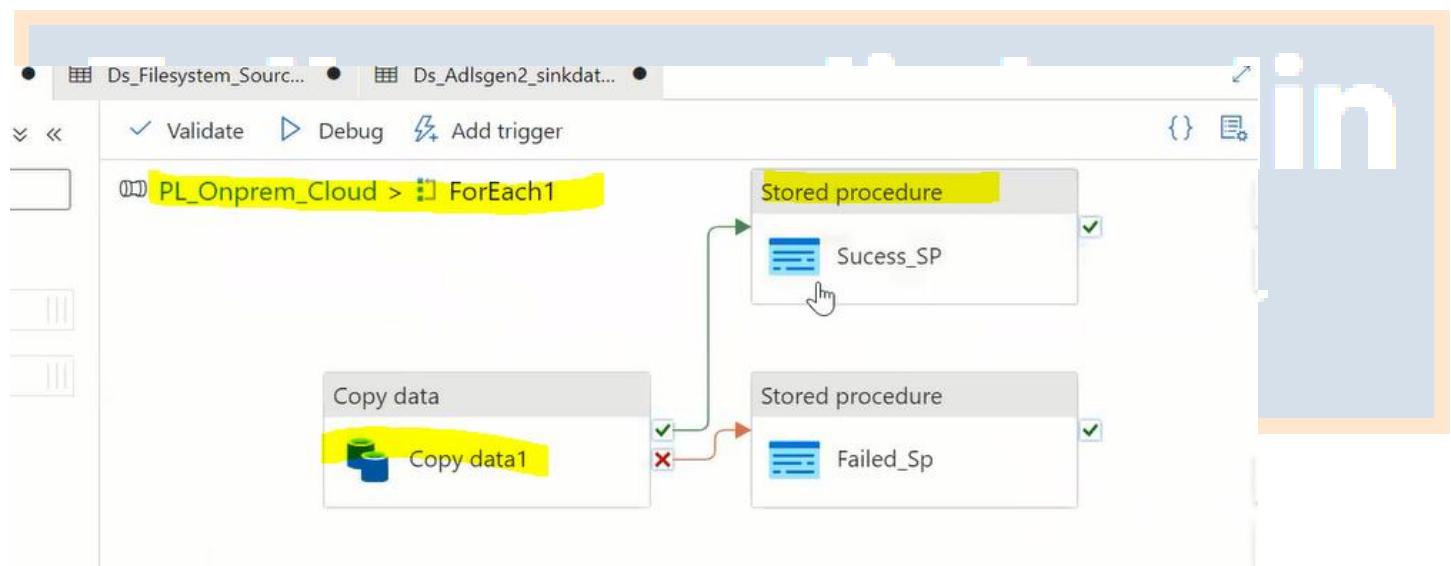
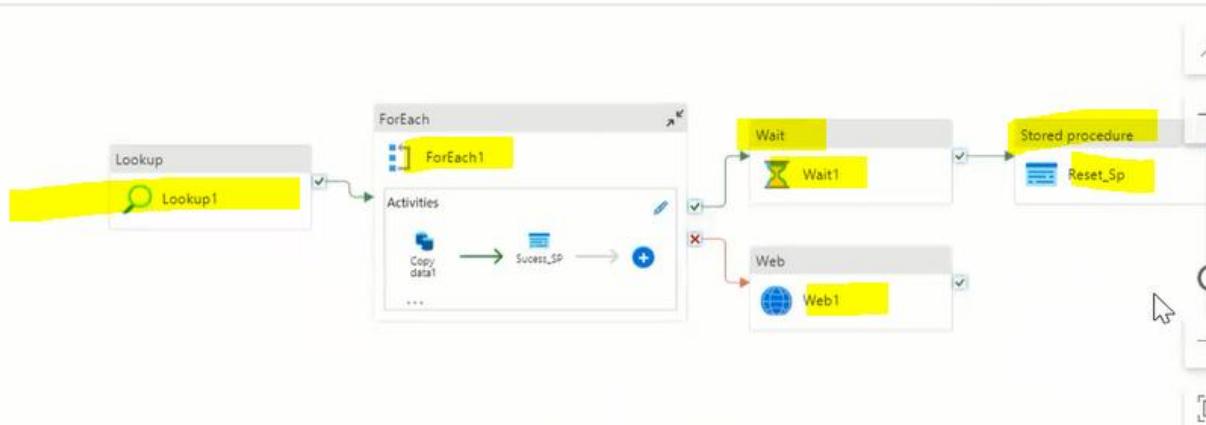
This screenshot shows a logic app workflow. It starts with a trigger 'When a HTTP request is received'. An arrow points down to an action 'HTTP GET URL'. The 'HTTP GET URL' action has the URL: https://prod-68.eastus.logic.azure.com:443/workflows/3e4ae71155c9498... and a method: GET.

## Step → go to ADF → on failed of for each take a web activity

This screenshot shows a detailed configuration for a 'Web' activity. The activity is part of a larger workflow. The 'Settings' tab is selected. Under the 'URL' section, the URL is set to https://prod-68.eastus.logic.azure.com:443/. The 'Method' is set to GET. The 'Authentication' dropdown is set to None. There are also sections for Headers and Advanced settings.

Follow on linkedin  
 @Shivakiran kotur

Small change → add the wait activity after for each loop → 30 secs



Here add the wild card and \* as there are folder in source in RDP where the files are added



Follow on linkedin  
@Shivakiran kotur

If any failure like the change in file name or metadata occurs email is triggered.

**Step 14 → perform data quality checks and move cleansed data from Raw layer to bronze layer.**

Data quality checks are performed in data bricks → Create Azure databricks service in Azure.

Data source is ADLS container → create a mount point to connect to container (since we are using only one container and inside that container we are moving data from raw folder to bronze folder only 1 mount point is enough)

For mount ADLS → required service → ADB, Azure key vault and SPN → i.e App registration (where we create a new client network and from there extract client secret value, client ID, tenant ID and create the key vault secrets for the same)

Create app registration → go to app directory → app registration → once created → go to secrets and certificate → new client secret → copy the secret value and keep as it will be encrypted once web is closed



Follow on linkedin  
@Shivakiran kotur

The screenshot shows the Microsoft Azure portal's 'Default Directory' section under 'App registrations'. On the left, a sidebar lists 'Overview', 'Preview features', 'Diagnose and solve problems', 'Manage' (with options like 'Users', 'Groups', 'External identities', etc.), and 'App registrations'. The 'App registrations' option is highlighted. The main area displays a table of registered applications:

Display name	Application (client) ID	Created on	Certificates & secret
AR armtemplates	375f5ed8-7e40-4c1d-8e79-7bcd16a...	5/7/2023	Current
AZ azuredevopsaccount	e0ec0f2f-63ec-41a1-a1ec-701d8139...	6/2/2023	Current
IN Infrdevhdfc	9d96a7dc-b8cf-4492-8b44-1b93ffe3...	5/6/2023	Current
MR mrrgarm	761c7d73-a63a-4b4a-a093-4900c36...	5/8/2023	Current

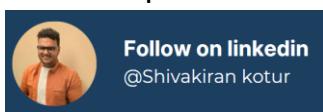
The screenshot shows the 'Certificates & secrets' blade for the 'projectrole' application. It includes tabs for 'Certificates (0)', 'Client secrets (1)', and 'Federated credentials (0)'. The 'Client secrets (1)' tab is selected, showing a single entry for a new client secret named 'test' with an expiration date of 2/26/2024. The secret value is partially visible as 'xCh8Q~36Q3mqwY63A4WVjsfsjceilK...'. A success message at the top right states 'Successfully updated application projectrole credentials'.

Step → go to your Adls storage account → I am role → create a role for storage Blob data contributor → assign that to app regist that i.e created above → review and assign

The screenshot shows the 'Add role assignment' blade for the 'Storage Blob Data Contributor' role. The 'Members' tab is selected. Under 'Selected role', 'Storage Blob Data Contributor' is chosen. Under 'Assign access to', 'User, group, or service principal' is selected. The 'Members' section shows a list of users, groups, or service principals, with 'projectrole' being selected. The 'Selected members' list contains 'projectrole'. At the bottom, there are 'Review + assign' and 'Next' buttons.

**Step 15 → go to ADB → create notebook →**

Create dbutilities widgets (once executes the widgets box appears at top through which we can filter the particular data and run the entire notebook)



```
dbutils.widgets.text(processeddate,"")
dbutils.widgets.text(foldername,"")
```

step → create the ADLS mount point

```
configs = {"fs.azure.account.auth.type": "OAuth",
    "fs.azure.account.oauth.provider.type": "org.apache.hadoop.fs.azurebfs.oauth2.ClientCredsTokenProvider",
    "fs.azure.account.oauth2.client.id": dbutils.secrets.get(scope="adlsgenkey",key="appid"),
    "fs.azure.account.oauth2.client.secret": dbutils.secrets.get(scope="adlsgenkey",key="apppwd"),
    "fs.azure.account.oauth2.client.endpoint": "https://login.microsoftonline.com/f5ea40f2-c7b8-4658-8d25-0aac8535e48c/oauth2/v2.0/token",
    "fs.azure.createRemoteFileSystemDuringInitialization": "true"}
```

```
dbutils.fs.mount(
source = "abfss://global@adlsgenstorageaccountny.dfs.core.windows.net/",
mount_point = "/mnt/global",
extra_configs = configs)
```

create the scope by adding attend of url as → #secrets/createScope

HomePage / Create Secret Scope

Create Secret Scope | Cancel | Create

A store for secrets that is identified by a name and backed by a specific store type. [Learn more](#)

Scope Name

Manage Principal

Azure Key Vault

DNS Name

Resource ID

Give the scope name → adlsgenkey

DNS name and resource ID → from azure key vault

onpremdtrvalt | Properties

Key vault

Search | Save | Discard changes | Refresh

Name: onpremdtrvalt

Sku (Pricing tier): Standard

Location: eastus

Vault URL: https://onpremdtrvalt.vault.azure.net/

Resource ID: /subscriptions/a67f73e7-b505-4ce3-8fbf-9...

Subscription ID: a67f73e7-b505-4ce3-8fbf-9...

Home > Default Directory | App registrations > projectrole

Search | Delete | Endpoints | Preview features

**Overview**

Got a second? We would love your feedback on Microsoft identity platform (previously)

**Essentials**

Display name: projectrole

Application (client) ID: 2e7fb133-b159-4282-ac06-d88b3b152540

Object ID: b79a03df-1742-45c8-8548-001b6c8a521b

Directory (tenant) ID: f5ea40f2-c7b8-4658-8d25-0aac8535e48c

Supported account types: My organization only

in mount point →

client.secret": → i.e apppwd the secret value copied from app registration

client id → application id

tenant id copy and paste in

fs.azure.account.oauth2.client.endpoint":  
"https://login.microsoftonline.com/<tenantid>/oauth2/v2.0/token"

source =

"abfss://<containername>@<storageaccountname>.dfs.core.windows.net/",

once execute comment all (select all in text and Ctrl+/ shortcut to comment all together)

**Step 16 → Aim is to move data from raw to bronze**

1 src\_path="/mnt/global/raw/"

Cmd 3

```
1 dbutils.widgets.text('processeddate','') # 2023/08/30
2 dbutils.widgets.text('foldername','') # cust
```

Command took 0.07 seconds -- by sravanazure72710@gmail.com at 8/30/2023, 6:45:22

Cmd 4

```
1 foldername=dbutils.widgets.get('foldername','')
2 pdate=dbutils.widgets.get('processeddate','')
```

Cmd 5

```
1 src_final_path =src_path+pdate+"/"+pdate
```

src\_path="/mnt/global/raw/"  
dest\_path="/mnt/global/bronze/"

dbutils.widgets.text(processeddate,"")  
dbutils.widgets.text(foldername,"")

foldername = dbutils.widgets.get('foldername')  
pdate = dbutils.widgets.get('processeddate')

```
print(foldername)
print(cdate)
src_final_path=src_path+foldername+"/"+pdate
print(src_final_path)
dest_final_path=dest_path+foldername+"/"+pdate
```



@Shivakiran kotur

```

print(dest_final_path)
#following is the code for cleaning the data
try:
    # Read data from source path
    df = spark.read.format("csv").option("header", True).load(src_final_path)

    # Count the number of rows in the source DataFrame
    src_count = df.count()
    print("Source count:", src_count)

    # Remove duplicates
    df1 = df.dropDuplicates()

    # Count the number of rows in the destination DataFrame
    dest_count = df1.count()
    print("Destination count:", dest_count)

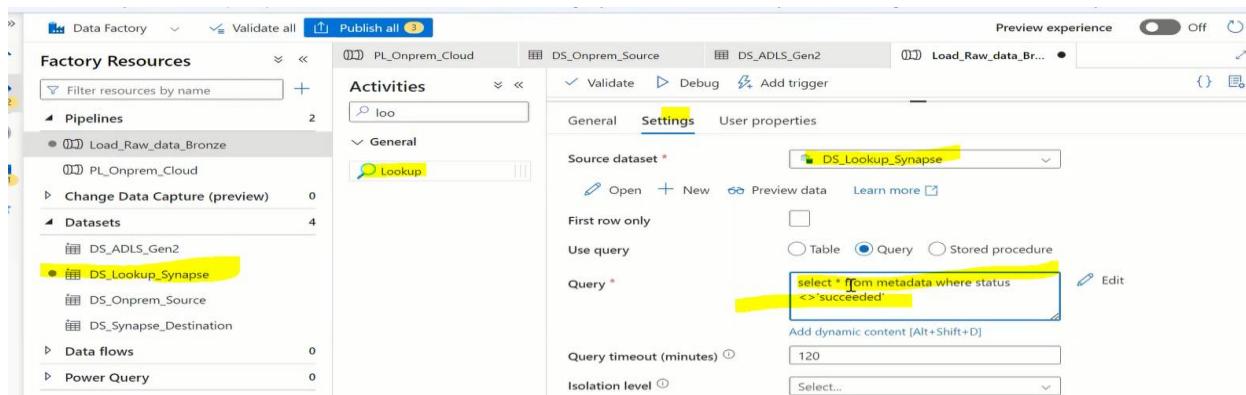
    # Write the cleaned data to the destination path
    df1.write.mode("overwrite").format("csv").option("header", True).save(dest_final_path)

    # Exit the notebook with success message and counts
    print("Success: Source count = " + str(src_count) + ", Destination count = " + str(dest_count))
except Exception as e:
    # Handle exceptions and exit with an error message
    dbutils.notebook.exit("Error: " + str(e))

```

Step 17 → go to ADF to create the pipeline for the movement of data from raw to bronze

Take the lookup activity → create the source dataset for synapse →



The screenshot shows the Azure Data Factory pipeline editor. On the left, the 'Factory Resources' sidebar lists Pipelines, Datasets, and Power Query. In the main area, a pipeline named 'Load\_Raw\_data\_Bronze' is selected. An 'Activities' list shows a 'Lookup' activity. The 'Settings' tab is active for the 'Lookup' activity. Under 'Source dataset', 'DS\_Lookup\_Synapse' is selected. The 'Query' field contains the following T-SQL:

```

select * from metadata where status >> 'succeeded'

```

The 'Isolation level' dropdown is set to 'Select...'.

Next take → foreach activity → take output of lookup → go inside for each → take notebook activity → create linked service → dataset for notebook

The screenshot shows the Azure Data Factory interface with two main windows open:

- New linked service (Left Window):**
  - Connect via integration runtime:** Self-IR (highlighted)
  - Account selection method:** From Azure subscription (highlighted)
  - Azure subscription:** Dotnet\_Dev\_ProjectA (highlighted)
  - Databricks workspace:** filesystemdata (highlighted)
  - Select cluster:** Existing instance pool (highlighted)
- New linked service (Right Window):**
  - AzureKeyVault1** (selected in dropdown)
  - Secret name:** databricksaccesstoken (highlighted)
  - Secret version:** Latest version
  - Choose from existing clusters:** Sravan N's Cluster (highlighted)
  - Annotations:** New
  - Parameters:**
  - Advanced:**

At the bottom, the status bar shows:

- \_Onprem\_Source
- DS\_ADLS\_Gen2
- ✓ Validate
- ▷ Debug
- ⚡ Add trigger

A message box indicates: "AzureDatabricks1 (Linked service) will be created when publishing."

The main canvas shows a flow starting with a **Lookup** activity (Lookup1) connected to a **ForEach** activity (ForEach1). The **ForEach1** activity contains a **Activities** section with a **Notebook1** activity (highlighted).

The **Azure Databricks** tab is selected in the settings pane, showing:

- Databricks linked service \***: AzureDatabricks1 (highlighted)
- Test connection**
- Edit**
- New**

Under setting → we have to pass 2 base parameters (as we have take processeddate and foldername as widgets in notebook)

The screenshot shows the Azure Data Factory pipeline interface. At the top, there are three tabs: DS\_Onprem\_Source, DS\_ADLG2, and Load\_Raw\_data\_Br... (highlighted). Below these are buttons for Validate, Debug, and Add trigger. The main area displays a pipeline flow starting with a DS\_ADLG2 activity, followed by a Load\_Raw\_data\_Bronze activity, and a ForEach1 activity. Inside the ForEach1 loop, there is a Notebook activity named Notebook1, which then branches to Success\_SP and Failure\_SP activities. The Load\_Raw\_data\_Bronze activity has a red error icon. The pipeline is currently in 'Running' status.

**Load\_Raw\_data\_Bronze > ForEach1**

**Notebook path \***

**Base parameters**

Name	Value
foldername	@item().storagepath
processeddate	@formatDateTime(utcnow(), 'yyyy/MM/dd')

`@formatDateTime(utcnow(), 'yyyy/MM/dd')`

Simialry to last pipeline → following the notebook activity take 2 store proc , 1 for success and other for failure add the parameter

Outside the for each take reset store proc and web activity for failure where we call the logic app

If any errors in notebook, pipeline fails and in output of the activity we get the notebook url, click on that and it directly goes to ADB, and one with highlighted one is having error fix it and run the pipeline again.



Follow on linkedin  
@Shivakiran kotur

The screenshot shows the Azure Data Bricks Pipeline status page. At the top, there are buttons for Validate, Debug, and Add trigger. Below that is an Output section with a 'Copy to clipboard' button. The output text is as follows:

```
{
  "runPageUrl": "https://adb-254324971627258.18.azuredatabricks.net/?o=254324971627258#http://478715158340812/run/2397",
  "runError": "TypeError: can only concatenate str (not \"int\") to str",
  "effectiveIntegrationRuntime": "Self-IR",
  "executionDuration": 16,
  "durationInQueue": {
    "type": "procedure",
    "Run start": "2023-08-30T19:50:21Z",
    "Run end": "2023-08-30T19:50:21Z"
  }
}
```

Below the output is a table of runs:

Notebook	Run start	C
Notebook1	2023-08-30T19:50:21Z	3
Failure_Sp	2023-08-30T19:50:09Z	7
Notebook1	2023-08-30T19:49:12Z	2

cannot edit here directly → so go to the main development branch and edit the code.

The screenshot shows the Azure Data Bricks Notebook output. The notebook name is ADF\_onprmdatfctry\_Load\_Raw\_data\_Bronze\_Notebook1\_4c78dbad-204ec5b0810035 run. The output shows two failed commands:

```
df1.write.format("csv").option("header",True).save(dest_final_path)
Command took 1.10 seconds
```

```
dbutils.notebook.exit("source count:"+src_count+" destination count:"+dest_count)
TypeError: can only concatenate str (not "int") to str
Command took 0.05 seconds
```

## Step 18 → to do transformation and move data from bronze layer to Silver layer

Below is the scripts to perform transformation in notebook from Bronze to silver, here only join transformation for cust table is performed

```
# Set source and destination paths
src_path = "/mnt/global/bronze/"
dest_path = "/mnt/global/silver/"

# Input widgets for folder name and processing date
dbutils.widgets.text('foldername', '')
dbutils.widgets.text('pdate', '')

try:
    # Get user input for folder name and processing date
```



```

foldername = dbutils.widgets.get('foldername')
pdate = dbutils.widgets.get('pdate')

print("Folder Name:", foldername)
print("Processing Date:", pdate)

# Create source and destination paths based on user input
src_final_path = src_path + foldername + "/" + pdate
print("Source Path:", src_final_path)

# Destination path for writing processed data
dest_final_path = dest_path + 'dim' + foldername
print("Destination Path:", dest_final_path)

# Load data from the source path
df = spark.read.format("csv").option("header", True).load(src_final_path)
src_count = df.count()
print("Source Count:", src_count)

# Display the DataFrame
df.show()

# Create a sample DataFrame (df11) - replace this with your actual data
df11 = spark.createDataFrame([(2, '78654345'), (3, '67865467')], ['cid', 'cphone'])
df11.show()

# Join dataframes if foldername is 'cust', otherwise use df as is
from pyspark.sql.functions import col

if foldername == 'cust':
    df1 = df.alias('a').join(df11.alias('b'), col('a.cid') == col('b.cid'), "inner").select('a.*',
'b.cphone')
    df1.show()
else:
    df1 = df

# Count rows in the destination DataFrame
dest_count = df1.count()

```



Follow on linkedin  
@Shivakiran kotur

```

# Write processed data to the destination path
df1.coalesce(1).write.mode("overwrite").format("csv").option("header",
True).save(dest_final_path)

print("Processing completed successfully.")
print("Source Count:", src_count)
print("Destination Count:", dest_count)
dbutils.notebook.exit("Processing completed successfully.")

except Exception as e:
    print("Error:", str(e))
    dbutils.notebook.exit("Error: " + str(e))

```

Create a pipeline similar to above raw to Bronze, change the notebook and provide the base parameters properly

### Step 19 → move data from Silver layer to Sql DW

```

print("Source Count:", src_count)
print("Destination Count:", dest_count)

# Load SQL data into the data warehouse
dbutils.widgets.text('foldername', '')

foldername = dbutils.widgets.get('foldername')
print("Folder Name:", foldername)

# Set source and destination paths for SQL data
src_path = "/mnt/global/silver/" + 'dim' + foldername
dest_path = "dim" + foldername
print("Source Path:", src_path)
print("Destination Path:", dest_path)

# Read data from the source path
df = spark.read.format("csv").option("header", True).load(src_path)
src_count = df.count()
print("Source Count:", src_count)

# Set Azure Storage account key

```



Follow on linkedin  
@Shivakiran kotur

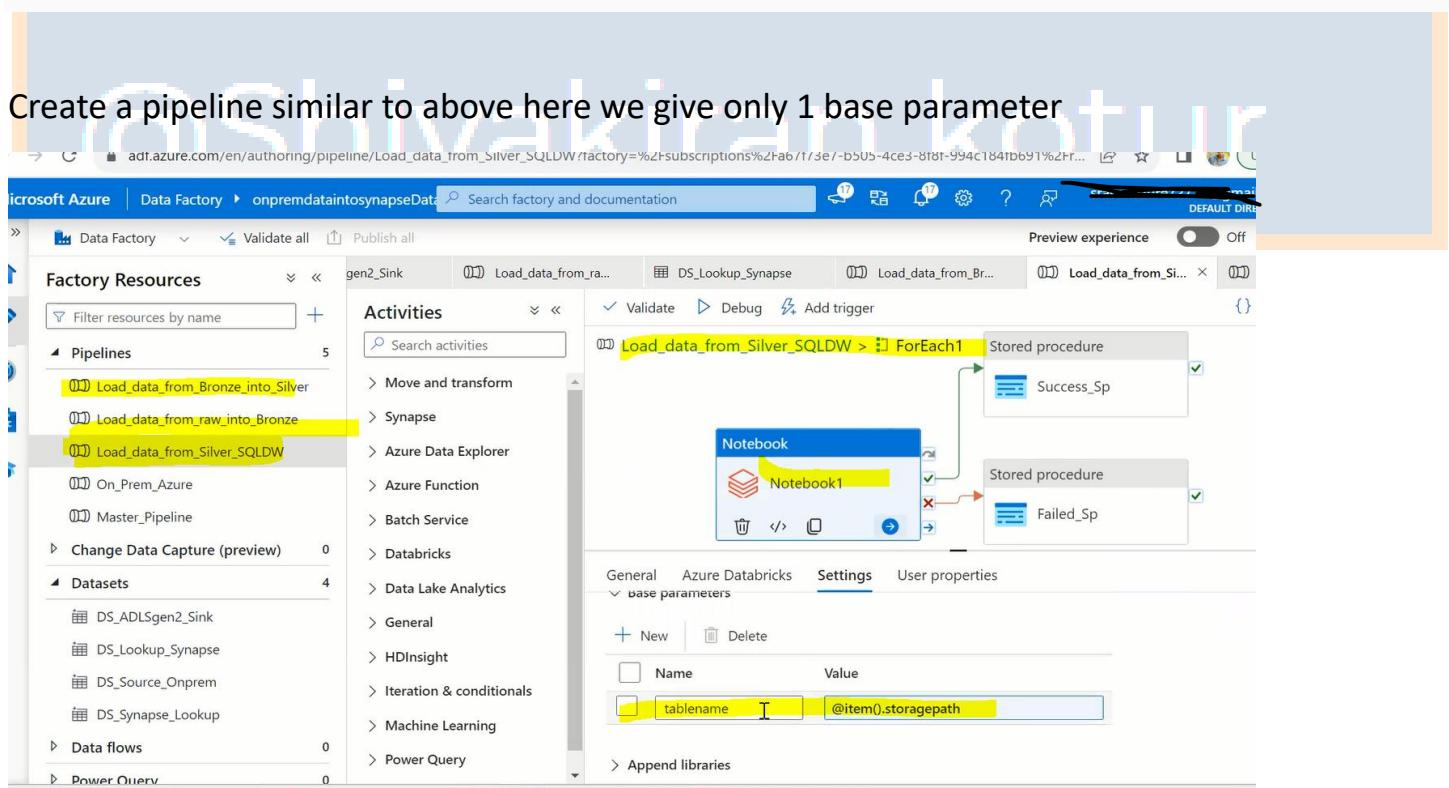
```

spark.conf.set("fs.azure.account.key.onpremdatasynasegen.dfs.core.windows.net",
"o82RdY56QpidiJOBzA0+c0xBYomGajKVXZ8oZKRr+TtVSjYOTI5+i6lVTmOFL5E73Ha5wJHe7aQ1+
ASstdFwNA==")

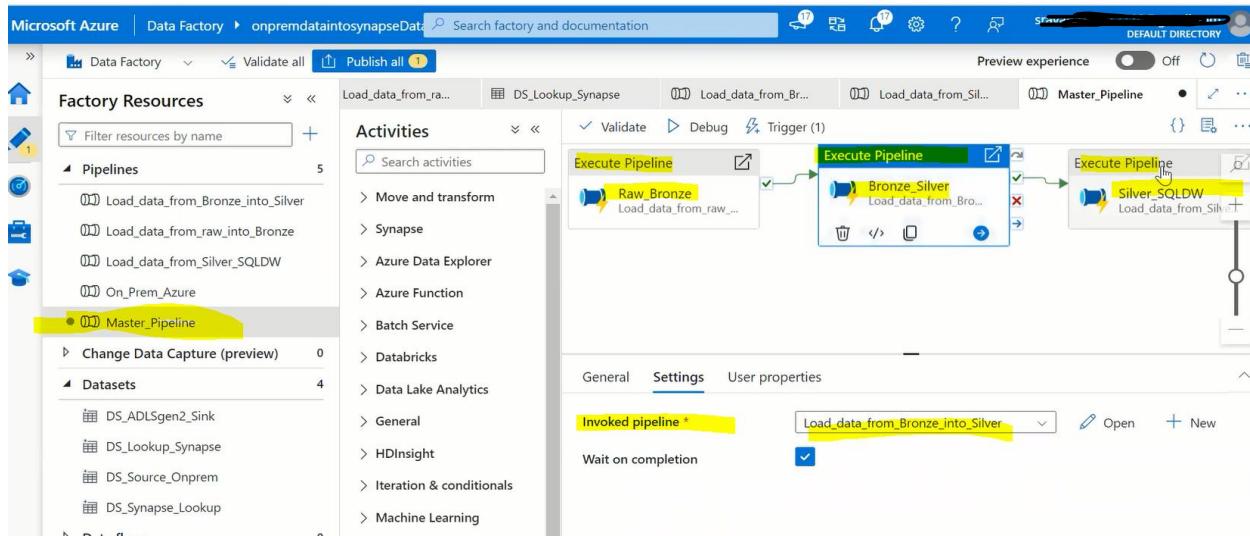
# Write data to SQL Data Warehouse (using JDBC connection from key vault)
df.write \
    .mode("overwrite") \
    .format("com.databricks.spark.sqldw") \
    .option("url", dbutils.secrets.get(scope="adlsngenkey", key="sqljdbcpwd")) \
    .option("dbtable", dest_path) \
    .option("tempDir",
"abfss://global@onpremdatasynasegen.dfs.core.windows.net/tmp/synapse") \
    .option("forwardSparkAzureStorageCredentials", "true") \
    .save()

# Display source count
print("Source Count:", src_count)
dbutils.notebook.exit("Source Count: " + str(src_count) + ", Destination Count: " +
str(dest_count))

```



## Step 20 → Create a master pipeline to execute all the pipeline using execute pipeline activity



Follow on linkedin  
@Shivakiran kotur



Follow on linkedin  
@Shivakiran kotur