# General Setup

**Install integration Runtime:**

1. Go to Integration Runtime and Click new
2. Give the name, create/ next
3. Go for manual setup and copy the keys
4. Install the integration runtime and run it.
5. Paste the key and then wait for it to get connected.
6. Once done click on launch configuration manager.
7. Refresh the page in Azure

**Connect Linked Service – File System:**

1. Select File System
2. Give a name and select Self Hosted Integration runtime as “Connect via Integration Runtime”
3. In corporate, the local drive is also hosted somewhere like: [\\servername\\sharedfolder\\[folder](file:///\\servername\\sharedfolder\\%5bfolder)]
4. Provide host, username and password
5. Test connection and use this link commands to resolve access not denied issue. Stack Overflow: <https://stackoverflow.com/questions/76402958/azure-data-factory-linked-service-to-c-drive>
6. cd "C:\Program Files\Microsoft Integration Runtime\5.0\Shared\"
7. .\dmgcmd.exe -DisableLocalFolderPathValidation

**Connect Linked Service - ADLS Gen2:**

1. Create Storage Account
2. Choose primary service as Azure Blob Storage or ADLS Gen2 and redundancy as Locally Redundant Storage (LRS)
3. Hierarchical namespace, complemented by Data Lake Storage Gen2 endpoint, enables file and directory semantics, accelerates big data analytics workloads, and enables access control lists (ACLs) Enable Hierarchical namespace.
4. Review + Create

# On-Prem to ADLS Gen2

**Create Data Pipeline:**

1. Copy Data, give activity name and pipeline name
2. At source select dataset > new dataset > select format > select the linked service.
3. Give name, file path all necessary details. Check using Preview Data
4. At sink, select Azure Data Lake Storage Gen2, select type as Parquet (for better data handling)
5. Give name, linked service, file path (mandatory) and file name (optional)
6. Validate and Debug
7. To dynamically ingest files, use parameter. And in file path use @dataset.p\_filename

# GitHub to ADLS Gen 2 via API

**Create Data Pipeline:**

1. We can use Web api call.
2. We can use Copy activity. Create a Linked Service for that at 1st place
3. Select new dataset in source and select HTTP, use anonymous. Test and create the connection.
4. Now create a source and sink dataset in Copy activity. In Mapping, import schemas. Connect 2 activity by either success, failure, completion or skip.
5. Validate and Debug

# Azure SQL to ADLS Gen 2

**Create Azure SQL Database:**

1. Select which database you need to create – SQL Database, SQL Managed Database, SQL Virtual Machine
2. Select SQL Database. Before creating database, create a server.
3. While creating server, use both SQL and Microsoft Entra ID
4. Set Microsoft Entra Admin, Select the account.
5. Set Username and Password
6. Now come back to Create SQL Database Page, there, provide a name for the database.
7. Select Workload environment – Development or Production
8. Configure Database. Select Service Tier – General Purpose. Compute Tier – Serverless or Provisioned. Compute Hardware, select max vCores, min vCores, Auto Pause Delay, Data Max Size(GB)
9. Backup Storage Redundancy – LRS
10. Go to Networking. Select Connectivity Method, Firewall Rules, Connection Policy.
11. Review and Create

**Create data pipeline:**

1. Create Linked Service 1st. Use Azure SQL. Select Server name, database name, authentication type – SQL authentication (provider username, password). Test connection and create.
2. Now select Copy data Activity: Select Azure SQL.
3. In source, give name, linked service. Click on create.
4. Use Query – select query

Incremental data loading

1. Create directory – monitor – add two new directory : emptyjson and lastload
2. Create empty.json and upload in emptyjson directory
3. In data factory, create a variable and set variable as @utcnow(), get the timestamp
4. Upload empty.json in lastload directory and rename as lastload.json. edit that.
5. Now create lookup activity. Create dataset and set the path.
6. Now modify the source query as

**SELECT \* FROM [dbo].[FactBookings]**

**WHERE booking\_date > '@{activity('Lastload').output.firstRow}'\**

Another Lookup activity and rename it as latestload. Select dataset as Azure SQL. Use query as **SELECT MAX(booking\_date) AS latestload**

**FROM [dbo].[FactBookings];**

Modify the Copy Data Source query as well.

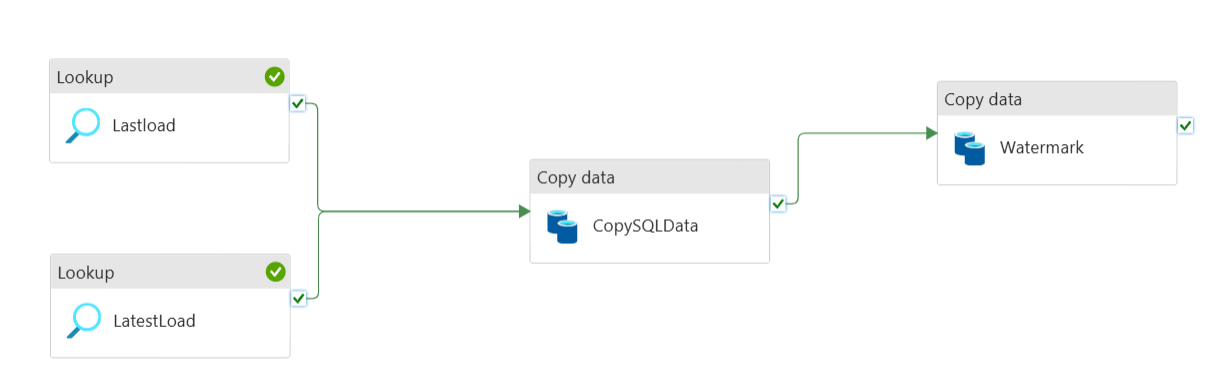
**SELECT \* FROM [dbo].[FactBookings]**

**WHERE booking\_date > '@{activity('Lastload').output.firstRow.lastload}'**

**AND booking\_date <= '@{activity('LatestLoad').output.firstRow.latestload}';**

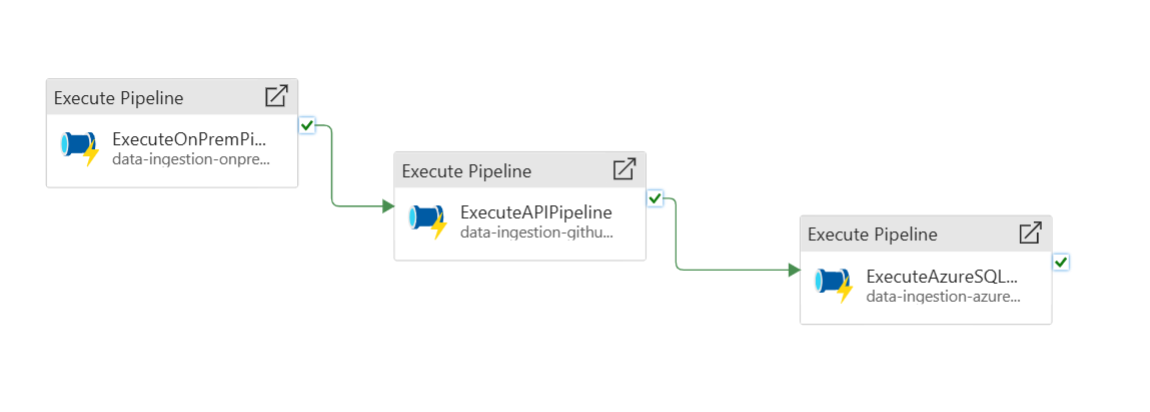
1. Now Tricky part: another copy activity. Give latestload. Basically for this Watermark is doing is taking the value from azure sql latest data and copying to latestloadjson.

[Lastload Montinor lastload json] -> (Latestload Azure SQL query) -> [Lastestload after copy -> Monitor lastload json]



**Create Orchestration – Aggregate all pipeline:**

1. Select Execute Pipeline Activity.
2. Provide name, parameter, invoked pipeline. Connect all.
3. Validate and Debug



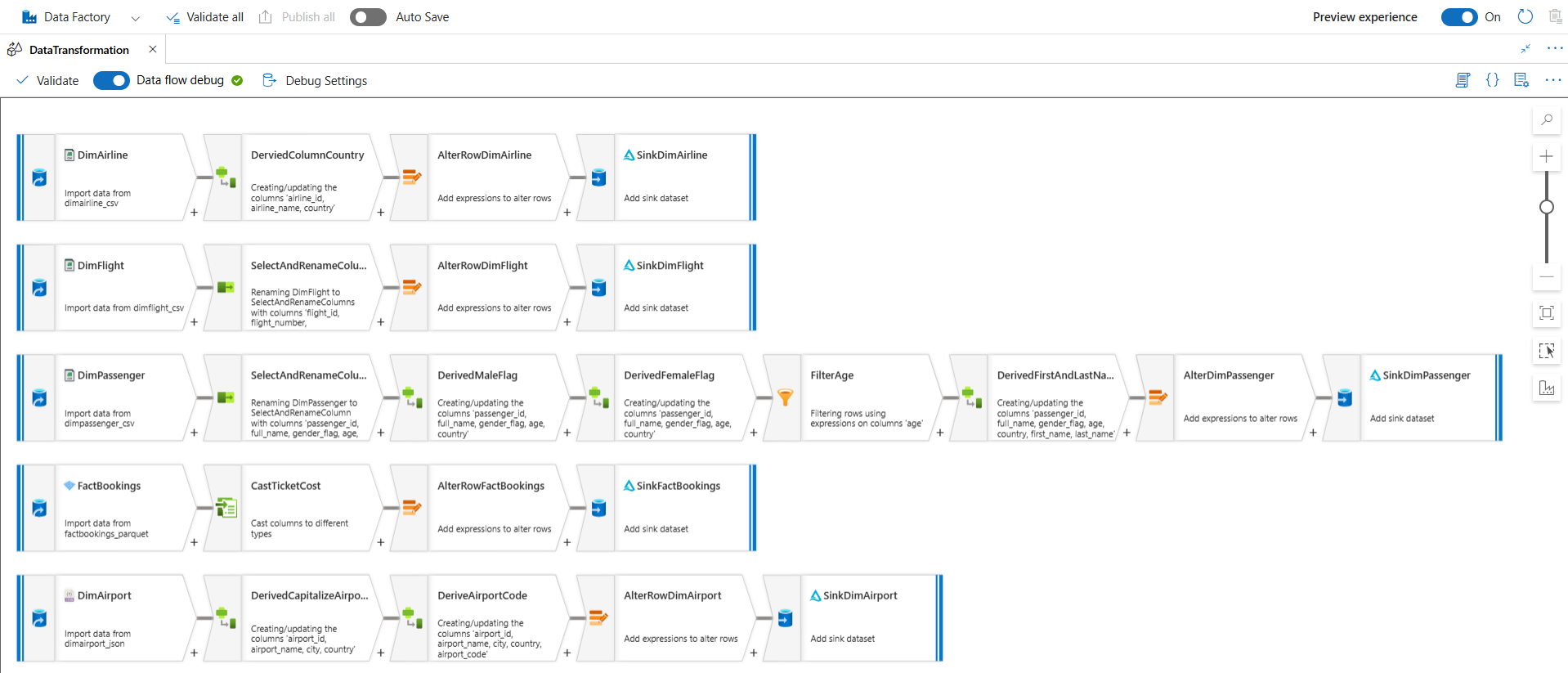
# Data Flow

## Data Transformation

1. Create a new Data flow and rename it.
2. Source Type – Dataset and Dateset – Create new dataset from ADLS Gen 2
3. Go to Projection tab and select import projection. (mandatory step for transformation)
4. Turn on data flow debug for Data Preview
5. Click on + and add transformation as per business need.

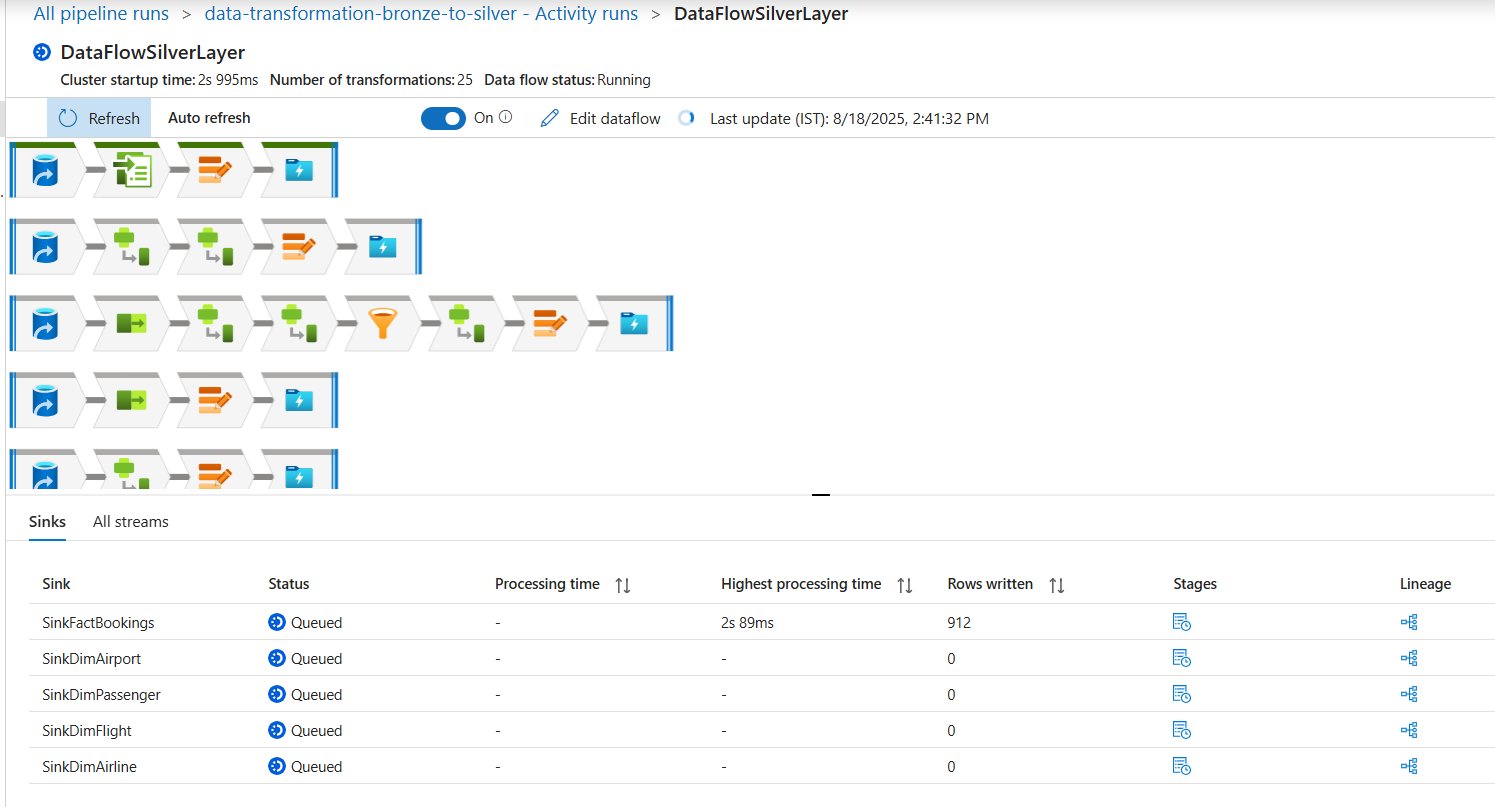
## Data Sink

1. Create activity Alter Row
2. Use Alter Row condition as Upsert (update + insert) if. Condition can be 1=1
3. Select activity Sink. Sink type – Inline. Inline dataset type – Delta, Allow schema drift
4. In settings, provide folder path, Update method as Allow Upsert.
5. Key Coloums as primary key.

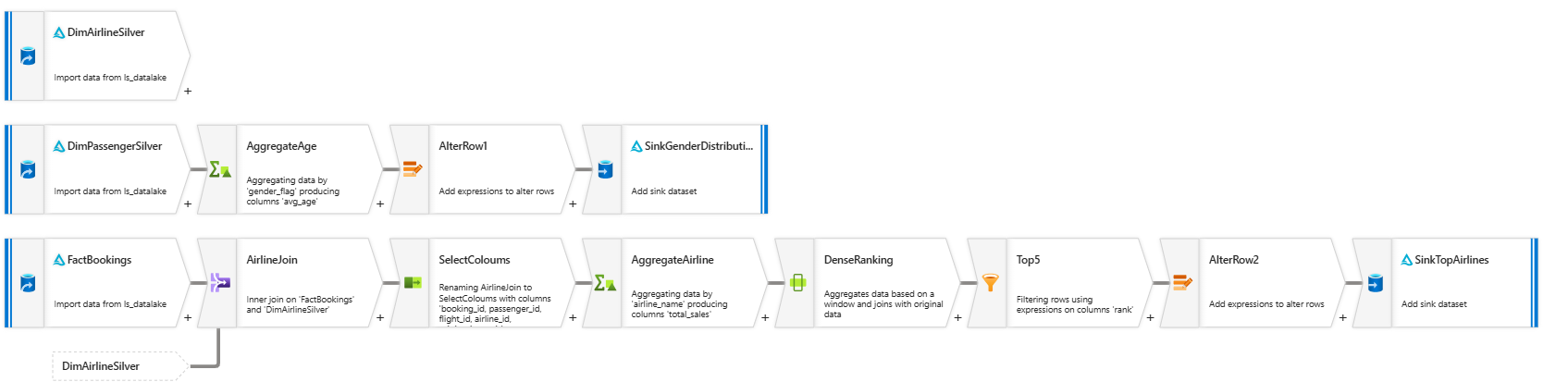


Now, its time to create a pipeline.

1. Simply create a pipeline and in settings select the data transformation pipeline.
2. Validate and Debug



1. In Gold Layer data, use Delta Lake in Source Settings and Inline dataset type as Delta and Linked Service.
2. Provide Source Option – Path.
3. Import Projection (mandatory step)
4. Select Join activity and provide all the necessary fields like left and right stream, join type, join condition etc.
5. After Join, Select coloums, aggregate, window function, Dense Rank etc as per business need.



That’s all. Connect GitHub or Azure DevOps to save your code.