..

---

Project: Gold_Finance ETL

---

**Introduction**

Gold_Finance ETL: a robust data engineering project that automates financial time series data processing, transforms raw financial data into structured insights for business stakeholders and analysts.

**Table of Contents**



**Business_Problem_Solution**

| S/N | Business Problem |
|---|---|
| 1 | High volume of rapidly changing financial data |
| 2 | Unstructured formats requiring significant cleaning |
| 3 | Time-intensive manual workflows prone to errors |

| S/N | Business Solution |
|---|---|
| 1 | Fully automated workflows for daily data integration and analysis |
| 2 | Aggregated metrics for informed decision-making |

**Alpha Vantage Api**

++ TIME_SERIES_DAILY:

This API returns raw (as-traded) daily time series (date, daily open, daily high, daily low, daily close, daily volume) of the global equity specified, covering 20+ years of historical data. The OHLCV data is sometimes called "candles" in finance literature. If you are also interested in split/dividend-adjusted data, please use the Daily Adjusted API, which covers adjusted close values and historical split and dividend events.

Reference: https://www.alphavantage.co/documentation/

**API Parameters**

❙ **Required: function**

The time series of your choice. In this case, function=TIME_SERIES_DAILY

❙ **Required: symbol**

The name of the equity of your choice. For example: symbol=IBM

❙ Optional: outputsize

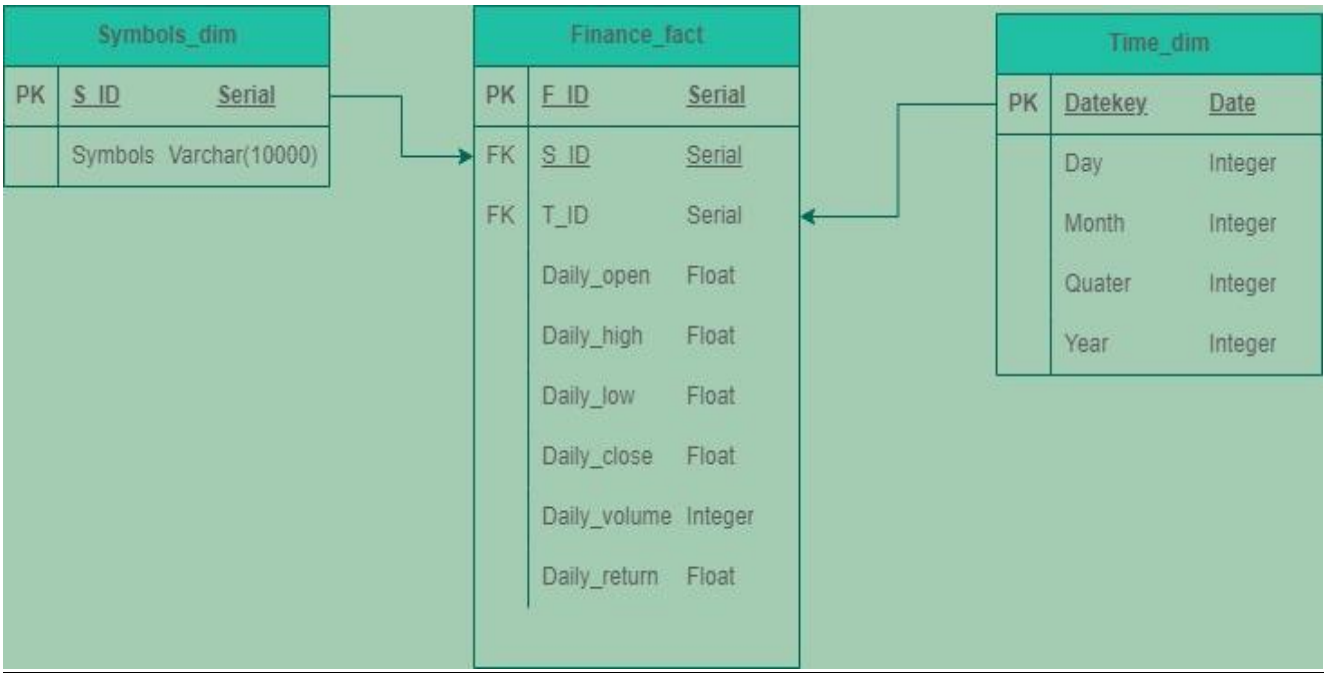
By default, outputsize=compact. Strings compact and full are accepted with the following specifications: compact returns only the latest 100 data points; full returns the full-length time series of 20+ years of historical data. The "compact" option is recommended if you would like to reduce the data size of each API call.

❙ Optional: datatype

By default, datatype=json. Strings json and csv are accepted with the following specifications: json returns the daily time series in JSON format; csv returns the time series as a CSV (comma separated value) file.

❙ **Required: apikey**

**ERD_Schema**

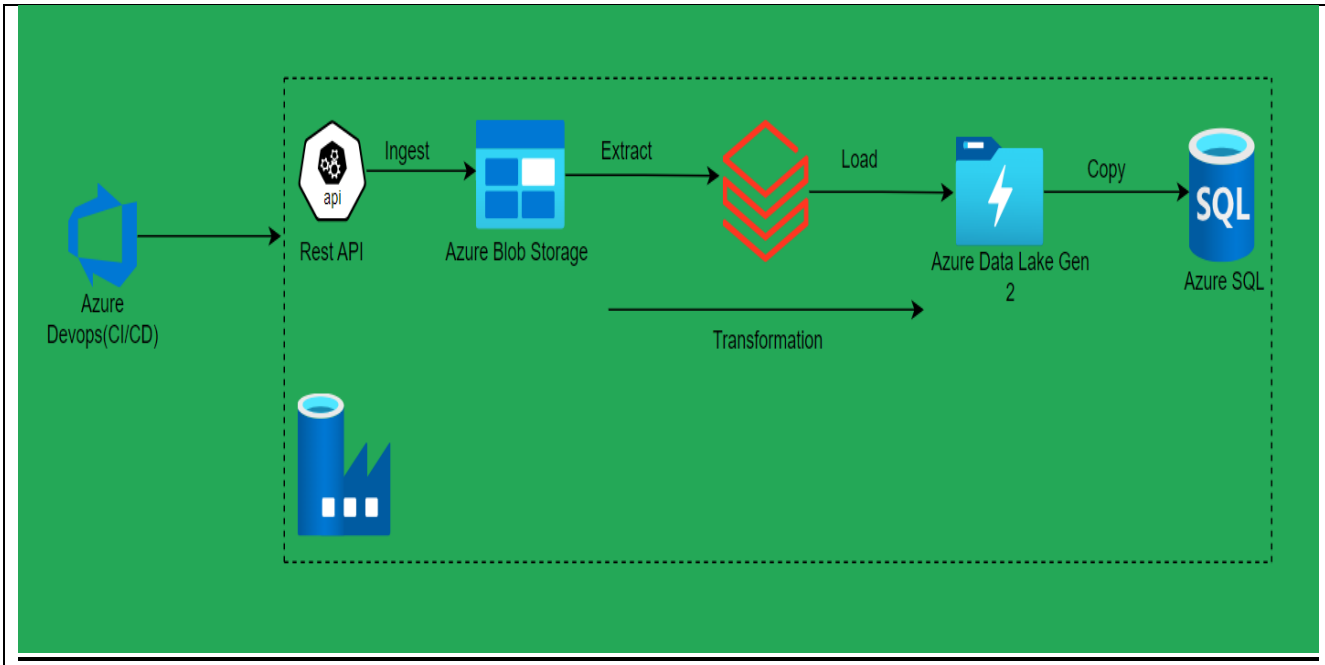



---

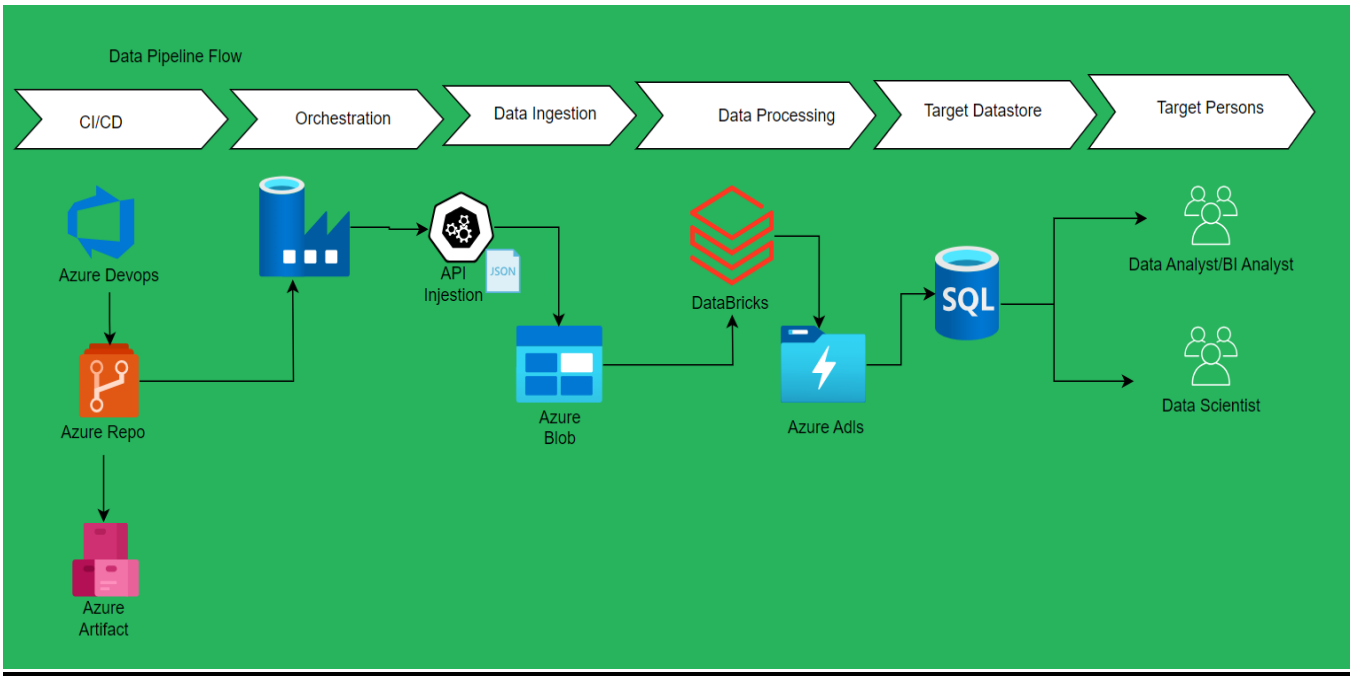
1.0 Azure Project: Gold_Finance ETL

---

**Azure_Solution_Architecture**

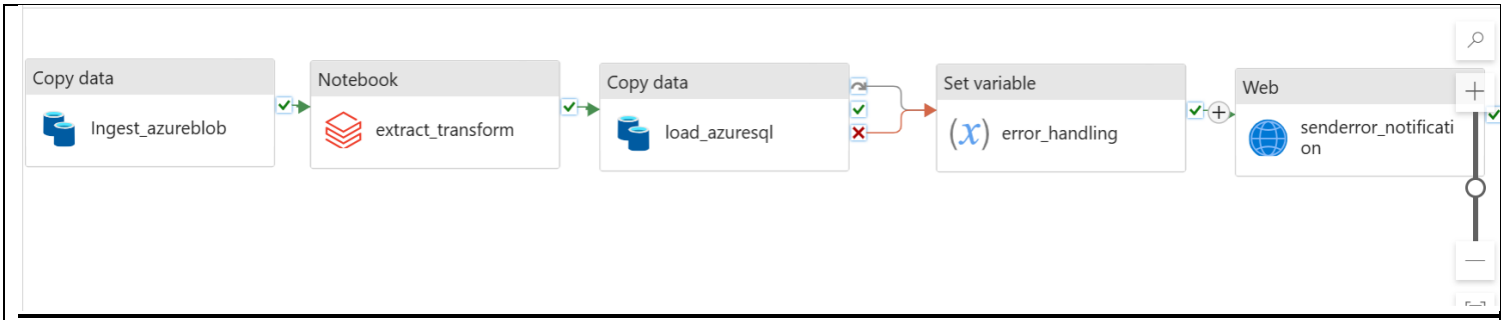


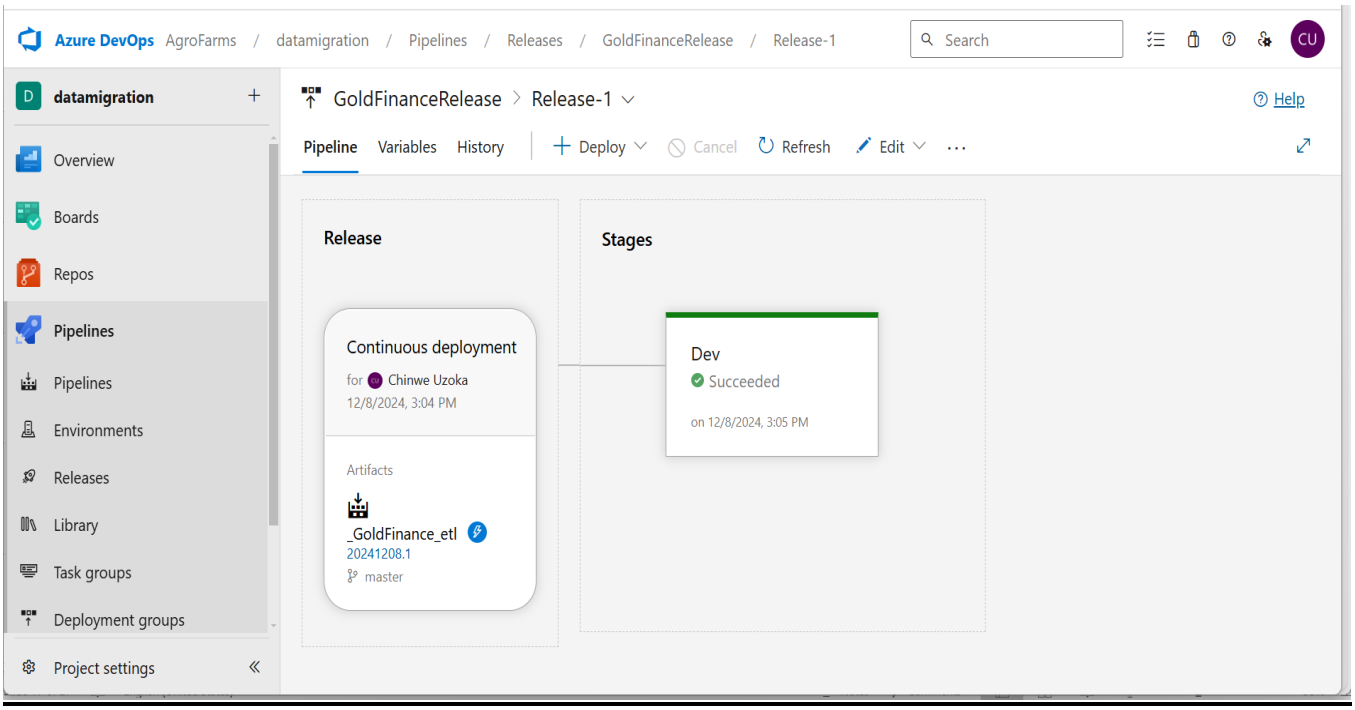

## Data_Pipeline_Flow

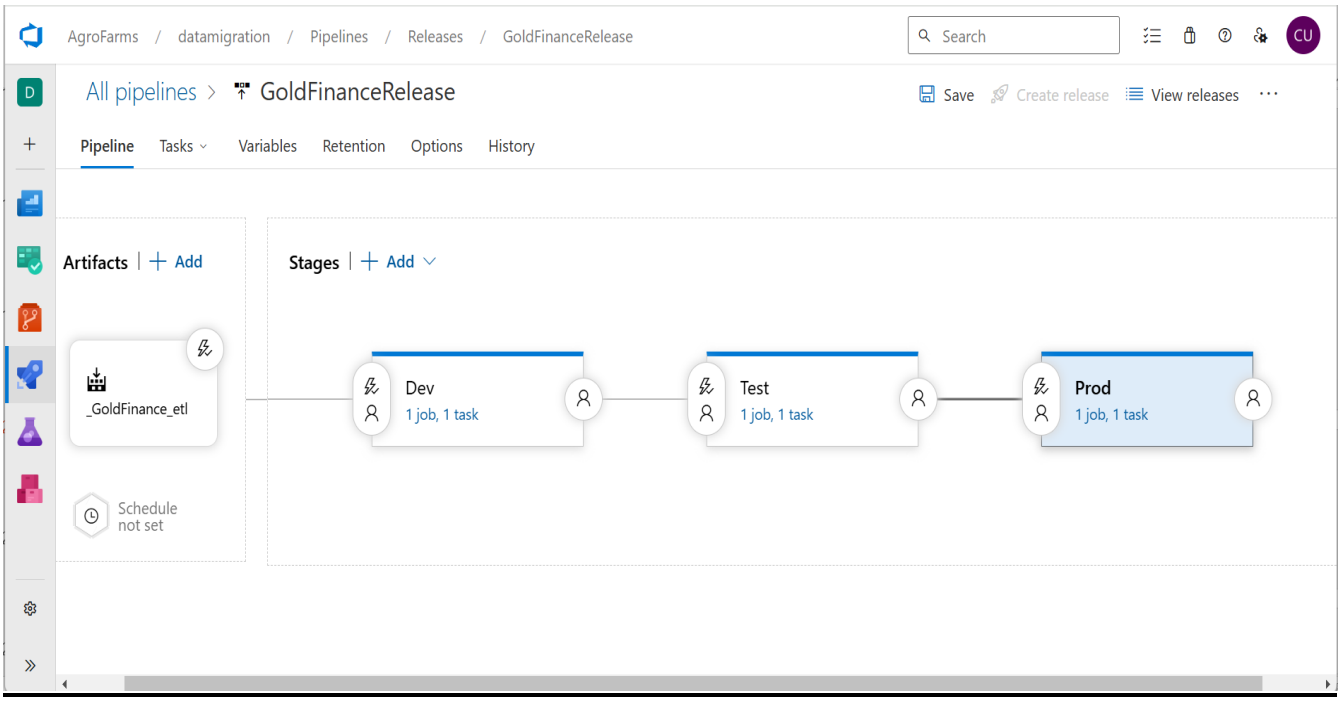



## Azure_Project_Tech_Stack

| S/N | Tech Stack | Details |
|---|---|---|
| 1 | Azure Storage Account | Data lake, Datawarehouse |
| 2 | Databricks | Transformation |
| 3 | Azure Data Factory | Orchestration |
| 4 | Azure Logic Apps | Error Handling and Notification |
| 5 | Azure DevOps | CI/CD |



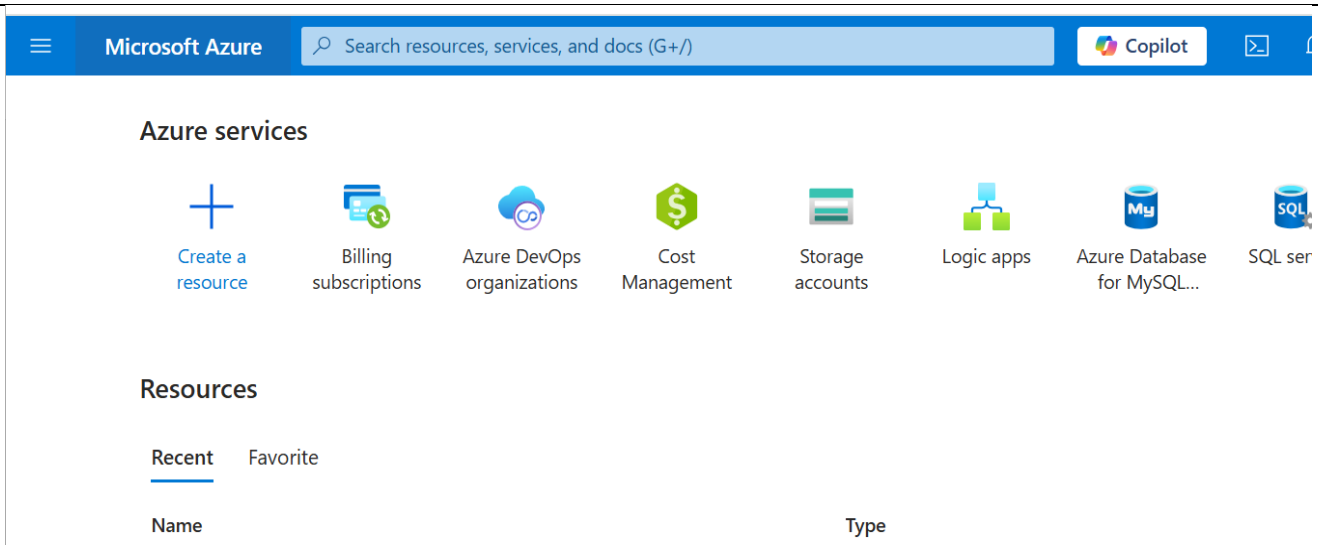




**+++ Azure_Storage_Setup**

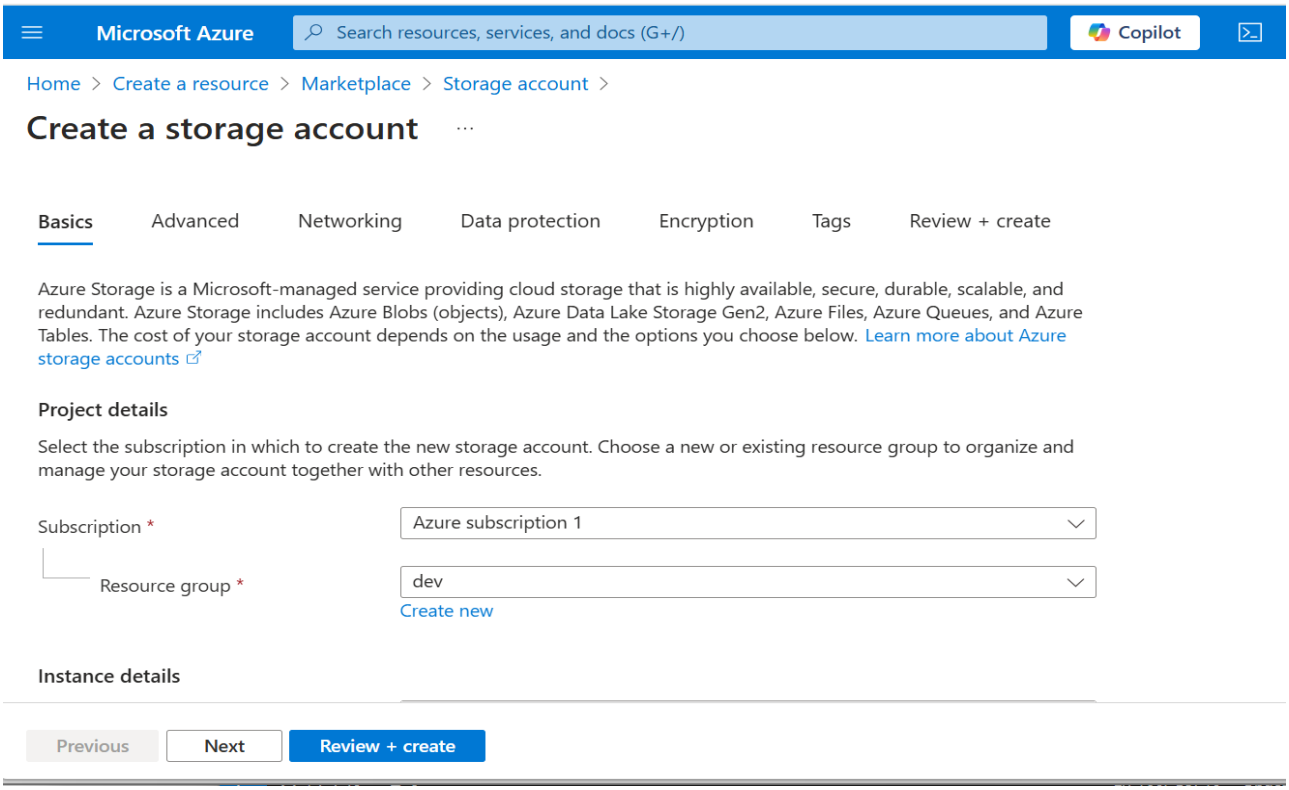
Create Storage Account and create Containers for both blob storage and adls gen 2

Create a Storage Account:

- Click on create a resource

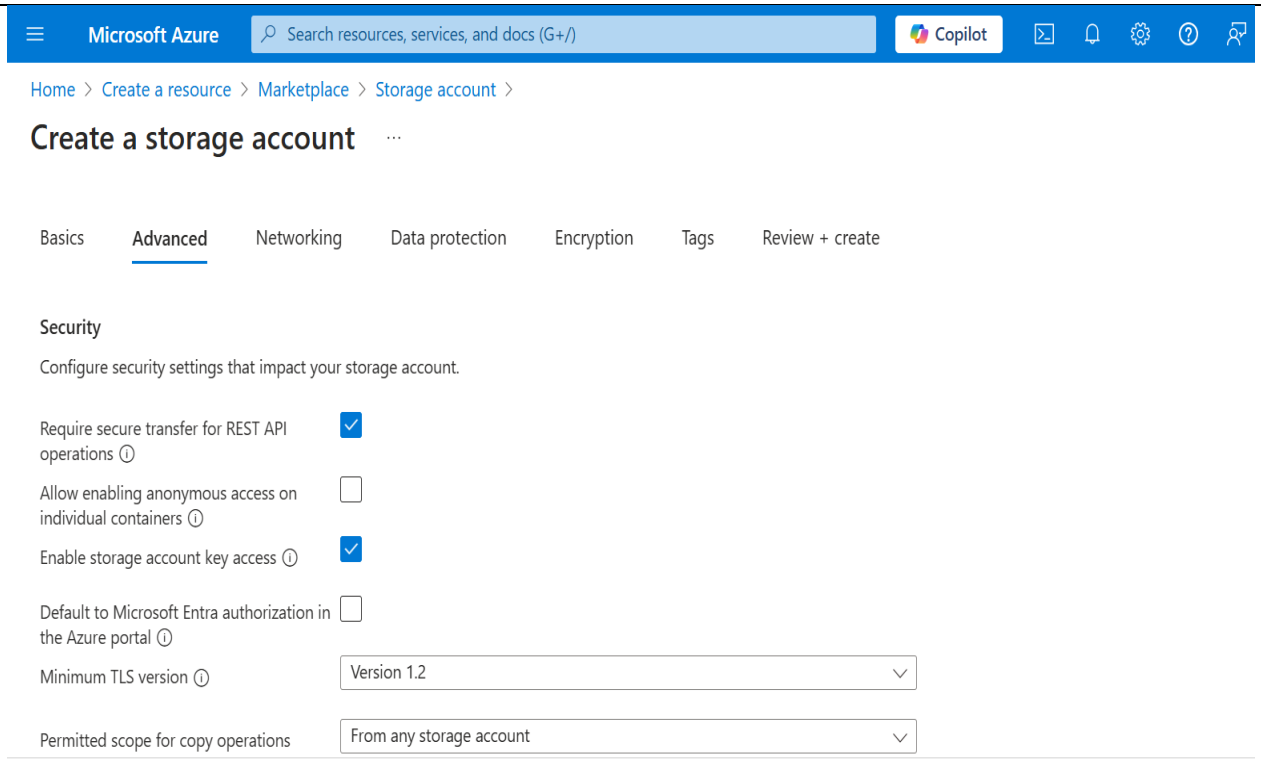


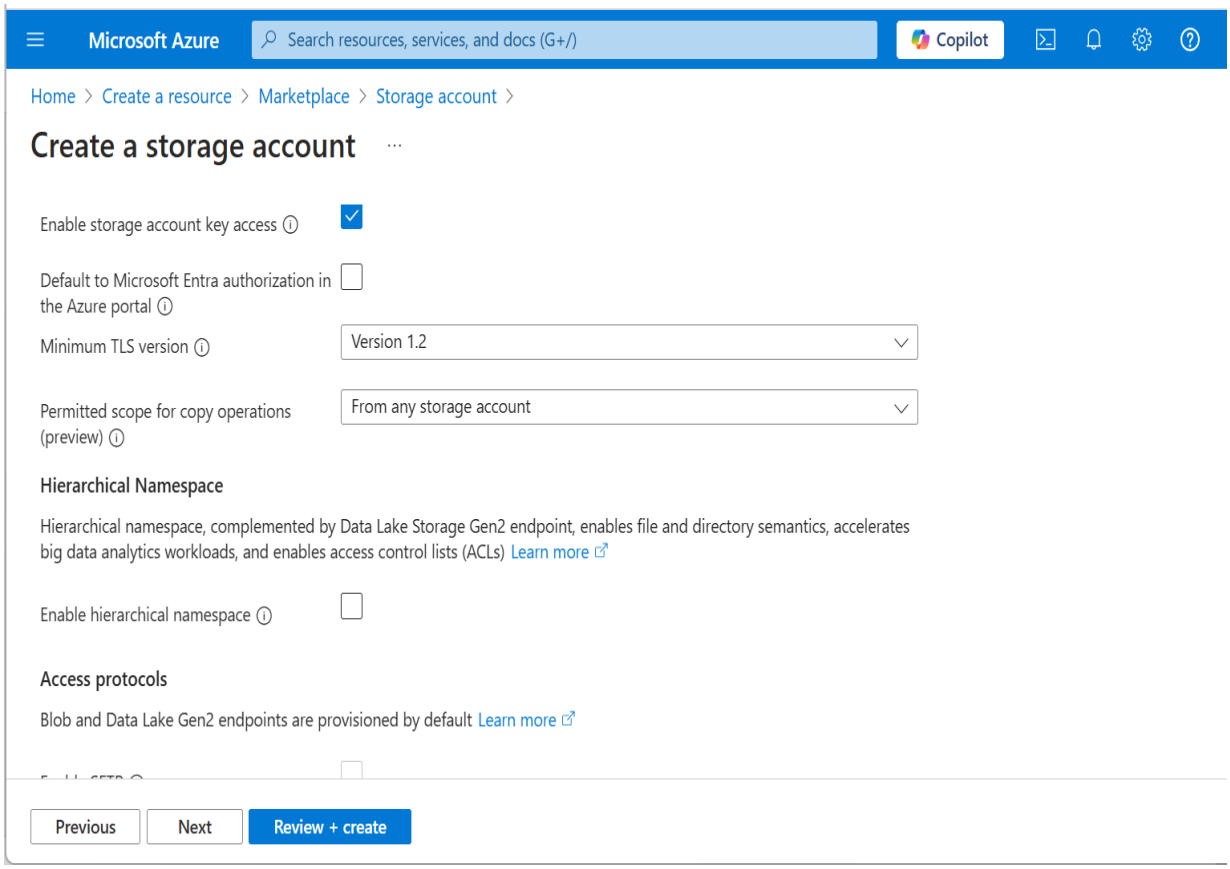

- Type storage account in the search bar and select storage account, fill the below as desired with respect to business use case.



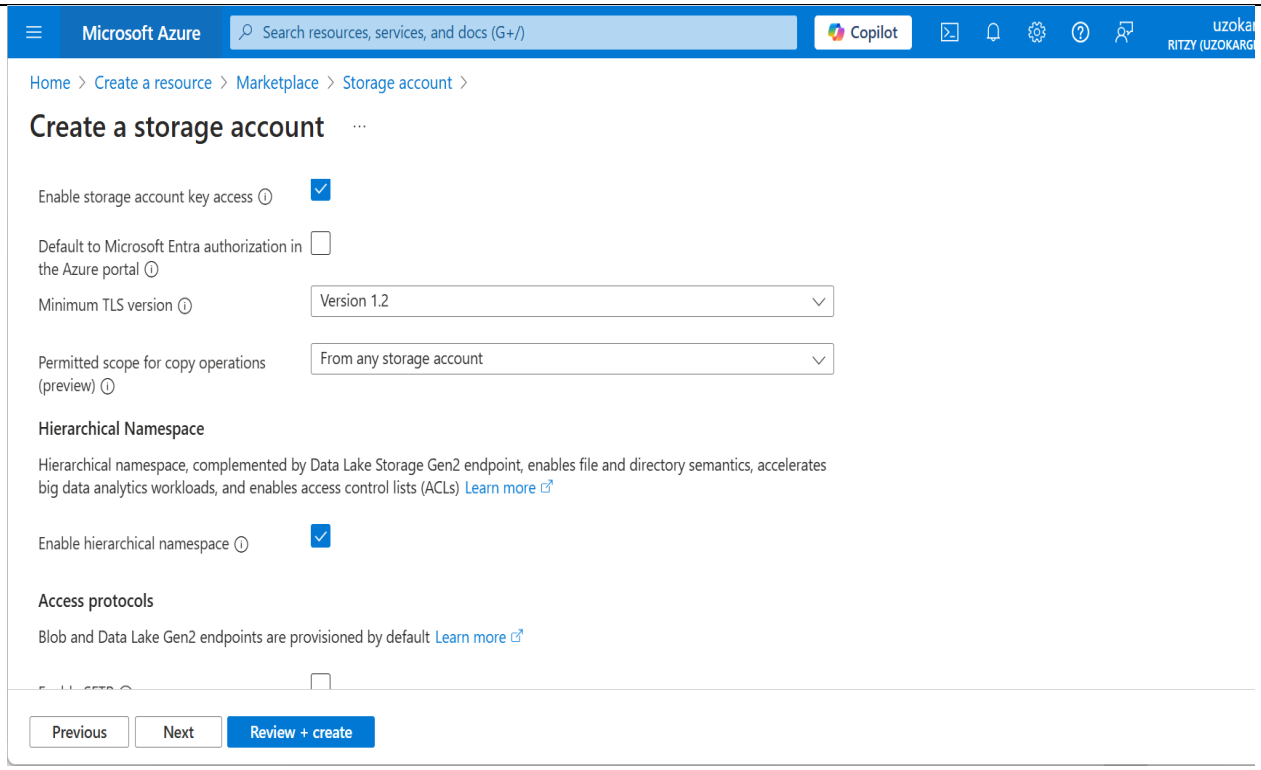

Create Storage account for both blob storage and adls gen 2

- While creating a blob storage account make sure to uncheck the “enable hierarchical namespace”

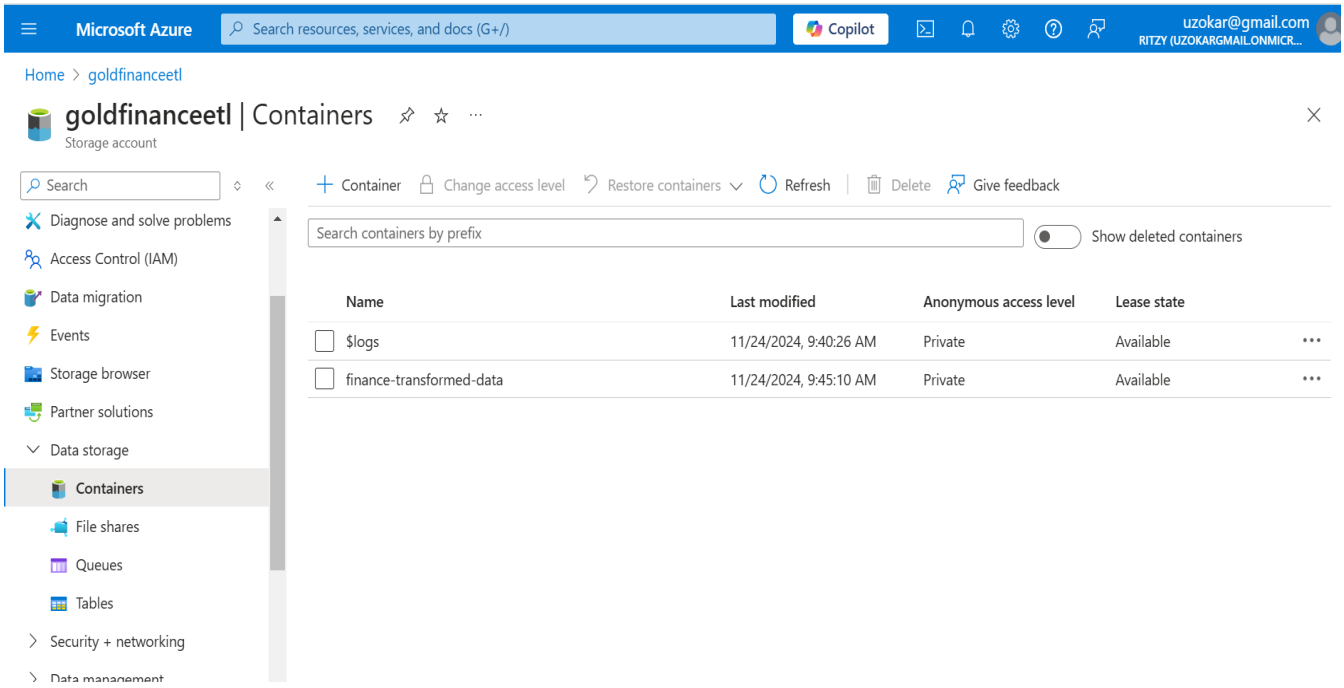


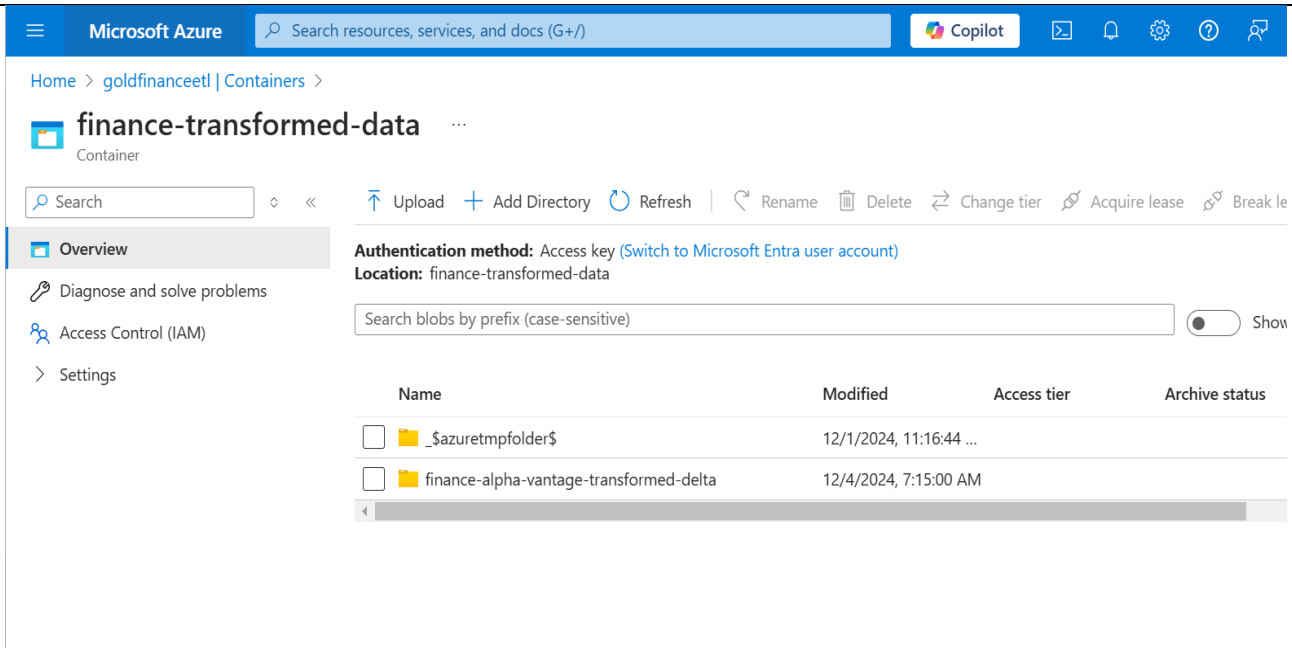

- While creating an adls Gen 2 storage account make sure to check the “enable hierarchical namespace”



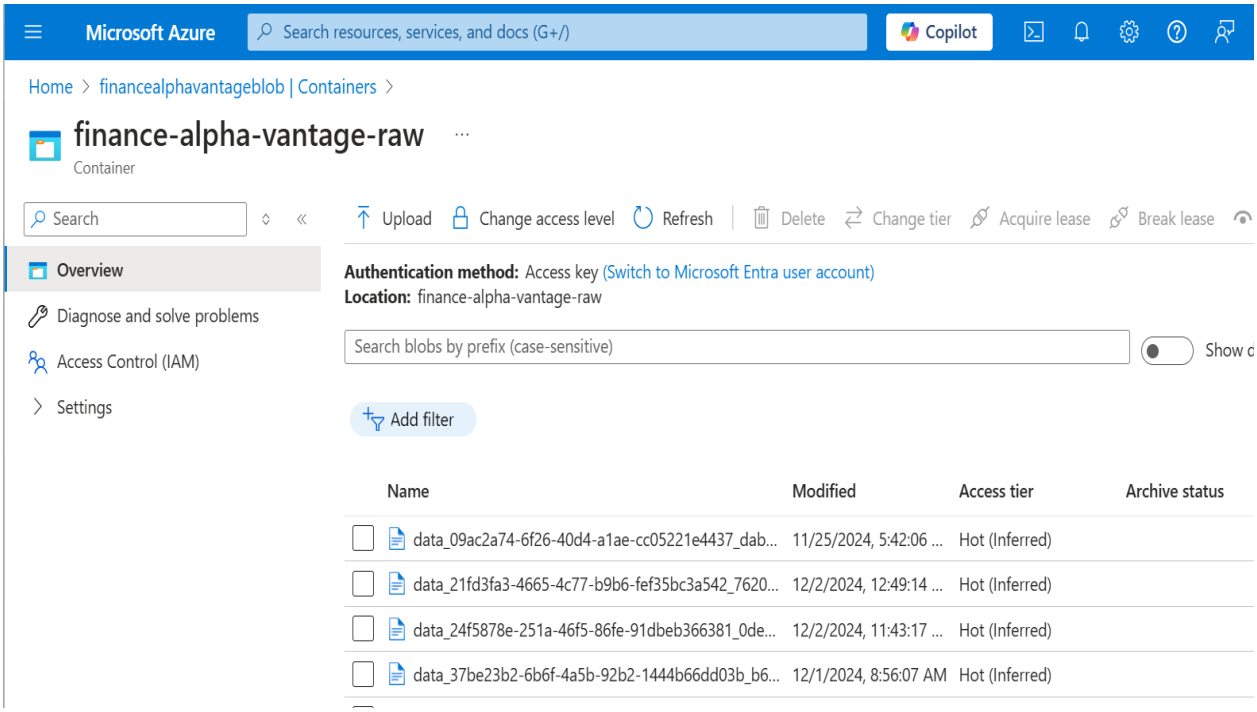

- Create a container for both blob storage and adls gen 2




- An adls gen 2 Container created



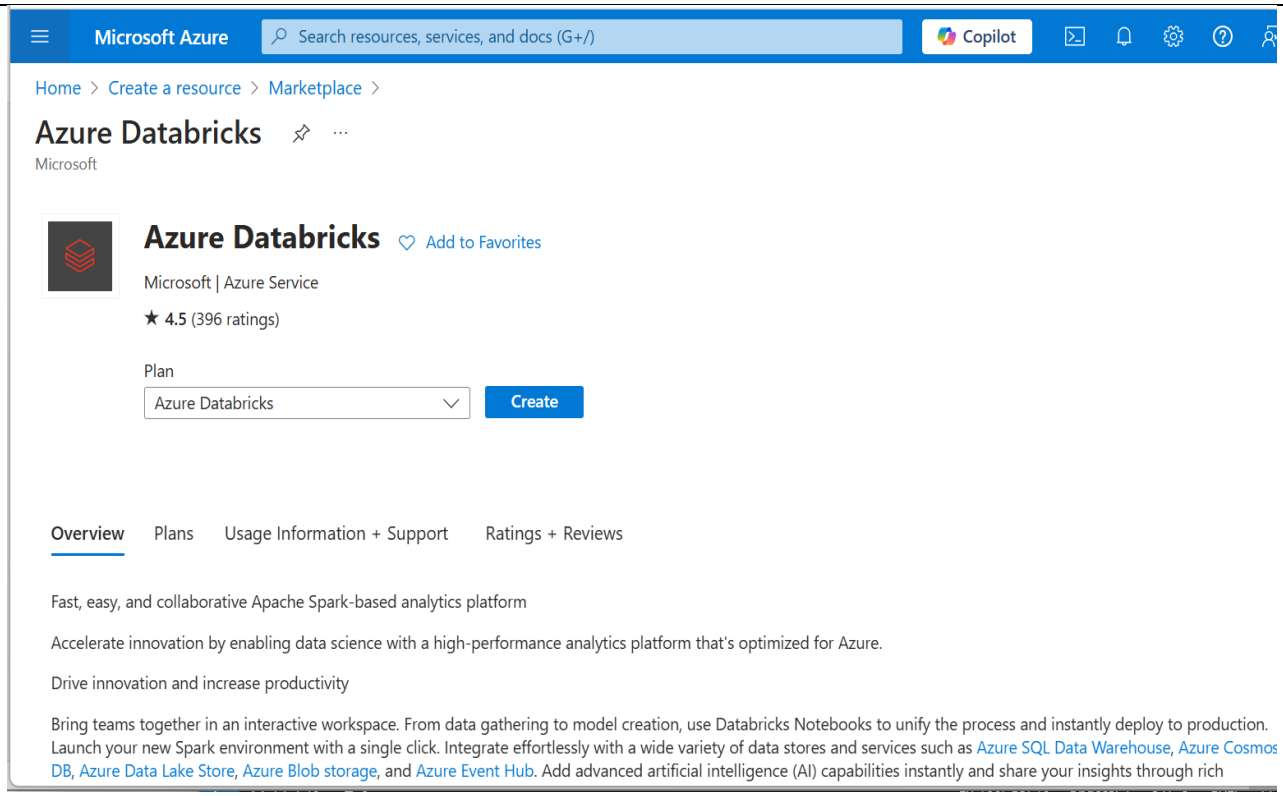

- A blob storage container created

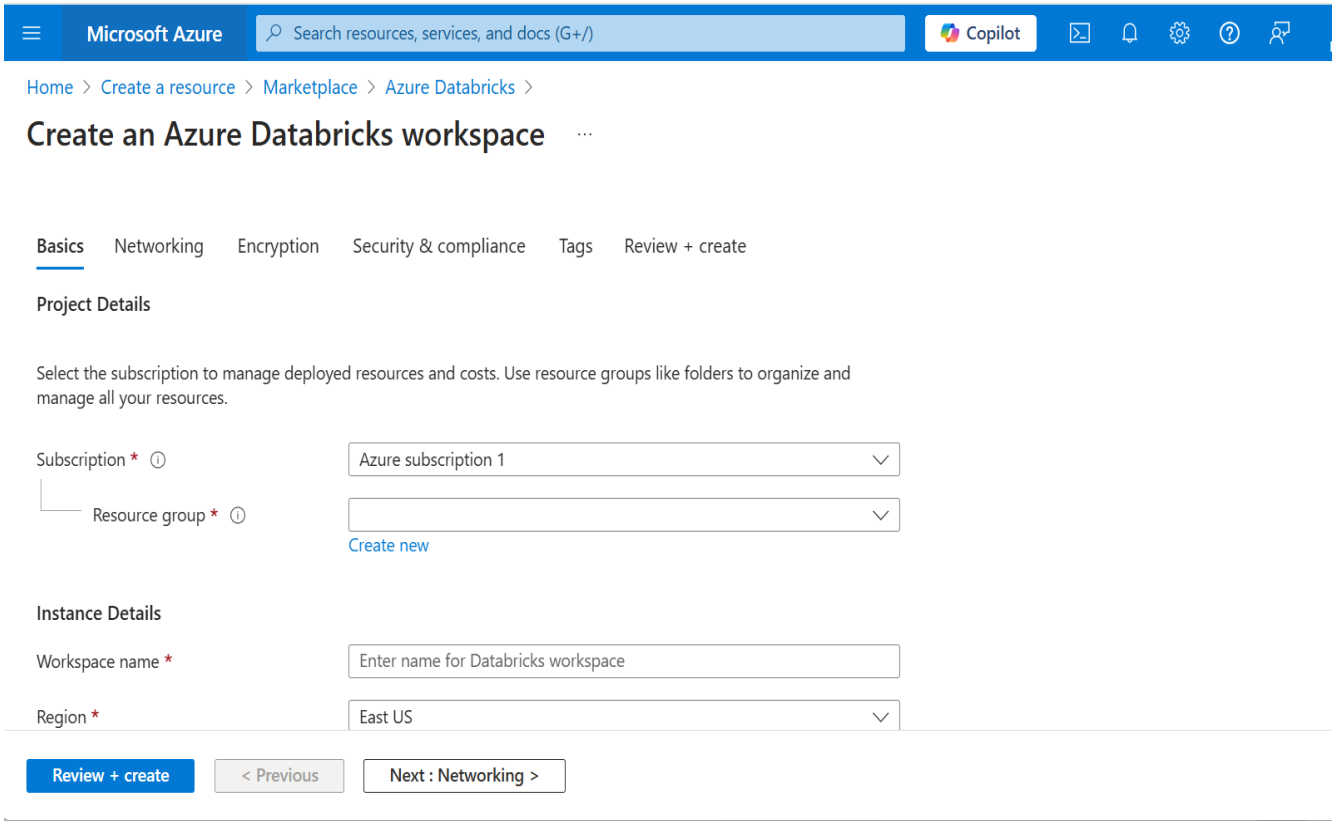


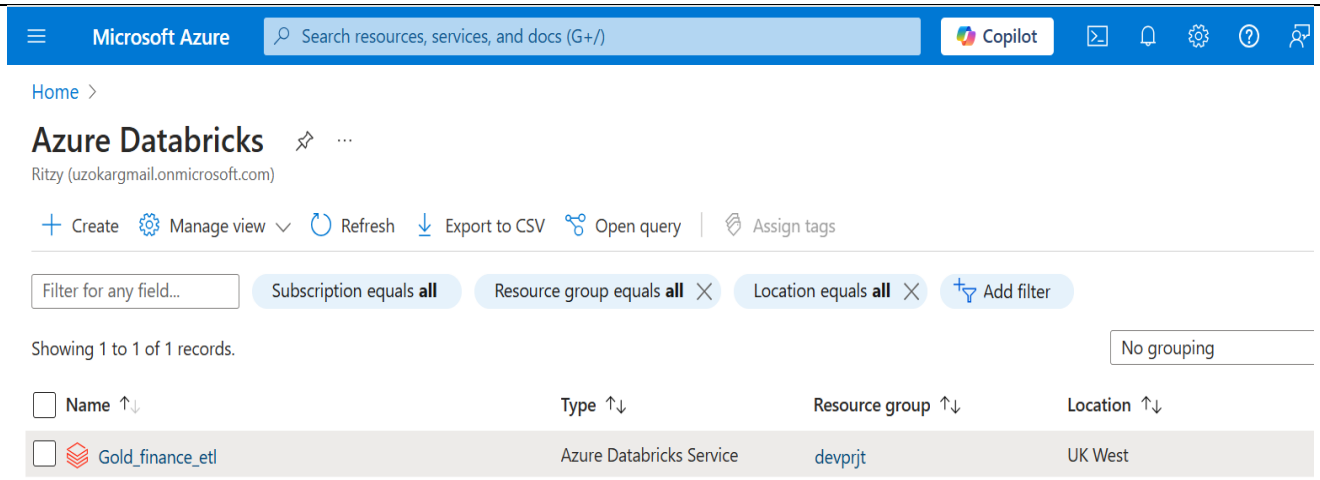

**+++ Azure Databricks Setup**

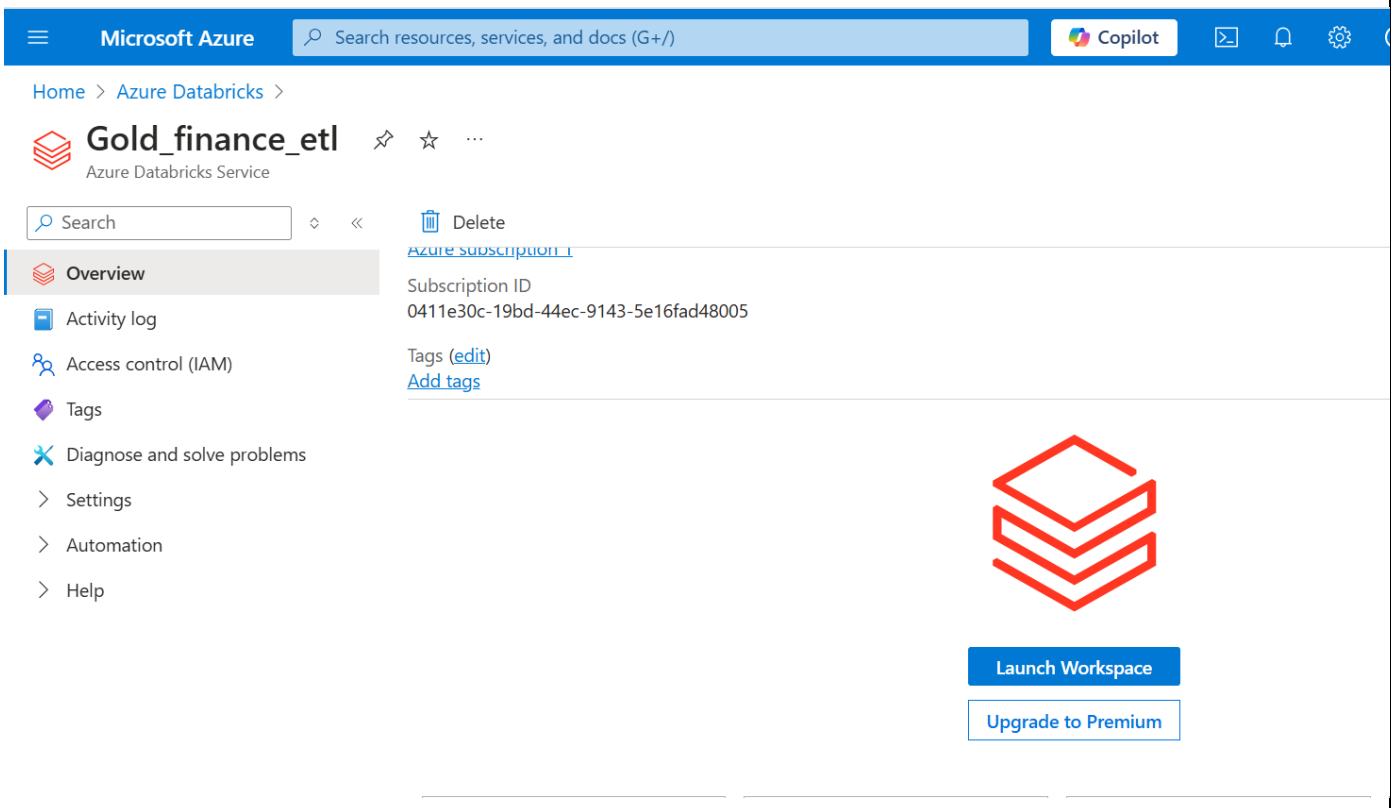
Create Azure Databrick's Workspace

- Click on create a resource search for Azure Databricks and select Azure Databricks.


Microsoft Azure
Search resources, services, and docs (G+/)
Copilot
Home > Create a resource > Marketplace >
Azure Databricks
Microsoft
Azure Databricks
Add to Favorites
Microsoft | Azure Service
★ 4.5 (396 ratings)
Plan
Azure Databricks
Create
Overview
Plans
Usage Information + Support
Ratings + Reviews
Fast, easy, and collaborative Apache Spark-based analytics platform
Accelerate innovation by enabling data science with a high-performance analytics platform that's optimized for Azure.
Drive innovation and increase productivity
Bring teams together in an interactive workspace. From data gathering to model creation, use Databricks Notebooks to unify the process and instantly deploy to production. Launch your new Spark environment with a single click. Integrate effortlessly with a wide variety of data stores and services such as Azure SQL Data Warehouse, Azure Cosmos DB, Azure Data Lake Store, Azure Blob storage, and Azure Event Hub. Add advanced artificial intelligence (AI) capabilities instantly and share your insights through rich


Microsoft Azure
Search resources, services, and docs (G+/)
Copilot
Home > Create a resource > Marketplace > Azure Databricks >
Create an Azure Databricks workspace
Basics
Networking
Encryption
Security & compliance
Tags
Review + create
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 subscription 1
Resource group *
Create new
Instance Details
Workspace name *
Enter name for Databricks workspace
Region *
East US
Review + create
< Previous
Next : Networking >

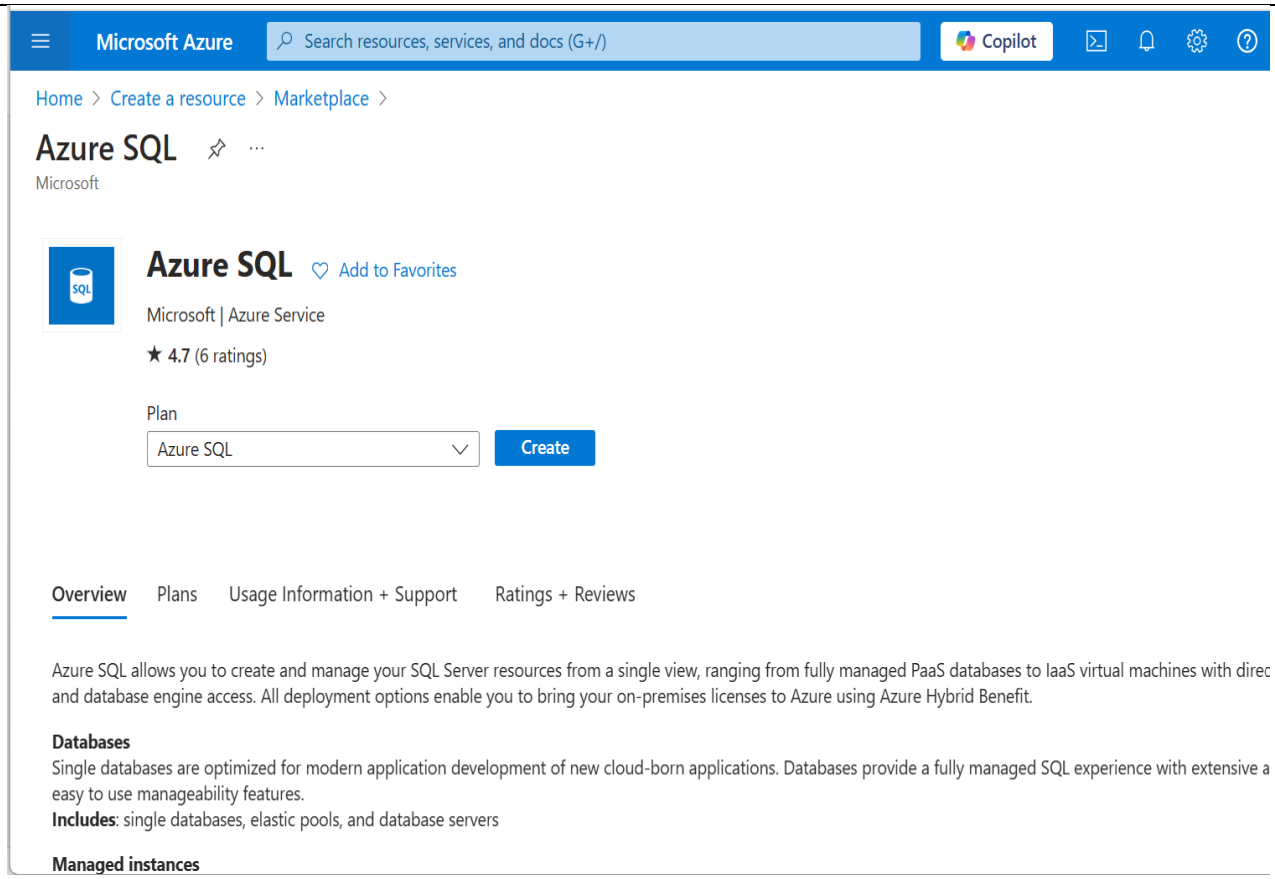


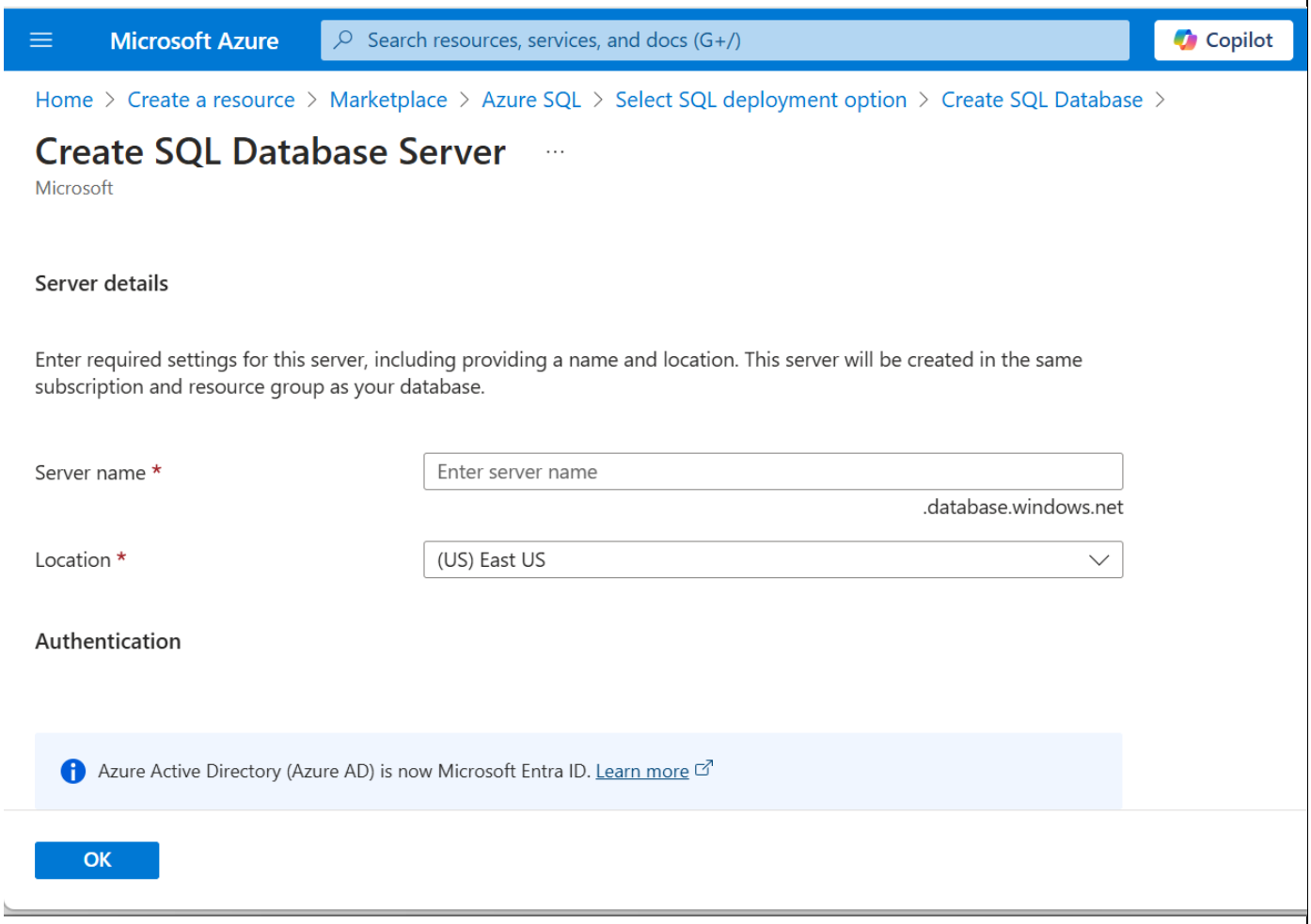




## +++ Azure_SQL_Server_Database_Setup

Create Azure SQL Server:

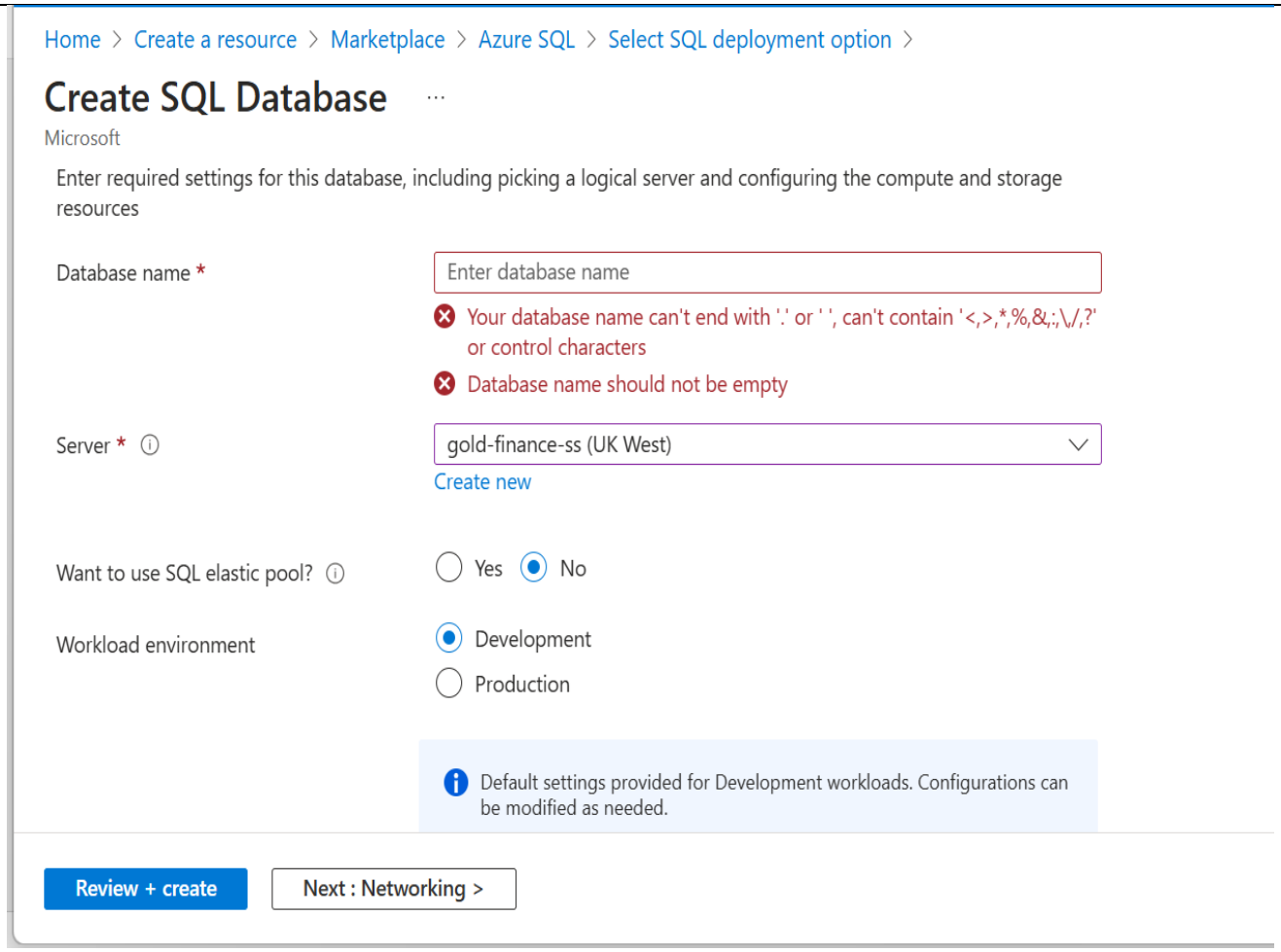
- Click on create a resource, search for azure sql server and select azure sql.




Create Azure SQL Database:

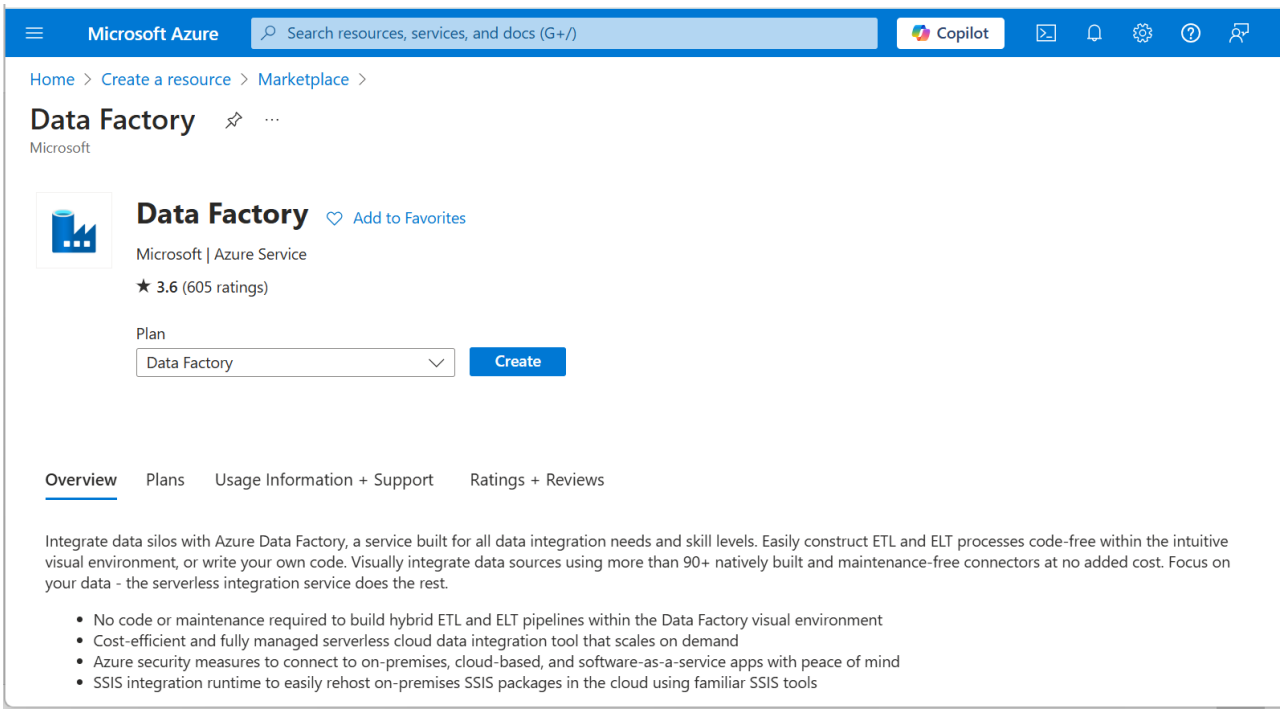
- Follow the prompt to create an sql database.

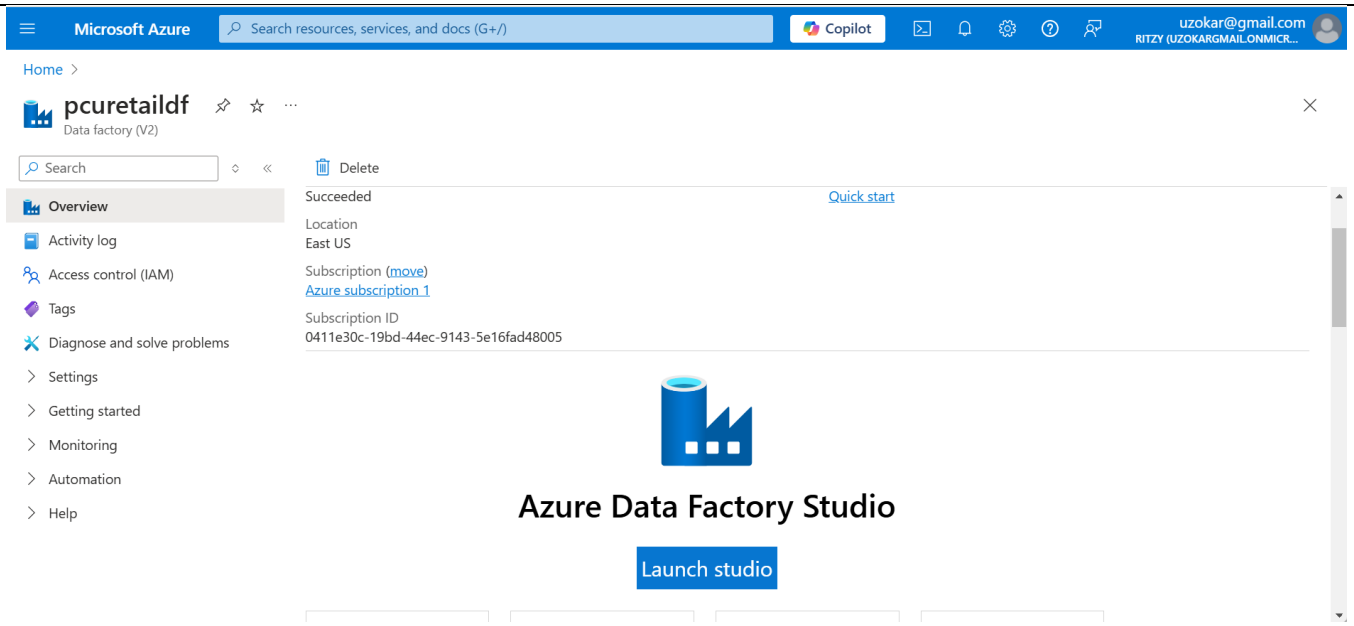



**+++ Setup Azure Data Factory**

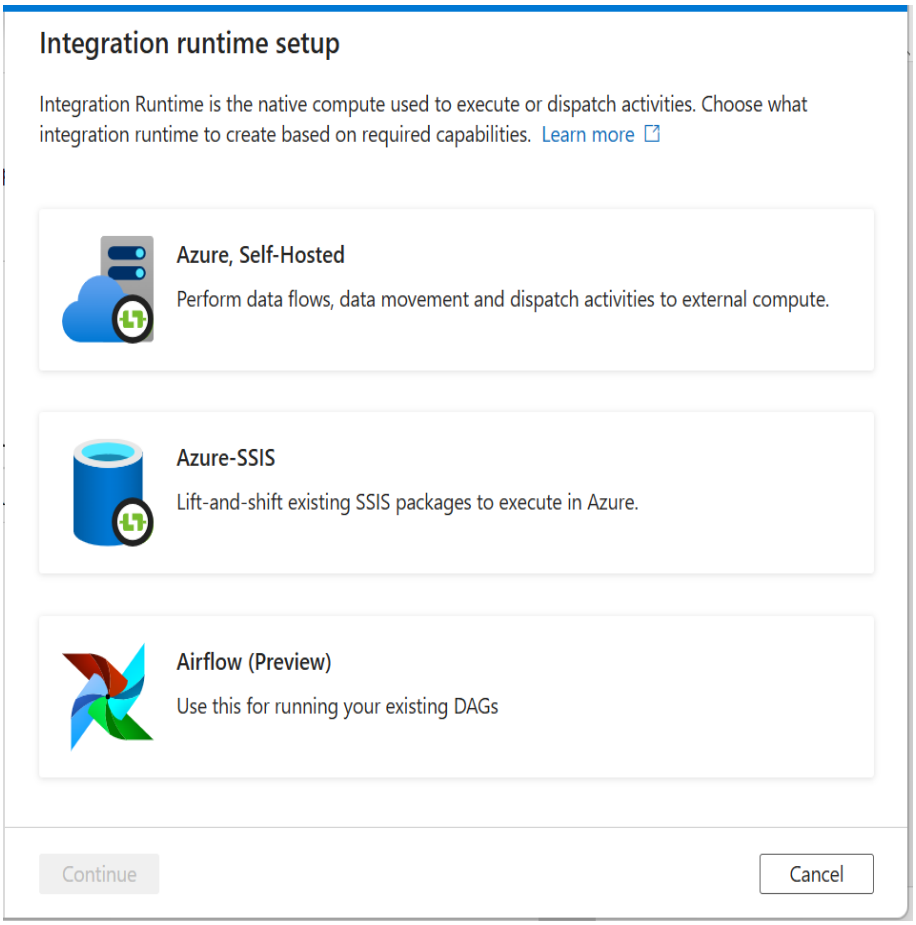
Create Azure Data Factory

- Click the select a resource, search with azure data factory and select azure data factory, create an azure data factory by following the prompts, open the azure data studio to launch the adf.

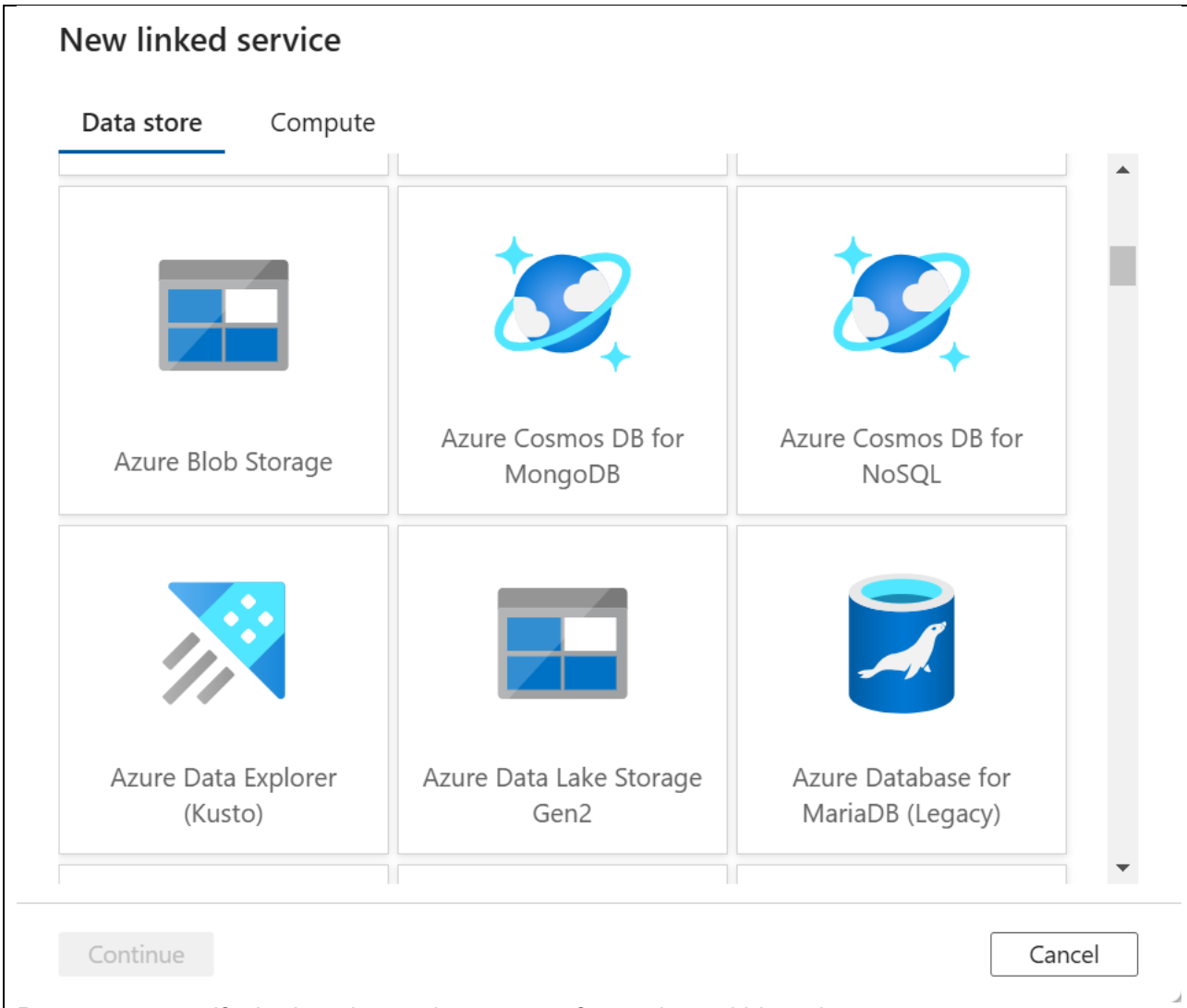





Integrated Runtime (IR):
- AutoResolve IR: To be used for all Azure services; the same IR can be used for all Azure services in a project and or multiple projects.
- Self-Hosted IR: To be used for onprem MySQL Server migration to Azure Cloud




Linked Service: defines the connection information to a data store or compute.




Dataset: to specify the location and structure of your data within a data store.

## New dataset

In pipeline activities and data flows, reference a dataset to specify the location and structure of your data within a data store. Learn more

Select a data store

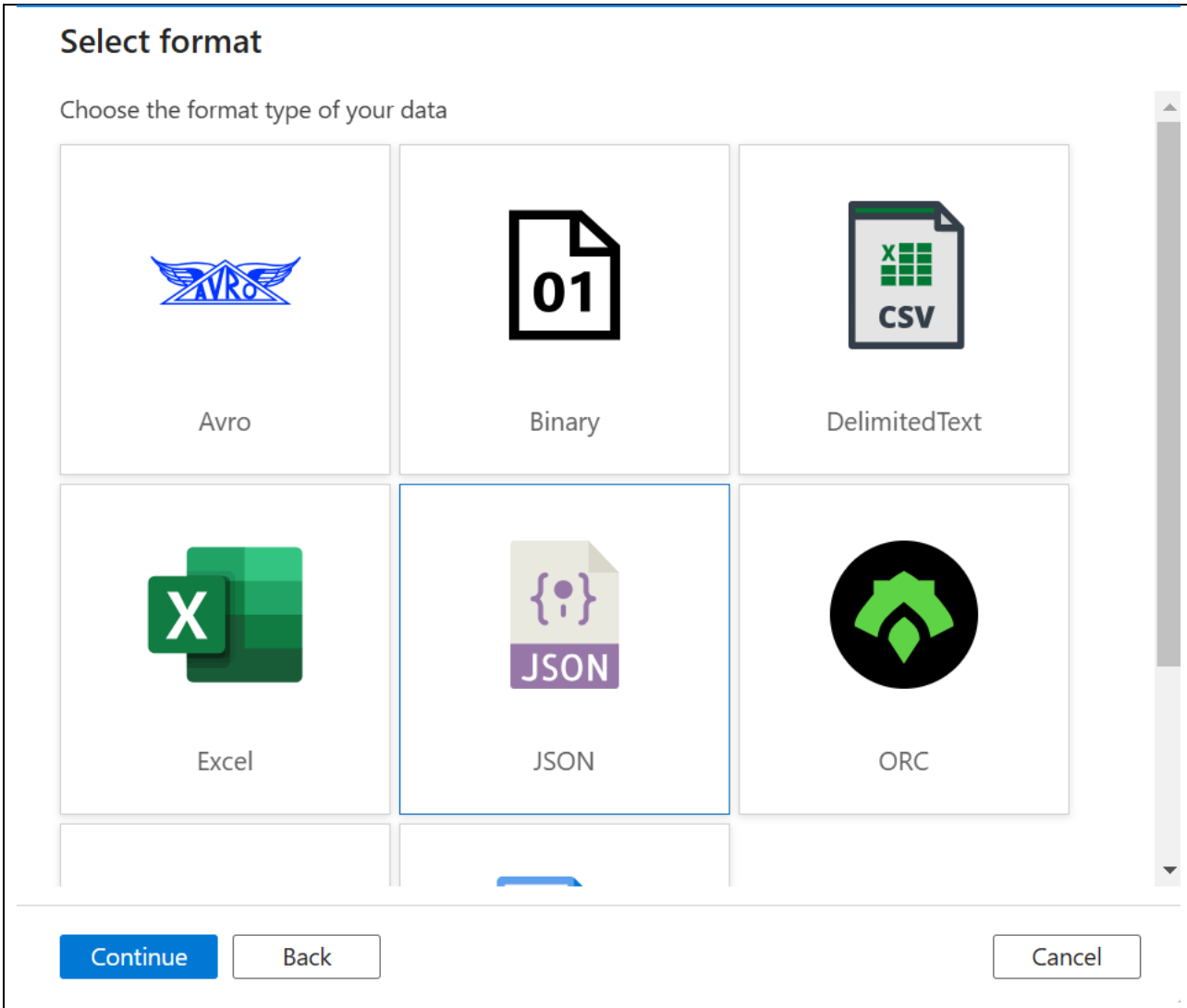
Search

All | Azure | Database | File | Generic protocol

Google BigQuery | Google Cloud Storage | Greenplum

HBase (Legacy) | HDFS | HTTP

Continue | Cancel

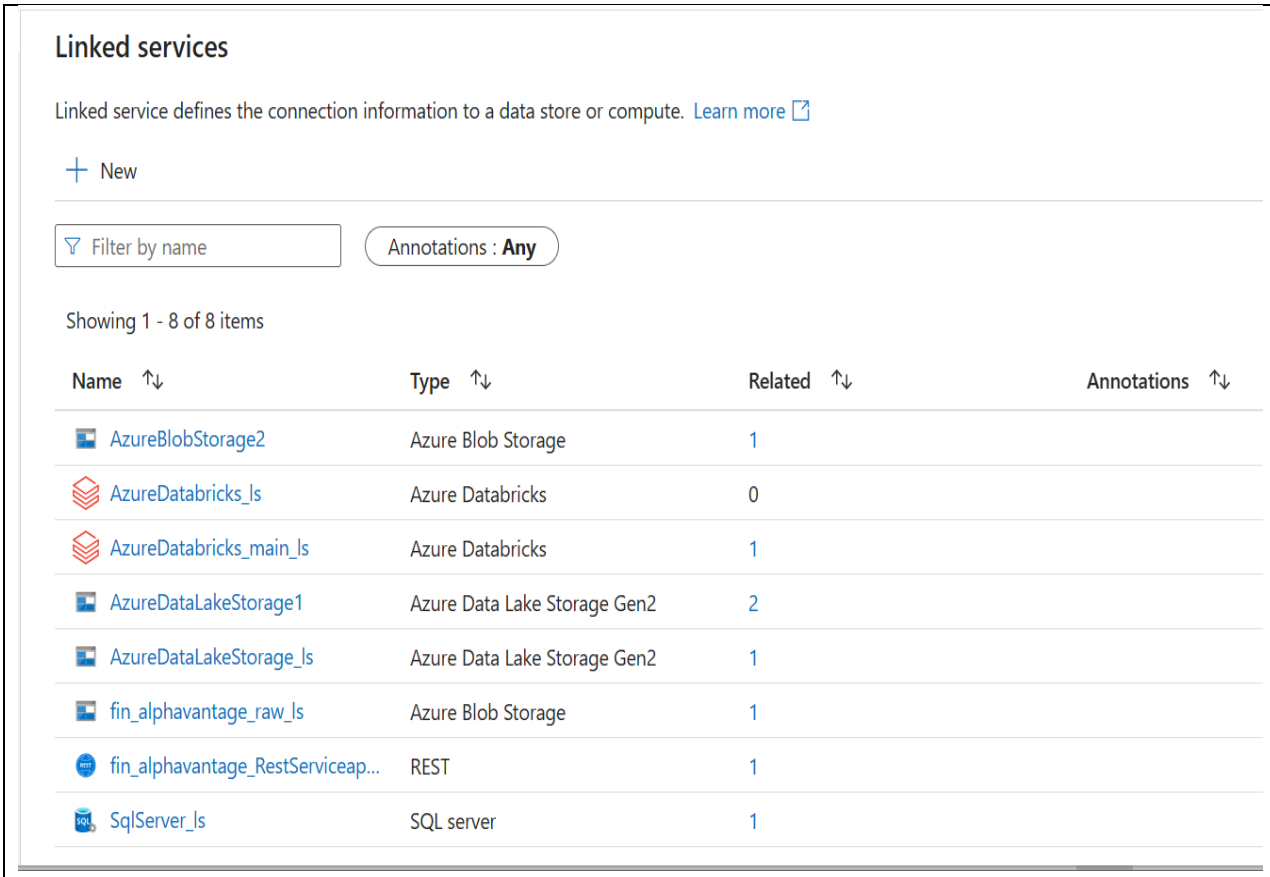


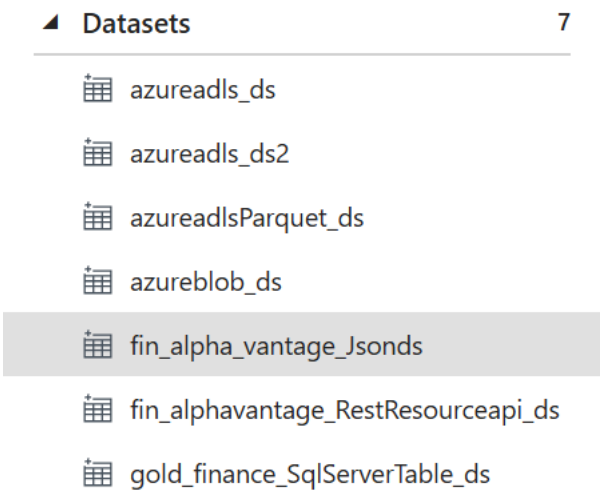

Create Linked Service for Azure Blob Storage:
Create Dataset for Azure blob Storage:

Create Linked Service for ADLS Gen 2:
Create Dataset for ADLS Gen 2:

Create Linked Service for Azure Databricks:
Create Dataset for Azure Databricks:

Create Linked Service for Azure SQL Database:
Create Dataset for Azure SQL Database:




++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

**Create_Azure_Data_Factory_Pipeline:**

**Each Time**

1) Create new Linked Service for each Azure service
2) Create new Dataset for each Azure service data format.

**Copy Activity**

The Copy Activity is used to move data from one source to a destination in the cloud.
Every Copy Activity has a source dataset and a sink dataset

**Ingestion Layer into Bronze-Layer (Azure Blob Storage)**

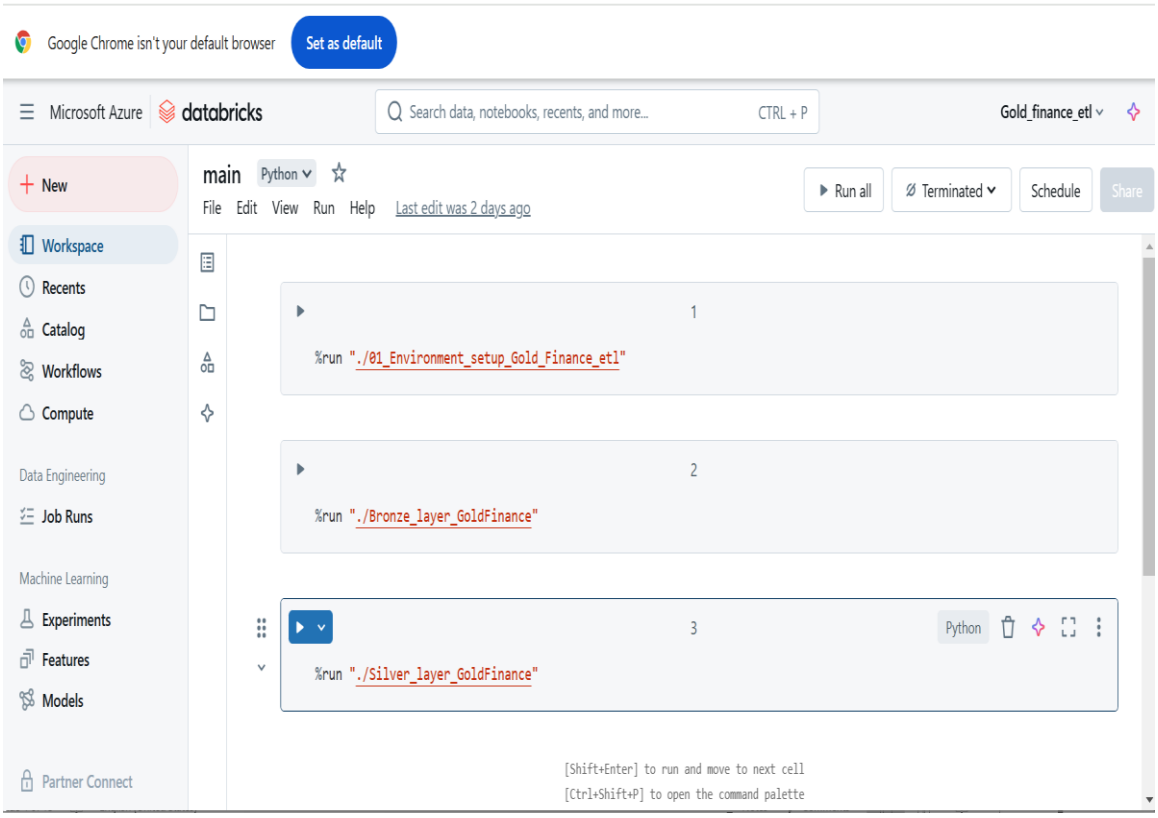
**Step 1:** Drag the copy activity to the pipeline workspace and configure the parameters

++Copy Activity Parameters: Select the respective datasets for the source and sink dataset. The copy Activity here ingests data from the Alpha vantage rest Api (Source dataset) to the azure blob storage (sink dataset)

**Extraction/Transformation Layer into Silver-Layer (Azure Databricks)**

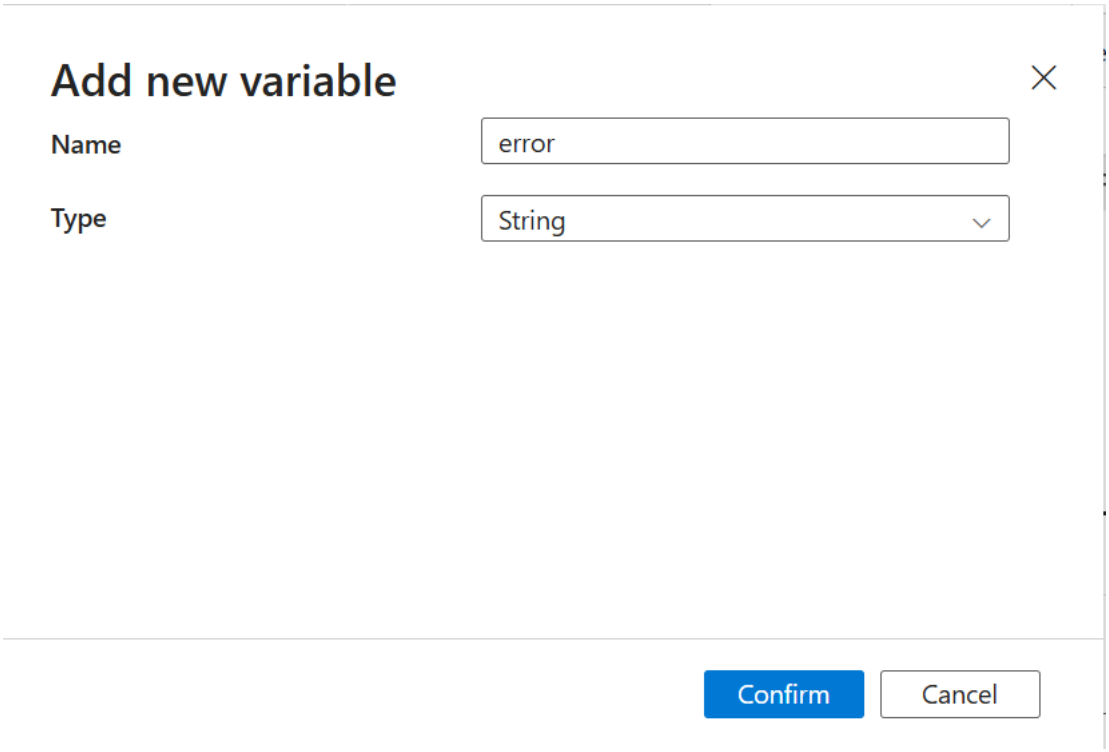
**Each Time**
1) Mount new data storage in databricks (Refer to Databricks Notebook 01: Environment Setup)
2) Update Azure Storage Details Configuration in Environment Setup & Bronze layer code

+++ Use modularized coding as seen below:



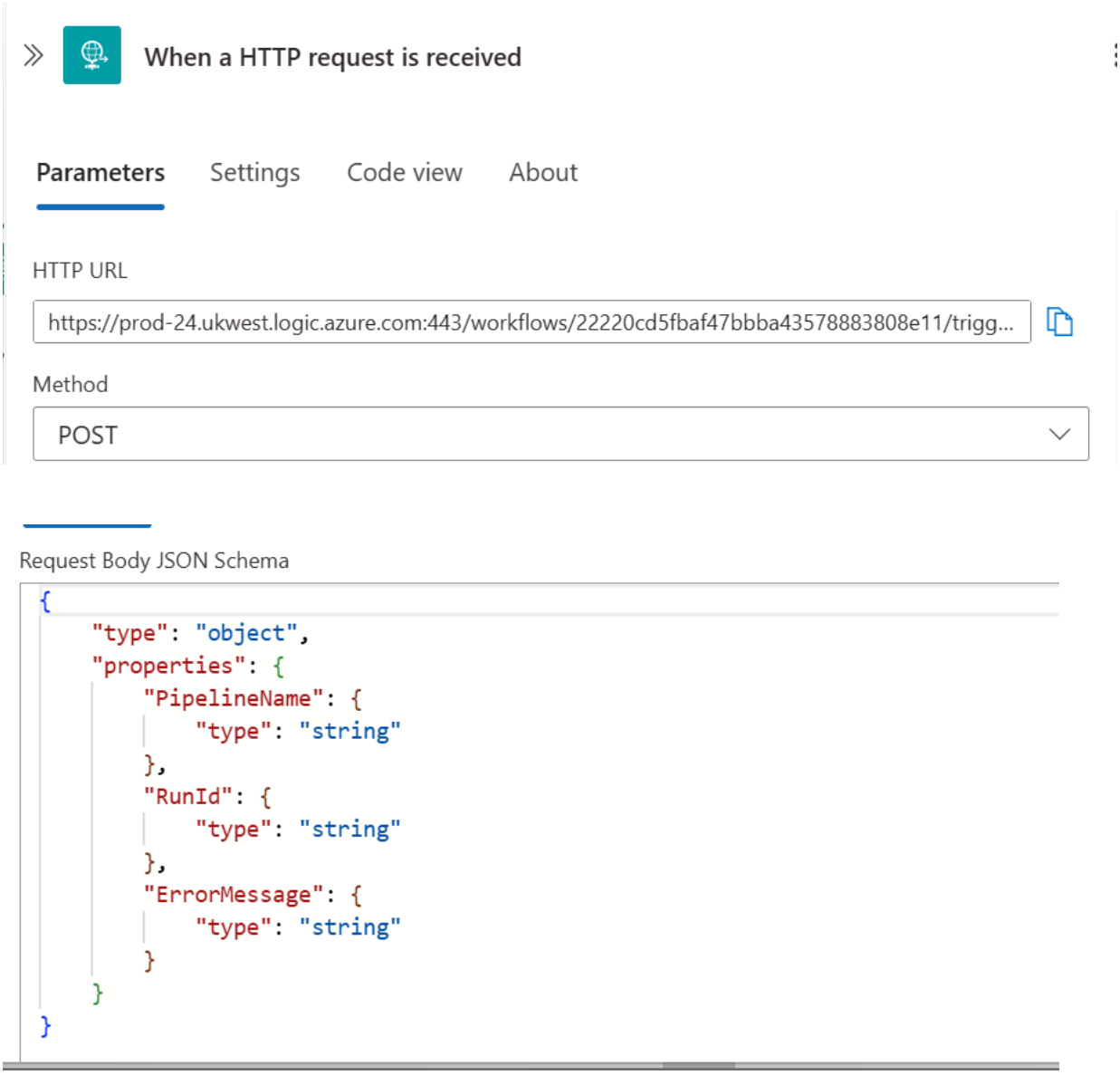

**Copy data from the Adls Gen 2 to Azure SQL**
The copy activity is used to copy data from the adls (source dataset) to azure sql database (sink dataset).

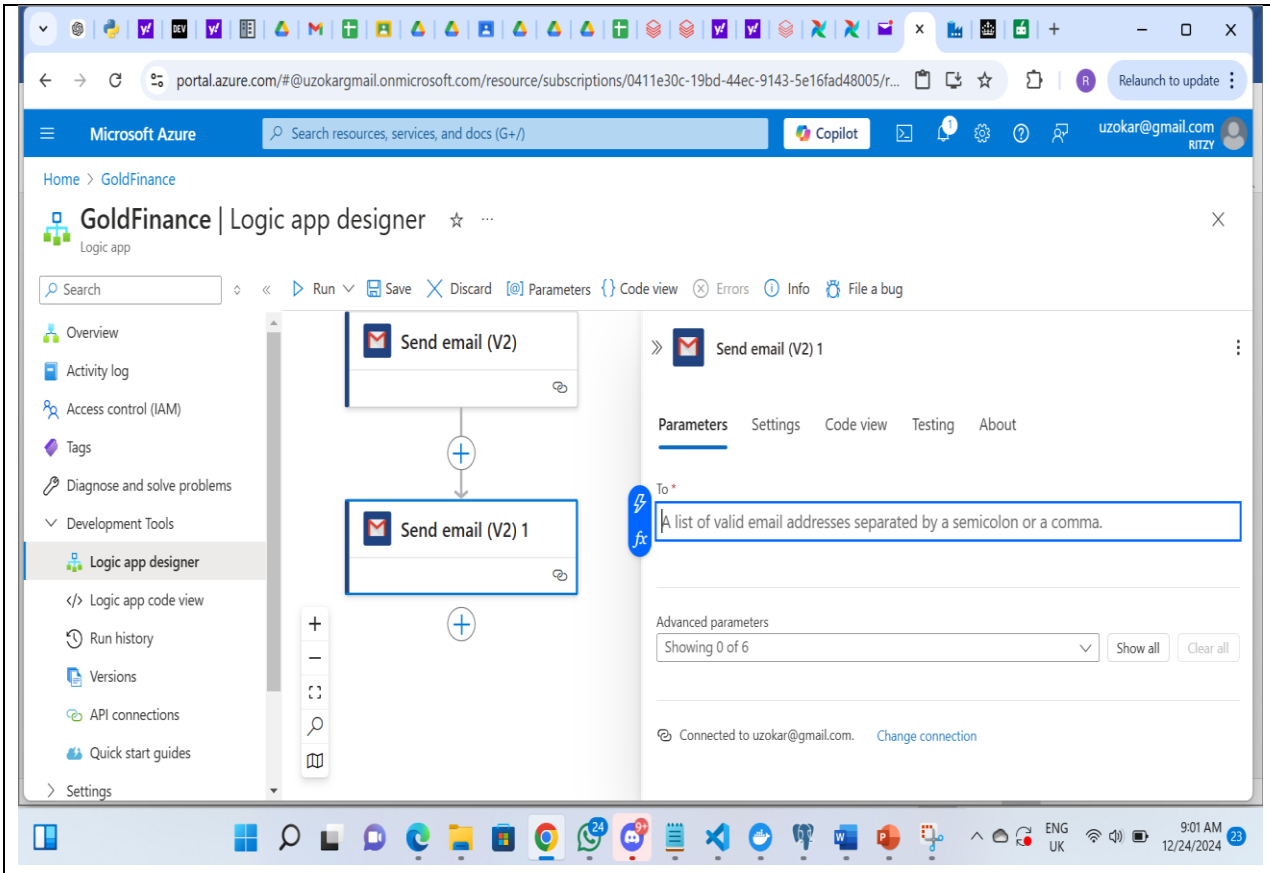
**+++ Error_Handling_Azure**
The Set Variable activity is used for error handling in adf.
First: define the variable error:

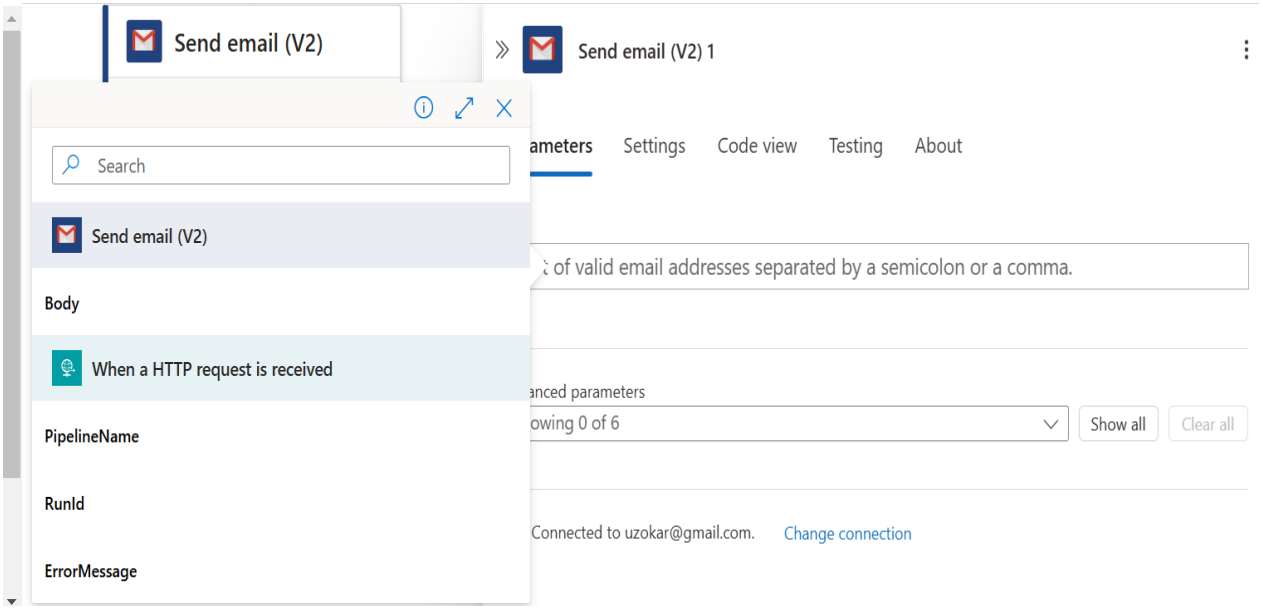


**+++ Error  Notification**

Logic Apps: Set up logic Apps, configure necessary email notification.





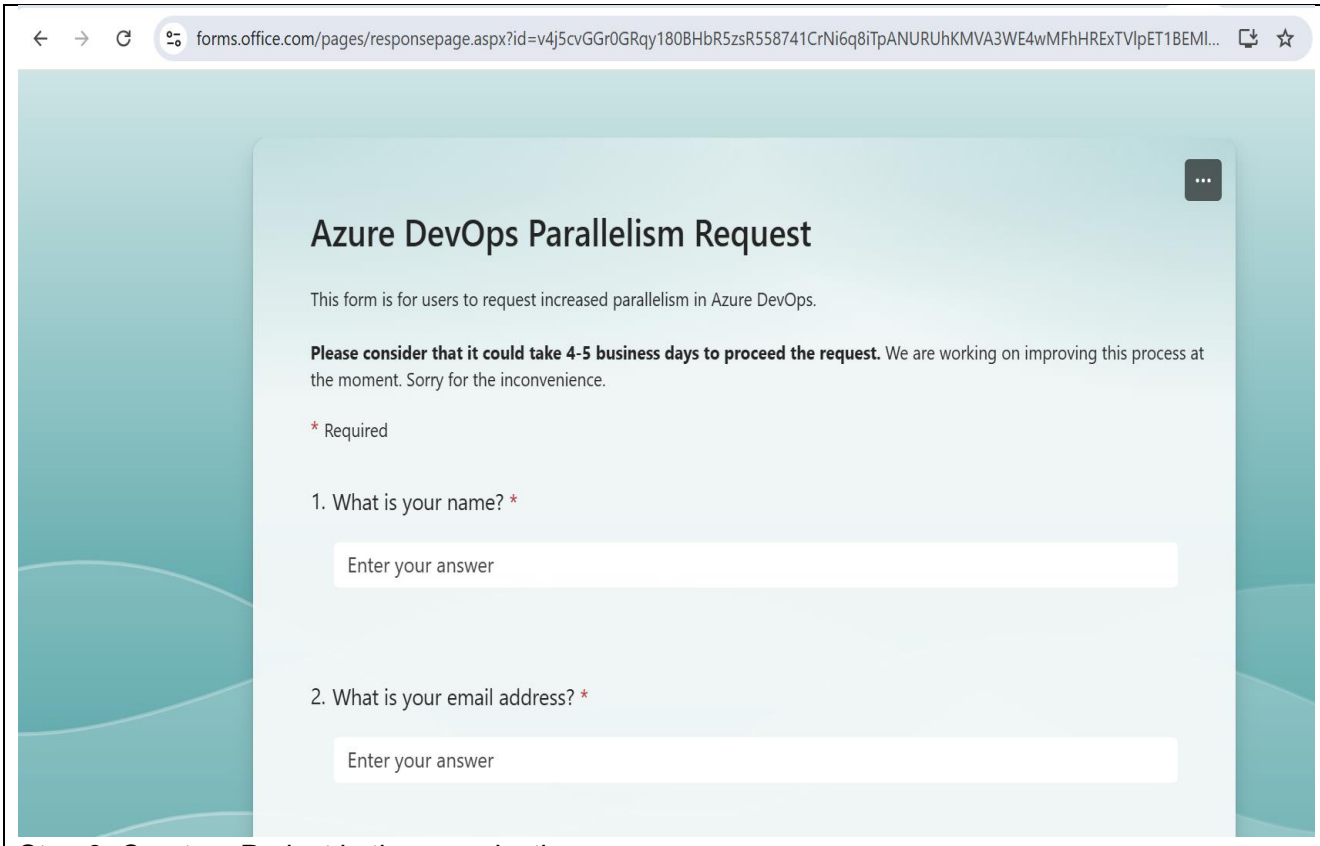




++ Web Activity:  Fill with Logic Apps parameters.

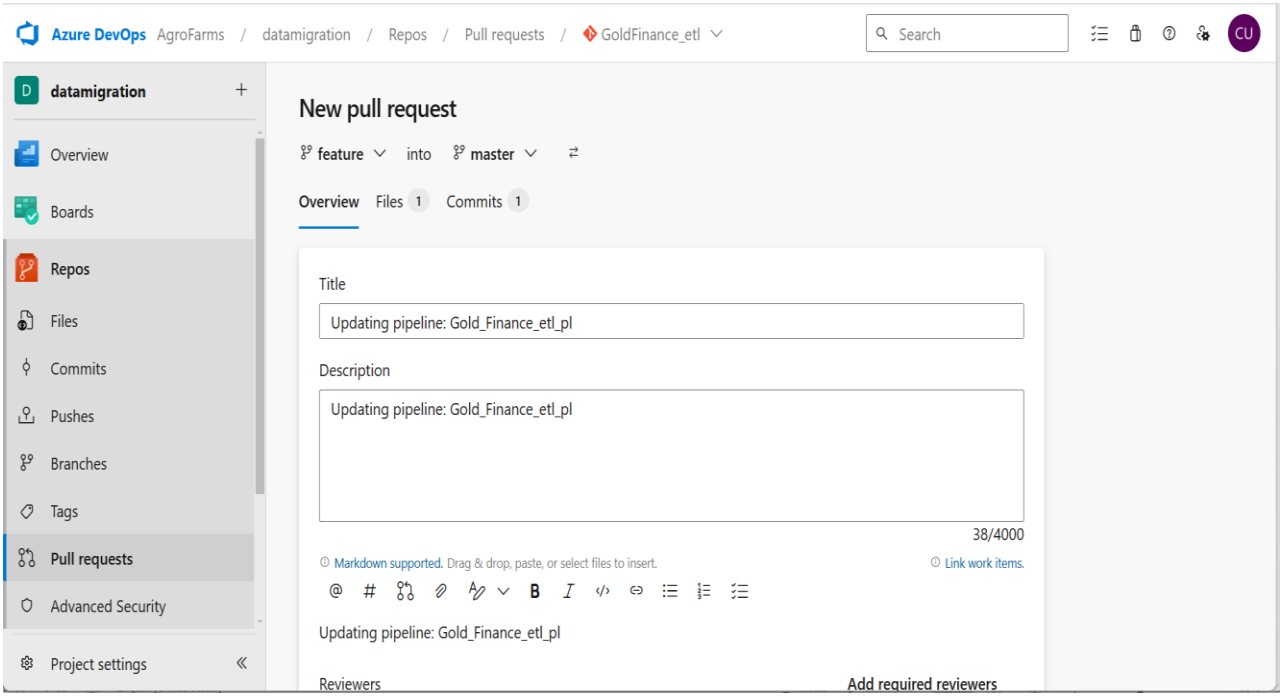
**+++ Azure  DevOPs  (CI/CD)**

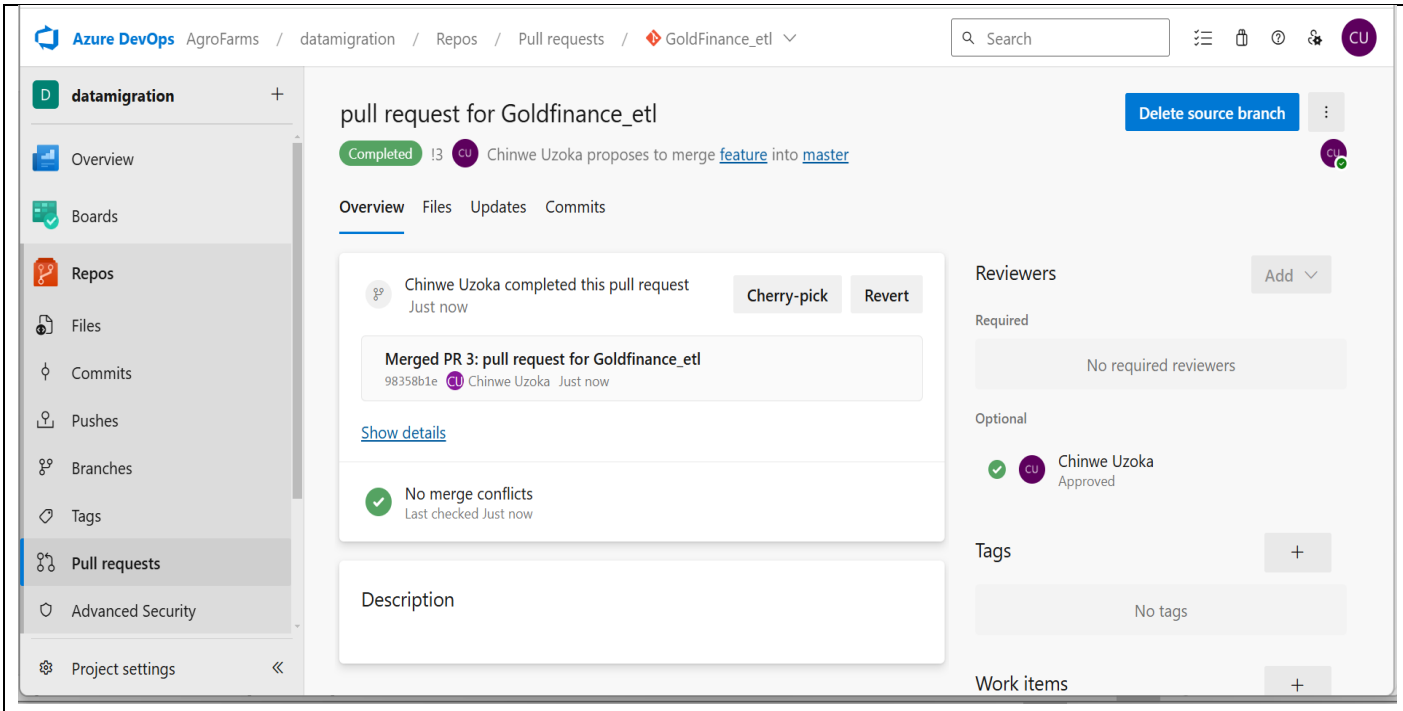
Step 1: Create an Azure DevOps Organization

Step 2: Fill this form to create a free parallelism request(https://aka.ms/azpipelines-parallelism-request) and submit for approval, it is only when it is approved that one can be able to work on the azure Organization.

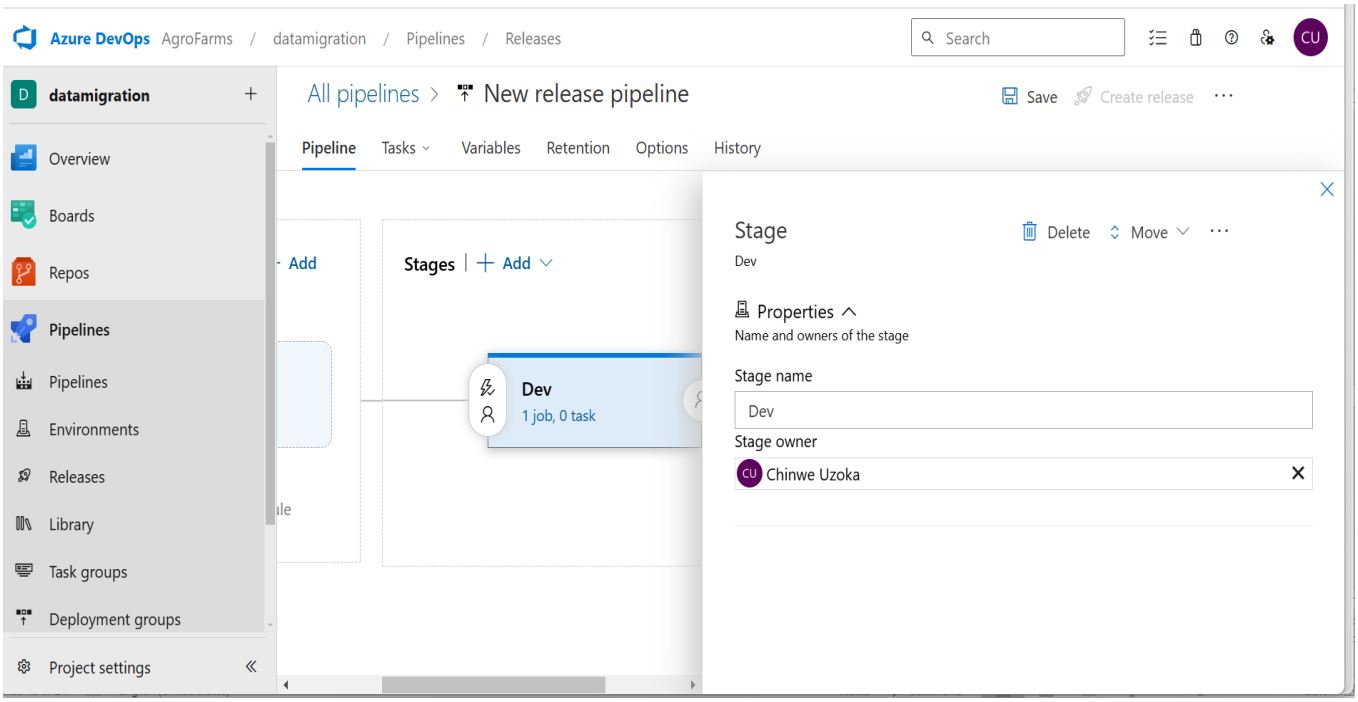


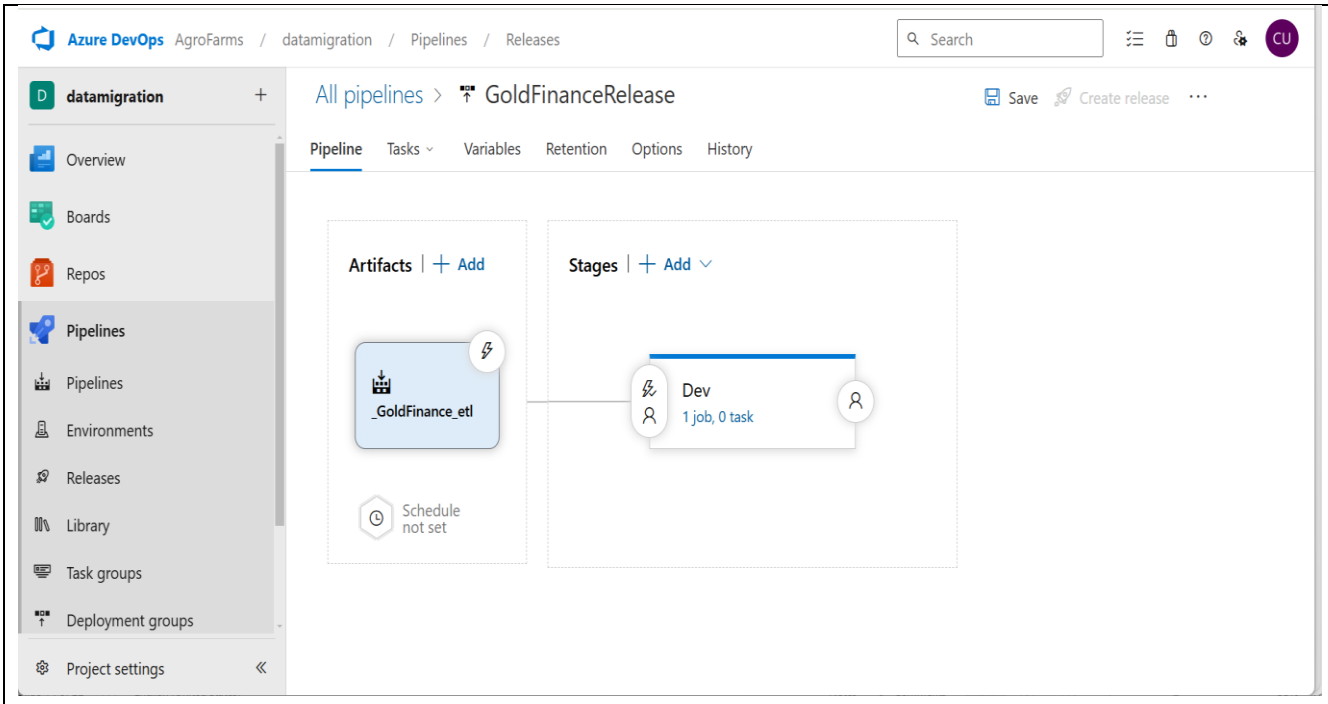

Step 3: Create a Project in the organization
Step 4: Create Repos for pull request and Merge to the main branch

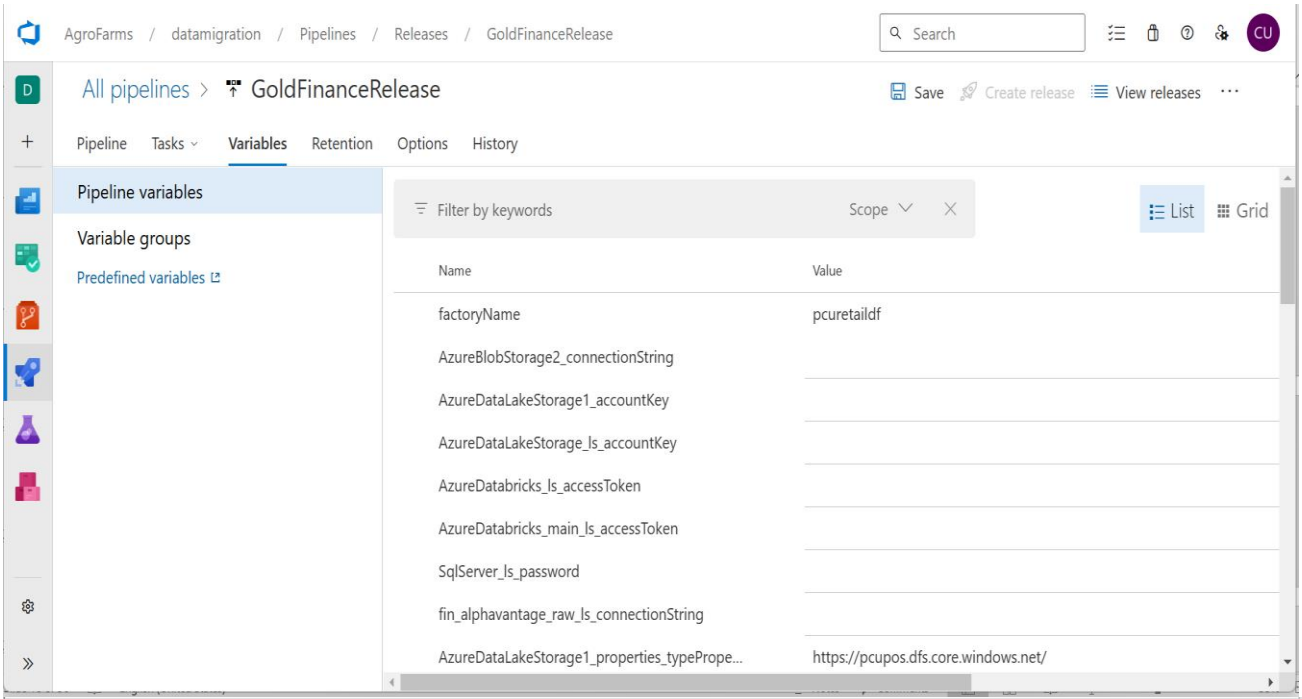


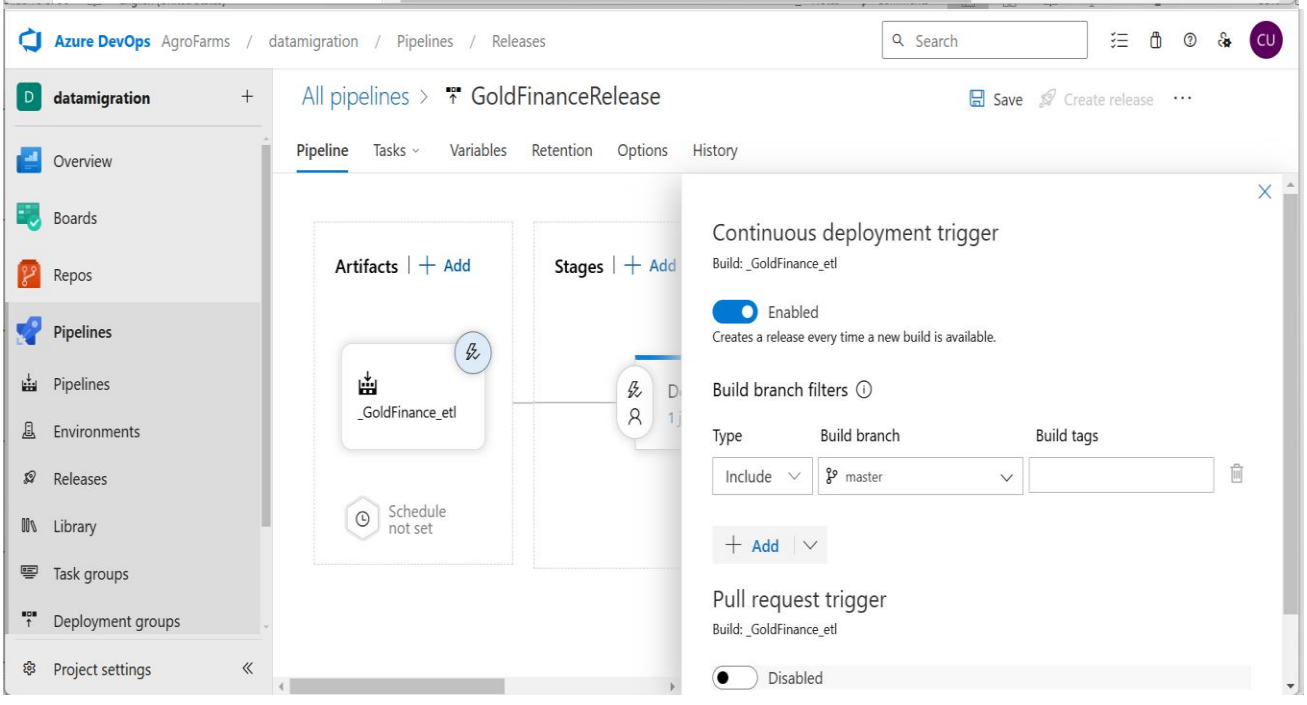


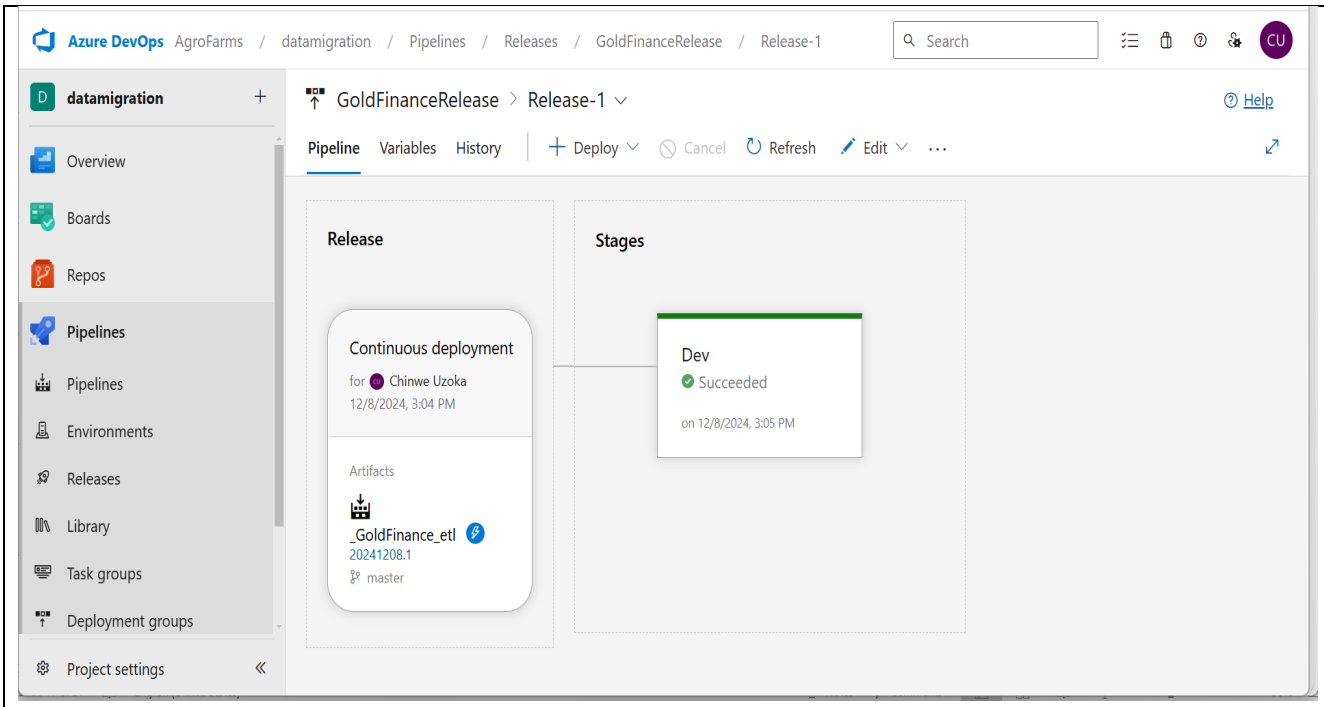

Step 4: adf_publish:1) Validate the pipeline 2) Generate Arm template and keep it in adf_publish branch 3) Deploy code to Adf Dev Live mode.
+++ Use the adf build.yml file to build the pipeline and generate an Arm template.

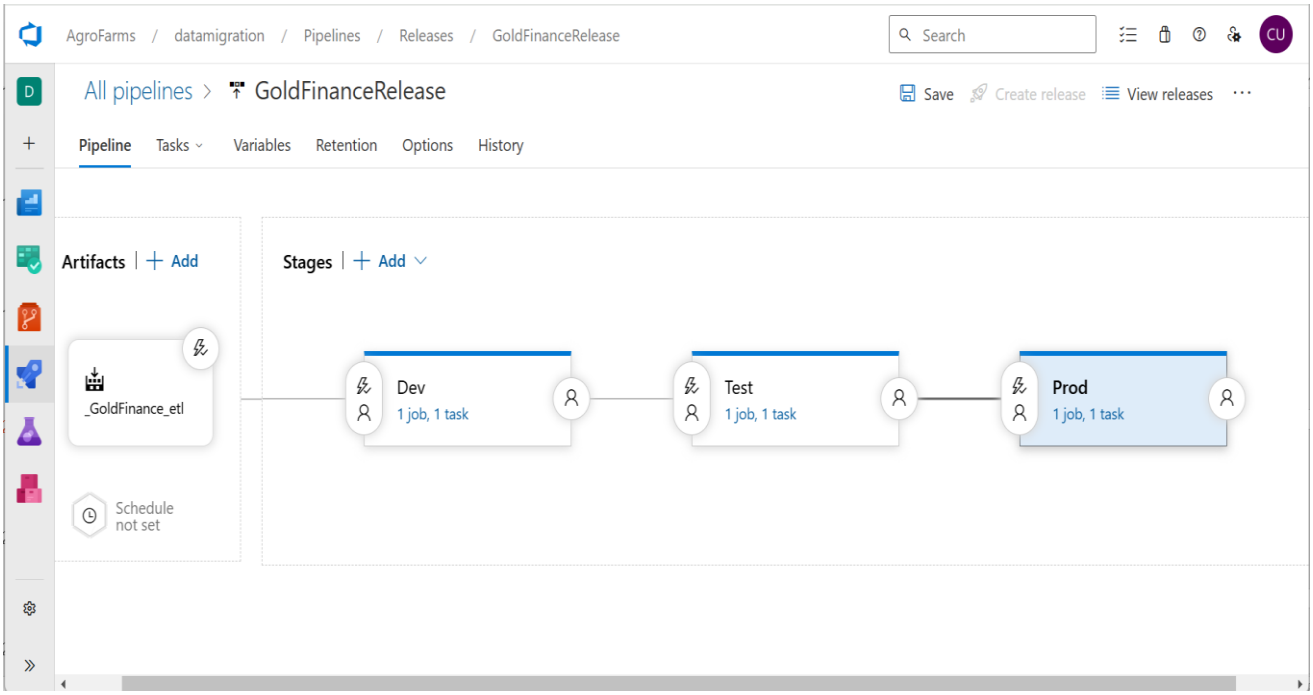



Azure DevOps AgroFarms / datamigration / Pipelines / Releases
Search
CU
datamigration
Overview
Boards
Repos
Pipelines
Pipelines
Environments
Releases
Library
Task groups
Deployment groups
Project settings
All pipelines > GoldFinanceRelease
Save
Create release
Pipeline
Tasks
Variables
Retention
Options
History
Artifacts | Add
Stages | Add
_GoldFinance_etl
Dev
1 job, 0 task
Schedule not set


AgroFarms / datamigration / Pipelines / Releases / GoldFinanceRelease
Search
CU
All pipelines > GoldFinanceRelease
Save
Create release
View releases
Pipeline
Tasks
Variables
Retention
Options
History
Pipeline variables
Variable groups
Predefined variables
Filter by keywords
Scope
List
Grid
Name
Value
factoryName
pcuretaildf
AzureBlobStorage2_connectionString
AzureDataLakeStorage1_accountKey
AzureDataLakeStorage_ls_accountKey
AzureDatabricks_ls_accessToken
AzureDatabricks_main_ls_accessToken
SqlServer_ls_password
fin_alphavantage_raw_ls_connectionString
AzureDataLakeStorage1_properties_typePrope...
https://pcupos.dfs.core.windows.net/


Azure DevOps AgroFarms / datamigration / Pipelines / Releases
Search
CU
datamigration
Overview
Boards
Repos
Pipelines
Pipelines
Environments
Releases
Library
Task groups
Deployment groups
Project settings
All pipelines > GoldFinanceRelease
Save
Create release
Pipeline
Tasks
Variables
Retention
Options
History
Artifacts | Add
Stages | Add
_GoldFinance_etl
Schedule not set
Continuous deployment trigger
Build: _GoldFinance_etl
Enabled
Creates a release every time a new build is available.
Build branch filters
Type
Build branch
Build tags
Include
master
Add
Pull request trigger
Build: _GoldFinance_etl
Disabled



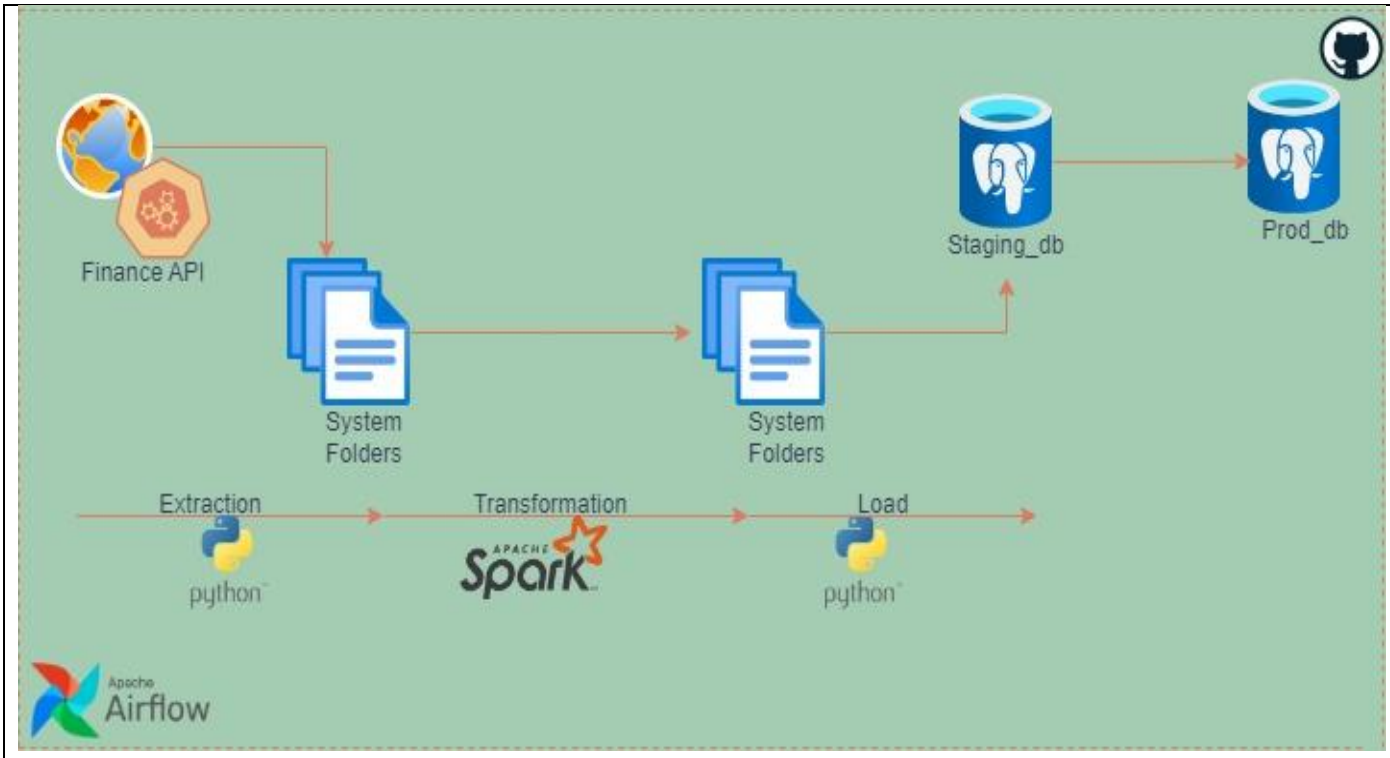




---

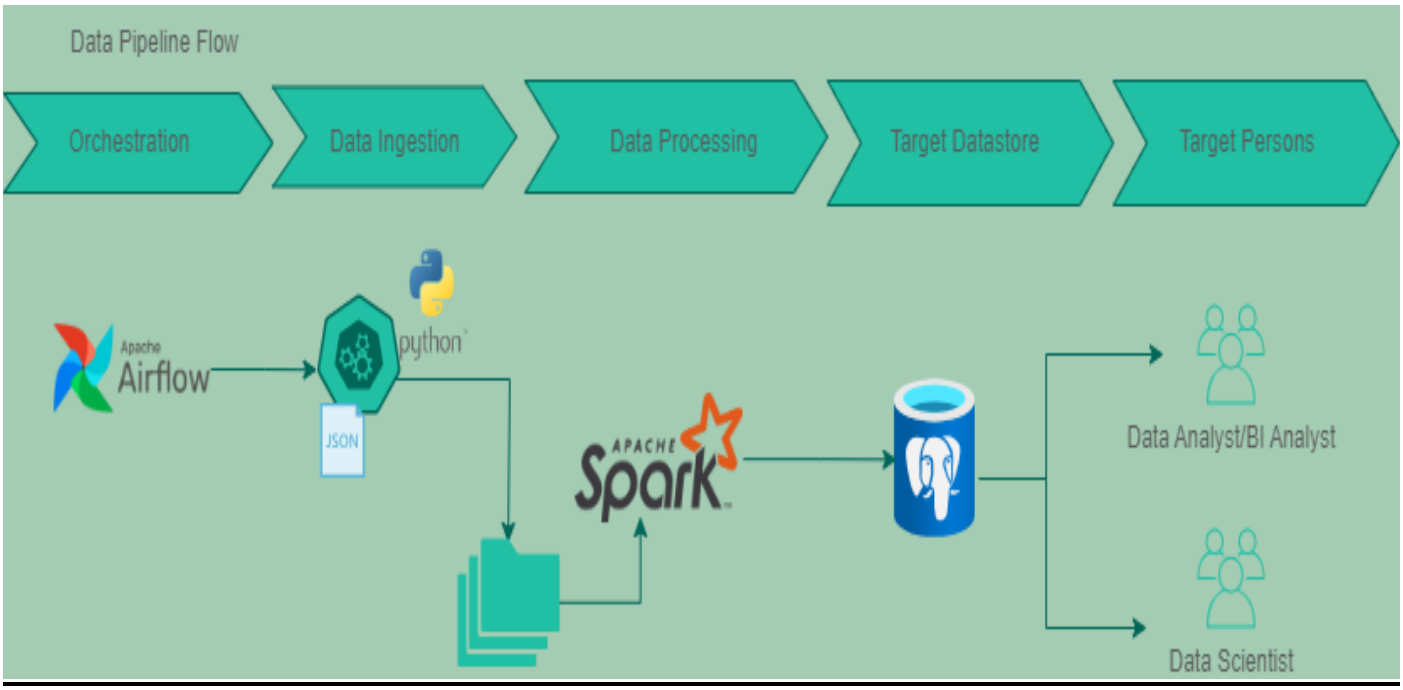
2.0 Onprem Project: Gold_Finance ETL

---

**Onprem_Solution_Architecture**




**Onprem_Data_Pipeline_Flow**




**Onprem_Project_Tech_Stack**

| S/N | Tech Stack | Details |
|---|---|---|
| 1 | Vscode | Python programming |
| 2 | Pyspark on Vscode | Transformation |
| 3 | Postgres | Database |
| 4 | Airflow | Orchestration |

**+++ Pyspark_Vscode**

- **Purpose**: Run Spark and custom Python scripts for data ingestion and transformation.
- **Aim**:
  - Writing Spark jobs to process large-scale datasets.

- Using Python scripts to ingest, transform and load data to PostgreSQL.

**+++PostgreSQL**

- **Purpose**: A relational database for storing metadata, structured data, and managing transactions.
- **Aim**:
  - Querying relational datasets using SQL.

Storing and retrieving structured data for analysis or visualization

**+++ Airflow**

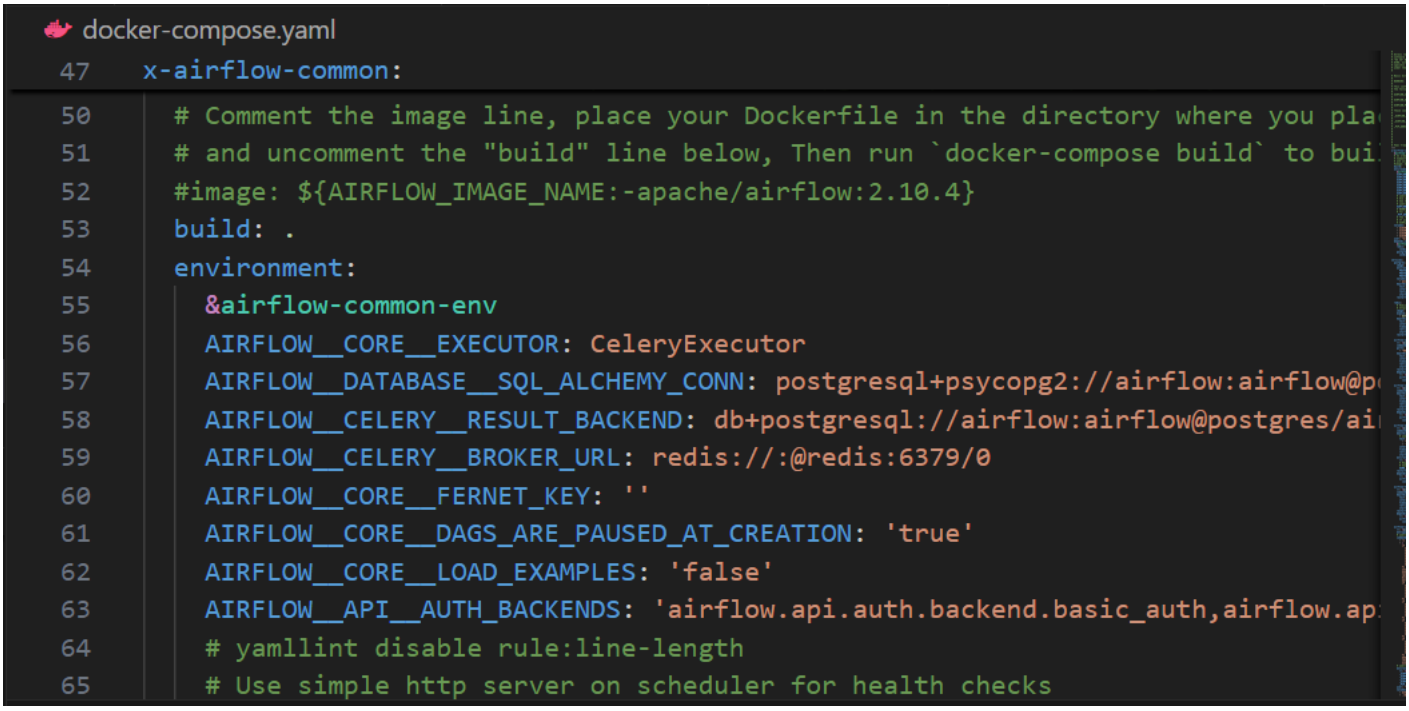
- **Purpose**: Workflow orchestration and ETL pipeline management.
- **Aim:**
  - Building and scheduling DAGs (Directed Acyclic Graphs) to automate workflows.
  - Managing and monitoring pipelines through the Airflow web interface.

- **Web Interface URL**: http://localhost:8080

**Step 1:** On vscode, open a bash terminal and paste the code below

curl -LfO 'https://airflow.apache.org/docs/apache-airflow/2.10.4/docker-compose.yaml'

Adjust the yaml file as per below screenshot




Step 2: Create a Dockerfile and a requirements.txt file.

Step 3: Paste the below in the bash terminal

mkdir -p ./dags ./logs ./plugins ./config

echo -e "AIRFLOW_UID=$(id -u)" > .env

Step 4: Create scripts as per below structure

```
myenv/
├── dags/
│   ├── etl_pipeline_dag.py
│   ├── modules/
│   │   ├── __init__.py
│   │   ├── extract.py
│   │   ├── transform.py
│   │   ├── load.py
│   │   ├── helpers.py
│   ├── configs/
│   │   ├── dev_config.json
│   │   ├── prod_config.json
│
├── tests/
│   ├── test_extract.py
│   ├── test_transform.py
│   ├── test_load.py
```

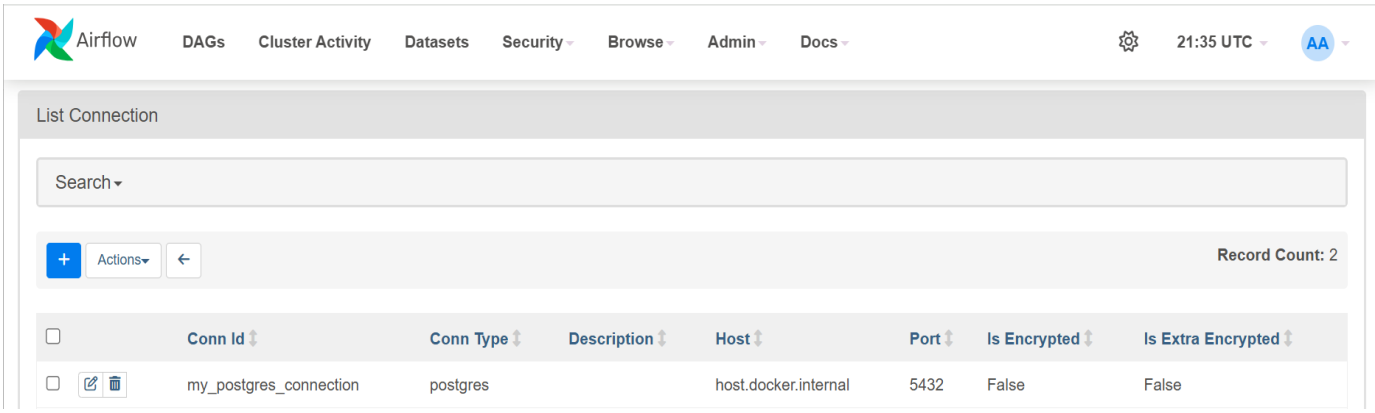
https://airflow.apache.org/docs/apache-airflow/stable/howto/docker-compose/index.html#fetching-docker-compose-yaml

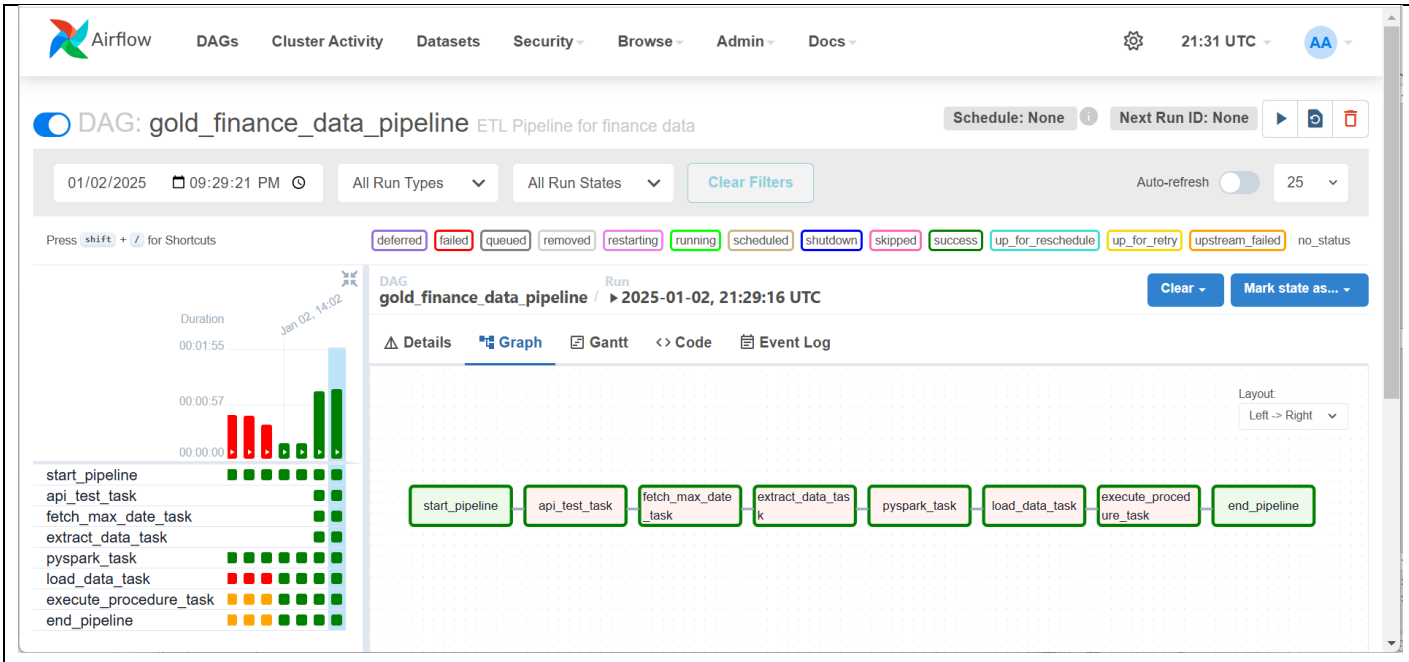
Step 5: Run the below command on bash terminal

docker compose up airflow-init

docker build .

docker compose up






## Linux Commands

rita@DESKTOP-1F00B3O:~/Finance_ETL/myenv$ ls -ld ./dags/tmp/raw

drwxrwxrwx 2 rita root 4096 Dec 28 17:54 ./dags/tmp/raw

rita@DESKTOP-1F00B3O:~/Finance_ETL/myenv$ ls -ld ./dags/tmp/transformed_data

drwxr-xr-x 2 rita root 4096 Dec 28 17:55 ./dags/tmp/transformed_data

rita@DESKTOP-1F00B3O:~/Finance_ETL/myenv$ chmod 777 ./dags/tmp/transformed_data

rita@DESKTOP-1F00B3O:~/Finance_ETL/myenv$ ls -ld ./dags/tmp/transformed_data

drwxrwxrwx 2 rita root 4096 Dec 28 17:55 ./dags/tmp/transformed_data

rita@DESKTOP-1F00B3O:~/Finance_ETL/myenv$ docker exec -u root -it myenv-airflow-worker-1 /bin/bash

root@77bcd09e75b5:/app# ls -ld /app/dags/tmp/raw

drwxrwxrwx 1 airflow root 4096 Dec 28 17:54 /app/dags/tmp/raw

rita@DESKTOP-1F00B3O:~/Finance_ETL/myenv$ docker ps

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

3673140a474c myenv-airflow-scheduler "/usr/bin/dumb-init …" 14 hours ago Up 9 minutes (healthy) 8080/tcp myenv-airflow-scheduler-1

77bcd09e75b5 myenv-airflow-worker "/usr/bin/dumb-init …" 14 hours ago Up 9 minutes (healthy) 8080/tcp myenv-airflow-worker-1

e89d6ad6b639 myenv-airflow-webserver "/usr/bin/dumb-init …" 14 hours ago Up 9 minutes (healthy) 0.0.0.0:8080->8080/tcp myenv-airflow-webserver

b979ae8ff5c3 myenv-airflow-triggerer "/usr/bin/dumb-init …" 14 hours ago Up 9 minutes (healthy) 8080/tcp myenv-airflow-triggerer-1

a03182e4283a redis:7.2-bookworm "docker-entrypoint.s…" 4 days ago Up 10 minutes (healthy) 6379/tcp myenv-redis-1

5b6d7ef142a6 postgres:13 "docker-entrypoint.s…" 4 days ago Up 10 minutes (healthy) 5432/tcp myenv-postgres-1

(myenv) rita@DESKTOP-1F00B3O:/tmp$ mkdir -p /tmp/spark_temp,

sudo chmod 777 /tmp/spark_temp

---

Project: Gold_Finance ETL: End

---

Team Alpha