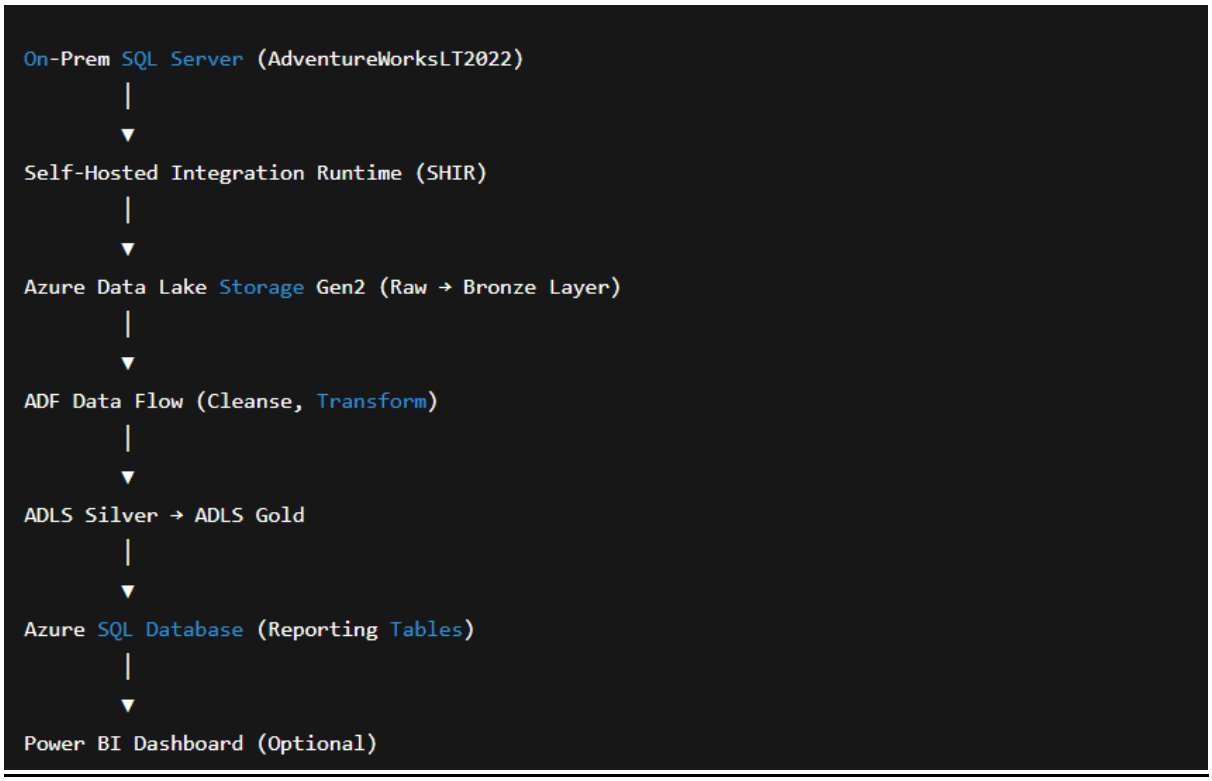
Architecture Overview

Complete Azure Data Factory (ADF) project using your on-prim AdventureWorksLT2022 **database and a** Self-Hosted Integration Runtime (SHIR) to move data into Azure Data Lake Storage Gen2 (ADLS) **and** Azure SQL Database**.**

1. Overall Architecture Overview



2. Required Azure Components

Component	Purpose
Azure Data Factory	Orchestration engine
Self-Hosted IR	Connect on-prem SQL Server
Azure Data Lake Storage Gen2	Staging and transformation zone
Azure SQL Database	Final curated data store
(Optional) Power BI	Reporting

3. Create Linked Services

Linked Service	Connection Type	Notes
LS_SQL_AdventureWorksLT	On-Prem SQL Server	Use Self-hosted IR
LS_ADLS_Raw	Azure Data Lake Storage Gen2	Store raw data
LS_ADLS_Silver	Azure Data Lake Storage Gen2	Store cleansed data
LS_AzureSQL_Gold	Azure SQL Database	For reporting tables

4. Create Datasets

Dataset Name	Source/Sink	Example Path/Table
DS_SQL_Customer	SQL Server	[SalesLT].[Customer]
DS_SQL_Product	SQL Server	[SalesLT].[Product]
DS_ADLS_Raw_Customer	ADLS	/raw/customer/{date}/customer.csv
DS_ADLS_Silver_Sales	ADLS	/silver/sales/{date}/sales.csv
DS_AzureSQL_CustomerSalesSummary	Azure SQL DB	dbo.CustomerSalesSummary

5. Create Metadata Table (In SQL Server)

ADF_Metadata_Config
.sql

6. Create Pipelines

1. PL_Master_AdventureWorksETL

   Main orchestrator pipeline.

   Activities:

   Lookup: Read table list from ADF_Metadata_Config

   ForEach: Loop through each table

   Inside ForEach → Execute PL_Bronze_Load

   Execute Pipeline: PL_Silver_Transform

   Execute Pipeline: PL_Gold_Load

   Web Activity: Notify Success (Teams or Email)

2. PL_Bronze_Load

   Extract data from SQL → ADLS Raw

   Activities:

   A. Copy Activity:

   - Source: SQL Server Table
   - Sink: ADLS /raw/{tablename}/{date}/data.csv

   B. **Stored Procedure:** Insert audit record into `ADF_Audit_Log`

ADF_Audit_Log.sql

3. PL_Silver_Transform
   Clean & standardize data using **Mapping Data Flow**.

   Example Transformations:
   - Join SalesOrderHeader + SalesOrderDetail
   - Derive TotalAmount = UnitPrice * Quantity
   - Remove null customers
   - Sink to /silver/sales/

4. PL_Gold_Load
   Aggregate & load curated data to Azure SQL DB.

   Mapping Data Flow:
   - Read /silver/sales/
   - Group By: CustomerID
   - Derive: TotalSales, AverageOrderValue
   - Sink: Azure SQL DB table dbo.CustomerSalesSummary

7. Add Triggers

Trigger Type	Purpose
Schedule Trigger	Daily at 1:00 AM
Event Trigger (optional)	On new file arrival in /raw/

8. Add Parameters and Variables

Name	Type	Used In	Example
TableName	String	All pipelines	"Customer"
DestinationPath	String	Copy Activity Sink	/raw/customer/{date}/
RunDate	String	All pipelines	@utcNow()

9. Monitoring and Logging

- Use ADF monitor tab for run history.
- Store audit logs in SQL via Stored Procedure.
- Send notifications using Web Activity (e.g., Teams or Logic App).

Optional Enhancements

- Row Count Validation Activity – Compare SQL vs ADLS row counts
- Delete Activity – Clean ADLS before each run
- If Condition – Skip table if no data returned
- Wait/Until – Wait for dependent files
- Parameterized Data Flows – Reuse transformations dynamically

Result

- We will have 3 layers (Raw → Silver → Gold)
- End-to-end orchestration using ADF
- Metadata-driven dynamic pipeline
- All ADF components covered
- Logs and notifications automated
- Ready for Power BI dashboard creation