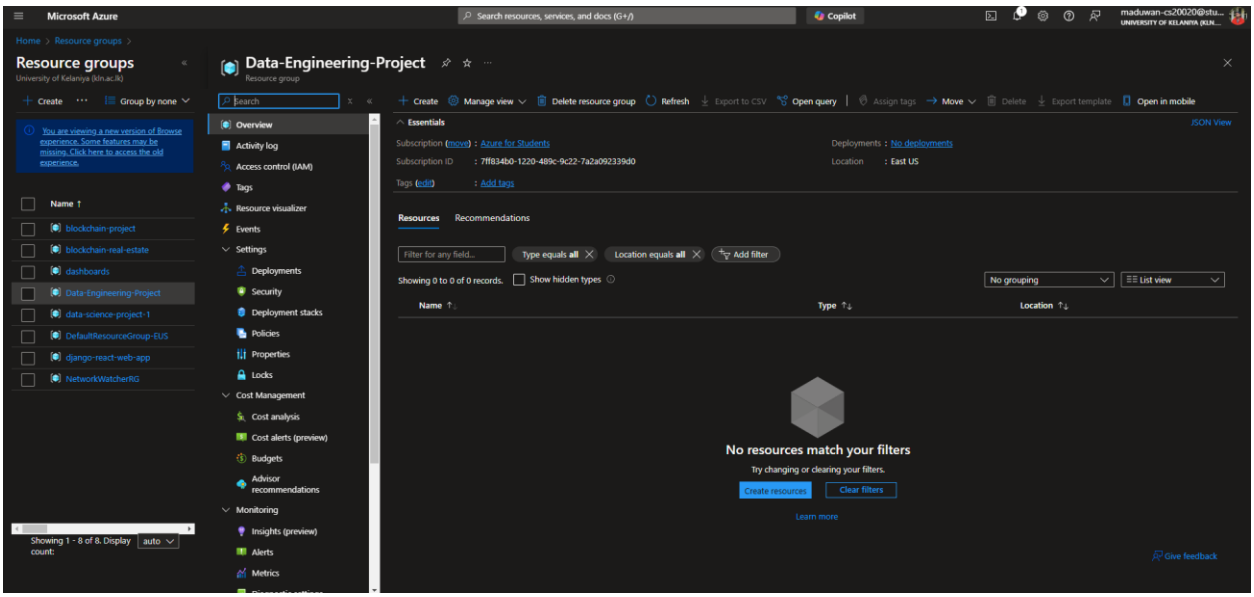
Phase 1

Why use this dataset?

Because we can work with several datasets and combine these datasets.

1. Create resource group



2. Create resources

2.1 Create Storage Account (Data Lake)

An Azure storage account contains all of your Azure Storage data objects: blobs, files, queues, and tables.

The storage account provides a unique namespace for your Azure Storage data that is accessible from anywhere in the world over HTTP or HTTPS.

Azure Blob Storage vs. Data Lake

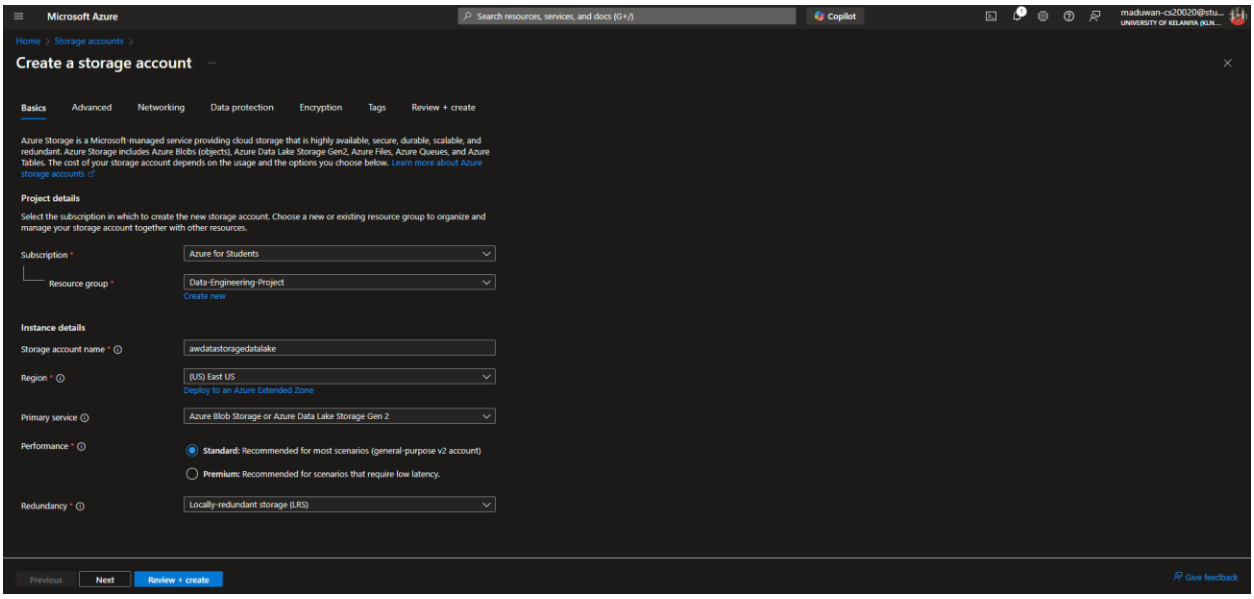
Azure Blob Storage is one of the most common Azure storage types. It's an object storage service for workloads that need high-capacity storage. Azure Data Lake is a storage service intended primarily for big data analytics workloads.

Blob Storage (binary large object) -- is ideal for large amounts of **unstructured data**, such as text, videos, photos, application back-end data and backup data.

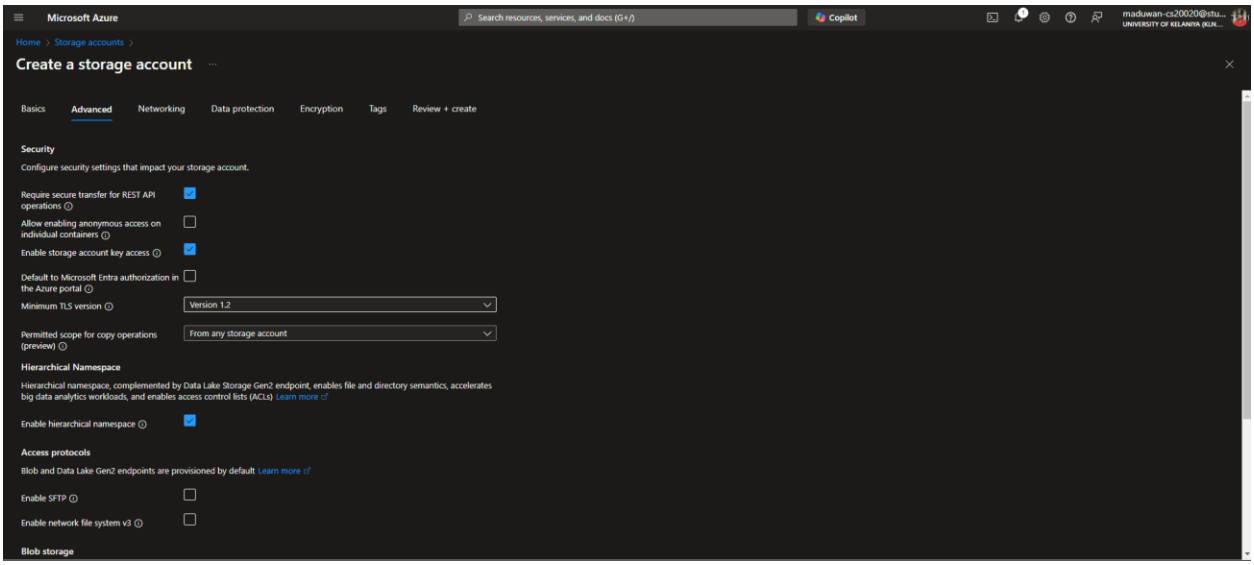
Azure Data Lake storage is currently separated into Gen1 and Gen2 options. Microsoft will retire Data Lake Gen1 storage in February 2024, and all customers using it must migrate to Gen2 before this date.

Azure Data Lake Gen1 is a storage service that's optimized for big data analytics workloads. Its **hierarchical file system** can store machine learning data, including log files, as well as interactive streaming analytics. It is performance-tuned to run large-scale analytics systems that require massive throughput and bandwidth to query and analyze large amounts of data.

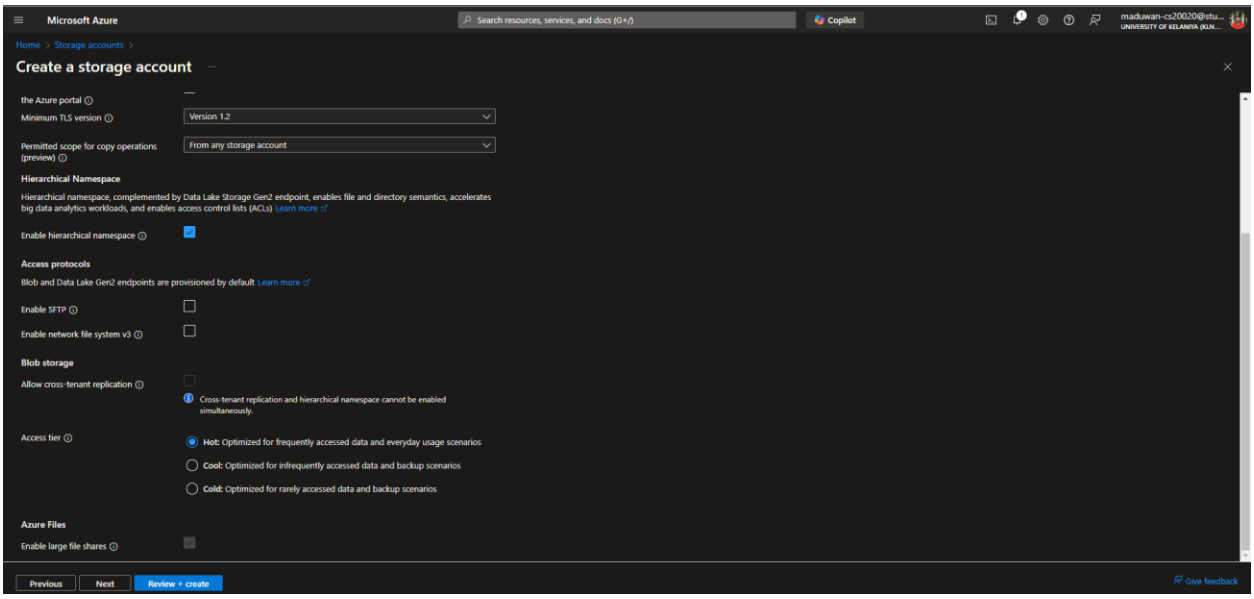
Azure Data Lake Gen2 converges the features and capabilities of Data Lake Gen1 with Blob Storage. It inherits the file system semantics, file-level security and scaling features of Gen1 and builds them on Blob Storage. This results in a low-cost, tiered-access, high-security and high availability big data storage option.

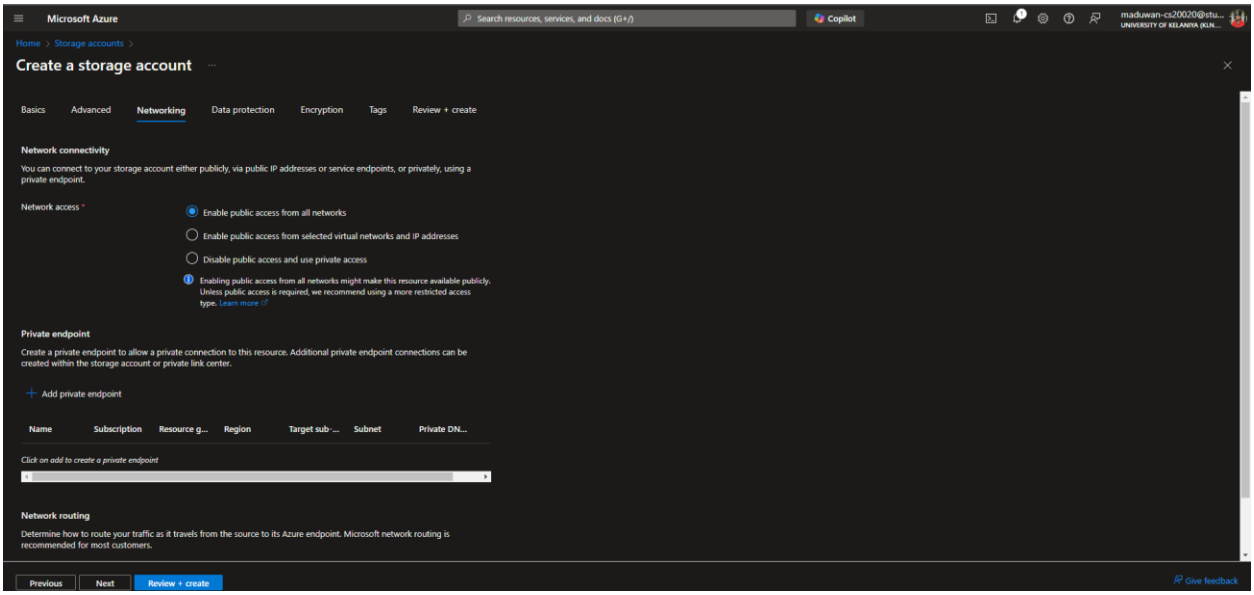


Click Next button to create Data Lake. Other wise it will create a Blob Storage.



Put the tick on “Enable Hierarchical Namespace”



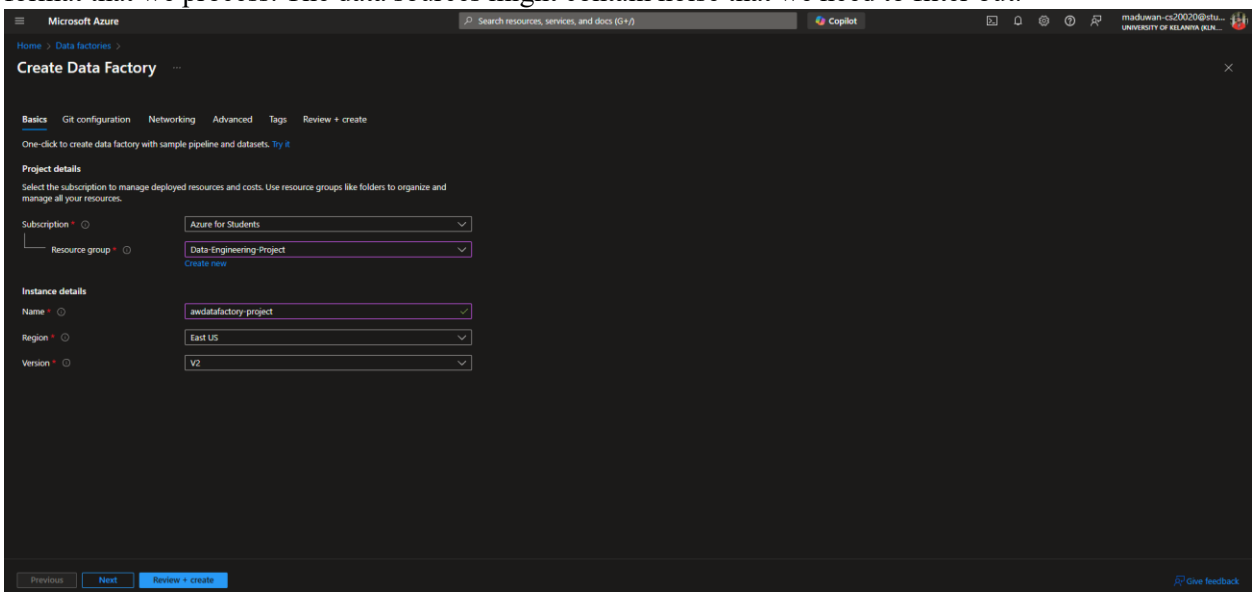


Click “review + create”

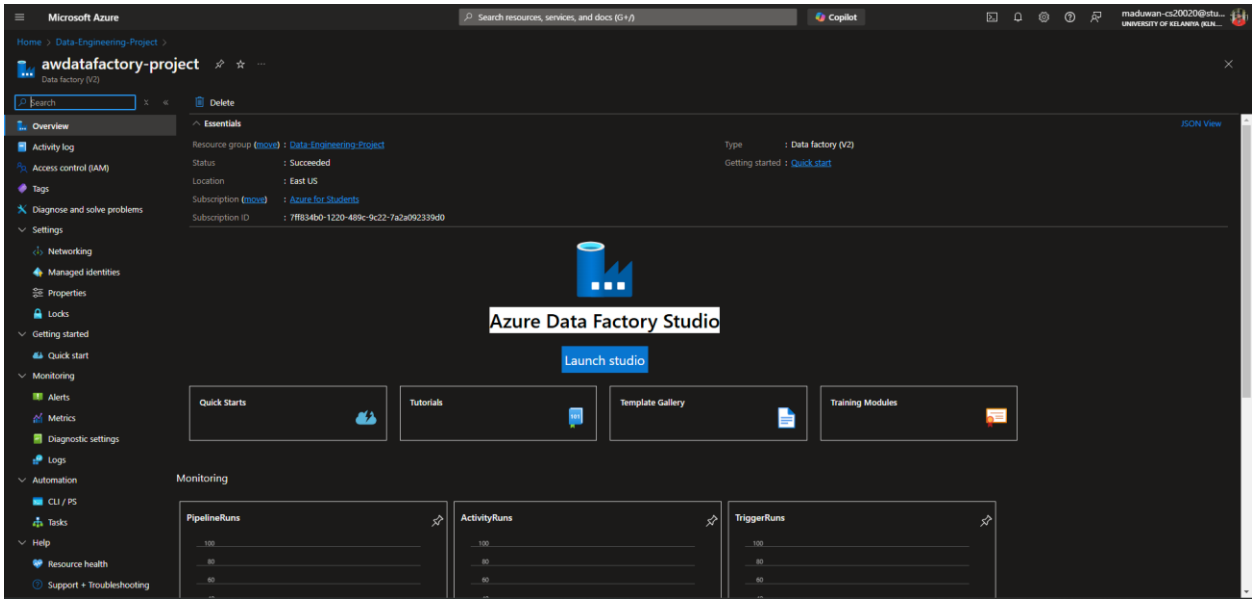
Next, click “create”

2.2 Create Data Factory

The aim of Azure Data Factory is to fetch data from one or more data sources and convert them into a format that we process. The data sources might contain noise that we need to filter out.



Next, go to the Azure Data Factory and click on “Launch Studio”

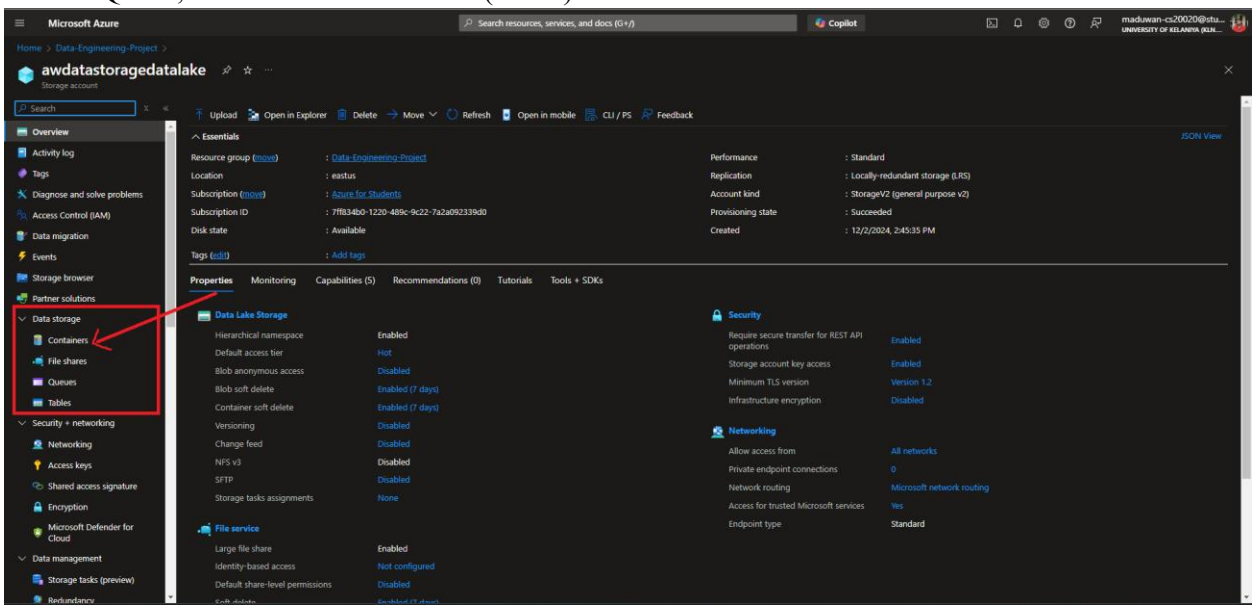


Let this tab open.

2.3

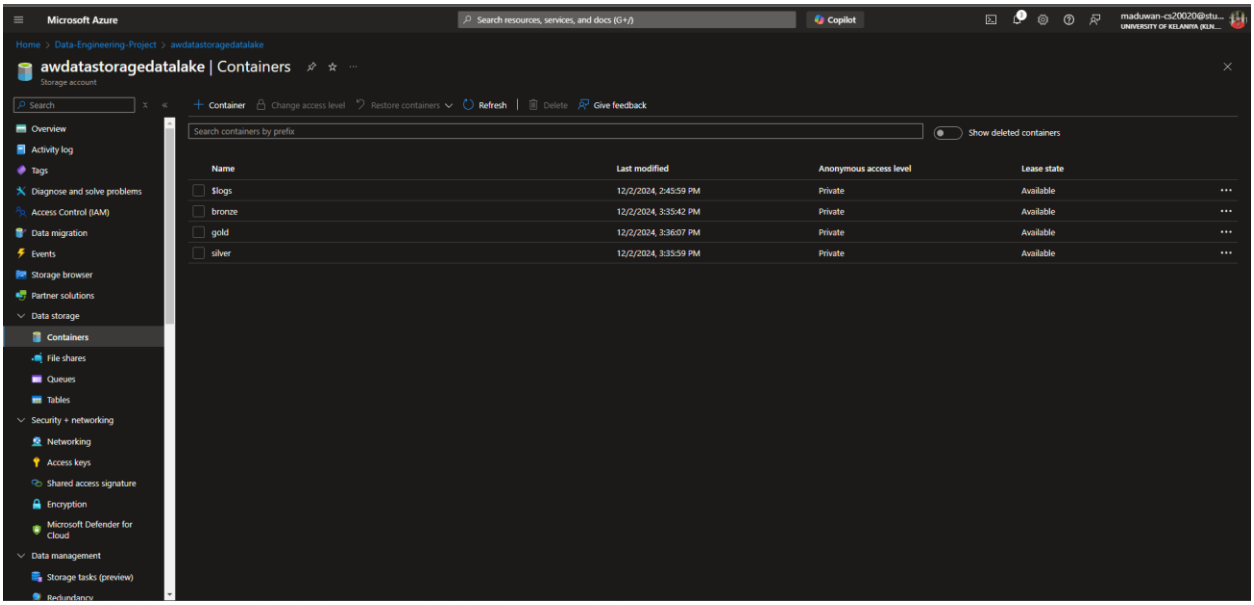
Now go to the resource group and click on the storage account resource.
Go to the "Container" tab.

- Queue, stores Unstructured data (JSON).



Now we should create three Containers for Bronze, Silver & Gold.

- Bronze – Raw data Store
- Silver – Transformed Data
- Gold – Serving



2.3 Create Static pipeline

In here we are going to make a Link service in between the **GitHub repo (API service)** and the raw data store(Data Lake -Bronze).

We get the data from API service. That is our dataset is stored as a GitHub repo.

Now create API for data ingestion from GitHub repository and save it in bronze layer.

To create a pipeline, we need source and destination.

- Go to ADF and click create pipeline.
- Give a name for pipeline.
- Click the “Move and transform” and drag “Copy Data”.
- Give a name for activity. Then we need source and sink(Destination)
- First create **Two** Link service.

    a. With Github API
    b. With Storage Account (Data Lake)

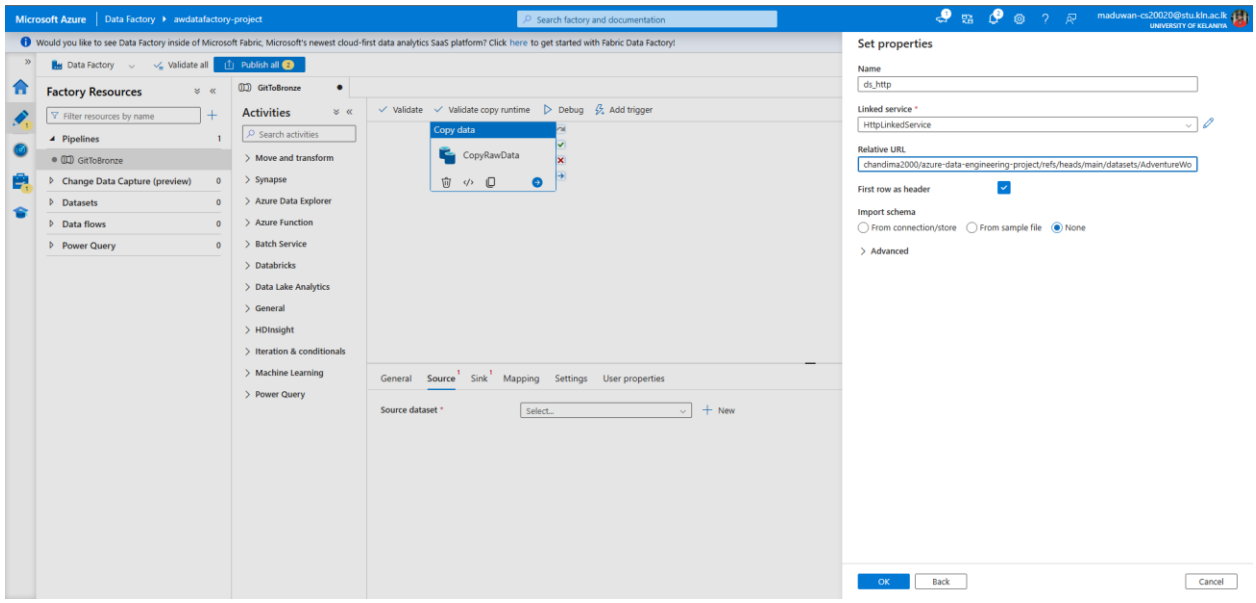
- Go to Manage tab -> Linked Service

    a. Click “+ new” -> select “HTTP” -> click continue
    Give a name
    Give base url. That is main part of the dataset of the github repo.
    Select authentication type as “Anonymous”.

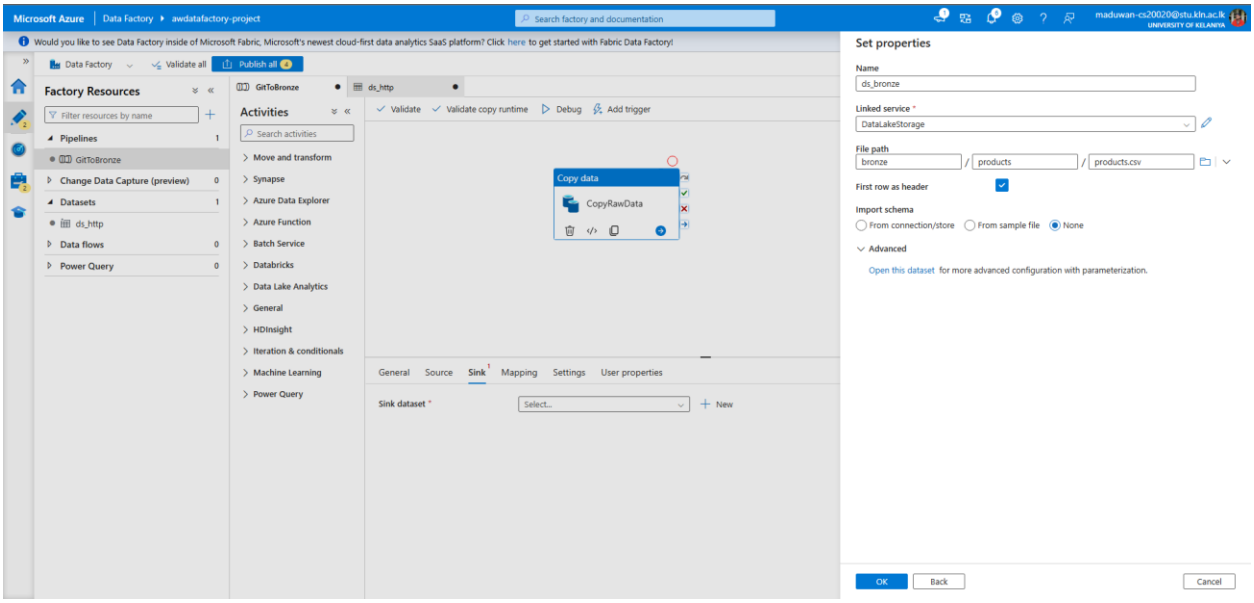
b. Click “+ new” -> select “Data Lake Gen 2” -> click continue
Give a name
Select subscription
Select storage account name which is created before.

- Next, create the dataset.

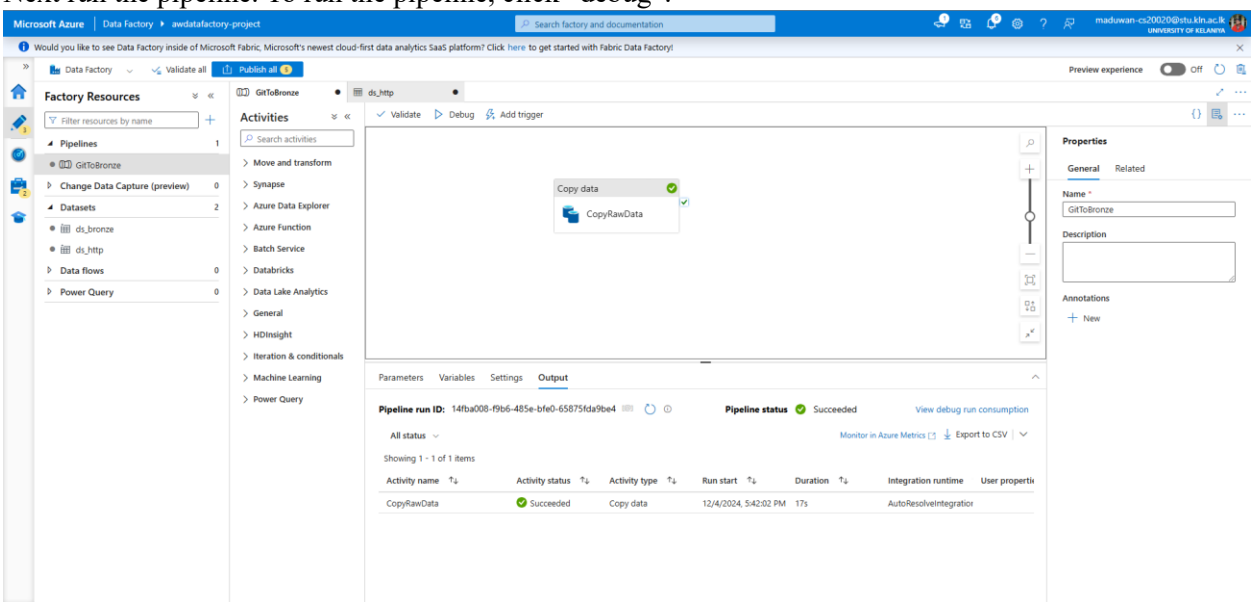
1. First create the source. That is github.
   Go to Author tab -> click “CopyRawData” activity -> click source -> click “+new”
   Select “HTTP” -> select the dataset format (Here, CSV)
   Give a name -> select the Linked service which is created before.
   Give relative url.



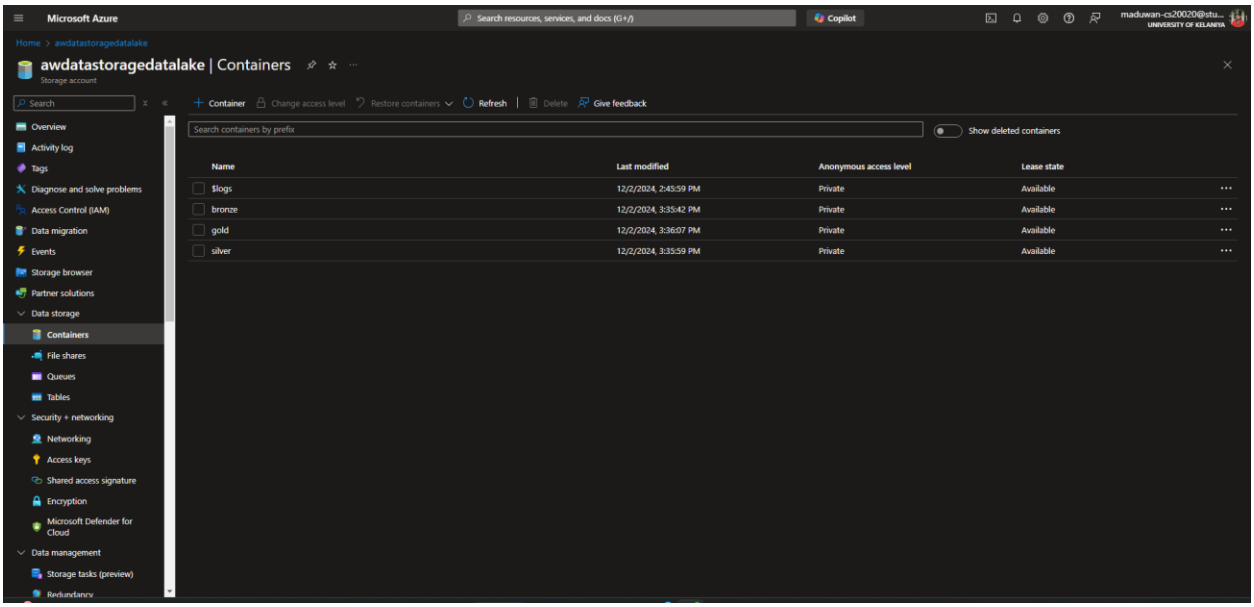
2. Next create the Sink (Destination).
   click sink -> click “+new”
   Select “Data Lake Gen 2” -> select the dataset format (Here, CSV)



- Next run the pipeline. To run the pipeline, click “debug”.

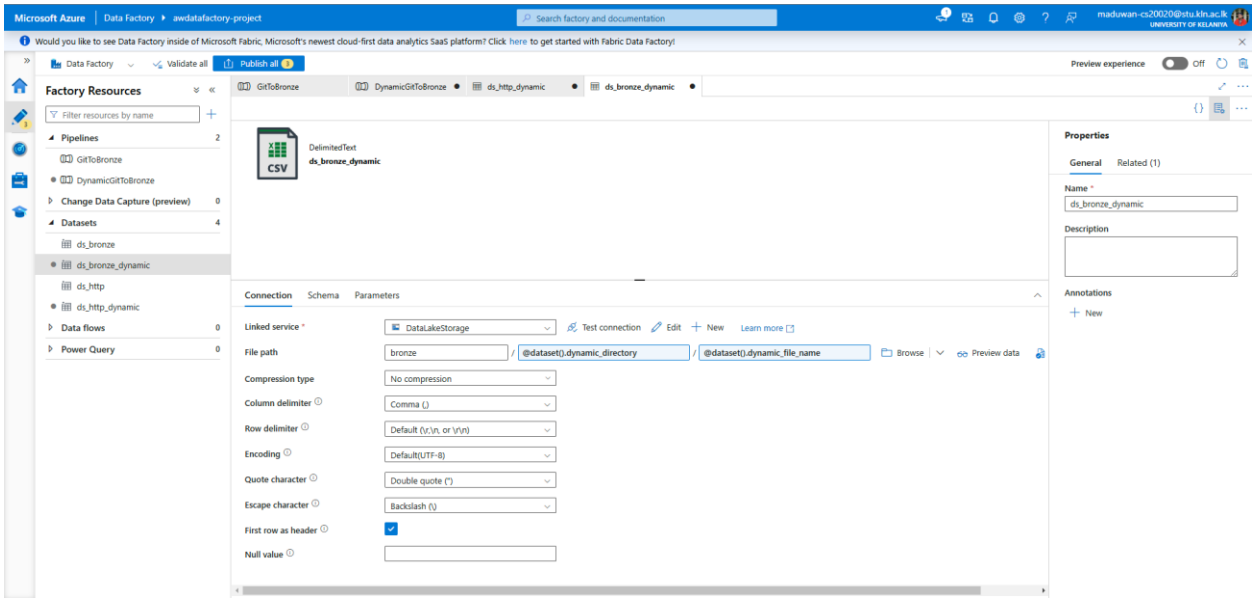


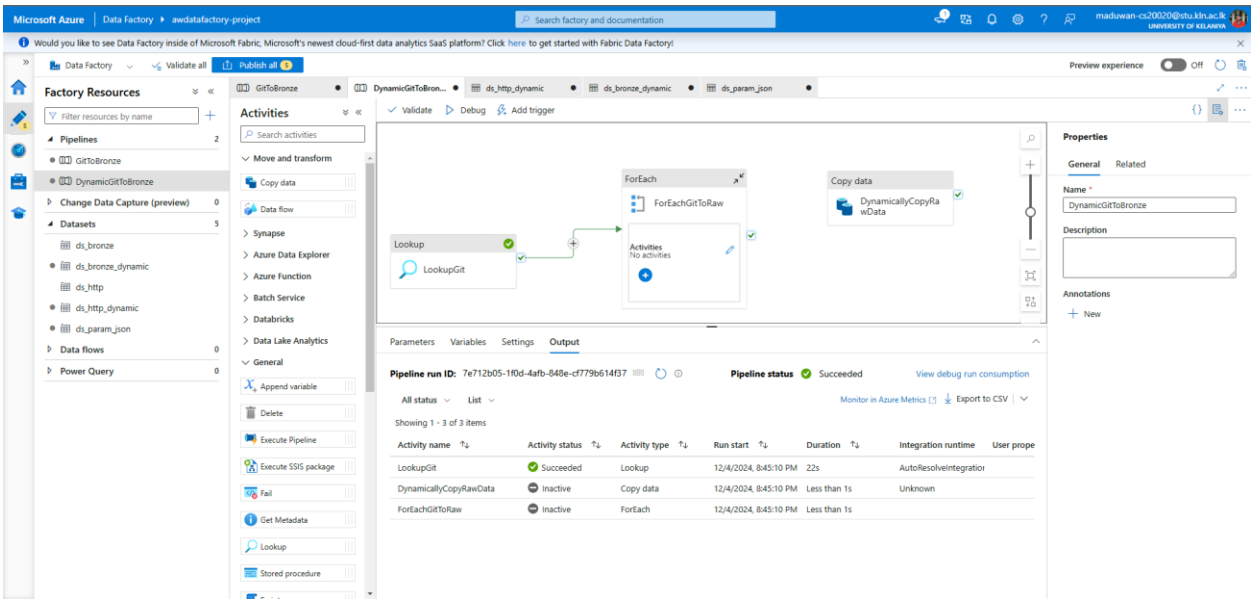
- Now we can see the raw data in bronze layer. Click bronze and then see the data.

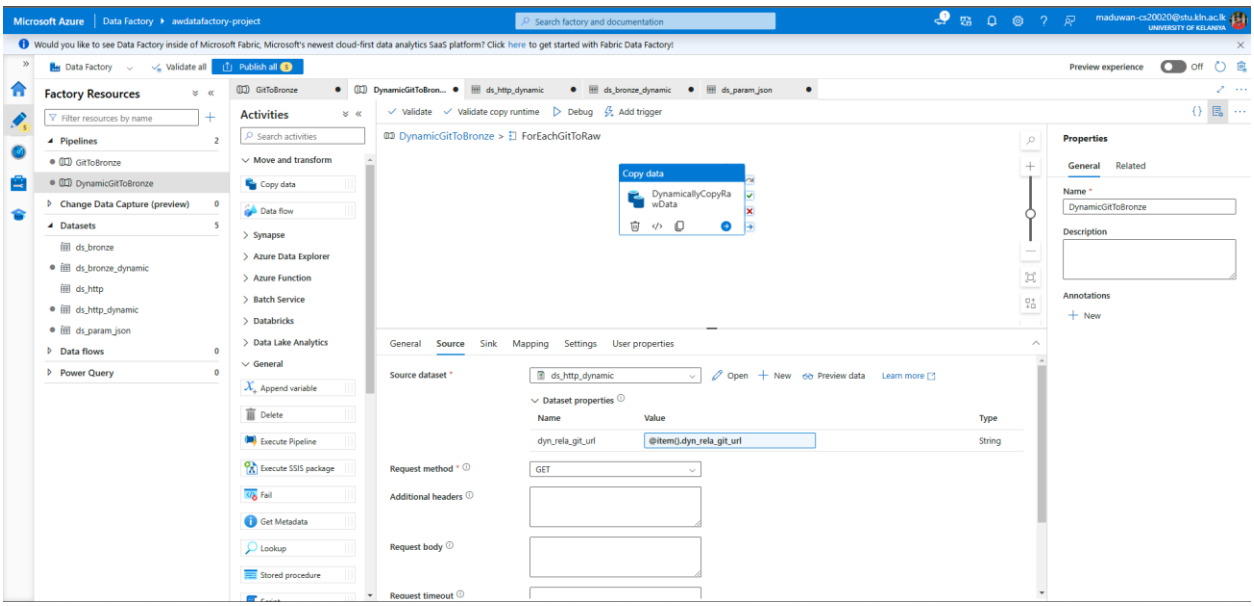


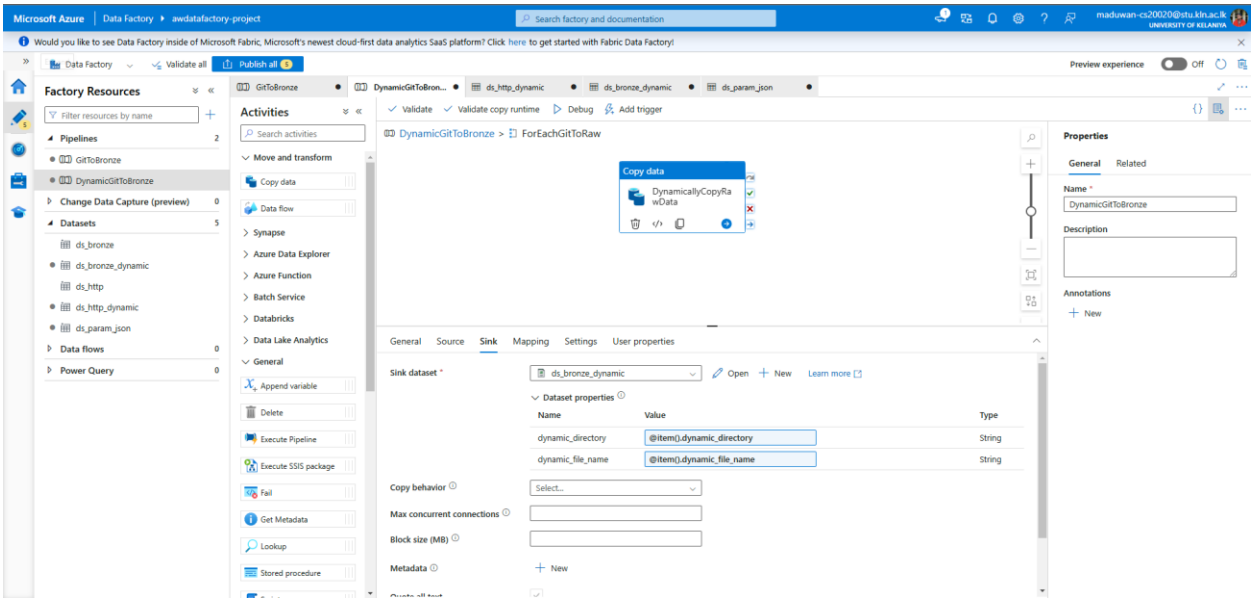
2.4 Create Dynamic pipeline

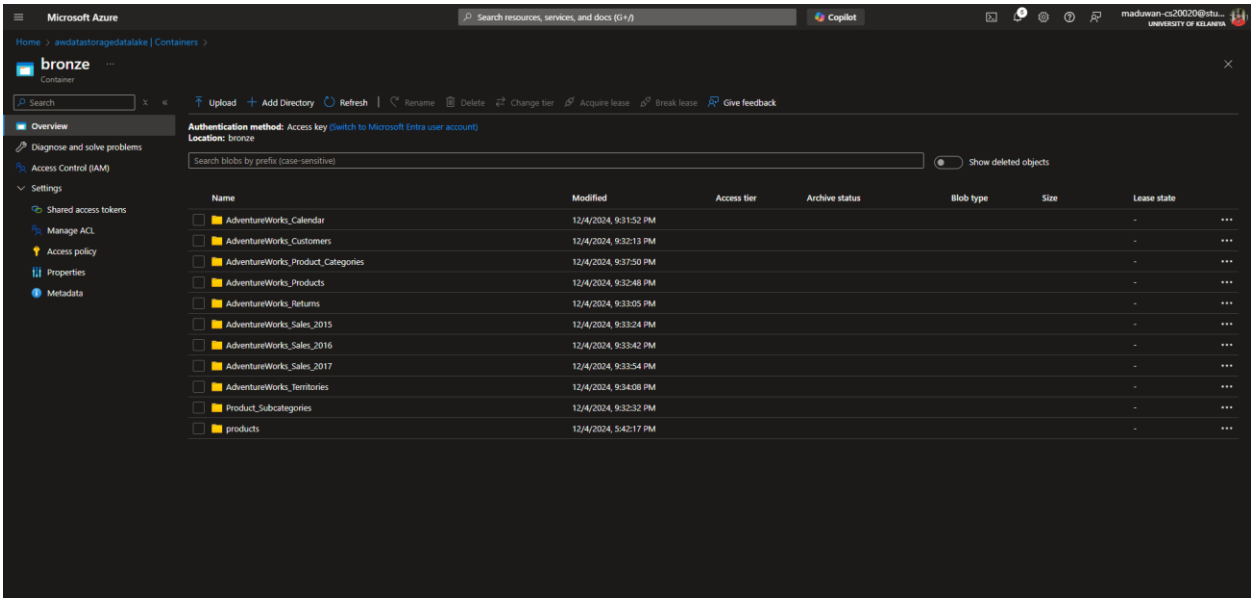
Create dynamic parameters




Microsoft Azure
Data Factory
awdatafactory-project
Factory Resources
Pipelines
GitToBronze
DynamicGitToBronze
Change Data Capture (preview)
Datasets
ds_bronze
ds_bronze_dynamic
ds_http
ds_http_dynamic
ds_param_json
Data flows
Power Query
Activities
Move and transform
Copy data
Data flow
Synapse
Azure Data Explorer
Azure Function
Batch Service
Databricks
Data Lake Analytics
General
Append variable
Delete
Execute Pipeline
Execute SSIS package
Fail
Get Metadata
Lookup
Stored procedure
Lookup
LookupGit
ForEach
ForEachGitToRaw
Activities
No activities
Copy data
DynamicallyCopyRawData
Properties
General
Related
Name
DynamicGitToBronze
Description
Annotations
New
Parameters
Variables
Settings
Output
Pipeline run ID: 7e712b05-1f0d-4afb-848e-cf779b614f37
Pipeline status Succeeded
View debug run consumption
Showing 1 - 3 of 3 items
Activity name
Activity status
Activity type
Run start
Duration
Integration runtime
LookupGit
Succeeded
Lookup
12/4/2024, 8:45:10 PM
22s
AutoResolveIntegratior
DynamicallyCopyRawData
Inactive
Copy data
12/4/2024, 8:45:10 PM
Less than 1s
Unknown
ForEachGitToRaw
Inactive
ForEach
12/4/2024, 8:45:10 PM
Less than 1s


Microsoft Azure
Data Factory
awdatafactory-project
Factory Resources
DynamicGitToBronze > ForEachGitToRaw
Copy data
DynamicallyCopyRawData
Properties
General
Related
Name
DynamicGitToBronze
Description
Annotations
New
General
Source
Sink
Mapping
Settings
User properties
Source dataset
ds_http_dynamic
Open
New
Preview data
Learn more
Dataset properties
Name
Value
Type
dyn_rela_git_url
@item().dyn_rela_git_url
String
Request method
GET
Additional headers
Request body
Request timeout


Microsoft Azure
Data Factory
awdatafactory-project
Search factory and documentation
Would you like to see Data Factory inside of Microsoft Fabric, Microsoft's newest cloud-first data analytics SaaS platform? Click here to get started with Fabric Data Factory!
Data Factory
Validate all
Publish all
Preview experience
Off
Factory Resources
Filter resources by name
Pipelines
GitToBronze
DynamicGitToBronze
Change Data Capture (preview)
Datasets
ds_bronze
ds_bronze_dynamic
ds_http
ds_http_dynamic
ds_param_json
Data flows
Power Query
Activities
Search activities
Move and transform
Copy data
Data flow
Synapse
Azure Data Explorer
Azure Function
Batch Service
Databricks
Data Lake Analytics
General
Append variable
Delete
Execute Pipeline
Execute SSIS package
Fail
Get Metadata
Lookup
Stored procedure
Validate
Validate copy runtime
Debug
Add trigger
DynamicGitToBronze > ForEachGitToRaw
Copy data
DynamicallyCopyRawData
General
Source
Sink
Mapping
Settings
User properties
Sink dataset *
ds_bronze_dynamic
Open
New
Learn more
Dataset properties
Name
Value
Type
dynamic_directory
@item().dynamic_directory
String
dynamic_file_name
@item().dynamic_file_name
String
Copy behavior
Select...
Max concurrent connections
Block size (MB)
Metadata
New
Properties
General
Related
Name *
DynamicGitToBronze
Description
Annotations
New


Microsoft Azure
Search resources, services, and docs (G+/)
Copilot
Home > awdatastoragedatalake | Containers >
bronze
Container
Search
Overview
Diagnose and solve problems
Access Control (IAM)
Settings
Shared access tokens
Manage ACL
Access policy
Properties
Metadata
Upload
Add Directory
Refresh
Rename
Delete
Change tier
Acquire lease
Break lease
Give feedback
Authentication method: Access key (Switch to Microsoft Entra user account)
Location: bronze
Search blobs by prefix (case-sensitive)
Show deleted objects
Name
Modified
Access tier
Archive status
Blob type
Size
Lease state
AdventureWorks_Calendar 12/4/2024, 9:31:52 PM
AdventureWorks_Customers 12/4/2024, 9:32:13 PM
AdventureWorks_Product_Categories 12/4/2024, 9:37:50 PM
AdventureWorks_Products 12/4/2024, 9:32:48 PM
AdventureWorks_Returns 12/4/2024, 9:33:05 PM
AdventureWorks_Sales_2015 12/4/2024, 9:33:24 PM
AdventureWorks_Sales_2016 12/4/2024, 9:33:42 PM
AdventureWorks_Sales_2017 12/4/2024, 9:33:54 PM
AdventureWorks_Territories 12/4/2024, 9:34:08 PM
Product_Subcategories 12/4/2024, 9:32:32 PM
products 12/4/2024, 5:42:17 PM

Phase-2

Azure Data Bricks

It is an Apache spark based analytics platform.

Apache Spark is a powerful open-source data processing engine written in Scala, designed for large-scale data processing.

Apache Spark is written in Scala programming language. To support Python with Spark, Apache Spark Community released a tool, PySpark.

PySpark is the Python API for Apache Spark.

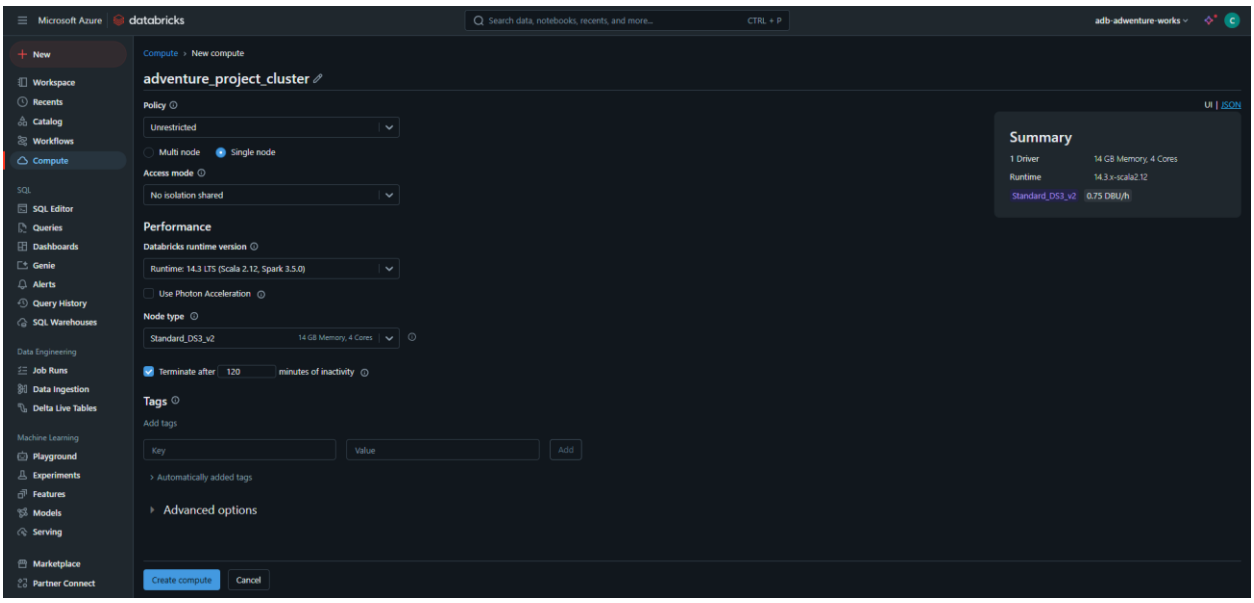
Spark itself is not a programming language but provides APIs in several programming languages to interact with its engine: Python (PySpark), Scala (native language of Spark), Java, R, SQL (for Spark SQL)

Features:-

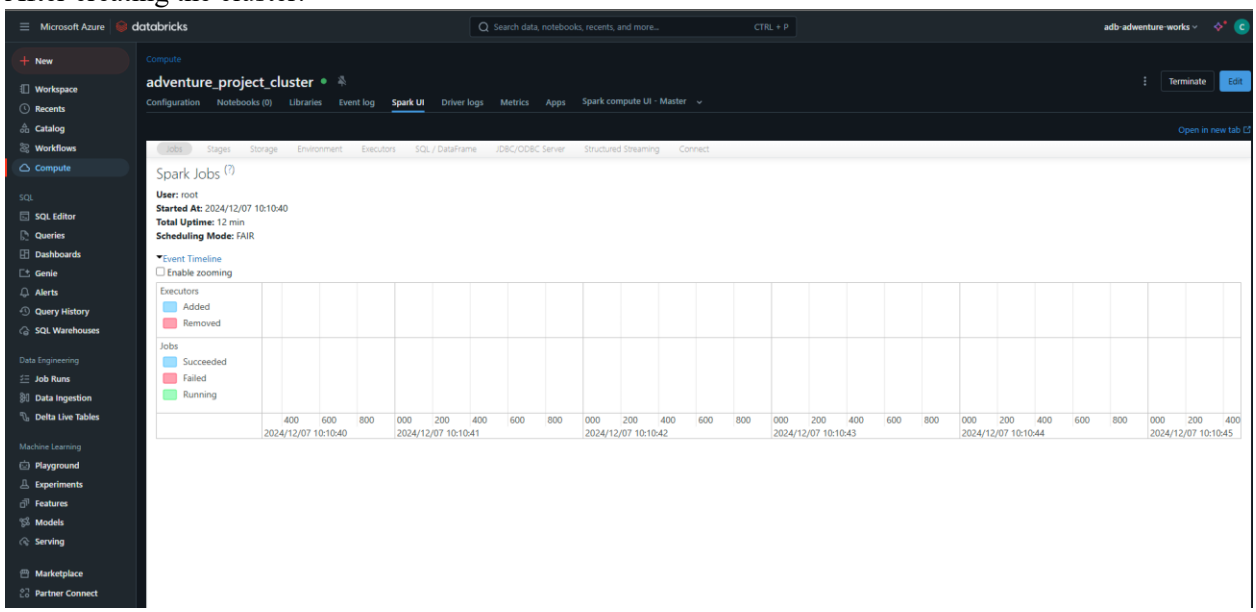
1. High performance computing
2. Notebooks
3. Switch between programming languages (Spar sql, Scala, PySpark)
4. ETL
5. Model Training

- Create a Cluster

  Click “compute” -> fill like below.



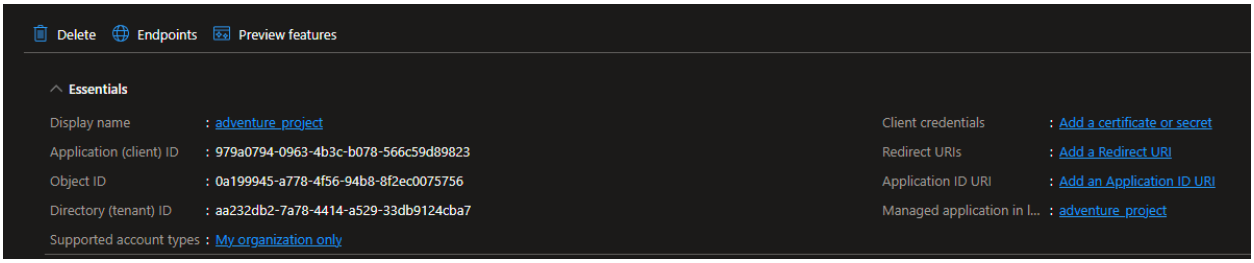
After creating the cluster:-



- Create Storage Access

Step 1:- Create an Application

search “Microsoft Entra ID” → App registration -> + new registration -> complete the registration



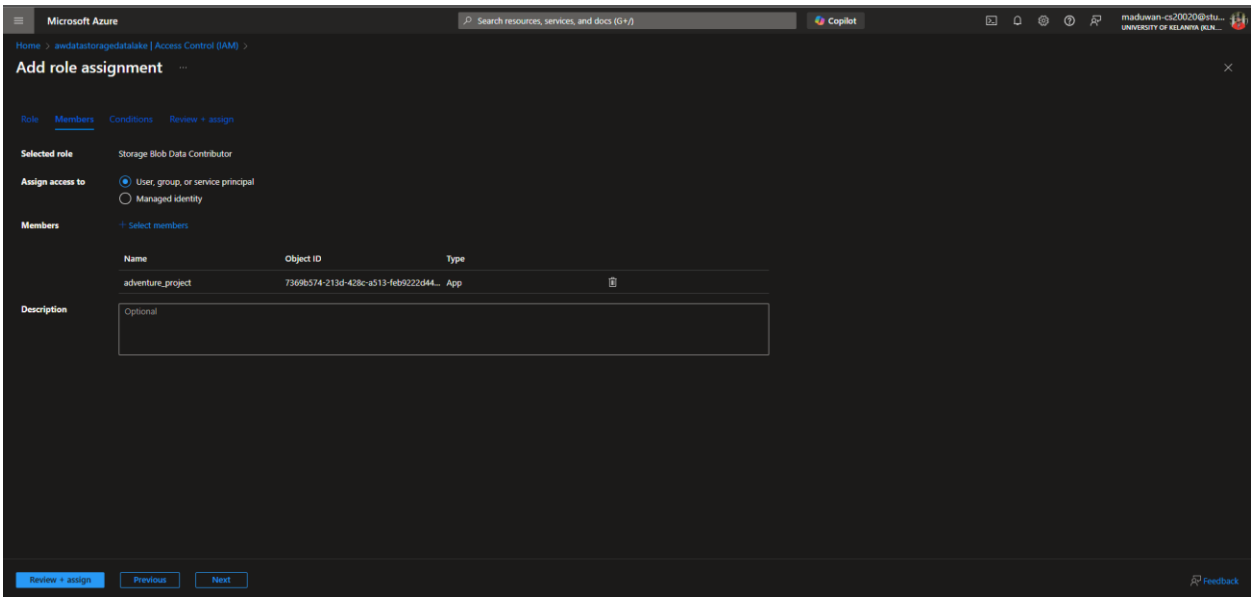
Simply copy these informations.

Next, create secrets.

Click “ Certificate & Secrets” -> + New client Secrets -> copy the secret-value and secret-id

Step 2 :- Assigning roles to this application

Goto home and select “Storage Account” -> Access Control -> + Add -> search “Storage Bolb Data Contributor” -> click “select members” -> select the application name which was created earlier -> click “Review + Assign”



Step 3 :-

Go to “data bricks” -> click “workspace” (because need to create a folder) -> click "folder” -> create a folder -> click “notebook” (because In this folder need to create a notebook) -> connect to the cluster

Do the transformation tasks in notebook and import data into silver layer. With this end our Phase 2.

What is Parquet?

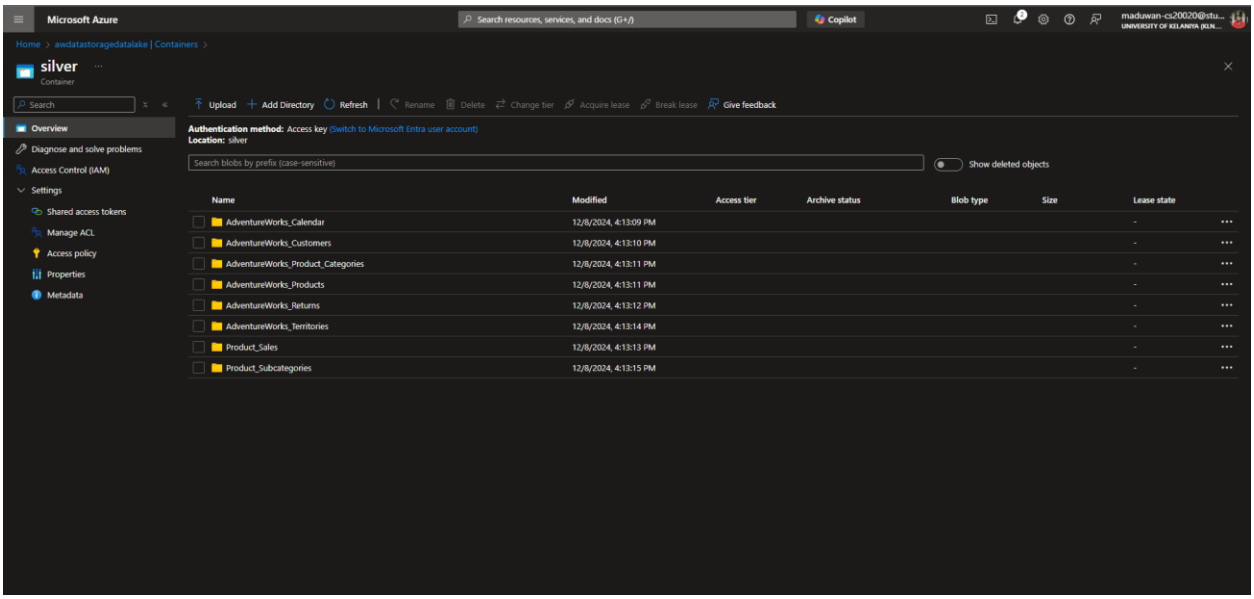
Apache Parquet is an open source, column-oriented data file format designed for efficient data storage and retrieval. It provides efficient data compression and encoding schemes with enhanced performance to handle complex data in bulk.

Characteristics of Parquet

- Free and open-source file format.
- Language agnostic.
- Column-based format - **files are organized by column**, **rather than by row(CSV files)**, which saves storage space and speeds up analytics queries.
- Used for analytics (OLAP) use cases, typically in conjunction with traditional OLTP databases.
- Highly efficient data compression and decompression.
- Supports complex data types and advanced nested data structures.

The open-source Delta Lake project builds upon and extends the Parquet format.

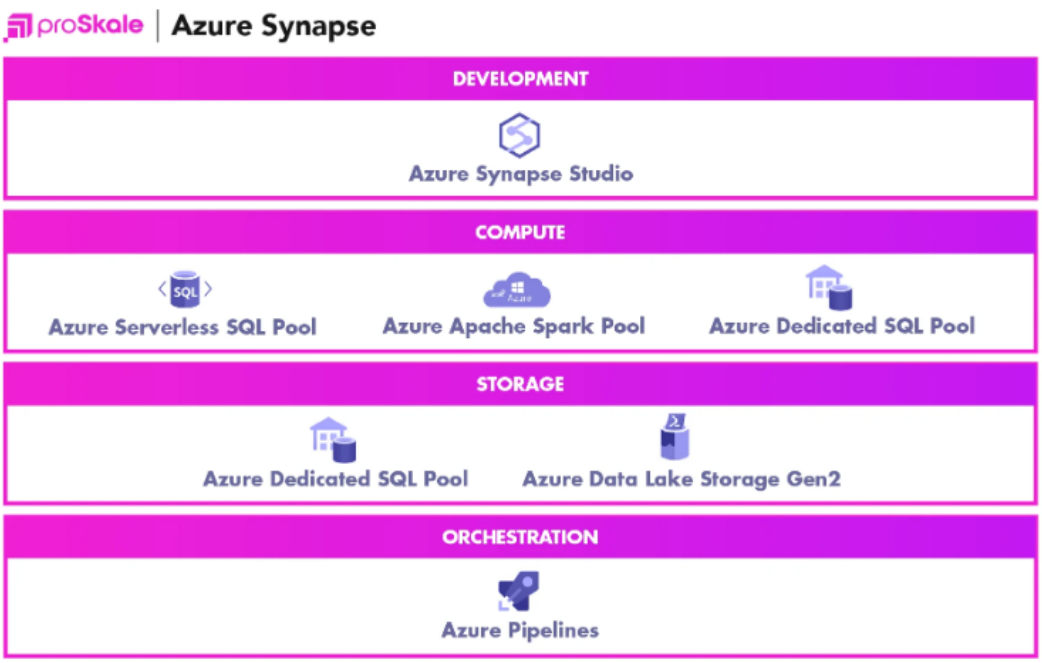
Silver Layer file system



Phase – 3

In here, we are going to serving the data. That is Gold Layer.

Azure Synapse Analytics is a comprehensive data integration, analytics, and data warehousing platform provided by Microsoft Azure. It combines enterprise data warehousing with big data analytics to offer a unified environment for processing, managing, and analyzing large volumes of structured and unstructured data.



<u>Key Features of Azure Synapse Analytics</u>

Data Integration:

- Azure Synapse Pipelines: Data integration service that allows you to create ETL (Extract, Transform, Load) workflows to move and transform data from various sources.
- Connects to on-premises, cloud-based, and hybrid data sources.

Data Warehousing:

- Offers **dedicated SQL pools** (provisioned resources) for massive parallel processing (MPP) of data.
- Serverless SQL pools provide on-demand querying capabilities for ad hoc analysis without requiring dedicated resources.

Big Data Analytics:

- Integrated with Apache Spark pools for distributed big data processing.
- Can process and analyze data stored in Azure Data Lake or other storage systems.

Data Exploration and Analysis:

- Allows querying across structured, semi-structured, and unstructured data.
- Supports T-SQL, making it accessible for users familiar with SQL Server.

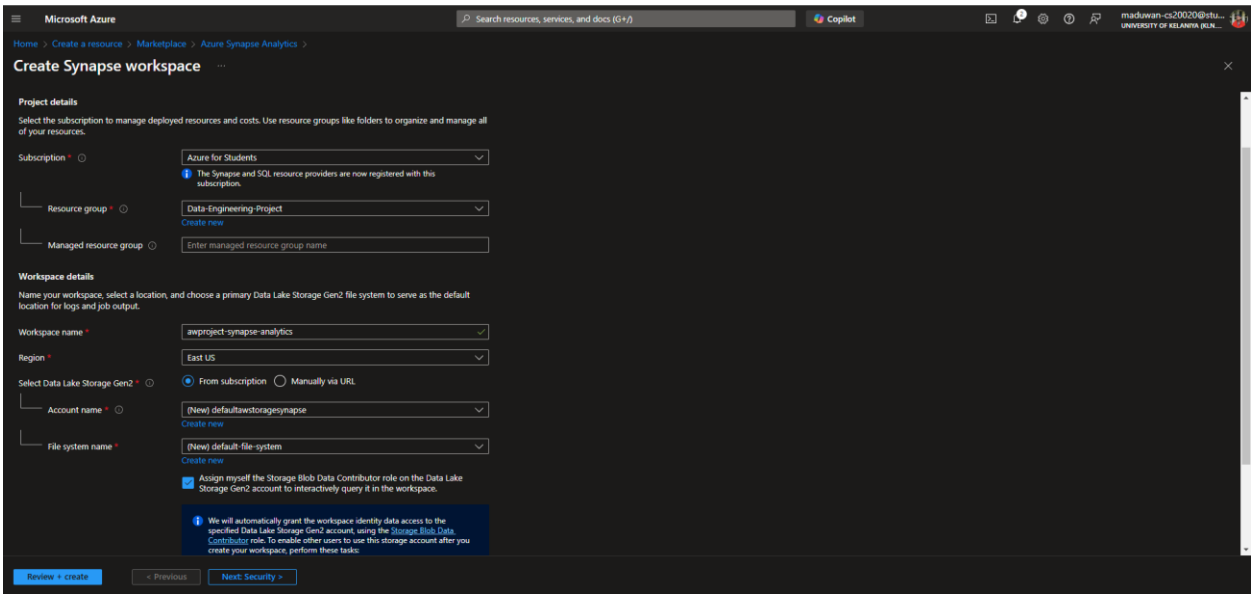
Data Visualization:

- Seamlessly integrates with Power BI for creating interactive dashboards and reports.
- Enables real-time business intelligence.

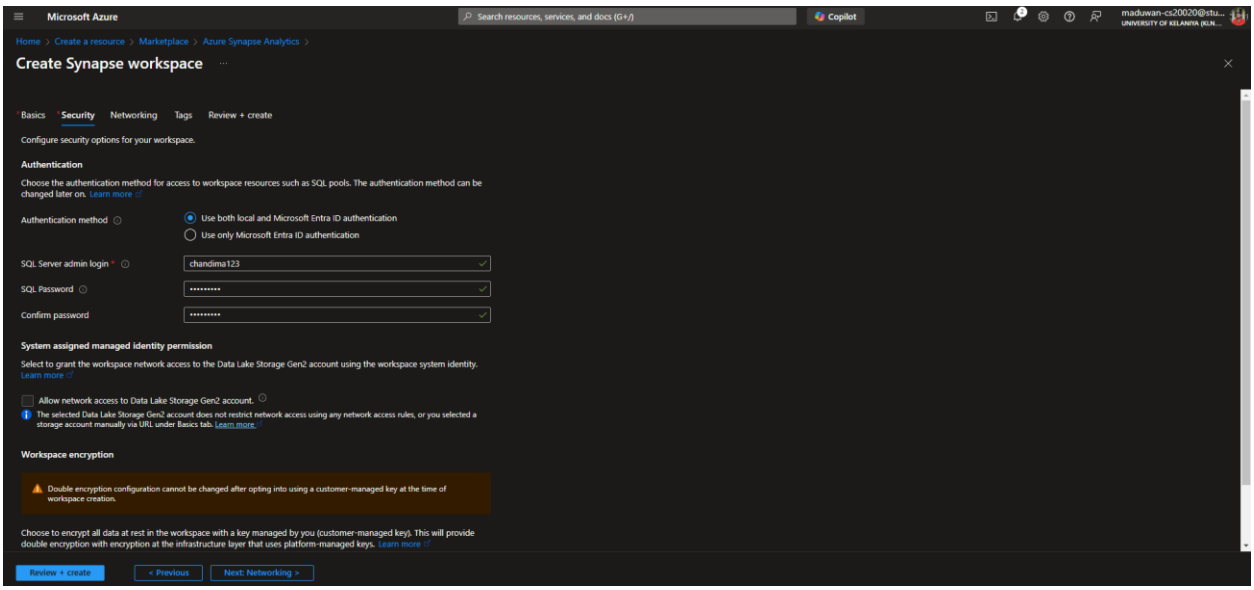
Security and Governance:

- Built-in security features like role-based access control (RBAC), managed identities, and data encryption.
- Integration with Azure Purview for data governance and lineage tracking.

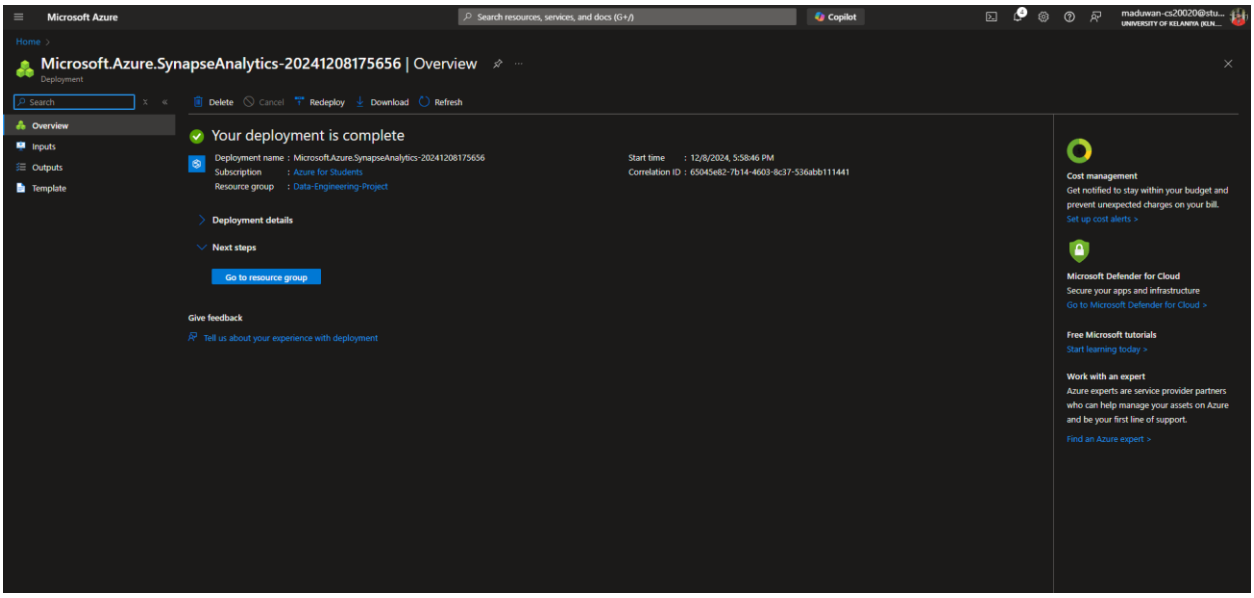
Click ‘Create resource” -> search “Synapse Analytics” -> fill the form.(Here we should create a new Storage account for synapse analytics)



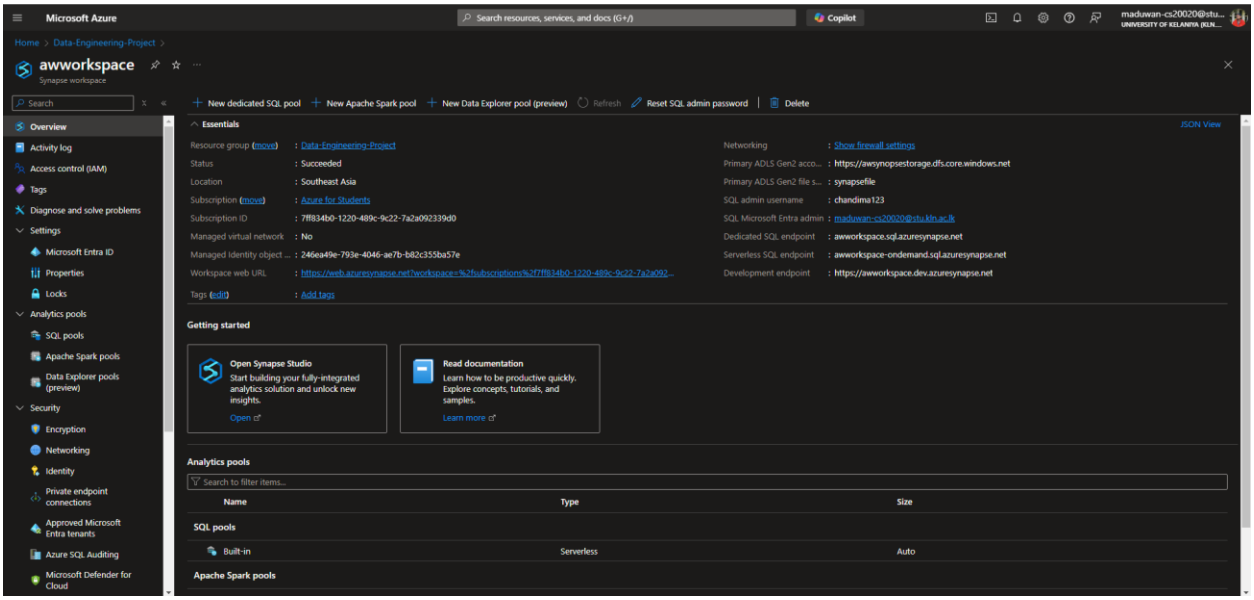
Configure the dedicated SQL pool

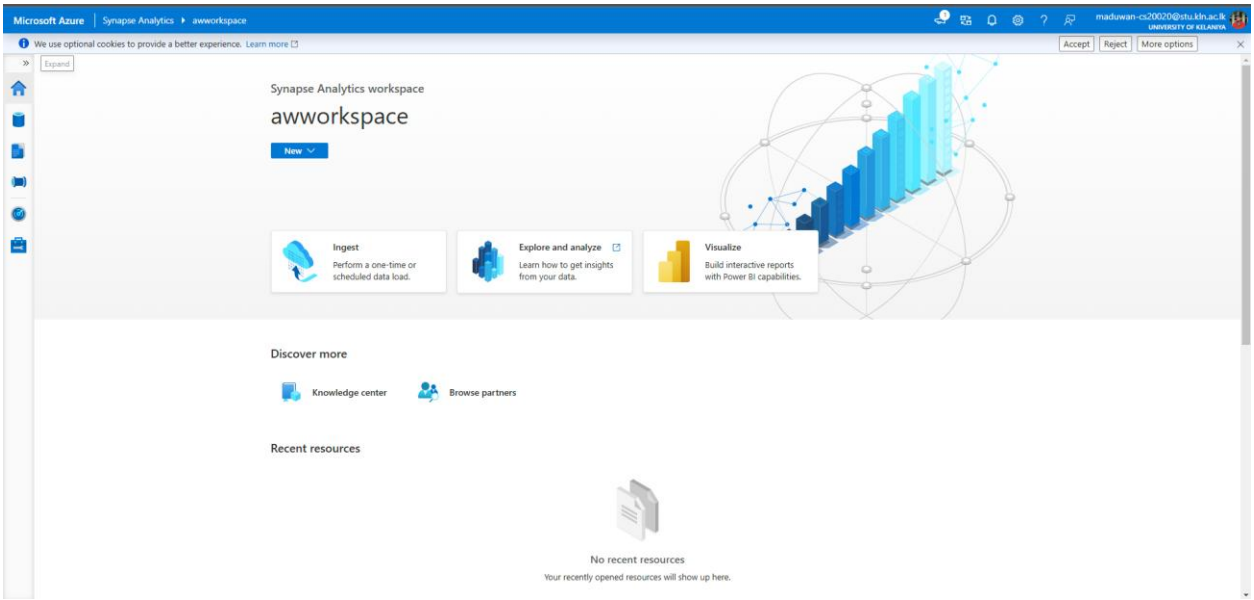


Then click “review + create”.

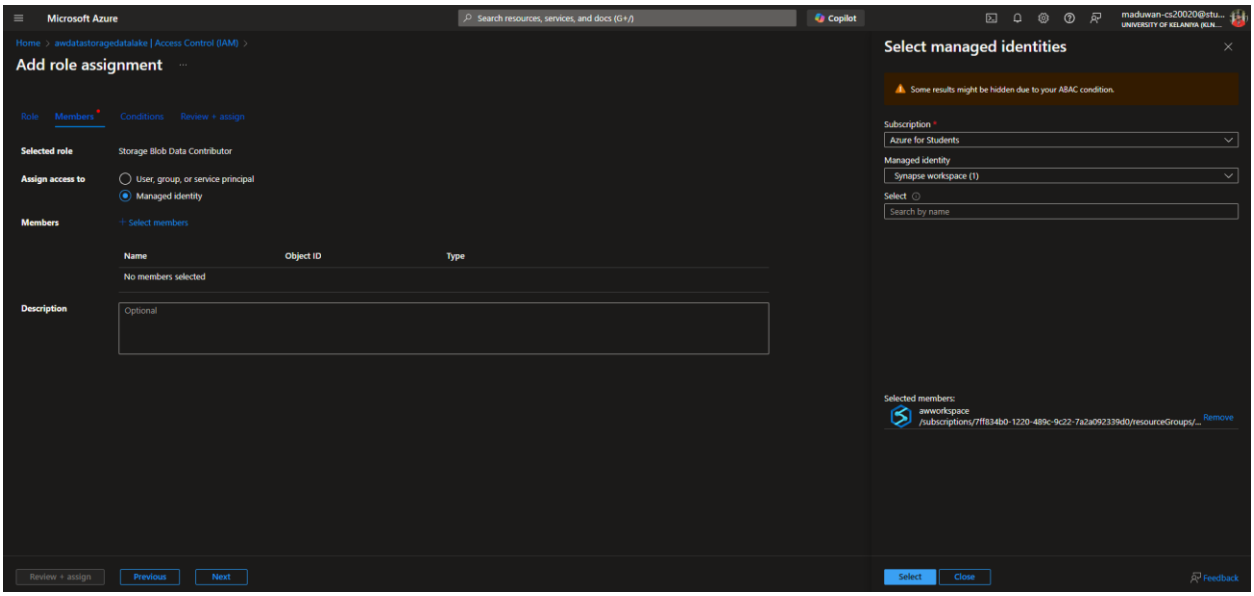


Synapses Workspace Homepage



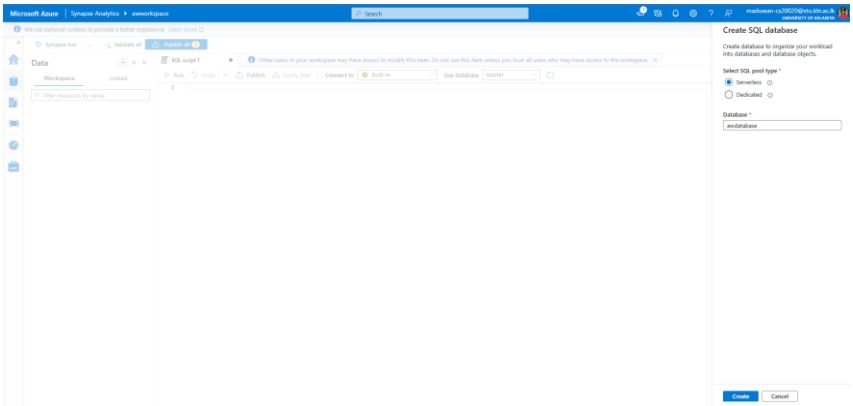


1. Goto “storage account” -> click “Access Control” -> click “+ add” -> search “Storage blob data contributor” -> do following

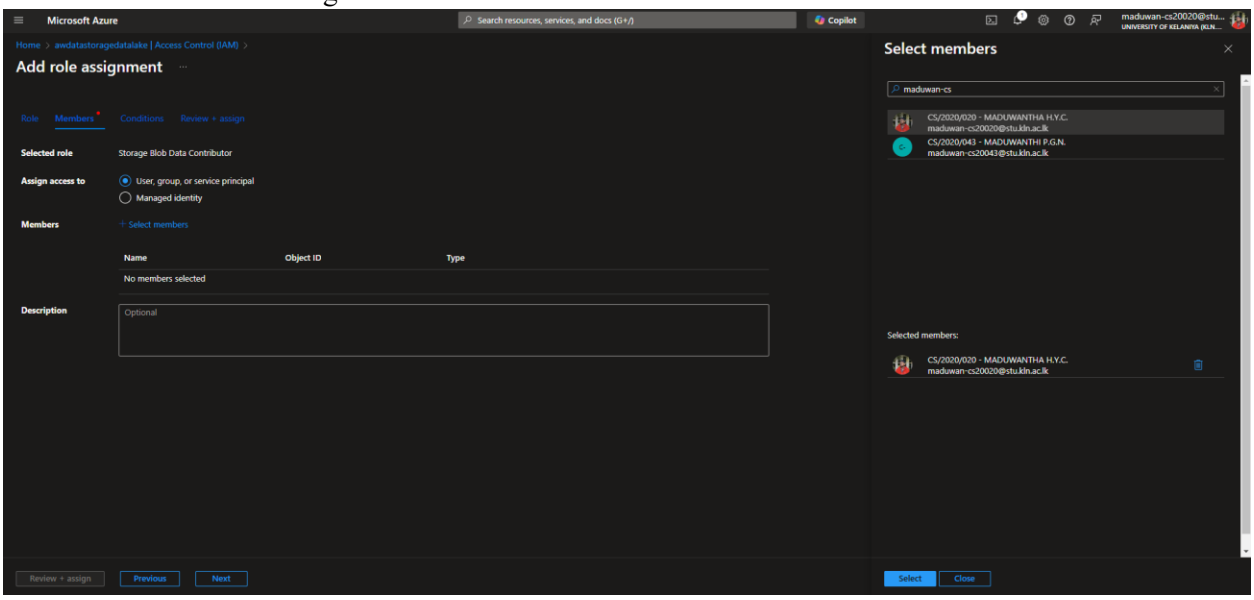


Lastly click “ Review + Assign”

2. Goto Azure synapse Analysis window -> select “develop” tab -> select “ + ” -> select “SQL Script”
3. Now we want to create a Database. Goto “Data” tab -> select “SQL database”
4.



5. Again, we need to create a Access control -> click “+ add” -> search “Storage blob data contributor” -> do following



- How to show data using power BI?
    - All data should be displayed using Views. We create Views on top of the Query. For that we need Schemas. These views stores in gold layer. Then power BI use gold layer.

Lastly show the Data using Power BI.

Goto synapse workspace and copy the serverless SQL endpoint