NC STATE UNIVERSITY

Independent Study

Building a Scalable Data Migration and Transformation Pipeline Using Azure

Sakshi Basapure
200534164

Problem Statement

Develop a cloud-based solution that automates the ingestion of on-premise data, transforms it into a clean and structured format, and stores it in a scalable cloud environment.

Objectives and BRs

BR 1: Ensure secure automated **data migration** between on-premise systems and cloud.

BR 2: Develop a **transformation** layer to convert raw data into clean, structured formats.

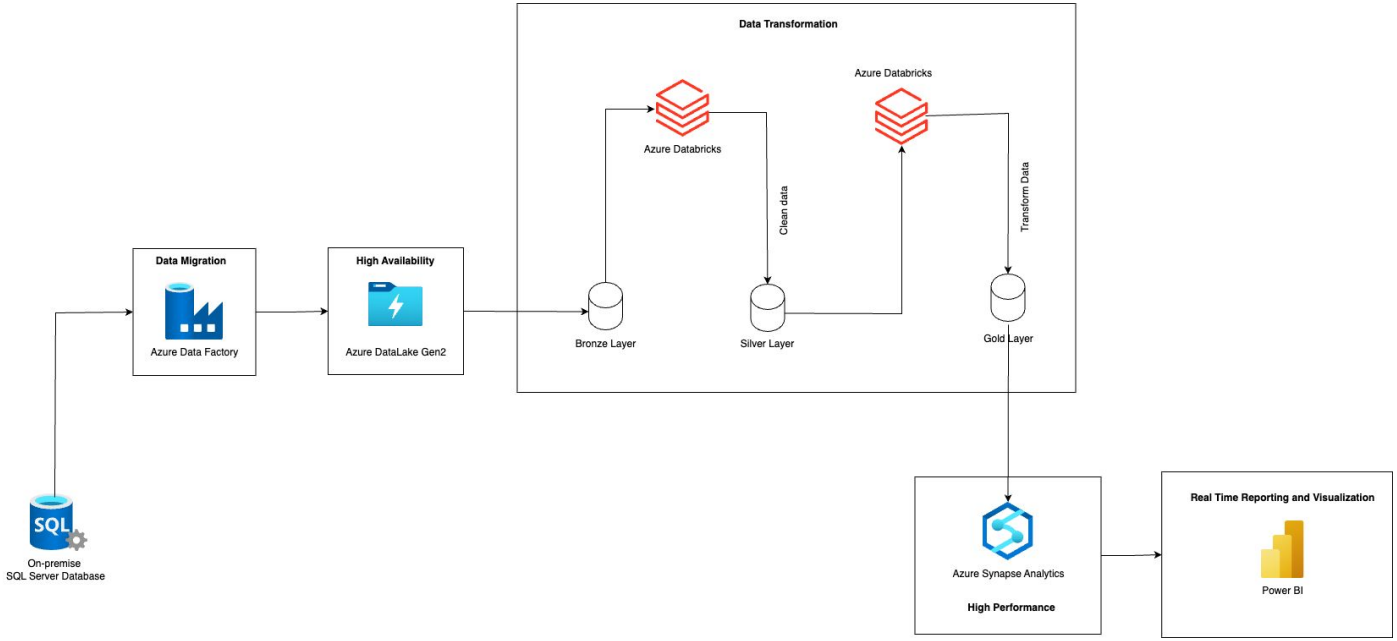
BR 3: Maintain **high availability** throughout the data pipeline.

BR 4: Maintain **high performance** throughout the data pipeline.

BR 5: Enforce **governance and compliance** with **data security** regulations.

BR 6: Implement **real-time reporting and visualization** capabilities, providing business users with actionable insights through intuitive dashboards.

Architecture



Implementation

BR 1: Ensure secure **automated data migration** between on-premise systems and cloud.

TR 1.2: Utilize a secure connection method to connect to on-premise data sources.

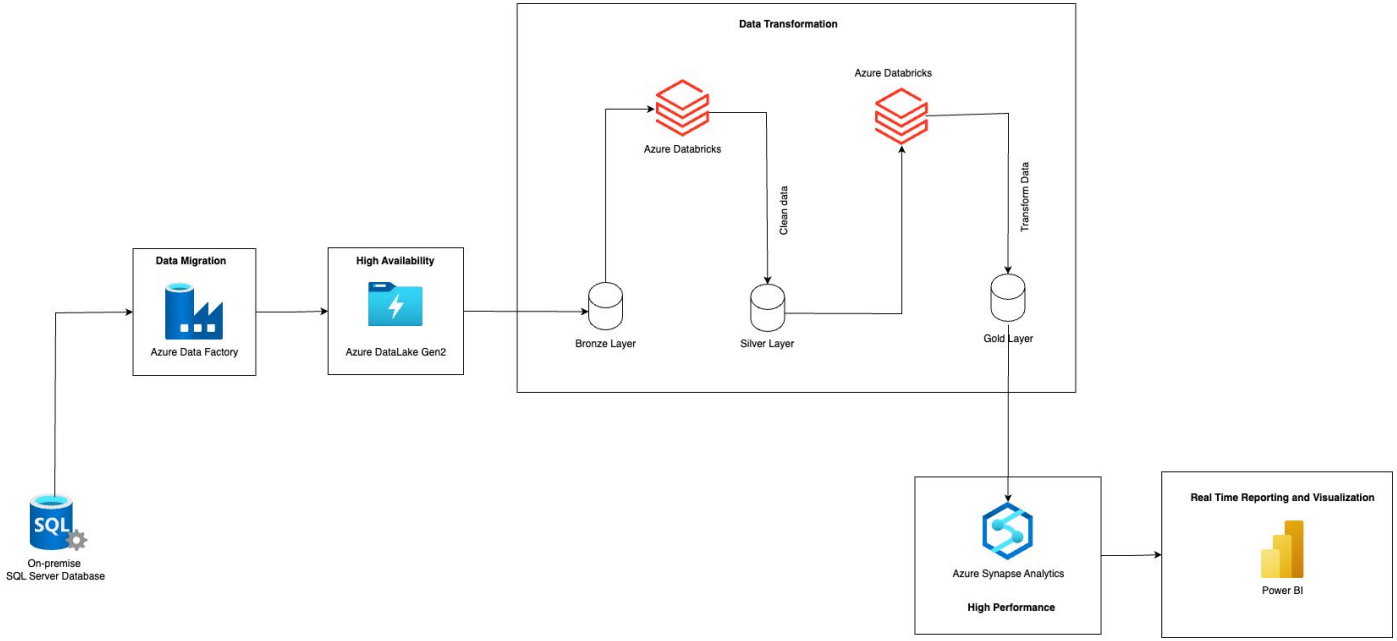
BR 2: Develop a **transformation** layer to convert raw data into clean, structured formats.

TR 2.1: Design and implement ETL (Extract, Transform, Load) processes for data cleansing and normalization.

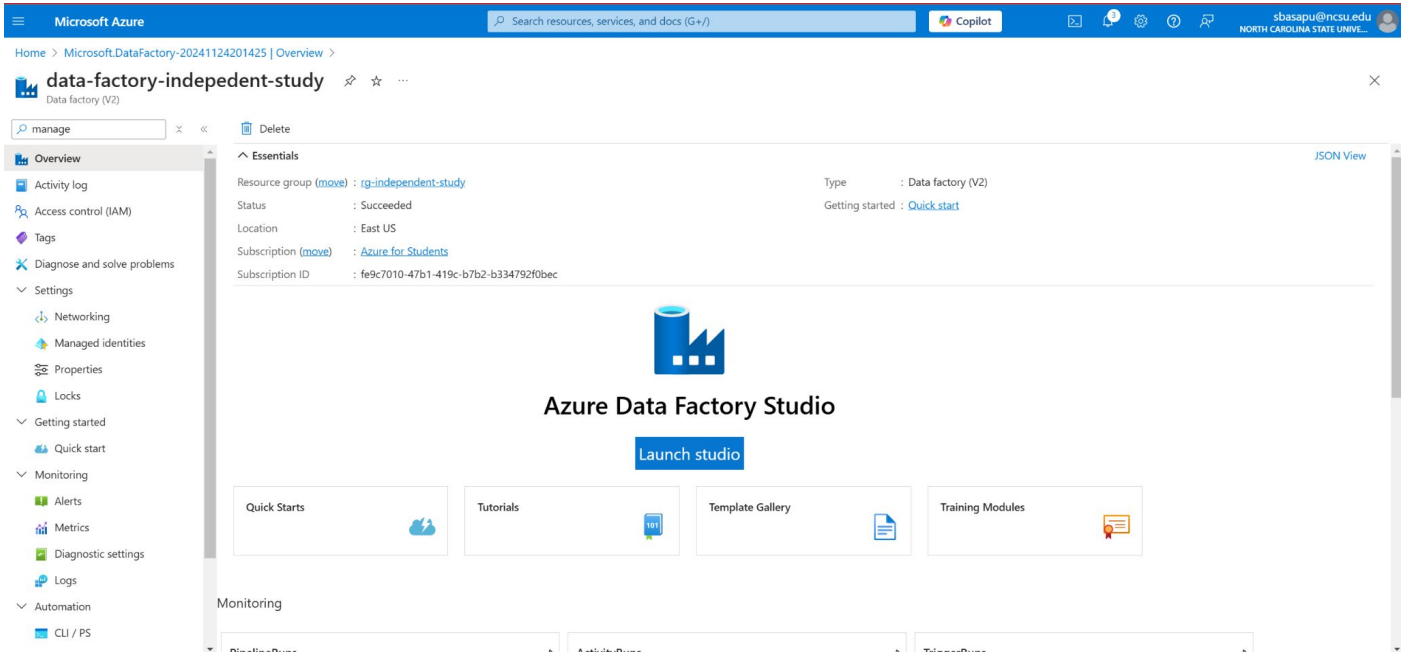
BR 3: Maintain **high availability** throughout the data pipeline.

TR 3.1: Configure redundant storage and automatic failover mechanisms to ensure continuous data availability.

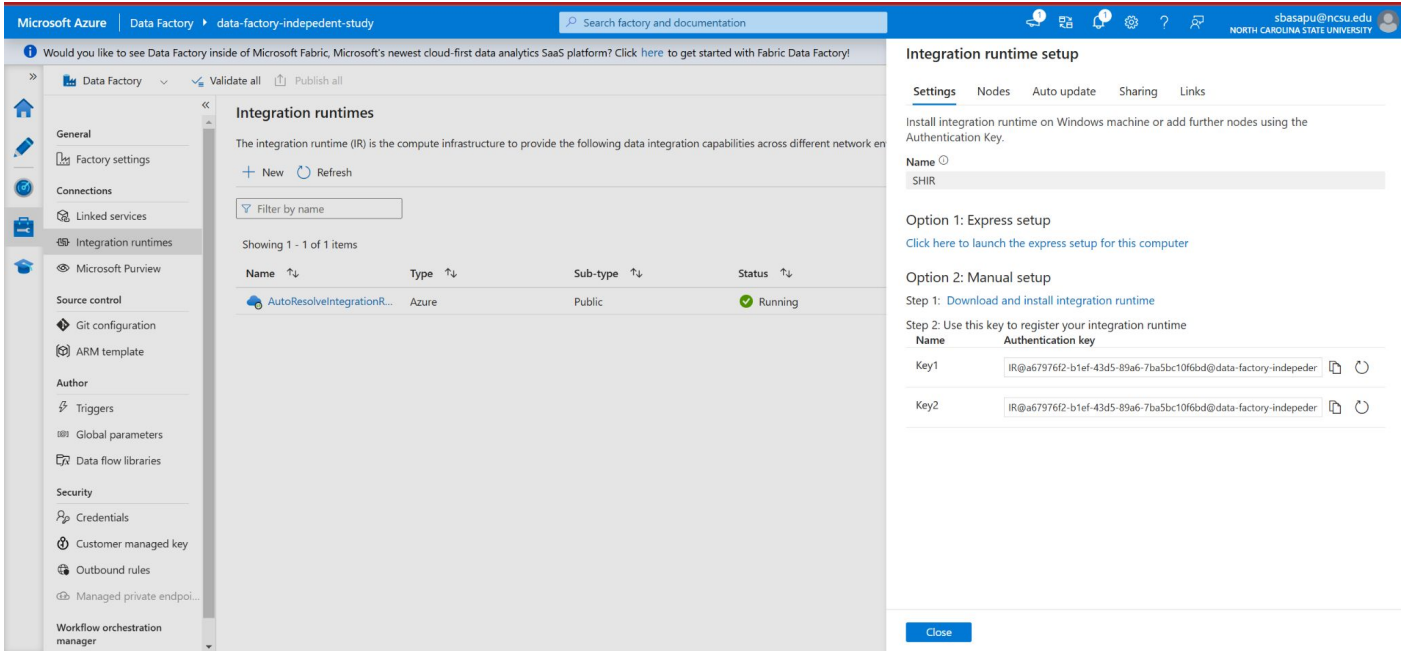
Architecture



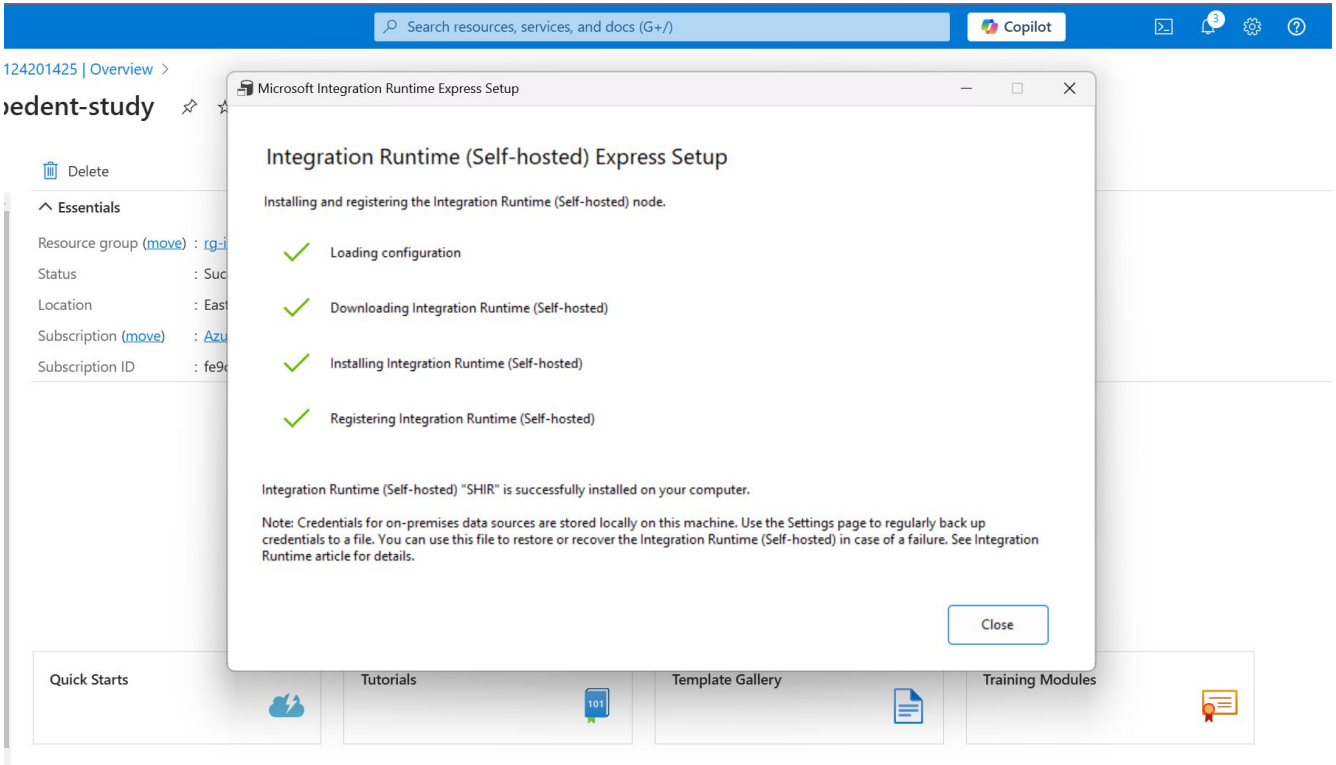
BR1: Data Migration



BR1: Data Migration



BR1: Data Migration



BR1: Data Migration

Integration runtimes

The integration runtime (IR) is the compute infrastructure to provide the following data integration capabilities across different network environment. Learn more

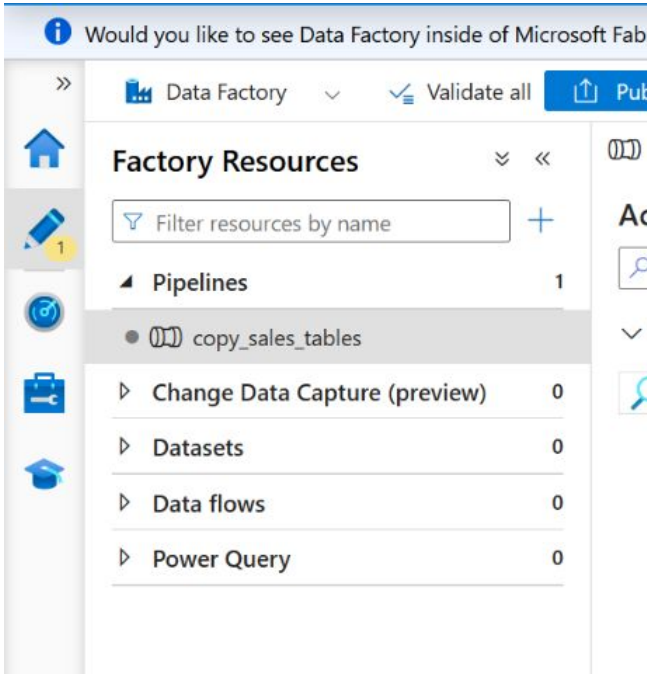
+ New Refresh

Filter by name

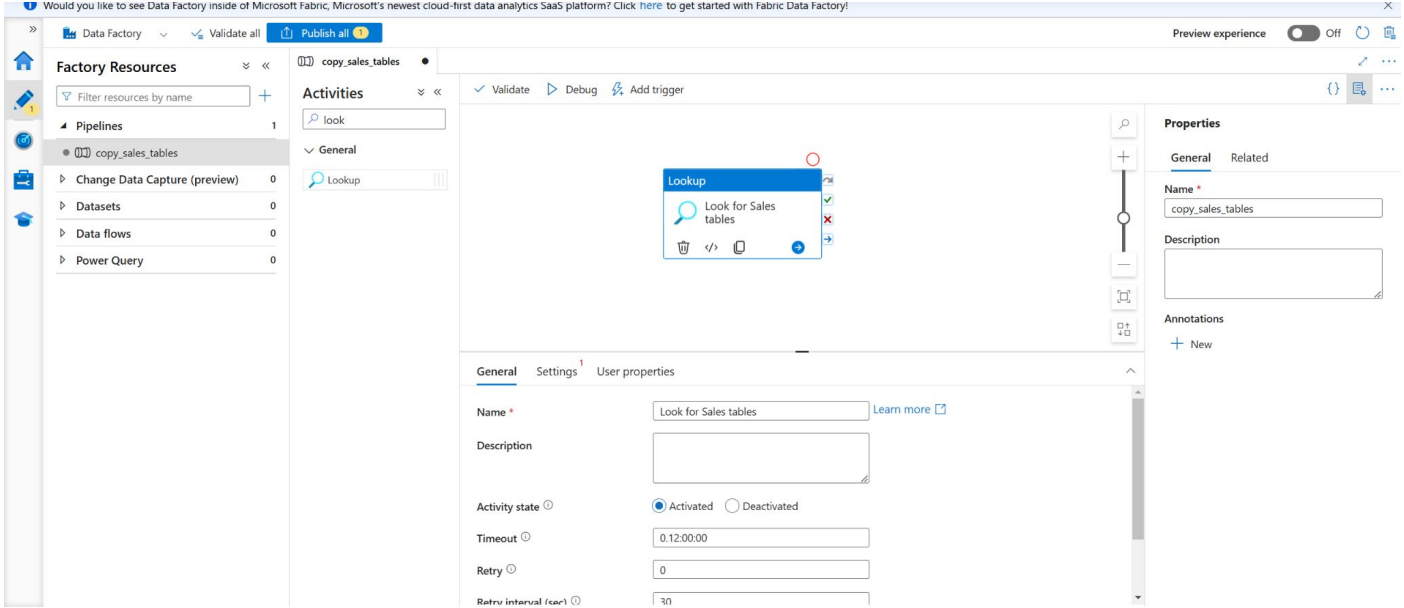
Showing 1 - 2 of 2 items

Name	Type	Sub-type	Status	Related	Region	Version
AutoResolveIntegrationR...	Azure	Public	Running	0	Auto Resolve	---
SHIR	Self-Hosted	---	Running (Limited)	0	---	5.46.9020.1

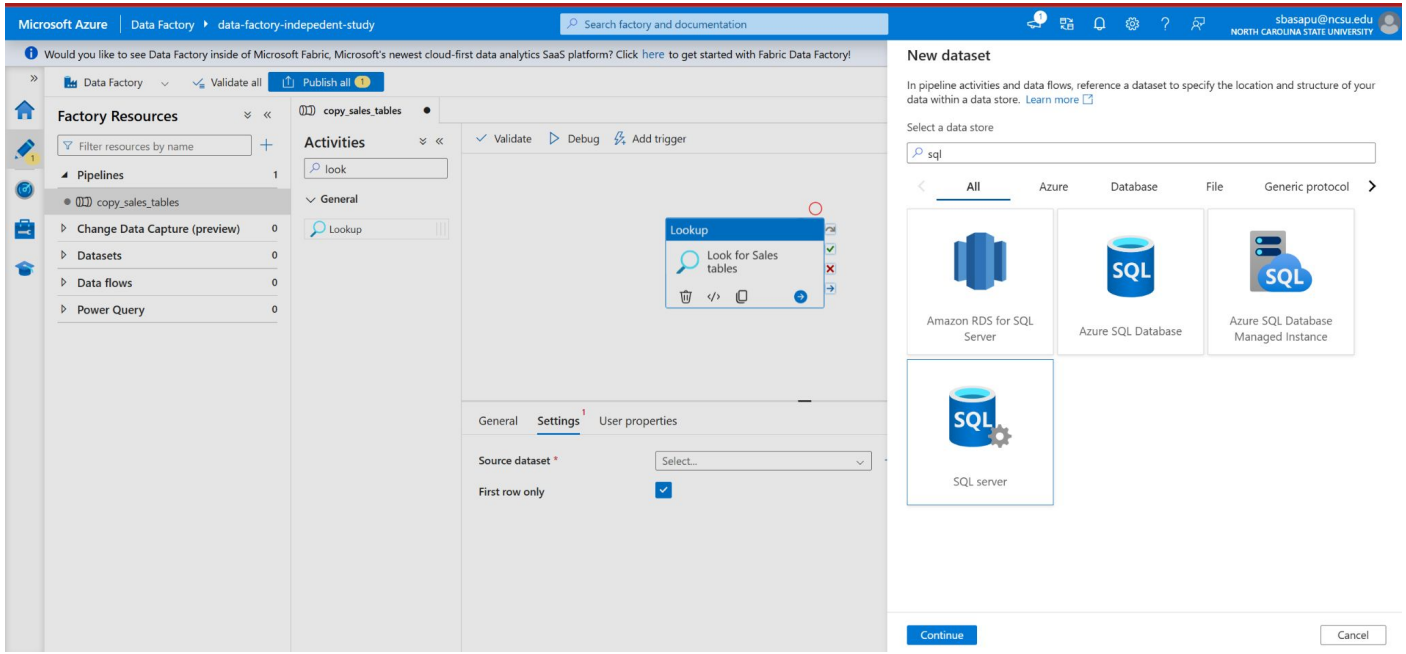
BR1: Data Migration



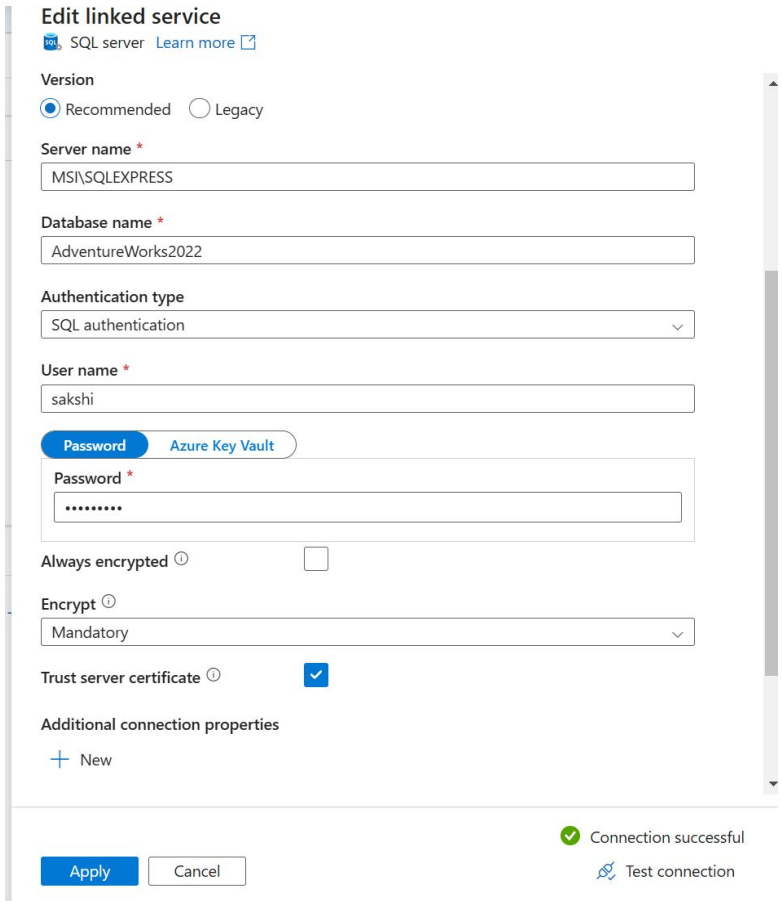
BR1: Data Migration



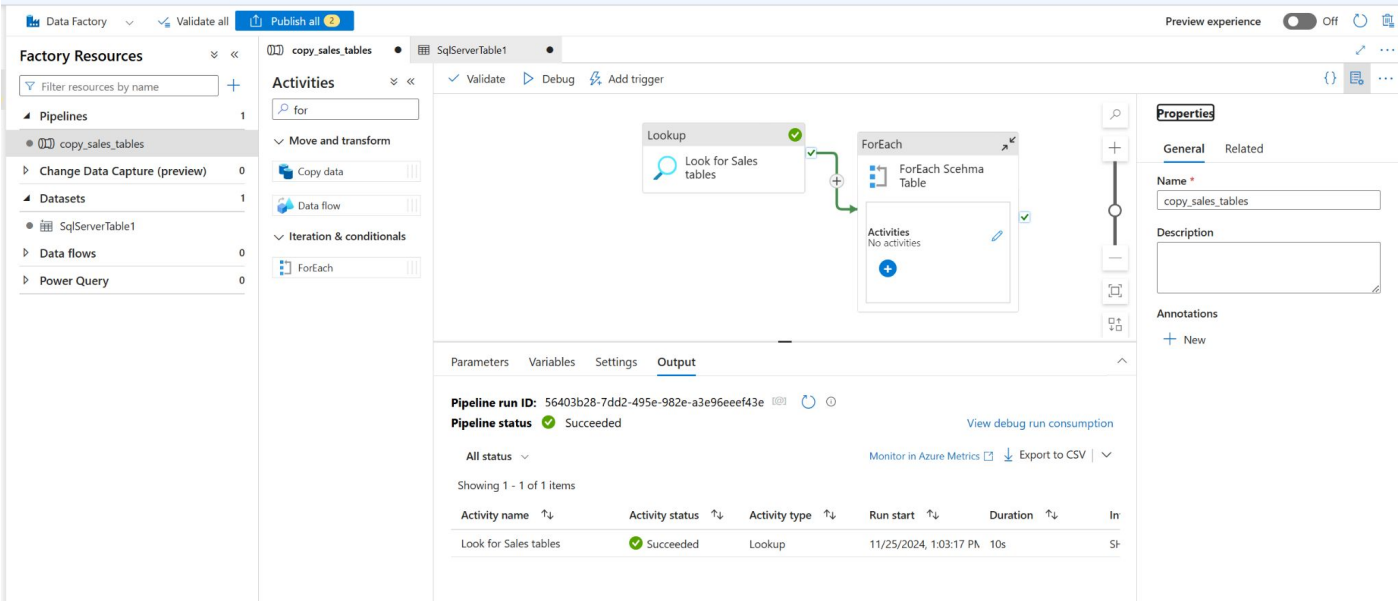
BR1: Data Migration



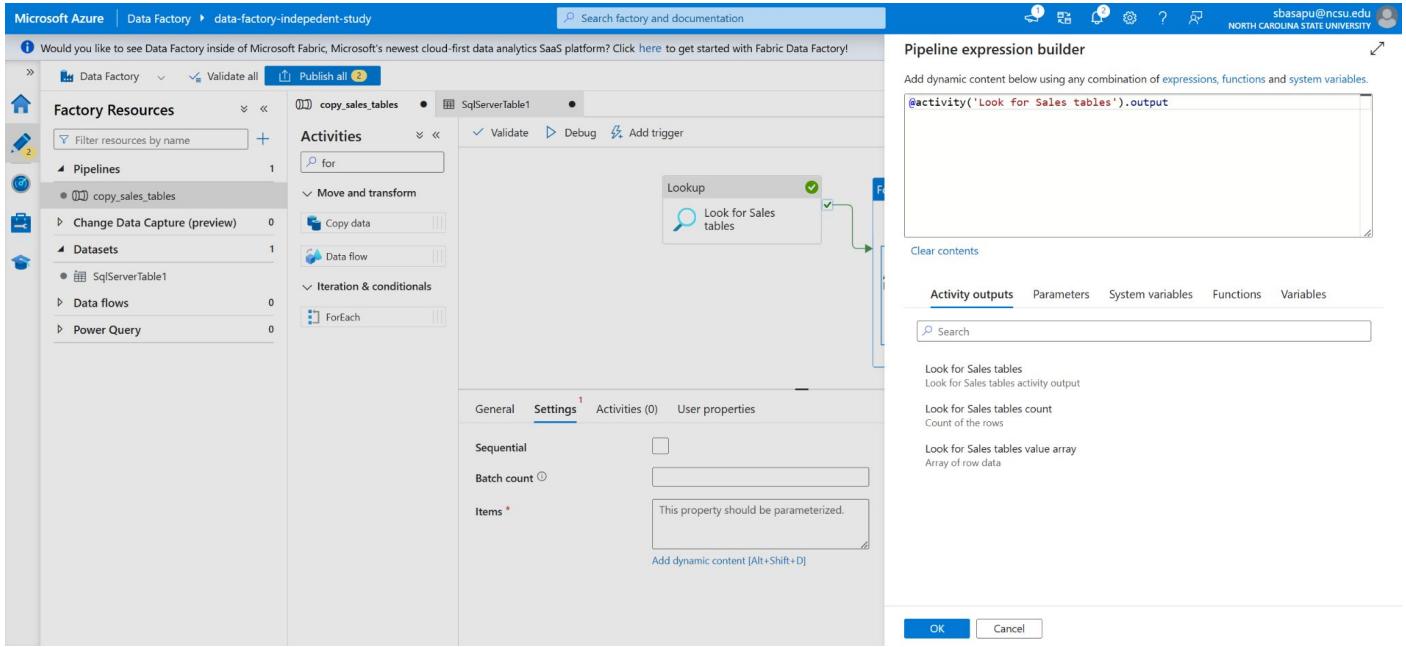
BR1: Data Migration



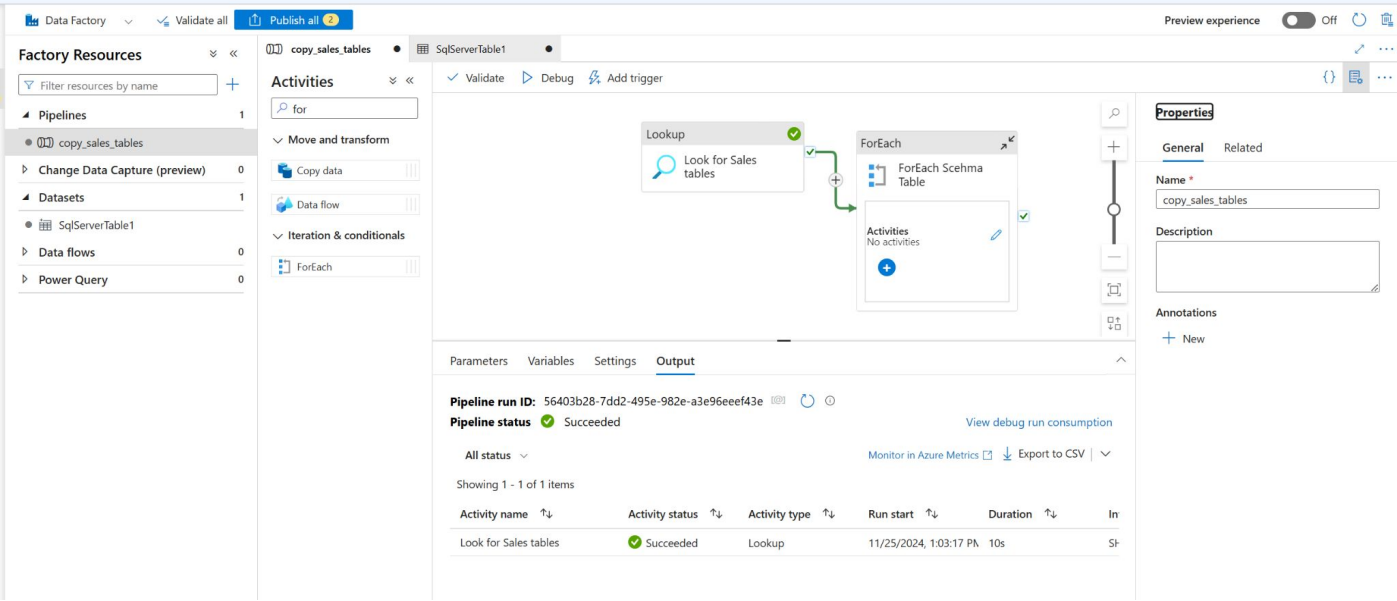
BR1: Data Migration



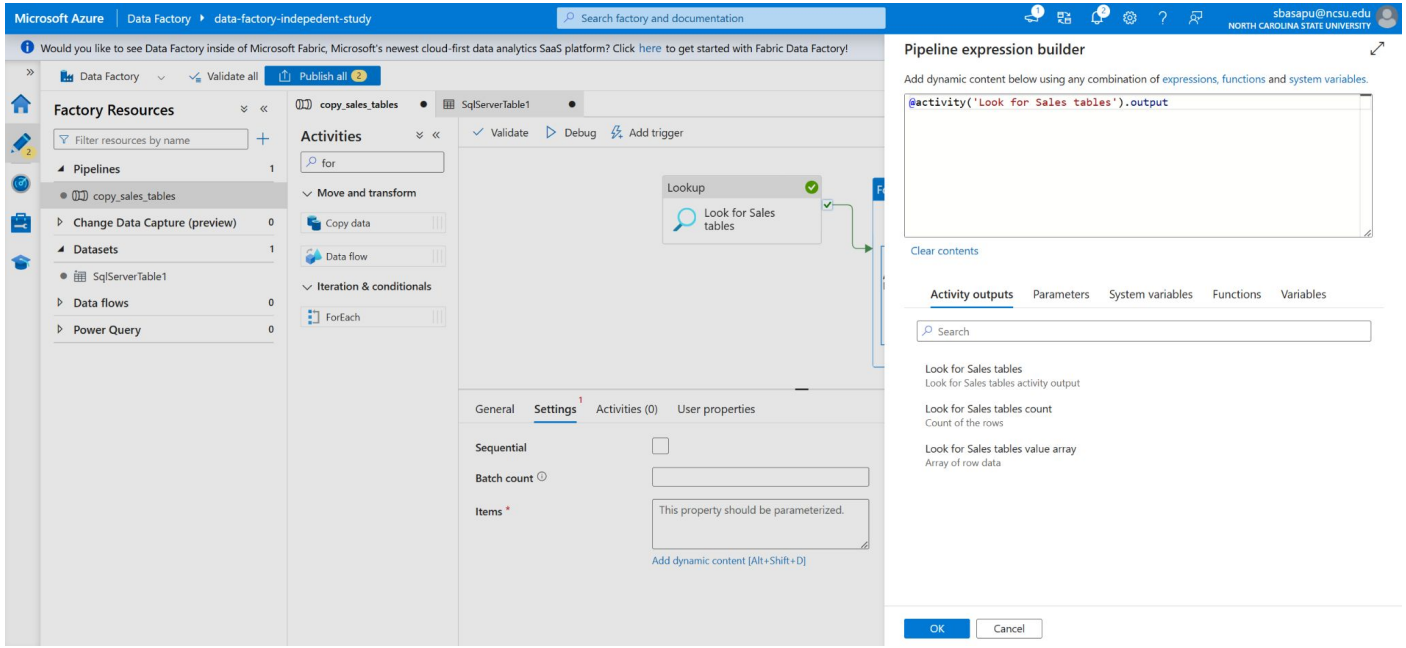
BR1: Data Migration



BR3: High Availability



BR3: High Availability



BR3: High Availability

Microsoft Azure | Data Factory ▸ data-factory-indepedent-study

Search factory and documentation

sbasapu@ncsu.edu
NORTH CAROLINA STATE UNIVERSITY

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 3 | Preview experience Off

Factory Resources

Filter resources by name

- Pipelines 1
  - copy_sales_tables
- Change Data Capture (preview) 0
- Datasets 2
  - SqlServerCopy
  - SqlServerTable1
- Data flows 0
- Power Query 0

copy_sales_tables | SqlServerTable1

Activities

Search activities

- Move and transform
  - Copy data
  - Data flow
- Synapse
- Azure Data Explorer
- Azure Function
- Batch Service
- Databricks
- Data Lake Analytics
- General
- HDInsight
- Iteration & conditionals
- Machine Learning
- Power Query

Validate | Validate copy runtime | Debug | Add trigger

copy_sales_tables > ForEach Schema Table

Copy data

Copy Each Table

Properties

General | Related

Name *: copy_sales_tables

Description

Annotations

+ New

General | Source | Sink | Mapping | Settings | User properties

Source dataset *: SqlServerCopy

Open | + New | Preview data | Learn more

Use query: Table | Query | Stored procedure

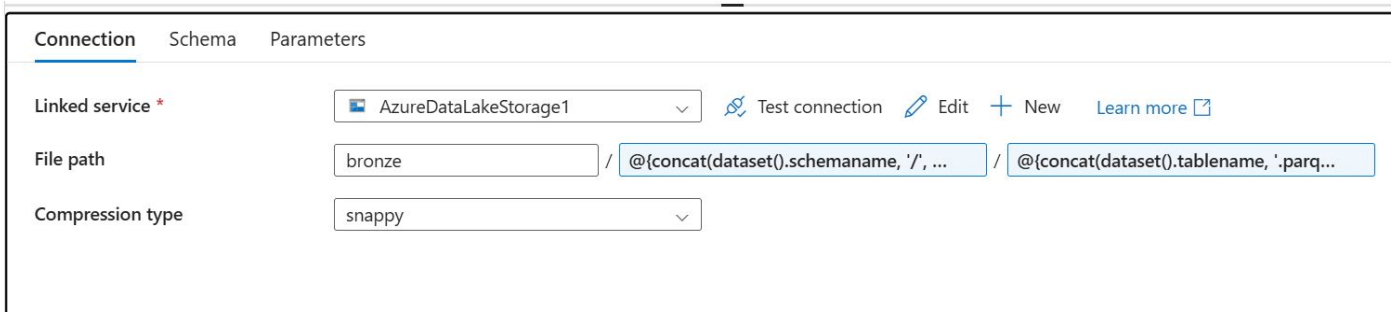
Query: @{concat('SELECT * FROM ', item().Sc...

Query timeout (minutes): 120

Isolation level: Select...

Partition option: None | Physical partitions of table | Dynamic range

BR3: High Availability



BR3: High Availability

Microsoft Azure | Data Factory ▸ data-factory-indepedent-study

Search factory and documentation

sbasapu@ncsu.edu
NORTH CAROLINA STATE UNIVERSITY

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!

Dashboards

Runs
- Pipeline runs
- Trigger runs
- Change Data Capture (previ...

Runtimes & sessions
- Integration runtimes
- Data flow debug

Notifications
- Alerts & metrics

Rerun | Cancel | Refresh | Update pipeline | List | Gantt

Activity runs

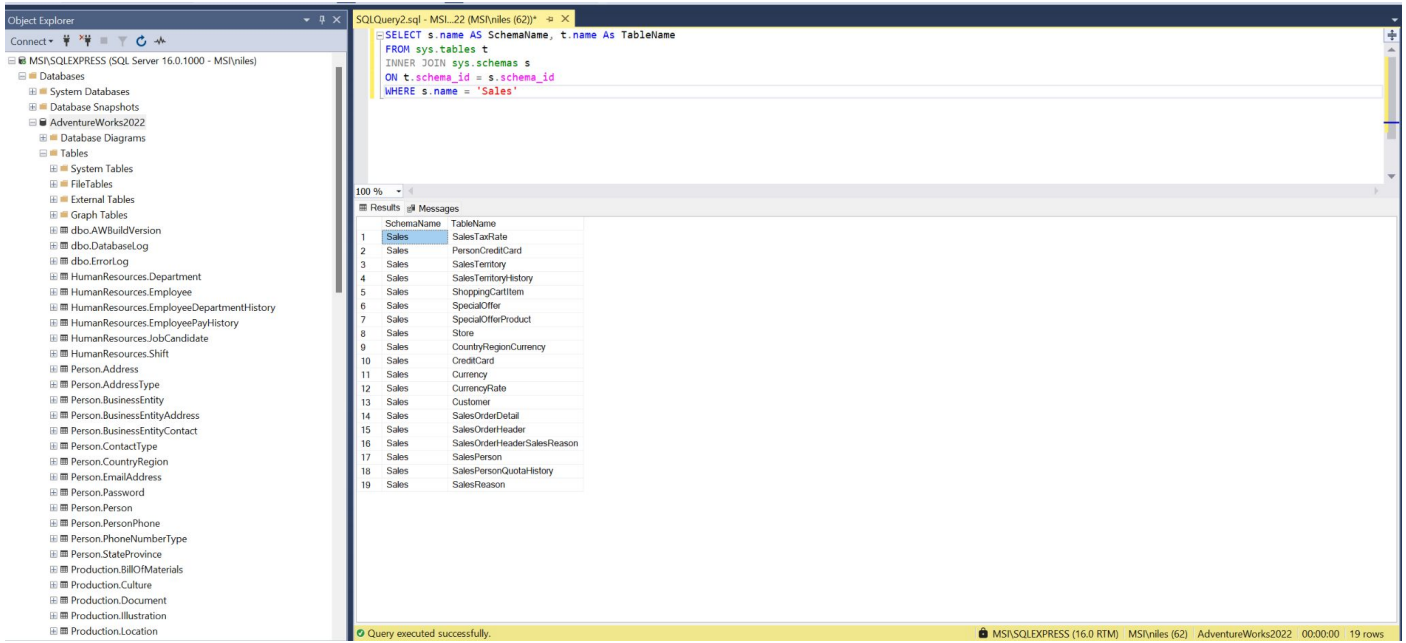
Pipeline run ID ed4593cb-f037-497a-b8ef-251fff55821d

All status | List | Monitor in Azure Metrics | Export to CSV

Showing 1 - 21 items

Activity name	Activity status	Activity type	Run start	Duration	Integration runtime	User properties	Activity run ID	Log
Look for Sales tables	Succeeded	Lookup	11/28/2024, 3:53:19 PM	12s	SHIR		5deb458c-177b-4397-8732-1350f03d9a90	
ForEach Scehma Table	Succeeded	ForEach	11/28/2024, 3:53:31 PM	1m 51s			7b59ec6a-bc8c-4b95-b8a0-0b18d26c5882	
Copy Each Table	Succeeded	Copy data	11/28/2024, 3:53:32 PM	1m 24s	SHIR		65a3db97-f9ea-450e-92a5-b551f982d2da	
Copy Each Table	Succeeded	Copy data	11/28/2024, 3:53:32 PM	44s	SHIR		12f3ea6c-1d92-4f01-bdcb-66368d663c19	
Copy Each Table	Succeeded	Copy data	11/28/2024, 3:53:32 PM	49s	SHIR		85a961f0-2b2b-42f9-9025-5e1937177f5e	
Copy Each Table	Succeeded	Copy data	11/28/2024, 3:53:32 PM	52s	SHIR		e76c6758-eb61-42d1-a2f9-1c2cb443038f	
Copy Each Table	Succeeded	Copy data	11/28/2024, 3:53:32 PM	48s	SHIR		6ae2727e-3469-4b06-bbed-76d86cc7e00a	
Copy Each Table	Succeeded	Copy data	11/28/2024, 3:53:32 PM	48s	SHIR		dfdbfd32-9299-41c4-a809-be49eb45d31b	
Copy Each Table	Succeeded	Copy data	11/28/2024, 3:53:32 PM	48s	SHIR		3406d430-733c-4e85-8b6d-f5b47c862b91	
Copy Each Table	Succeeded	Copy data	11/28/2024, 3:53:32 PM	46s	SHIR		af713751-3bcd-42a4-b8a9-5125dd540c5c	
Copy Each Table	Succeeded	Copy data	11/28/2024, 3:53:32 PM	46s	SHIR		f3a4ad08-9757-4e17-9676-aebeb6fccb80	
Copy Each Table	Succeeded	Copy data	11/28/2024, 3:53:32 PM	1m 22s	SHIR		961353e4-b74a-4b7c-b5c1-a2fe43059613	
Copy Each Table	Succeeded	Copy data	11/28/2024, 3:53:32 PM	47s	SHIR		e3bbc72e-b3f4-45cd-a0bf-fcf8c8fa859a	
Copy Each Table	Succeeded	Copy data	11/28/2024, 3:53:32 PM	1m 23s	SHIR		e6b8d1d4-6509-4a33-8b09-7b96e14adef4	
Copy Each Table	Succeeded	Copy data	11/28/2024, 3:53:32 PM	49s	SHIR		4091122b-6bce-4376-8b25-a94050317d43	

BR3: High Availability



BR3: High Availability

Microsoft Azure | Search resources, services, and docs (G+/) | Copilot | sbasapu@ncsu.edu NORTH CAROLINA STATE UNIVE...

Home > indstudystorageaccount | Containers >

bronze
Container

Search

- Overview
- Diagnose and solve problems
- Access Control (IAM)
- Settings

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 / Sales

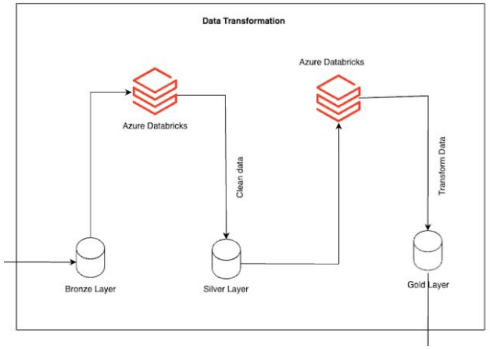
Search blobs by prefix (case-sensitive) | Show deleted objects

Name	Modified	Access tier	Archive status	Blob type	Size	Lease state
[..]						
CountryRegionCurrency	11/28/2024, 3:54:14 PM					-
CreditCard	11/28/2024, 3:54:16 PM					-
Currency	11/28/2024, 3:54:12 PM					-
CurrencyRate	11/28/2024, 3:54:16 PM					-
Customer	11/28/2024, 3:54:22 PM					-
PersonCreditCard	11/28/2024, 3:54:16 PM					-
SalesOrderDetail	11/28/2024, 3:54:22 PM					-
SalesOrderHeader	11/28/2024, 3:54:19 PM					-
SalesOrderHeaderSalesReason	11/28/2024, 3:54:17 PM					-
SalesPerson	11/28/2024, 3:55:18 PM					-
SalesPersonQuotaHistory	11/28/2024, 3:54:17 PM					-
SalesReason	11/28/2024, 3:54:14 PM					-
SalesTaxRate	11/28/2024, 3:54:53 PM					-
SalesTerritory	11/28/2024, 3:54:51 PM					-

BR2: Data Transformation

The data will be processed through **three layers**:

- **Bronze Layer**: Raw, unprocessed data stored in its original form.
- **Silver Layer**: Cleaned and transformed data, where data quality and consistency improvements are applied.
- **Gold Layer**: Fully refined and aggregated data, ready for reporting, advanced analytics, and business insights.



BR2: Data Transformation

Home >

rg-independent-study_ind-study-databricks | Overview

Deployment

Search

Overview
Inputs
Outputs
Template

Delete Cancel Redeploy Download Refresh

Your deployment is complete

Deployment name : rg-independent-study_ind-study-databricks
Subscription : Azure for Students
Resource group : rg-independent-study

Start time : 11/28/2024, 6:06:55 PM
Correlation ID : a53ff3dc-1117-450e-8420-109bdf61ea5d

Deployment details

Resource	Type	Status	Operation details
ind-study-databricks	Azure Databricks Service	OK	Operation details

Next steps

Go to resource

Give feedback

Tell us about your experience with deployment

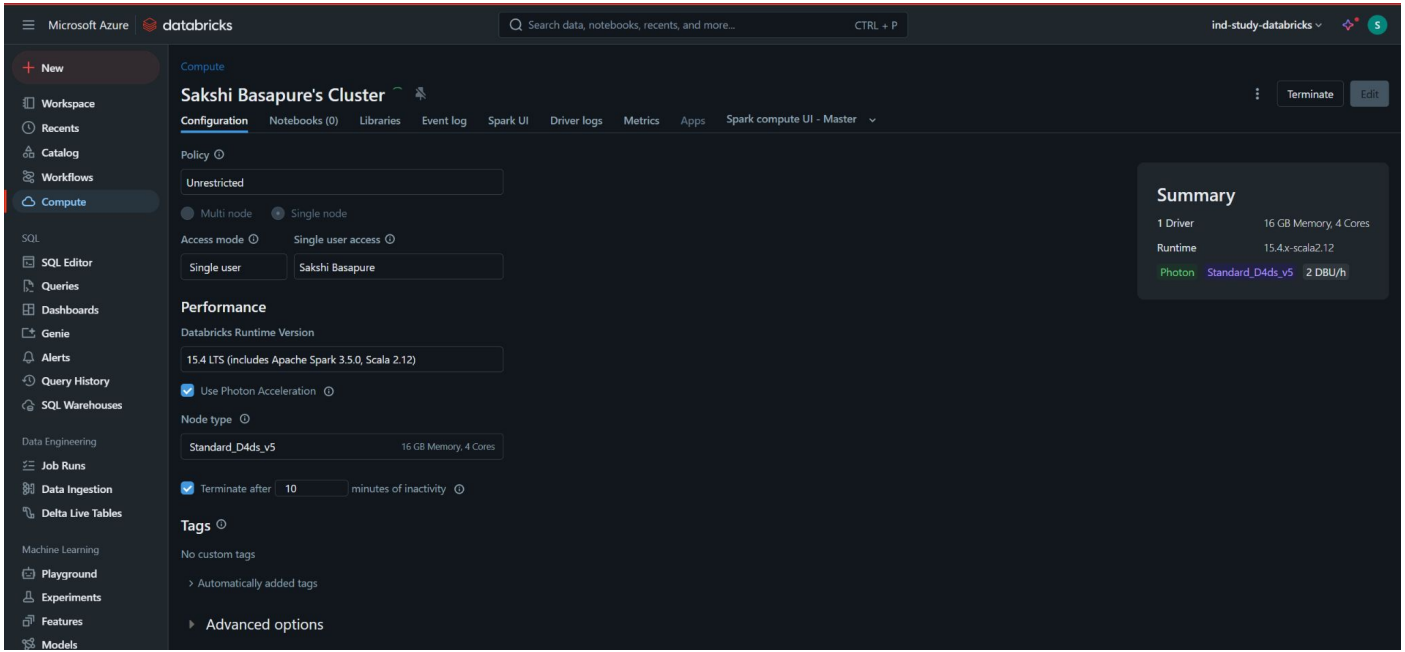
Cost management
Get notified to stay within your budget and prevent unexpected charges on your bill.
Set up cost alerts >

Microsoft Defender for Cloud
Secure your apps and infrastructure
Go to Microsoft Defender for Cloud >

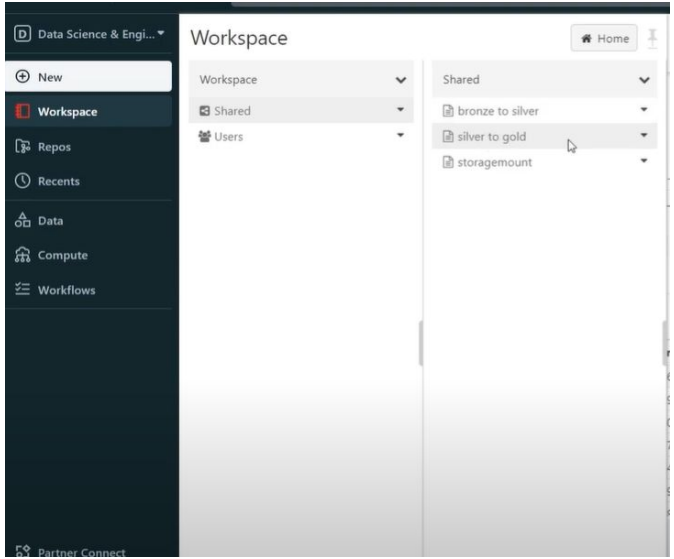
Free Microsoft tutorials
Start learning today >

Work with an expert
Azure experts are service provider partners who can help manage your assets on Azure and be your first line of support.
Find an Azure expert >

BR2: Data Transformation



BR2: Data Transformation



BR2: Data Transformation

StorageMount.ipynb

File Edit View Insert Runtime Tools Help Last edited on November 29

Share

+ Code + Text

Connect Gemini

This code is used for mounting Azure Data Lake containers to a Databricks environment in order to access, process, and manipulate the raw data stored in the Azure Data Lake. Specifically, the code mounts three different containers (Bronze, Silver, and Gold layers) into Databricks so that their data can be easily accessed for downstream processing and analysis.

```
[ ] # Configuration dictionary for mounting Azure Data Lake Storage Gen2
    configs = {
        # Authentication type set to CustomAccessToken for secure access
        "fs.azure.account.auth.type": "CustomAccessToken",

        # Specifies the class name of the token provider for Databricks passthrough authentication
        "fs.azure.account.custom.token.provider.class": spark.conf.get("spark.databricks.passthrough.adls.gen2.tokenProviderClassName")
    }

    # Mount the Azure Data Lake Storage Gen2 container to Databricks file system (DBFS)
    dbutils.fs.mount(
        source = "abfss://bronze@deprojdatalakegen2.dfs.core.windows.net/",  # Source URI of the ADLS Gen2 container
        mount_point = "/mnt/bronze",                                          # Mount point path in DBFS
        extra_configs = configs                                              # Additional con
```

```
True
```

```
[ ] dbutils.fs.ls("/mnt/bronze/SalesLT")
```

```
[FileInfo(path='dbfs:/mnt/bronze/SalesLT/Address/', name='Address/', size=0, modificationTime=1705744334000),
 FileInfo(path='dbfs:/mnt/bronze/SalesLT/Customer/', name='Customer/', size=0, modificationTime=1705744330000),
 FileInfo(path='dbfs:/mnt/bronze/SalesLT/CustomerAddress/', name='CustomerAddress/', size=0, modificationTime=1705744354000),
 FileInfo(path='dbfs:/mnt/bronze/SalesLT/Product/', name='Product/', size=0, modificationTime=1705744336000),
 FileInfo(path='dbfs:/mnt/bronze/SalesLT/ProductCategory/', name='ProductCategory/', size=0, modificationTime=1705744343000),
 FileInfo(path='dbfs:/mnt/bronze/SalesLT/ProductDescription/', name='ProductDescription/', size=0, modificationTime=1705744351000),
 FileInfo(path='dbfs:/mnt/bronze/SalesLT/ProductModel/', name='ProductModel/', size=0, modificationTime=1705744332000),
 FileInfo(path='dbfs:/mnt/bronze/SalesLT/ProductModelProductDescription/', name='ProductModelProductDescription/', size=0, modificationTime=1705744353000),
```

BR2: Data Transformation

BronzeToSilver.ipynb

File Edit View Insert Runtime Tools Help Last edited on November 29

Share

+ Code + Text

Connect Gemini

```python
# Iterate through each table name in the list 'table_name'
for i in table_name:
    # Construct the file path for reading data from the bronze layer
    path = "/mnt/bronze/SalesLT/" + i + "/" + i + ".parquet"

    # Load the parquet file into a DataFrame
    df = spark.read.format('parquet').load(path)

    # Retrieve the list of column names in the DataFrame
    columns = df.columns

    # Iterate through each column in the DataFrame
    for col in columns:
        # Check if the column name contains "Date" or "date" (case-sensitive)
        if "Date" in col or "date" in col:
            # Convert the column to a timestamp, adjust to UTC timezone,
            # and reformat the date as "yyyy-MM-dd"
            df = df.withColumn(
                col,
                date_format(
                    from_utc_timestamp(
                        df[col].cast(TimestampType()), "UTC"  # Cast column to timestamp and adjust timezone
                    ),
                    "yyyy-MM-dd"  # Format the date as 'year-month-day'
                )
            )

    # Define the output path for saving the transformed DataFrame to the silver layer
    output_path = "/mnt/silver/SalesLT/" + i + "/"

    # Write the transformed DataFrame to the specified output path in Delta format
    # Use 'overwrite' mode to replace any existing data at the target location
    df.write.format("delta").mode("overwrite").save(output_path)
```

BR2: Data Transformation

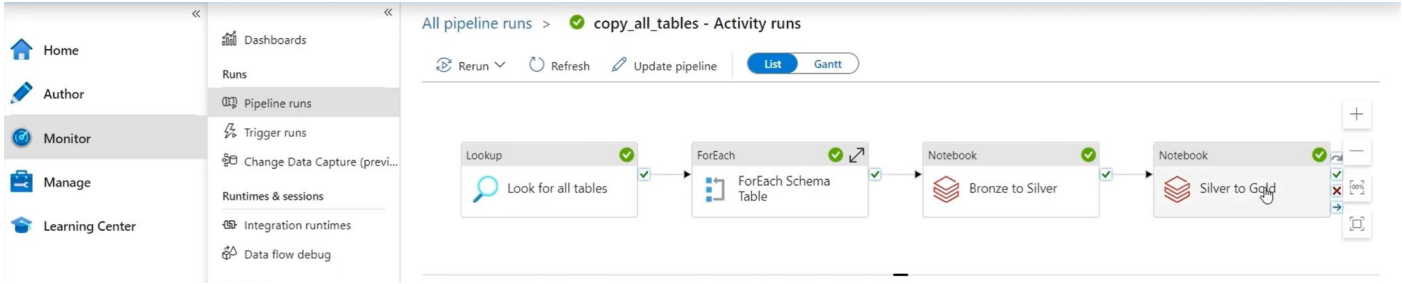
SilverToGold.ipynb

File Edit View Insert Runtime Tools Help Last edited on November 29

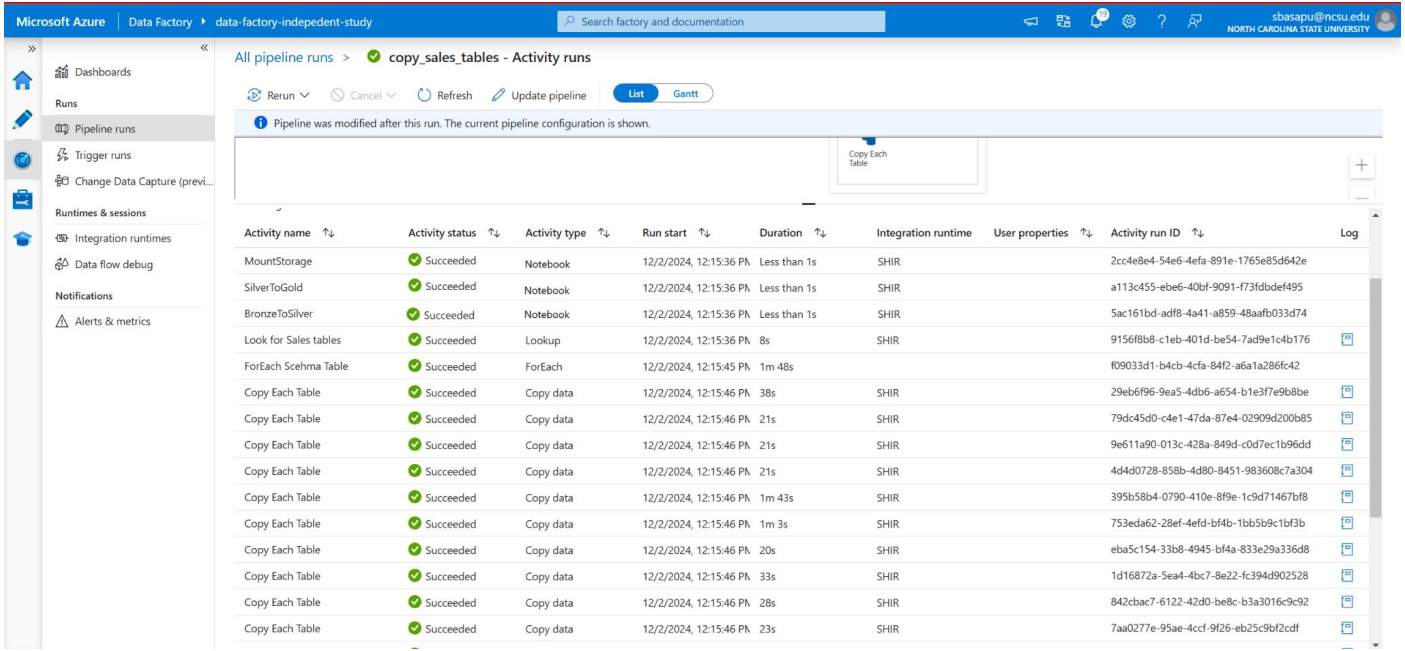
+ Code + Text

```python
# Iterate through each table name in the list 'table_name'
for name in table_name:
    # Define the path for reading data from the silver layer for the current table
    path = "/mnt/silver/SalesLT/" + name
    print(path)  # Print the path for debugging or logging purposes

    # Load the Delta table from the specified path into a DataFrame
    df = spark.read.format("delta").load(path)

    # Get the list of column names in the DataFrame
    column_names = df.columns

    # Iterate through each column name to rename it
    for old_col_name in column_names:
        # Convert the column name from CamelCase (e.g., "ColumnName") to snake_case (e.g., "column_name")
        # This logic inserts an underscore before any uppercase character that is not preceded by another uppercase character
        new_col_name = "".join([
            "_" + char if char.isupper() and not old_col_name[i-1].isupper() else char
            for i, char in enumerate(old_col_name)
        ]).lstrip("_")  # Remove any leading underscore added at the start of the column name

        # Rename the column in the DataFrame
        df = df.withColumnRenamed(old_col_name, new_col_name)

    # Define the output path for saving the transformed DataFrame to the gold layer
    output_path = "/mnt/gold/SalesLT/" + name + "/"

    # Write the transformed DataFrame to the specified output path in Delta format
    # 'overwrite' mode ensures that any existing data at the path is replaced
    df.write.format('delta').mode('overwrite').save(output_path)
```

BR2: Data Transformation



BR2: Data Transformation



Activity name	Activity status	Activity type	Run start	Duration	Integration runtime	User properties	Activity run ID	Log
MountStorage	Succeeded	Notebook	12/2/2024, 12:15:36 PM	Less than 1s	SHIR		2cc4e8e4-54e6-4efa-891e-1765e85d642e	
SilverToGold	Succeeded	Notebook	12/2/2024, 12:15:36 PM	Less than 1s	SHIR		a113c455-ebe6-40bf-9091-f73fdbdef495	
BronzeToSilver	Succeeded	Notebook	12/2/2024, 12:15:36 PM	Less than 1s	SHIR		5ac161bd-adf8-4a41-a859-48aafb033d74	
Look for Sales tables	Succeeded	Lookup	12/2/2024, 12:15:36 PM	8s	SHIR		9156f8b8-c1eb-401d-be54-7ad9e1c4b176	
ForEach Scehma Table	Succeeded	ForEach	12/2/2024, 12:15:45 PM	1m 48s			f09033d1-b4cb-4cfa-84f2-a6a1a286fc42	
Copy Each Table	Succeeded	Copy data	12/2/2024, 12:15:46 PM	38s	SHIR		29eb6f96-9ea5-4db6-a654-b1e3f7e9b8be	
Copy Each Table	Succeeded	Copy data	12/2/2024, 12:15:46 PM	21s	SHIR		79dc45d0-c4e1-47da-87e4-02909d200b85	
Copy Each Table	Succeeded	Copy data	12/2/2024, 12:15:46 PM	21s	SHIR		9e611a90-013c-428a-849d-c0d7ec1b96dd	
Copy Each Table	Succeeded	Copy data	12/2/2024, 12:15:46 PM	21s	SHIR		4d4d0728-858b-4d80-8451-983608c7a304	
Copy Each Table	Succeeded	Copy data	12/2/2024, 12:15:46 PM	1m 43s	SHIR		395b58b4-0790-410e-8f9e-1c9d71467bf8	
Copy Each Table	Succeeded	Copy data	12/2/2024, 12:15:46 PM	1m 3s	SHIR		753eda62-28ef-4efd-bf4b-1bb5b9c1bf3b	
Copy Each Table	Succeeded	Copy data	12/2/2024, 12:15:46 PM	20s	SHIR		eba5c154-33b8-4945-bf4a-833e29a336d8	
Copy Each Table	Succeeded	Copy data	12/2/2024, 12:15:46 PM	33s	SHIR		1d16872a-5ea4-4bc7-8e22-fc394d902528	
Copy Each Table	Succeeded	Copy data	12/2/2024, 12:15:46 PM	28s	SHIR		842cbac7-6122-42d0-be8c-b3a3016c9c92	
Copy Each Table	Succeeded	Copy data	12/2/2024, 12:15:46 PM	23s	SHIR		7aa0277e-95ae-4ccf-9f26-eb25c9bf2cdf	

BR2: Data Transformation

Upload | Add Directory | Refresh | Rename | Delete | Change tier | Acquire lease | Break lease | Give feedback

Authentication method: Access key (Switch to Azure AD User Account)
Location: silver / SalesLT / Customer

Search blobs by prefix (case-sensitive) | Show deleted objects

Name	Modified	Access tier	Archive status	Blob type	Size	Lease state
[..]						
_delta_log						-
part-00000-6b145ce2-1de8-4cfe-830f-3ca8696c86...	4/17/2023, 12:23:12 ...	Hot (Inferred)		Block blob	88.33 KiB	Available

Upload | Add Directory | Refresh | Rename | Delete | Change tier | Acquire lease | Break lease | Give feedback

Authentication method: Access key (Switch to Azure AD User Account)
Location: gold / SalesLT / Address

Search blobs by prefix (case-sensitive) | Show deleted objects

Name	Modified	Access tier	Archive status	Blob type	Size	Lease state
[..]						
_delta_log						-
part-00000-bd1df7a5-1ab5-467b-9c67-1cafceea4...	4/17/2023, 12:31:48 ...	Hot (Inferred)		Block blob	34.76 KiB	Available

Learnings

1. Importance of Secure Data Handling
2. Scalability in Cloud Architecture
3. Data Transformation and Layered Architecture
4. Real-Time Data Integration

Challenges

1. Design Challenges:
    a. Architectural Integration
    b. Balancing Performance and Cost Efficiency
2. Implementation Challenges:
    a. Setting Up Self-Hosted Integration Runtime (SHIR)
    b. Debugging Multi-Service Pipelines

Acknowledgement

A heartfelt thanks to Prof. Viniotis for their invaluable guidance and mentorship throughout this project.

The foundational knowledge from courses like "Cloud Computing Architecture" and "Advanced Cloud Computing Architecture" greatly contributed to designing and implementing this architecture.