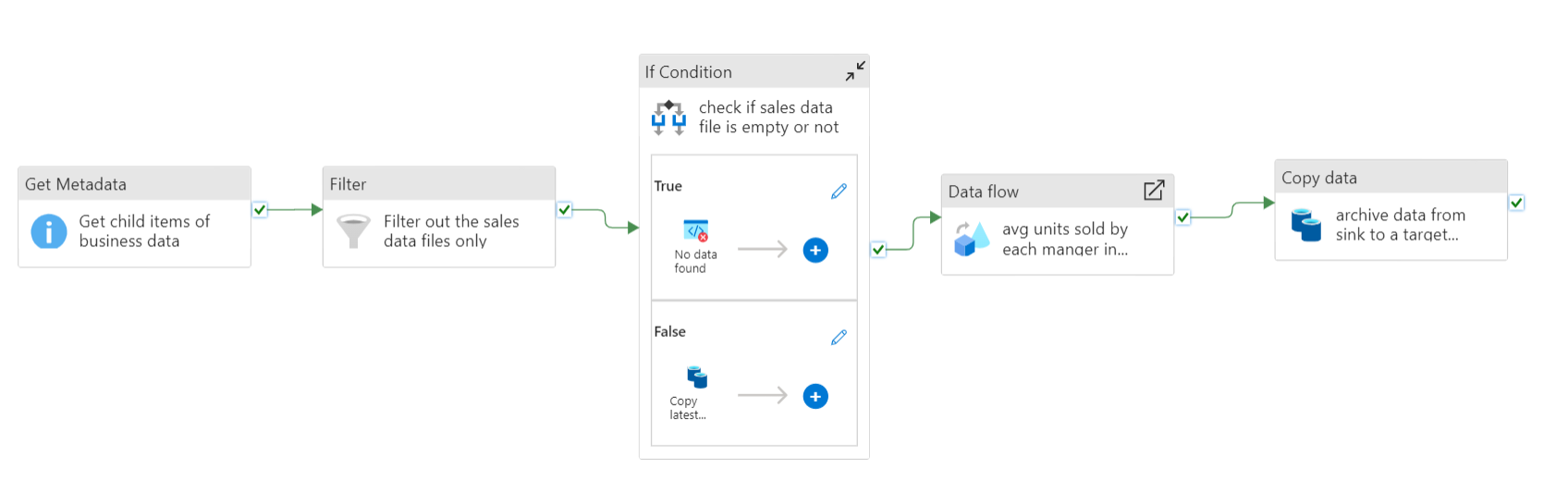
# PROJECT 4 – Create an Azure data factory Pipeline with data flow activitiy

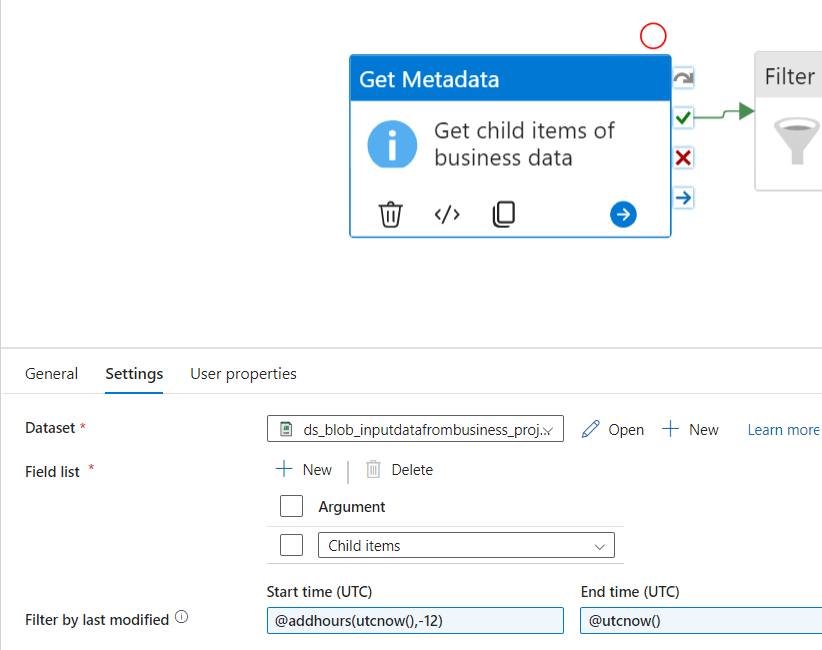
## **Scenario/Problem:** To obtain average no of units sold by each manager in central region using sales data.

## The input sales data will be uploaded at source end along with employees data by the vendor/business team in every 12 hrs. The sales data contains all region wise data (central, east and west). Group and sort the final result manager wise and make it available at target side making use of latest data uploaded at source end. Also archive the data to prevent loss of transformed data.

## **Solution:** Create a **ADF** pipeline that extract the sales data, transforms it and load in the target. For that we need a set of activities.



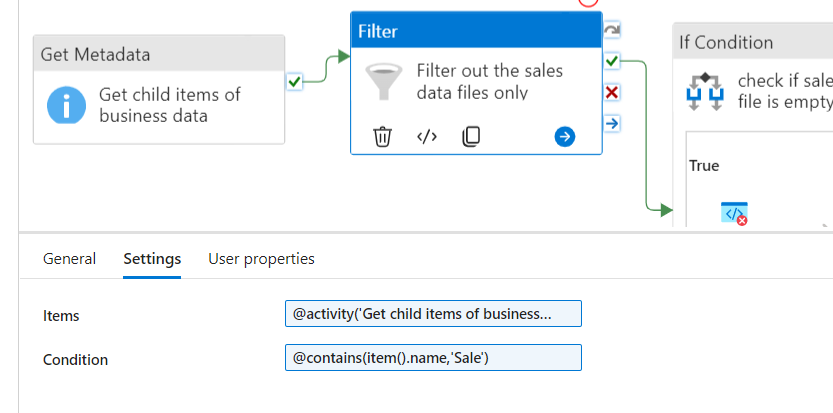
1. Get Metadata Activity:



* Source: business data container in blob storage that contains both sales and employee data.
* Argument: We are passing child items – it gives all child items present in container.
* Filter by Last modified: helps to take in the latest data for processing(here it takes in the modified file in last 12 hrs)

1. Filter Activity:

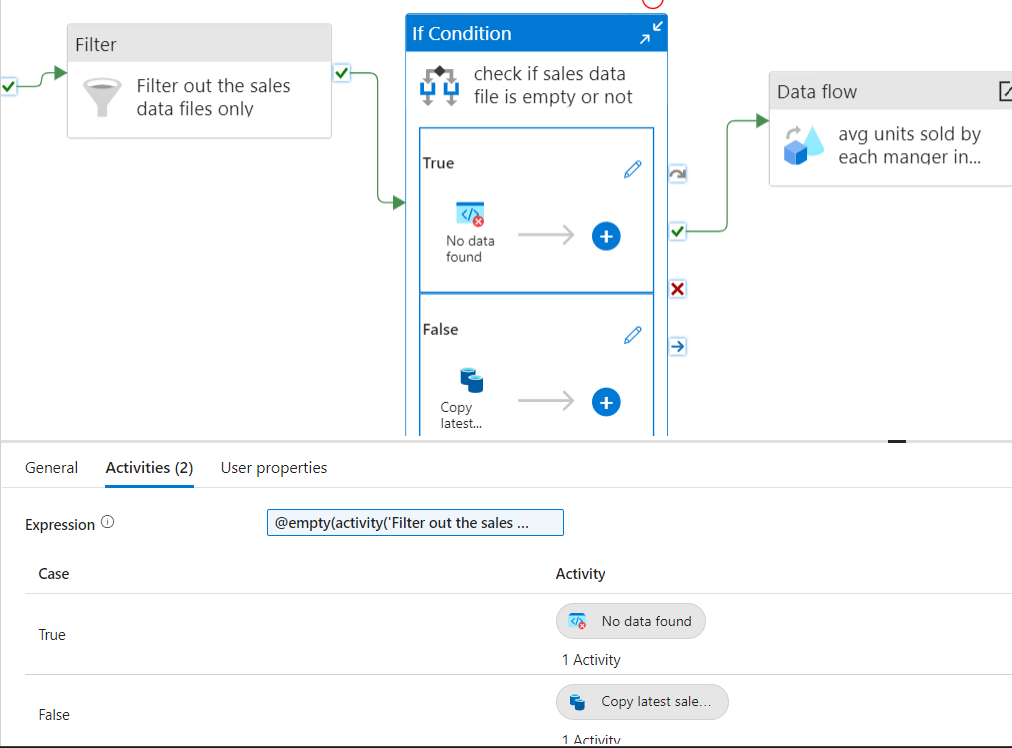
From the child items of Getmetadata activity, it filter out sales data only using condition(file name starting with ‘Sale’)



* Items: @activity('Get child items of business data').output.childItems
* Condition: @contains(item().name,'Sale')

1. If Condition Activity:

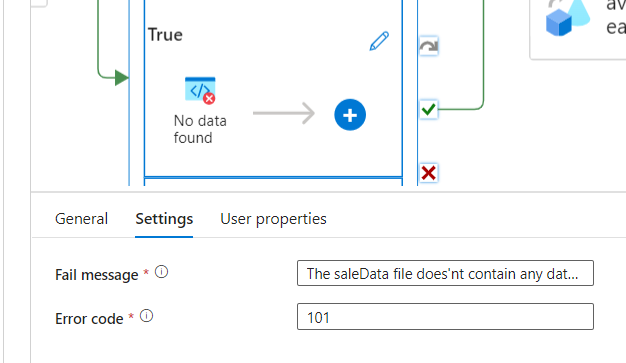
Check if the Incoming Sales data is empty or not.



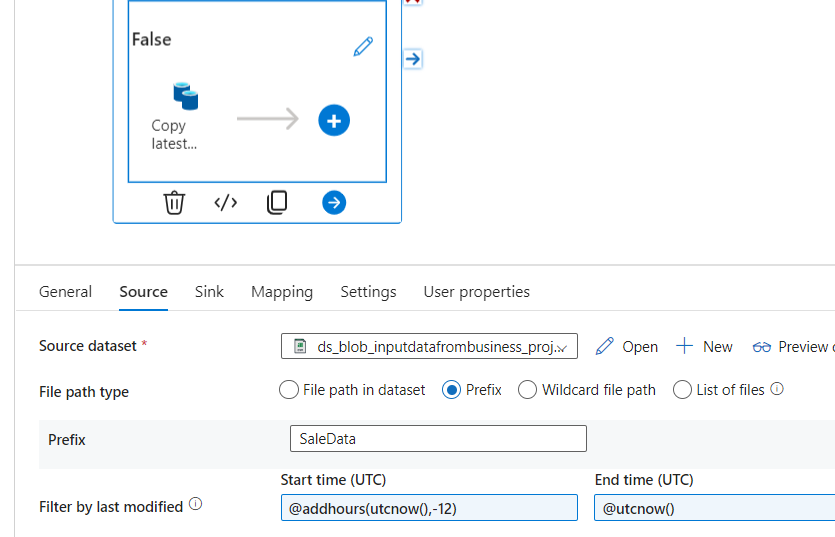
Expression: @empty(activity('Filter out the sales data files only').output)

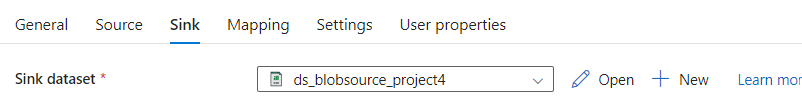
There are 2 activities inside If Condition Activity:

* If condition is true 🡪 Fail activity: which says no data is present in sales data file with an error code.



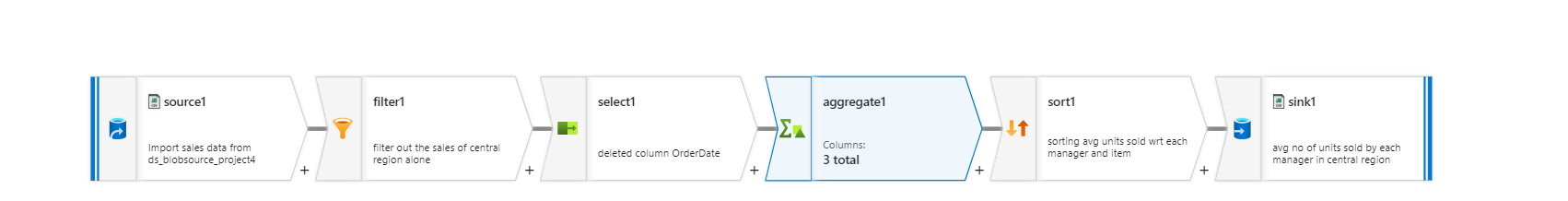
* If condition is false 🡪 Copy data from business container to source container and make data available for data flow transformations.

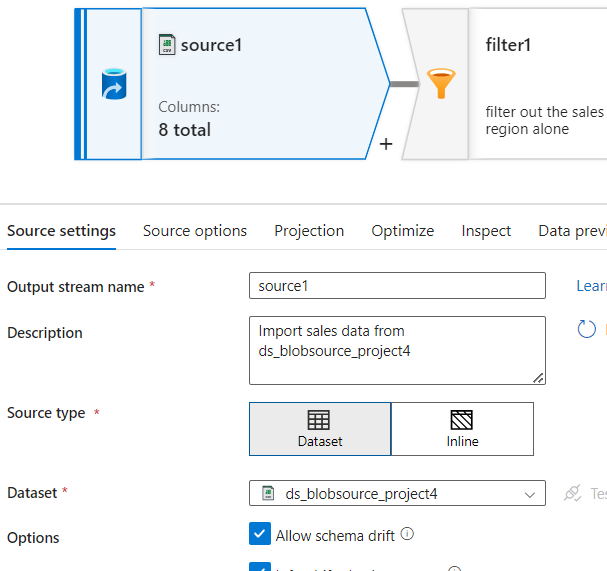




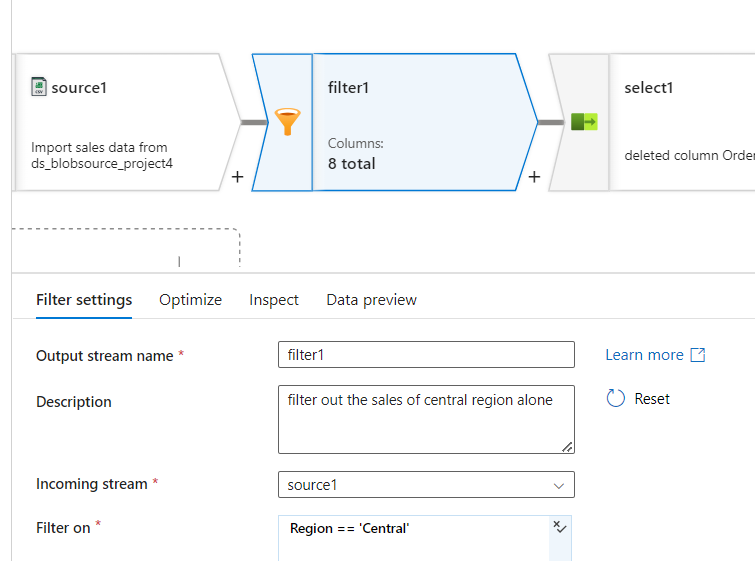
>>Prefix: takes in files starting with SaleData and copy to source container in blob storage.

>>Filter by last modified: helps to take the latest data for processing(here it takes in the modified file in last 12 hrs)

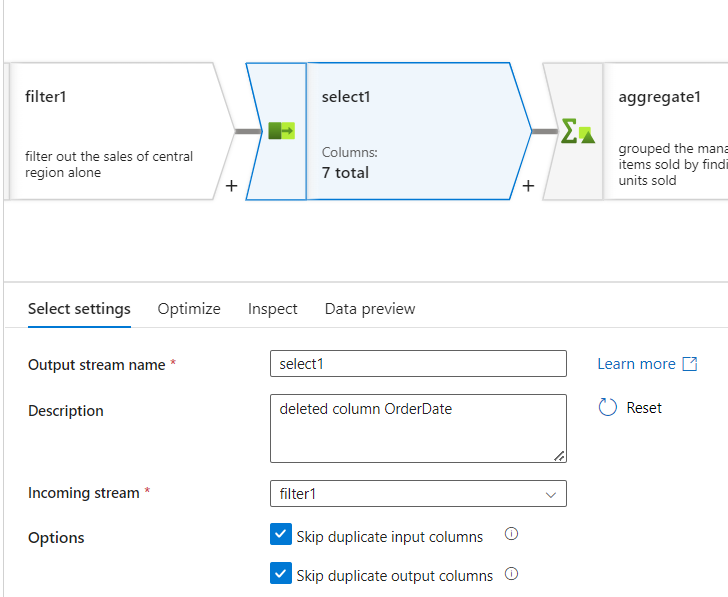
1. Data flow Activity
2. Source:



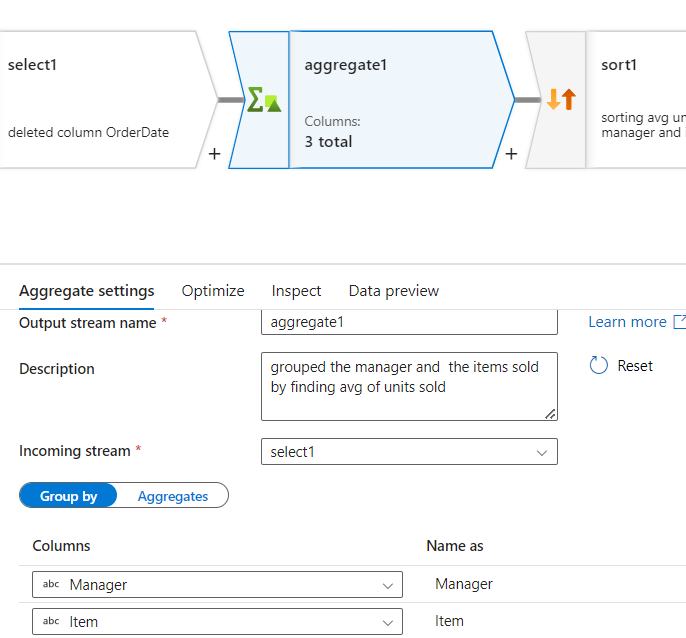
1. Filter:

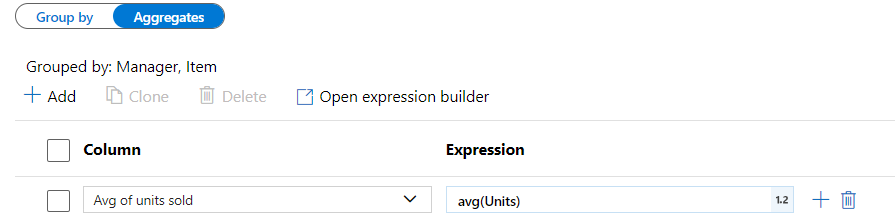


1. Select:

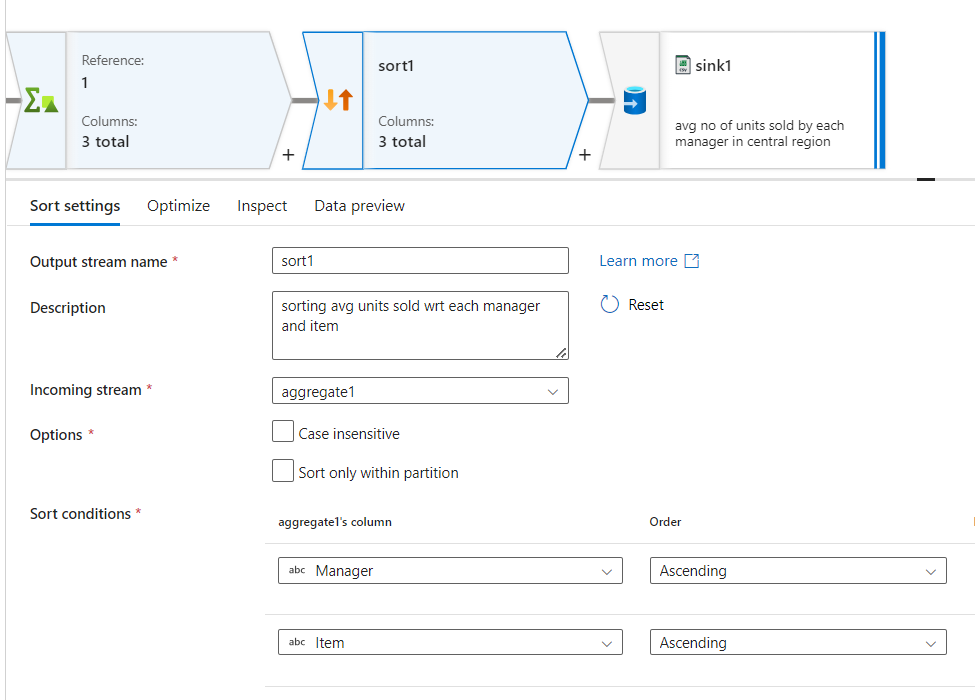


1. Aggregate:

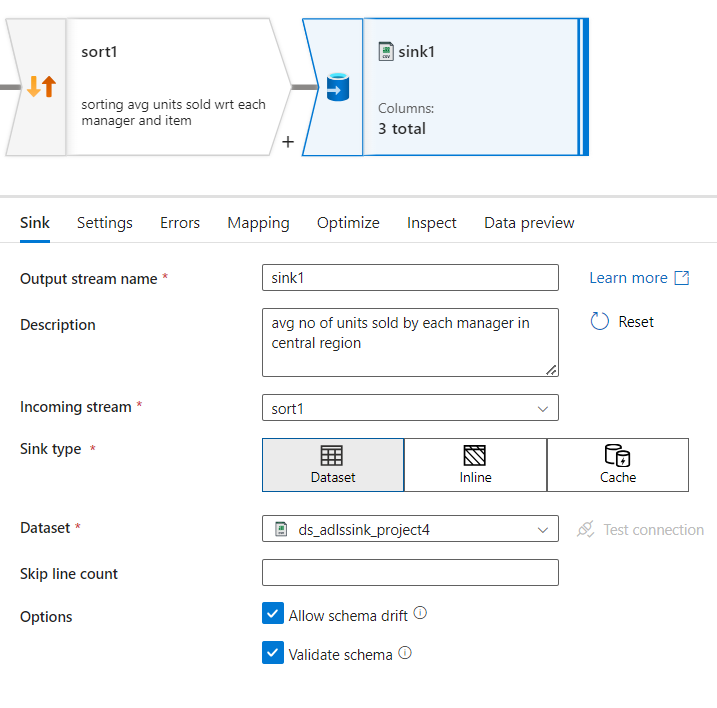




1. Sort:

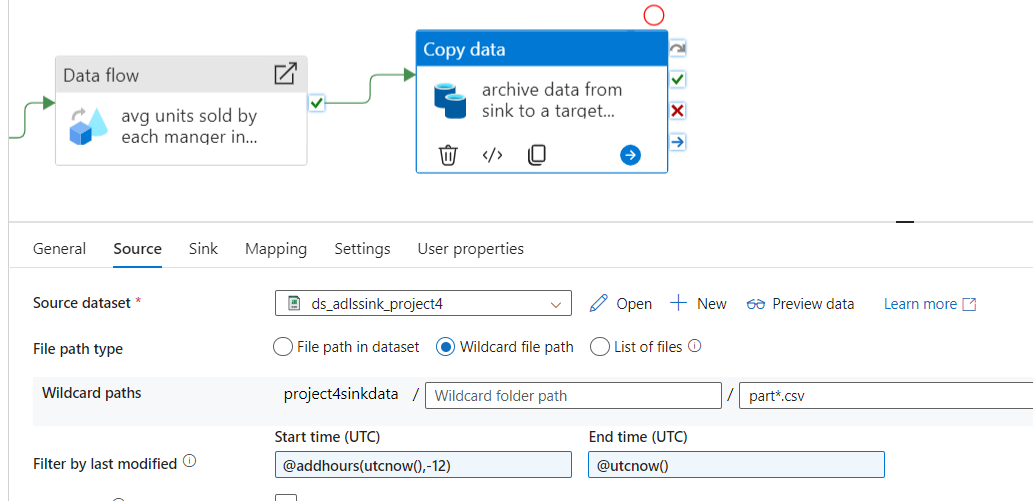


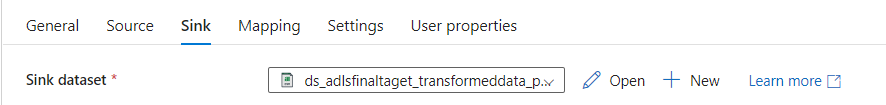
1. Sink:



1. Copy Data Activity:

To archive the data we copy it from sink container to final target container that contains transformed data. It is like creating back up of transformed data.

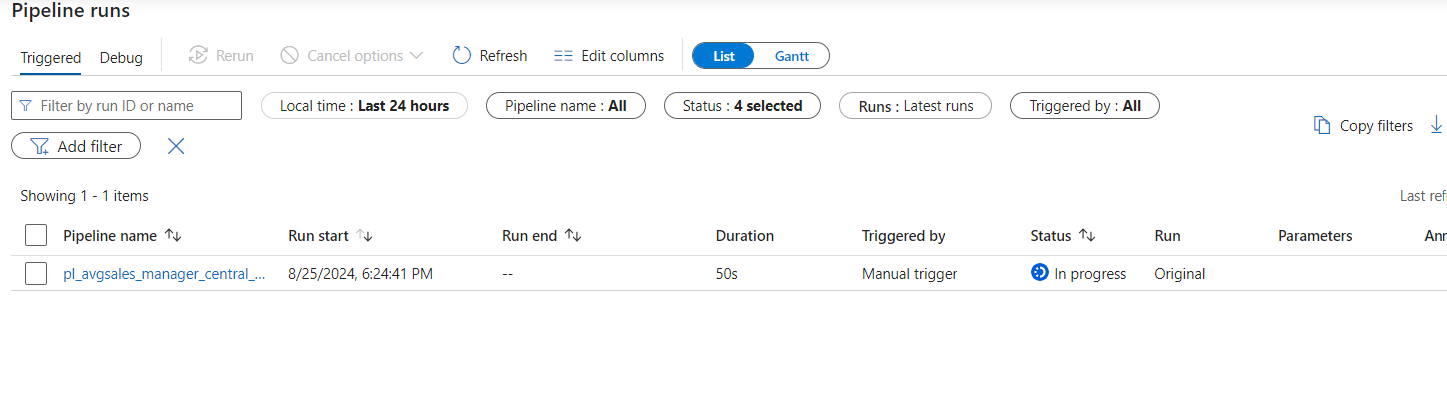


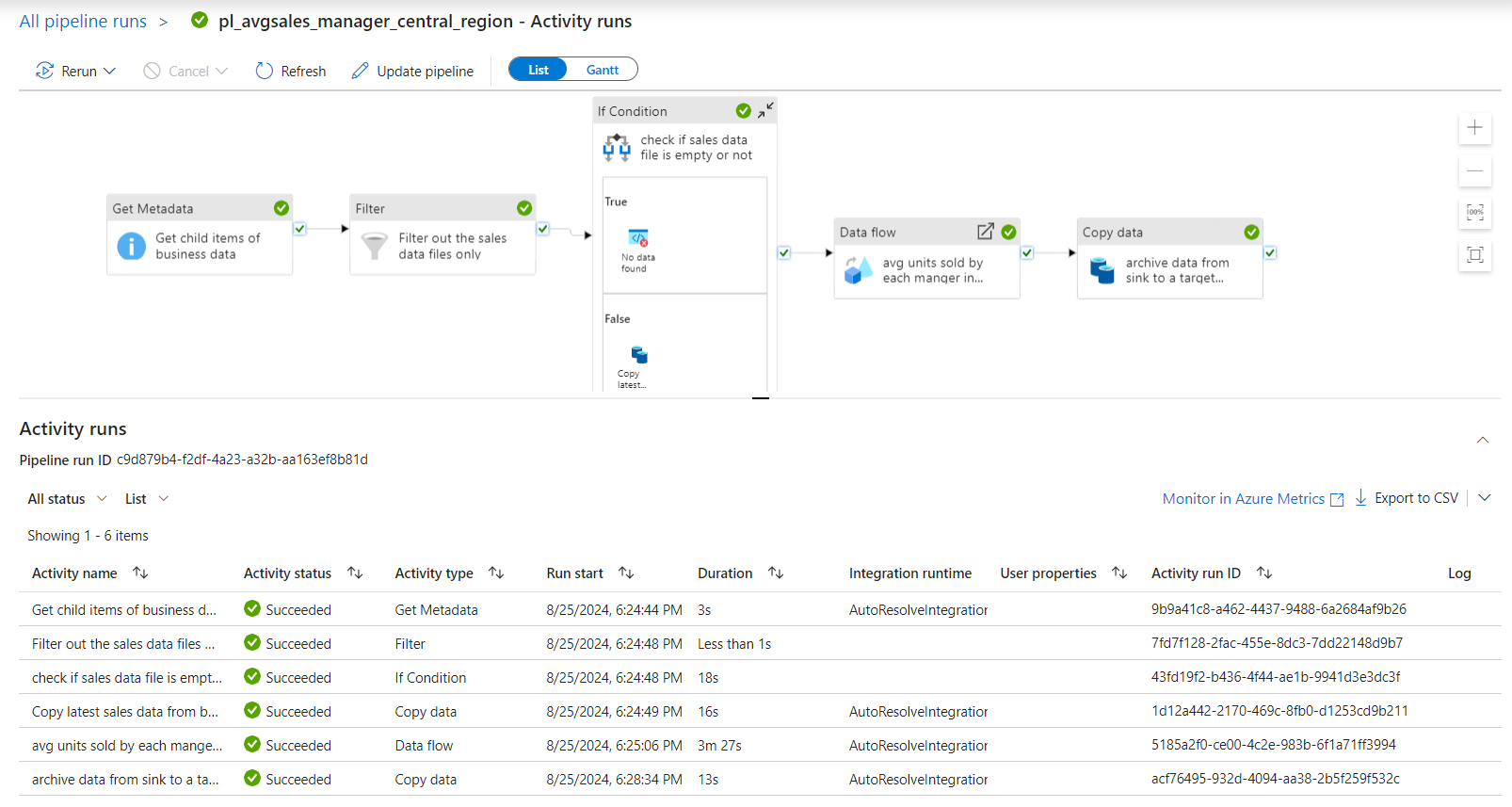


* Wildcard paths : takes in files starting with ‘part’ and copy to target container in ADLS.
* Filter by last modified: helps to take the latest data for copying (here it takes in the modified file in last 12 hrs)

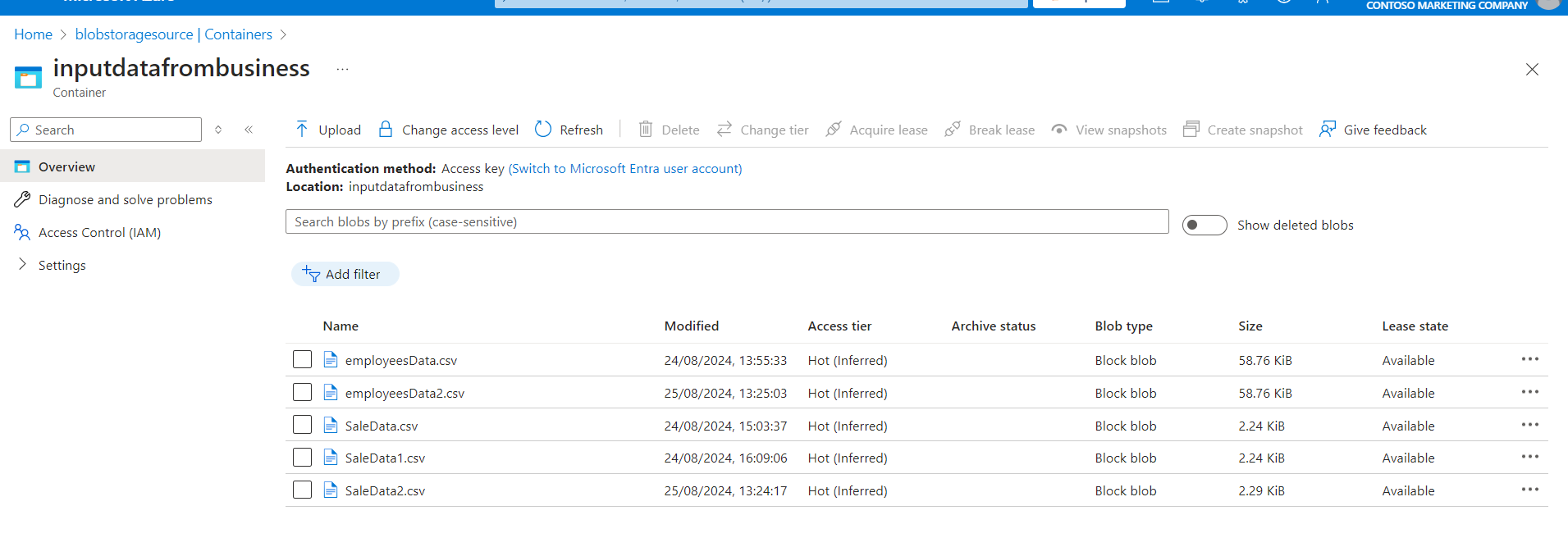
## Triggering pipeline and checking its status:

1. Triggered pipeline and checked monitor tab to see the progress. After sometime pipeline got completed.

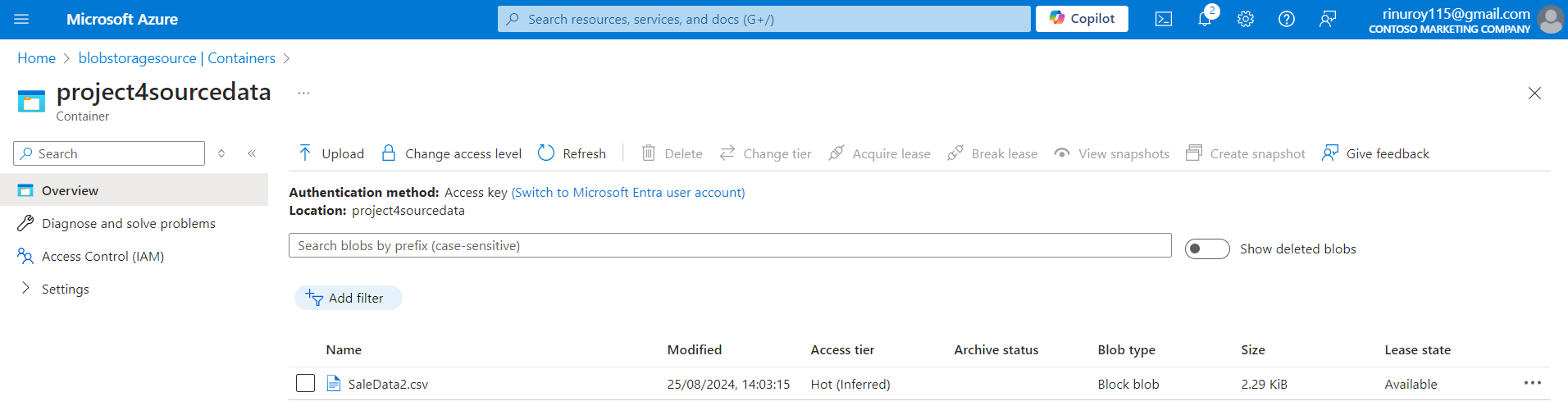




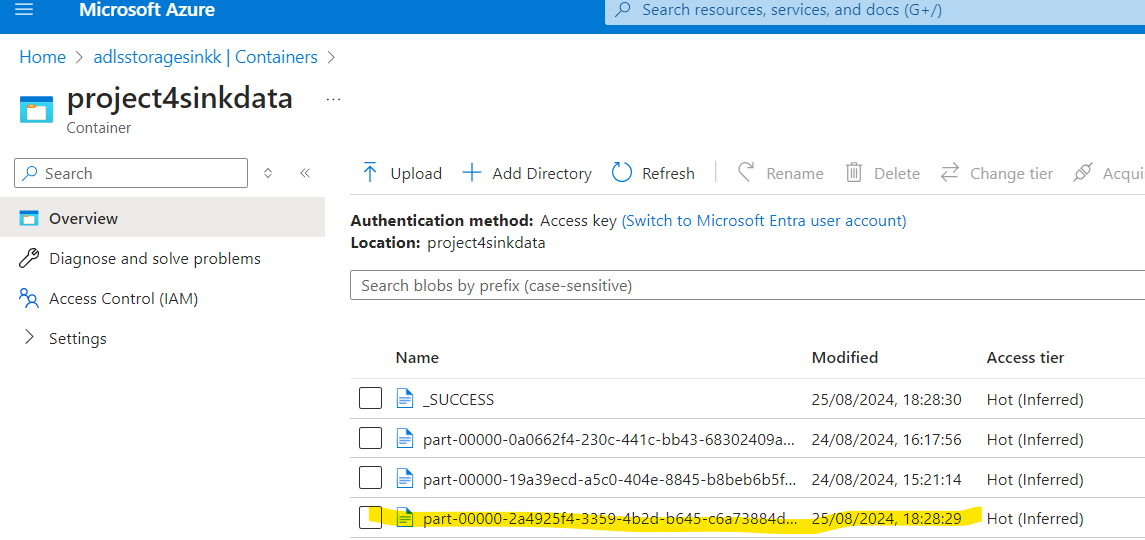
1. The flow of data movement is shown below:
2. First data is uploaded at blobstoragesource (in inputdatafrombusiness container). It contains sales and employee data.



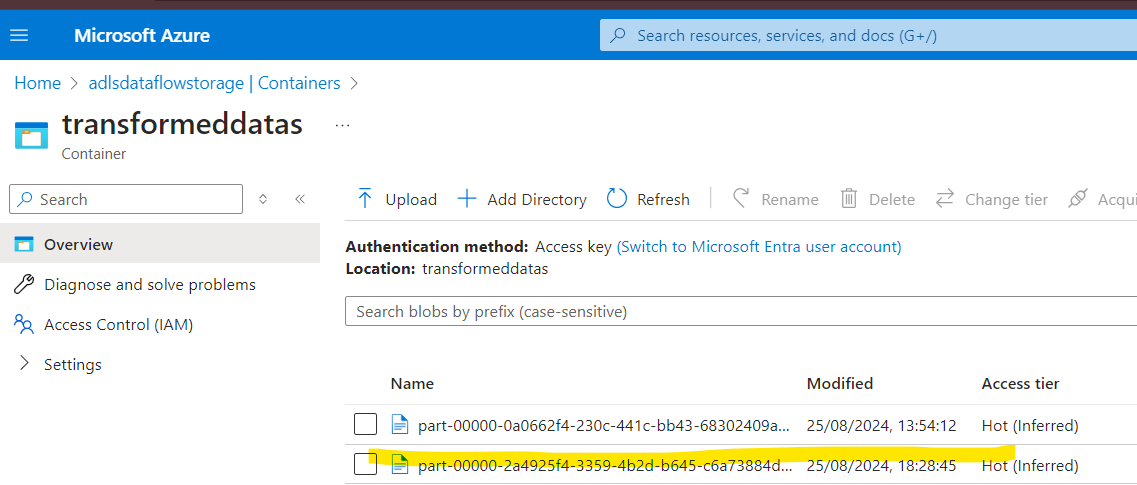
1. After If condition is satisfied latest SaleData is copied to another container named project4sourcedata in blobstoragesource. This act as source for data flow transformation.



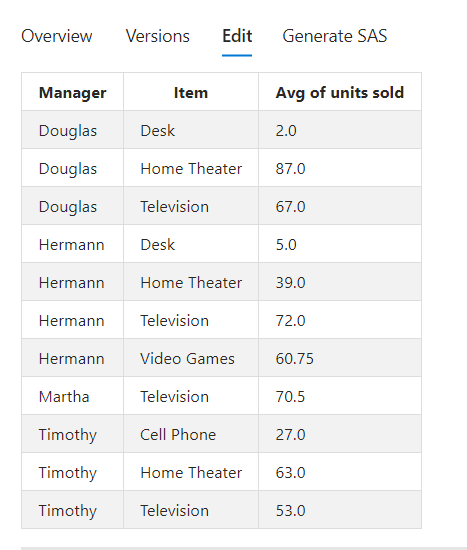
1. After data flow activity, transformed data is stored in an container named project4sinkdata in adlsstoragesink.



1. From there data is copied to Archive location – container: transformeddatas, Location: adlsdataflowstorage



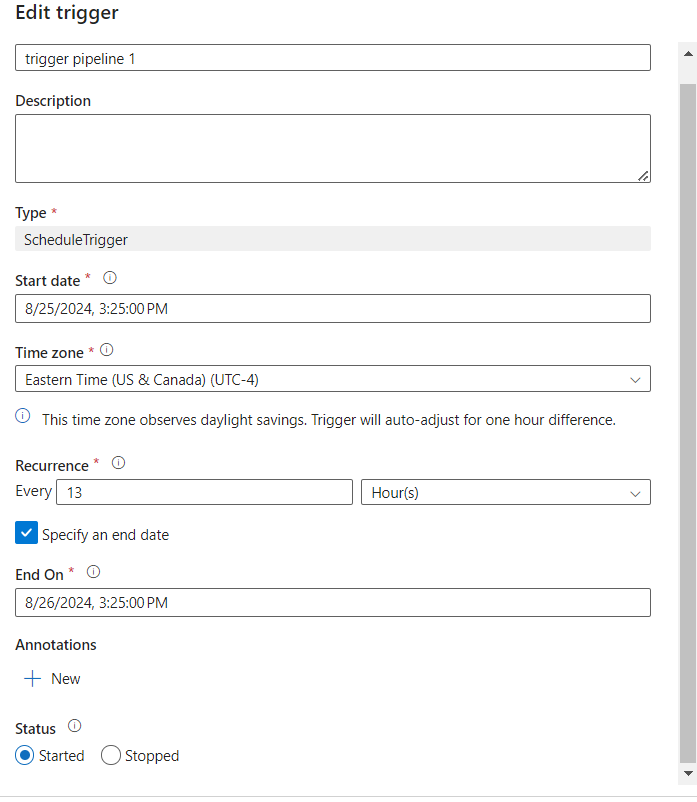
Final Output:



Scheduling Pipeline/ Scheduling Trigger:

Under trigger, add new trigger and schedule it.

Data gets updated in business container every 12 hrs



Creating Failure alert through email notification:

Under monitor tab >> Alerts & metrics >> create new alert rule for a single or group of pipelines.

