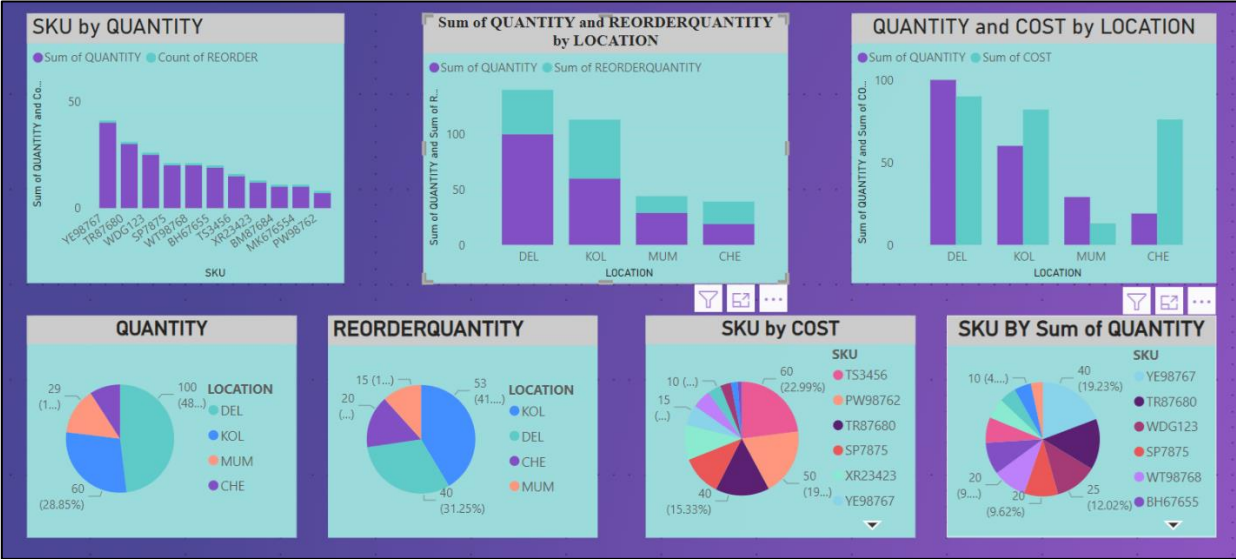


End-to-End Data Engineering for an Inventory management Database on SAP-HANA

21-08-2024 Mageshwaran.V , mageshwaran@dataeverconsulting.com



Contents

1. Introduction.....	4
2. Objective	4
3. Pre requisites	4
1. Azure cloud A/C	4
2. A Local System with.....	4
4. Solution Architecture	5
5. Solution Components.....	5
5.1 Sap Hana Database.....	5
5.2 HDBODBC Installation steps are table 2	5
5.3 Azure Cloud.....	6
5.4 Data Integration	6
5.5 Data Factory and pipeline	6
5.6 Virtual Machine on Azure	6
5.7 Integration Runtime [4]	6
5.8 Azure integration runtime	6
5.9 Self-hosted integration runtime (SHIR).....	6
5.10 Azure Blob Storage	6
5.11 Azure Data Lake Storage Gen2	7
5.12 Data Lake solution	7
5.13 Trigger	7
5.14 Azure DATA BRICKS.....	7
5.15 Azure Synapse Analytics	7
5.16 Azure storage account	7
5.17 ETL (Extract, Transform, Load)	7
5.18 SQL (Structured Query Language)	8
5.19 PYSPARK	8
5.20 Power BI.....	8
6. Inventory management dataset	8
6.1 SKU (Stock Keeping Unit)	8
6.2 DESCRIPTION.....	8
6.3 Bin Number	9
6.4 LOCATION.....	9
6.5 UNIT.....	9
6.6 Quantity.....	9

6.7 REORDERQUANTITY.....	9
6.8 COST	9
6.9 Inventory Value	9
6.10 REORDER.....	9
7. isql HDBODBC	11
8. Azure Local Files System to Data Transformation Cloud	12
9. TROUBLESHOOTING in Synapse Workspace:.....	17
10. ODBC for SAP HANA trouble shooting	18
11. Conclusion and Next steps	20
11.1 Sum of Quantity and Reorder Quantity by Location:	20
12. References:	21

End-to-End Data Engineering for an Inventory management Database on SAP-HANA

1. Introduction

End-to-End Data Engineering for an Inventory Management Database on SAP HANA involves designing, implementing, and optimizing data pipelines that seamlessly integrate, store, and analyse inventory data. Leveraging this process ensures real-time data processing, accurate inventory tracking, and efficient resource management. By enabling advanced analytics and reporting, it supports informed decision-making and enhances overall operational efficiency in inventory management

2. Objective

In our endeavour to do end-to-end data engineering projects, we have so far completed projects involving source data sets such as Json/AWS and On-Prem MS-SQL DB

In continuing with this effort, in this proof of concept (POC) our objective is to do an end-to-end data engineering involving SAP-HANA Database as source data set.

3. Pre requisites

1. Azure cloud A/C

- Data Factory
- Storage Account
- Data Bricks
- Synapse Analytics
- Virtual Machine

2. A Local System with

- Power BI Desktop
- ODBC
- SAP-HANA Express Client
- Self-Hosted Integration Runtime

4. Solution Architecture

This Section Presence the Architecture Diagram for an End-to-End Data Engineering solution of an Inventory management Database on SAP-HANA. The backbone solution architecture is a well-known data engineering pattern know as ETL. The ETL stages or indicated in the form of tasks in solution architecture. Also, different data layers are shown in this diagram.

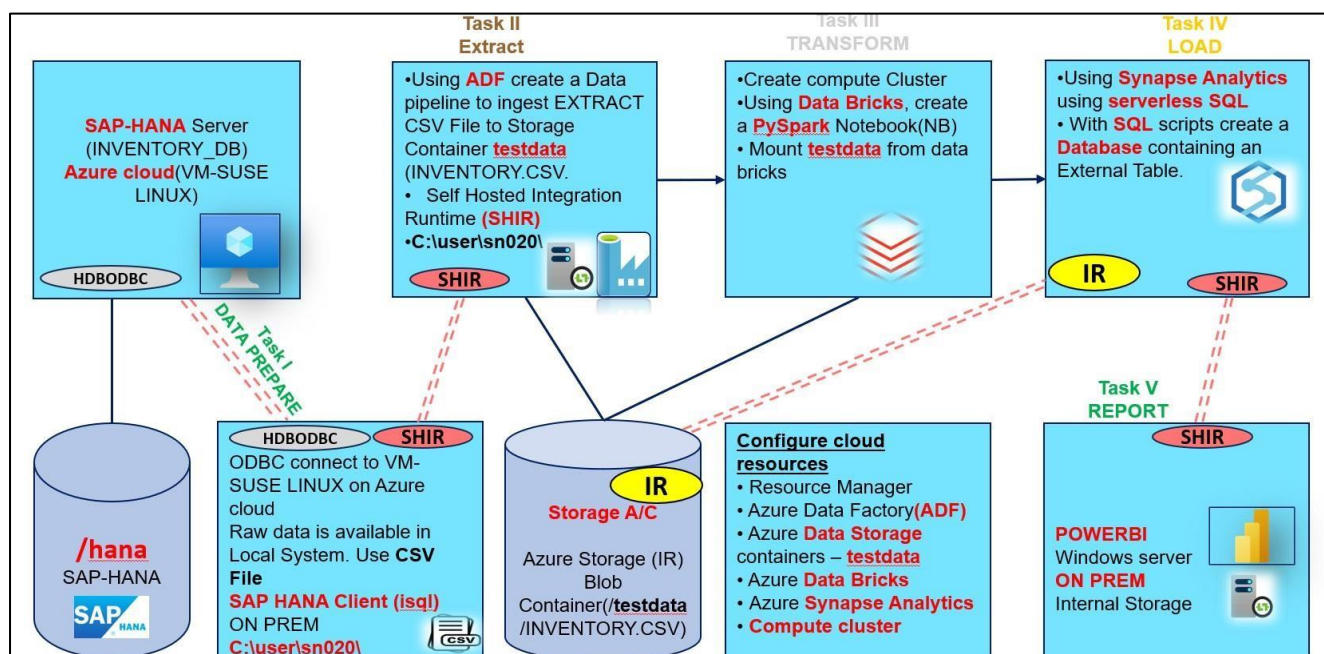


Figure 1: Solution Architecture Diagram

5. Solution Components

5.1 Sap Hana Database

Installation steps are in table 1 this version of sap Hana we are using does not include SAP applications are ABAP frame work. Our main motivation was to replicate the work reported in reference [1]. For want of licences, we did not pursue that line. Also, we would have saved some time if we had sap Hana studio but we could not download SAPHANA studio. To use this one requires licences

5.2 HDBODBC Installation steps are table 2

HDBODBC is the ODBC (Open Database Connectivity) driver provided by SAP HANA, enabling applications to connect to and interact with the SAP HANA database using the ODBC standard.

HDBODBC is typically included in the SAP HANA client installation and requires proper configuration, including setting up a DSN (Data Source Name) to connect to the SAP HANA server.

5.3 Azure Cloud

Azure is the only consistent hybrid cloud, delivers unparalleled developer productivity, provides comprehensive, multilayered security, including the largest compliance coverage of any cloud provider, and you'll pay less for Azure as AWS is five times more expensive than Azure for Windows Server and SQL Server.

5.4 Data Integration

SAP HANA Cloud integrates with various data sources, including SAP and non-SAP systems, cloud storage, and external databases. This integration capability supports data management and analytics across different platforms.

5.5 Data Factory and pipeline

A pipeline is a logical grouping of activities that together perform a task. A pipeline could contain a set of activities that ingest and clean log data, and then kick off a mapping data flow to analyse the log data.

ADF includes a rich set of data transformation activities to clean, transform, and enrich data during the ETL process.

Then, use a data flow activity or a Databricks Notebook activity to process and transform data from the blob storage to an Azure Synapse Analytics pool on top of which business intelligence reporting solutions are built.

5.6 Virtual Machine on Azure

An Azure Virtual Machine (VM) is an on-demand, scalable computing resource offered by Microsoft Azure that allows users to run various operating systems and applications, including SUSE Linux, in the cloud.

5.7 Integration Runtime [4]

The Integration Runtime (IR) is the compute infrastructure used by Azure Data Factory and Azure Synapse pipelines to provide the data flow and data movement activities. There are two types of IR as explained below

5.8 Azure integration runtime

Azure integration runtime provides a fully managed, serverless compute in Azure. Azure integration runtime provides the native compute to move data between cloud data stores in a secure, reliable, and high-performance manner.

5.9 Self-hosted integration runtime (SHIR)

If you want to perform data integration securely in a private network environment that doesn't have a direct line-of-sight from the public cloud environment, you can install a selfhosted IR in your on-premises environment. Currently, the self-hosted IR is only supported on a Windows operating system.

5.10 Azure Blob Storage

Azure Blob Storage is Microsoft's object storage solution for the cloud. Blob Storage is optimized for storing massive amounts of unstructured data. Unstructured data is data that doesn't adhere to a particular data model or definition, such as text or binary data

5.11 Azure Data Lake Storage Gen2

Azure Data Lake Storage Gen2 is a set of capabilities dedicated to big data analytics, built on Azure Blob Storage. Data Lake Storage Gen2 converges the capabilities of Azure Data Lake Storage Gen1 with Azure Blob Storage

5.12 Data Lake solution

Data lakes are distributed data stores that can hold very large volumes of diverse data. They can be used to store different types of data such as structured SQL, semi-structured, unstructured email, images, streaming sensor data, and so on.

A data lake solution usually comprises a storage layer, a compute layer, and a serving layer. The compute layers on cloud include Extract, Transform, Load (ETL); Batch; or Stream processing.

5.13 Trigger

A trigger is responsible for executing an Azure function and there are dozens of triggers to choose from. This module will show you some of the most common types of triggers and how to configure them to execute your logic

5.14 Azure DATA BRICKS

In Azure Databricks, while creating the cluster, you can select Enable Autoscaling and specify the number of Min Works and Max Workers. The cluster will automatically scale up and down between two numbers based on the load.

Azure Databricks provides the latest versions of Apache Spark and allows you to seamlessly integrate with open-source libraries. Spin up clusters and build quickly in a fully managed Apache Spark environment with the global scale and availability of Azure

In data bricks as part of transformation, we perform some simple operation like inserting new currency a new column

5.15 Azure Synapse Analytics

Synapse Analytics supports metadata sharing among its computational pools such as Spark and SQL. Spark databases and external tables that are created using parquet format can be easily accessed from the SQL pools.

Any database you create using Spark SQL can be directly accessed by the dedicated or serverless SQL pools in Synapse, provided both these pools have storage-level access to the newly created database.

5.16 Azure storage account

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.

5.17 ETL (Extract, Transform, Load)

ETL (Extract, Transform, Load) in Azure Data Engineering is a critical process that involves extracting data from various sources, transforming it into a usable format, and loading it into a target data store for analysis or further processing. In Azure, ETL is typically managed using services like Azure Data Factory, Azure Synapse Analytics for data warehousing, Azure Data

Lake for scalable storage, and Azure Databricks for advanced data transformation and machine learning tasks.

The process is essential for ensuring that data is clean, structured, and optimized for analytics, enabling organizations to derive actionable insights from vast amounts of data across different systems. Azure's ETL tools offer robust integration capabilities, scalability, and automation, making them vital for modern data engineering workflows.

5.18 SQL (Structured Query Language)

SQL (Structured Query Language) is a standardized programming language used for managing and manipulating relational databases. SQL is the foundation for interacting with database systems like MySQL, Microsoft SQL Server, PostgreSQL, and Oracle Database.

5.19 PYSPARK

PySpark is the Python API for Apache Spark, an open-source, distributed computing system that provides an easy-to-use interface for big data processing and analytics. PySpark allows users to leverage the power of Apache Spark's distributed computing capabilities while using Python, a popular programming language known for its simplicity and rich ecosystem of libraries. With PySpark, users can perform complex data transformations, run large-scale machine learning models, and process real-time data streams across a cluster of computers. It supports operations on Resilient Distributed Datasets (RDDs) and Data Frames, enabling efficient handling of large datasets in a scalable and fault-tolerant manner. PySpark is widely used in data engineering, data science, and big data analytics to process and analyse large datasets quickly and efficiently.

5.20 Power BI

Power BI is a business analytics service provided by Microsoft that enables users to visualize data and share insights. It allows users to connect to multiple data sources, transform and model data, and create interactive reports and dashboards. Power BI is used on-premises, as a tool for data analysis, reporting, and business intelligence.

6. Inventory management dataset

Getting a real manufacturing industry dataset is a big challenge. We also faced the same challenge. So, we took a representative dataset for our work. Given below is an inventory management dataset. However, we can follow the same process for a full dataset as well. We plan to do this in future

6.1 SKU (Stock Keeping Unit)

A unique identifier for each distinct product or item in inventory.

If a store sells three types of t-shirts (small, medium, large), each size might have its own SKU

6.2 DESCRIPTION

A brief text that describes the product or item associated with the SKU. This might include details like the product name, size, colour, or other distinguishing features.

6.3 Bin Number

A specific identifier used to denote the location of an item in the warehouse.

A1-B2 might refer to the first aisle, second bin on the shelf where a particular item is stored.

6.4 LOCATION

The physical location or warehouse where the item is stored. This could refer to different areas within a warehouse, different warehouses, or different geographic locations.

6.5 UNIT

The unit of measurement for the quantity of the item. This could be pieces, kilograms, Liters, or any other relevant unit.

6.6 Quantity

The amount of a specific item currently in stock.

A warehouse might have 500 units of a particular product in stock.

6.7 REORDERQUANTITY

The predefined quantity at which a reorder of the item should be triggered to prevent stockouts. This is often based on historical usage patterns and lead times.

6.8 COST

The cost of a single unit of the item. This could be the purchase cost from suppliers or the cost of manufacturing the item.

6.9 Inventory Value

The total monetary value of all the items in inventory, usually calculated by multiplying the quantity by the unit cost.

Stock level(quantity) and inventory value are of importance to the customer

6.10 REORDER

Indicates whether the item needs to be reordered based on current stock levels compared to the reorder quantity. This might be a simple yes/no or a numerical value indicating the quantity to reorder.

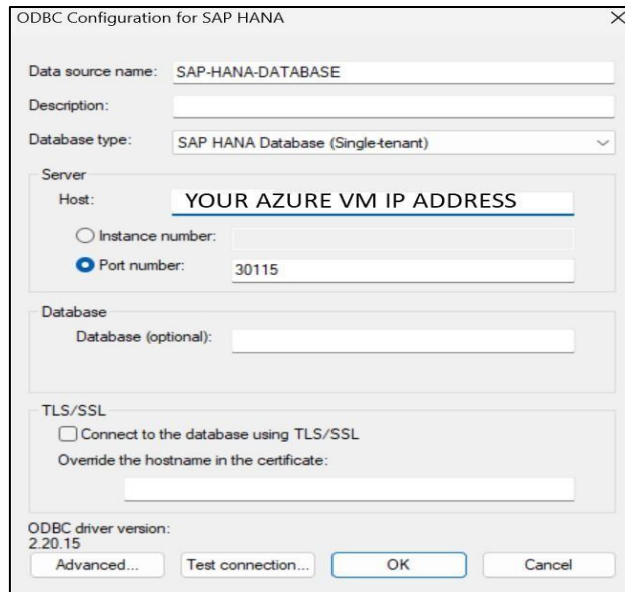


Figure 2: ODBC Connection to the Azure VM

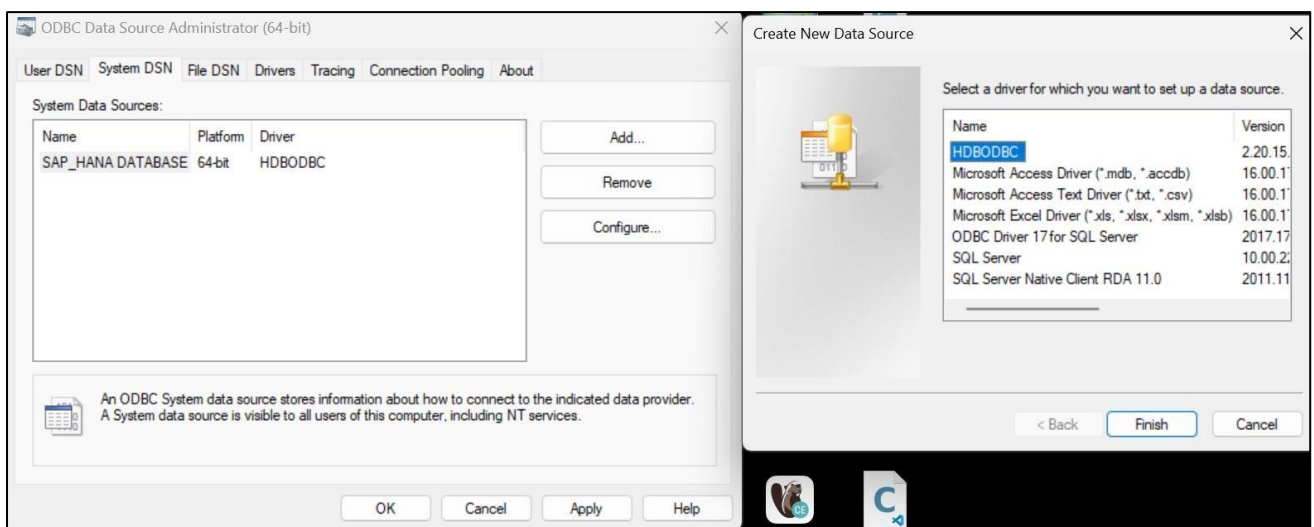


Figure 3: ODBC Connection configuration

7. isql HDBODBC

```
hxeadm@SAPHANA:/usr/sap/HXE/HDB01> isql HDBODBC SYSTEM *****
```

```
SQL> CREATE TABLE INVENTORY (SKU VARCHAR(50) PRIMARY
KEY,Description VARCHAR(255),BinNumber VARCHAR(50),Location
VARCHAR(100),Unit VARCHAR(50),Quantity INT,ReorderQuantity
INT,Cost DECIMAL(10, 2),InventoryValue DECIMAL(15, 2),Reorder
BOOLEAN);
SQLRowCount returns 0
SQL> INSERT INTO
INDUSTRY (SKU,DESCRIPTION,BINNUMBER,LOCATION,UNIT,QUANTITY,REORDER
QUANTITY,COST,INVENTORYVALUE,REORDER) VALUES ('SP7875','Item
1','T345','DEL','Each',20,10,30.00,600.00,FALSE);
SQLRowCount returns 1
SQL> INSERT INTO
INDUSTRY (SKU,DESCRIPTION,BINNUMBER,LOCATION,UNIT,QUANTITY,REORDER
QUANTITY,COST,INVENTORYVALUE,REORDER) VALUES ('TR87680','Item
2','T345','DEL','Each',30,15,40.00,1200.00,FALSE);
SQLRowCount returns 1
```

Like this all-other data can be inserted

```
SQL> SELECT * FROM INDUSTRY;
SQL> quit
```

SKU	DESCRIPTION	BINNUMBER	LOCATION	UNIT	QUANTITY	REORDERQUANTITY	COST	INVENTORYVALUE	REORDER
SP7875	Item 1	T345	DEL	Each	20	10	30	600	FALSE
TR87680	Item 2	T345	DEL	Each	30	15	40	1200	FALSE
MK676554	Item 3	T5789	DEL	Each	10	5	5	50	FALSE
YE98767	Item 4	T9876	DEL	Box(10 ct	40	10	15	600	TRUE
XR23423	Item 5	T098	CHE	Each	12	10	26	312	TRUE
PW98762	Item 6	T345	CHE	Each	7	10	50	350	FALSE
BM87684	Item 7	T349	MUM	Each	10	5	10	100	FALSE
BH67655	Item 8	T5789	MUM	Each	19	10	3	57	FALSE
WT98768	Item 9	T9875	KOL	Package(20	30	14	280	FALSE
TS3456	Item 10	T349	KOL	Each	15	8	60	900	FALSE
WDG123	Item 11	T349	KOL	Each	25	15	8	200	FALSE

Table 1: SAP -HANA database table

8. Azure Local Files System to Data Transformation Cloud

Step	Action	Remark
Step 1	Install Microsoft integration Runtime	HOST NAME = SHIRSAPHANA
Step 2	CREATE Resource Group	student-rg-n3
Step 3	Create Data Factory	SAPHANA-ADF
Step 4	Create Storage Account	saphanasa1
Step 5	Create Databricks	SAPHANA-DB
Step 6	Create Synapse Workspace	saphana-sws

Table 2: Key Data Engineering initial configurations tasks

storage configuration parameters:

```
C:\Windows\System32>cd C:\Program Files\Microsoft Integration Runtime\5.0\Shared
```

```
C:\Program Files\Microsoft Integration Runtime\5.0\Shared>.dmgcmd.exe DisableLocalFolderPathValidation
```

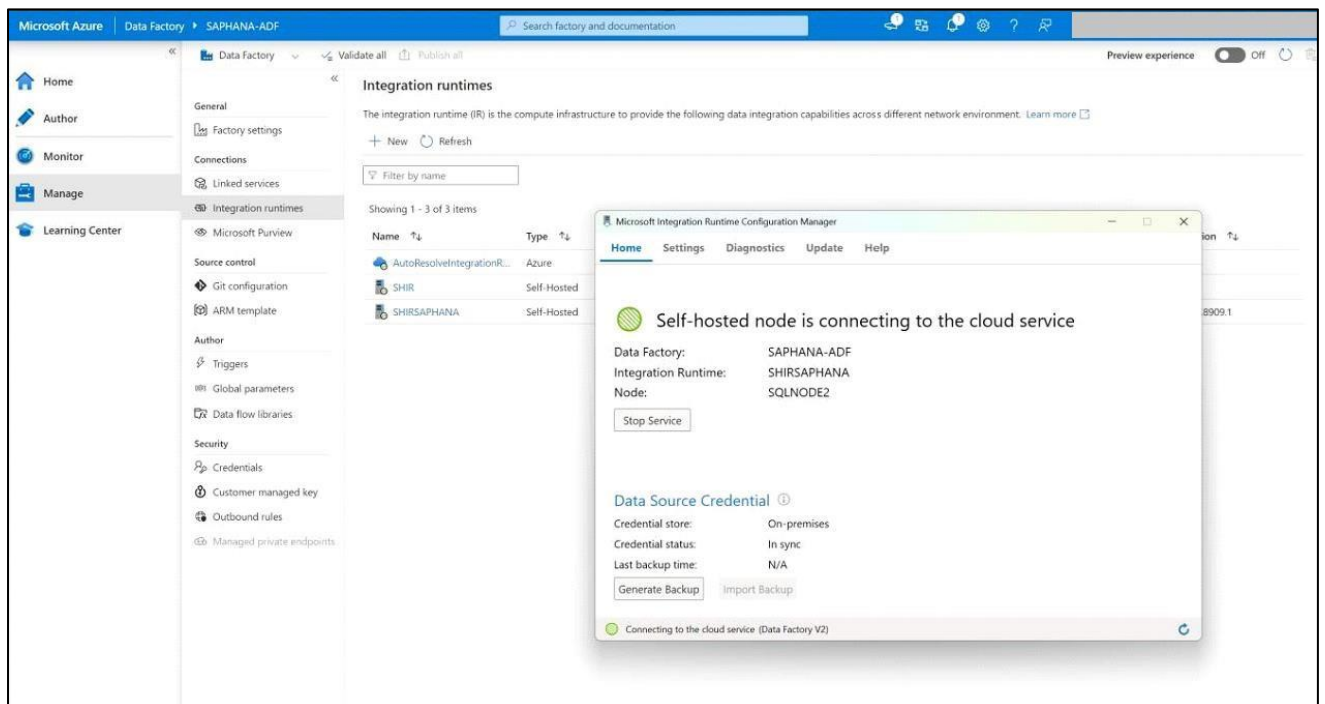






Figure 4: Self-hosted integration Run time installation


Edit linked service

 File system [Learn more](#) 

Name *

Description

Connect via integration runtime * ⓘ
 SHIRSAPHANA  

 The credentials are stored in the machines of self-hosted integration runtime if you don't choose to store them in Azure Key Vault.

Host * ⓘ

User name *

Authentication
 Password Azure Key Vault

Password *

Annotations
 + New
 > Parameters
 > Advanced ⓘ

Buttons:





Status:  Connection successful  Test connection



Figure 5: Testing Connection to the on-prem file system


Edit linked service


 Azure Blob Storage [Learn more](#) 

Name *
SAPHANA_BLOB

Description

Connect via integration runtime * [?]
 SHIRSAPHANA  

 The credentials are stored in the machines of self-hosted integration runtime if you don't choose to store them in Azure Key Vault.

Authentication type
Account key 

Connection string **Azure Key Vault**

Account selection method [?]
 From Azure subscription Enter manually

Storage account name *
saphanasa1

Storage account key **Azure Key Vault**

Storage account key *
.....

Partitioned DNS enabled [?]

Endpoint suffix
core.windows.net

Additional connection properties
+ New



 Connection successful
 Test connection

Figure 6: Connect via integration run time

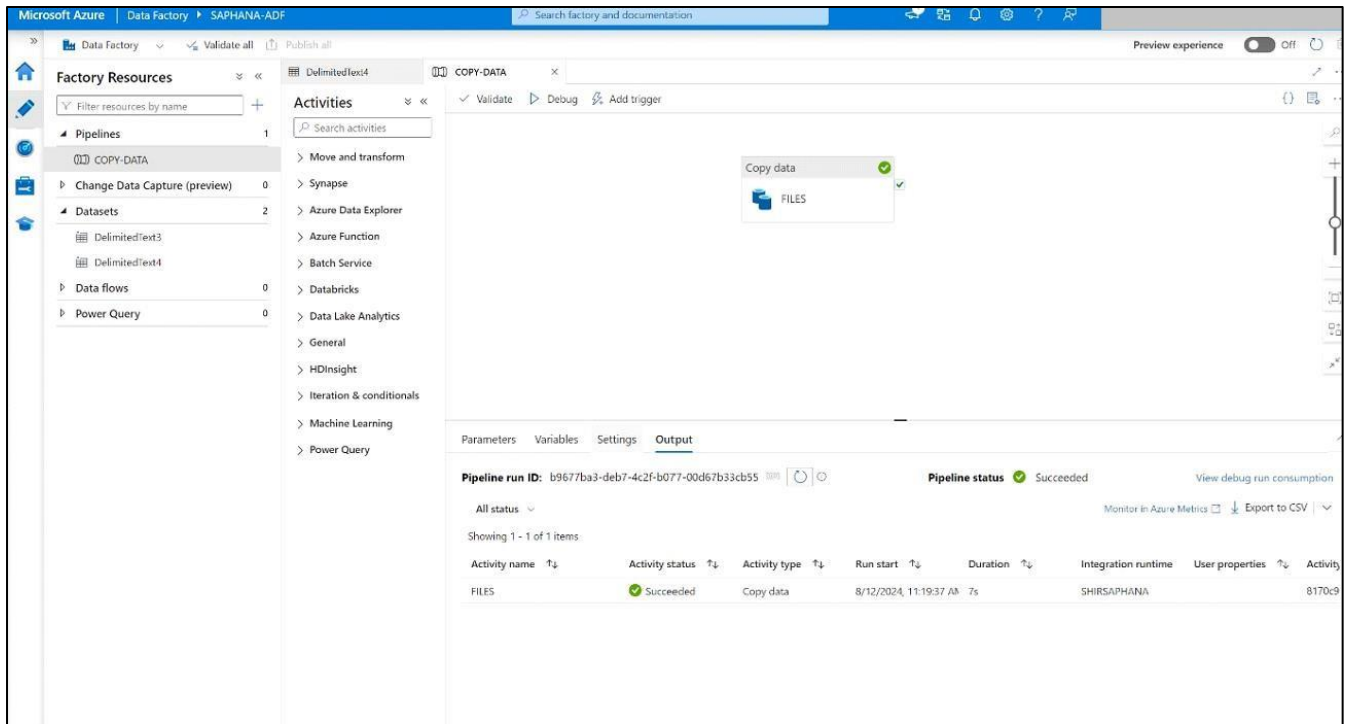


Figure 7: Create a pipeline for transforming data.

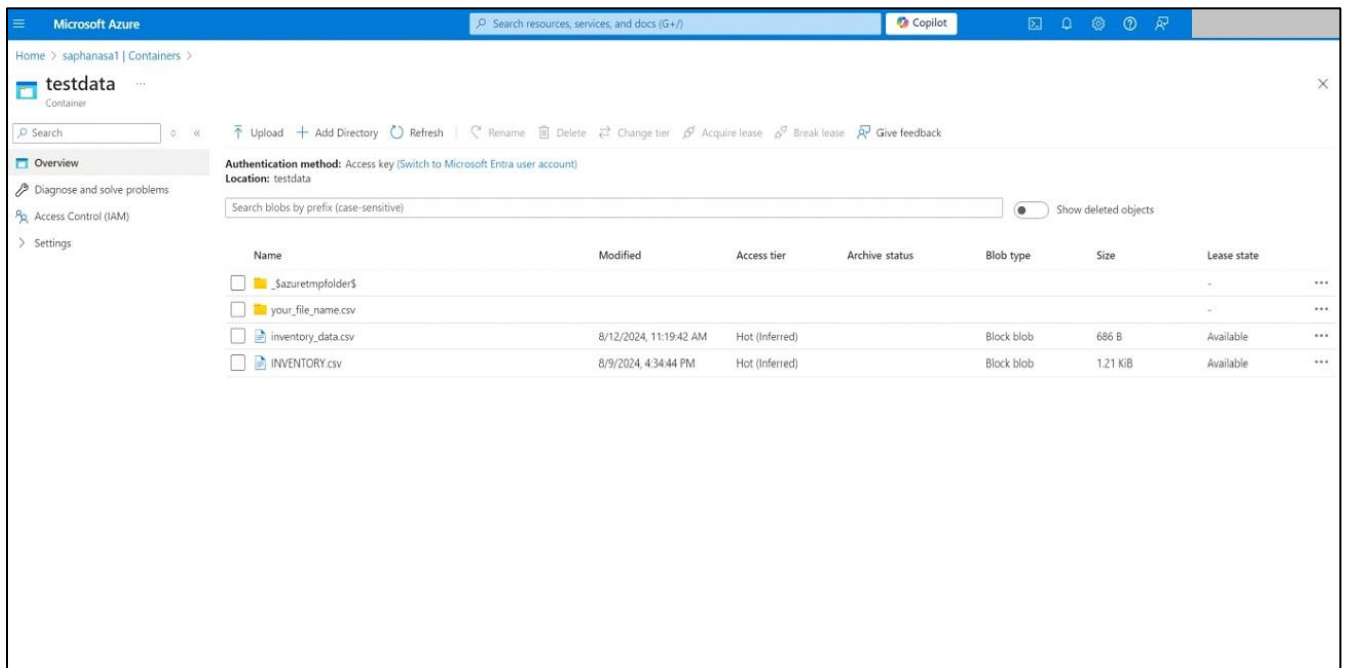


Figure 8: Create a storage container on Azure. Transfer file to the *testdata* in the container.

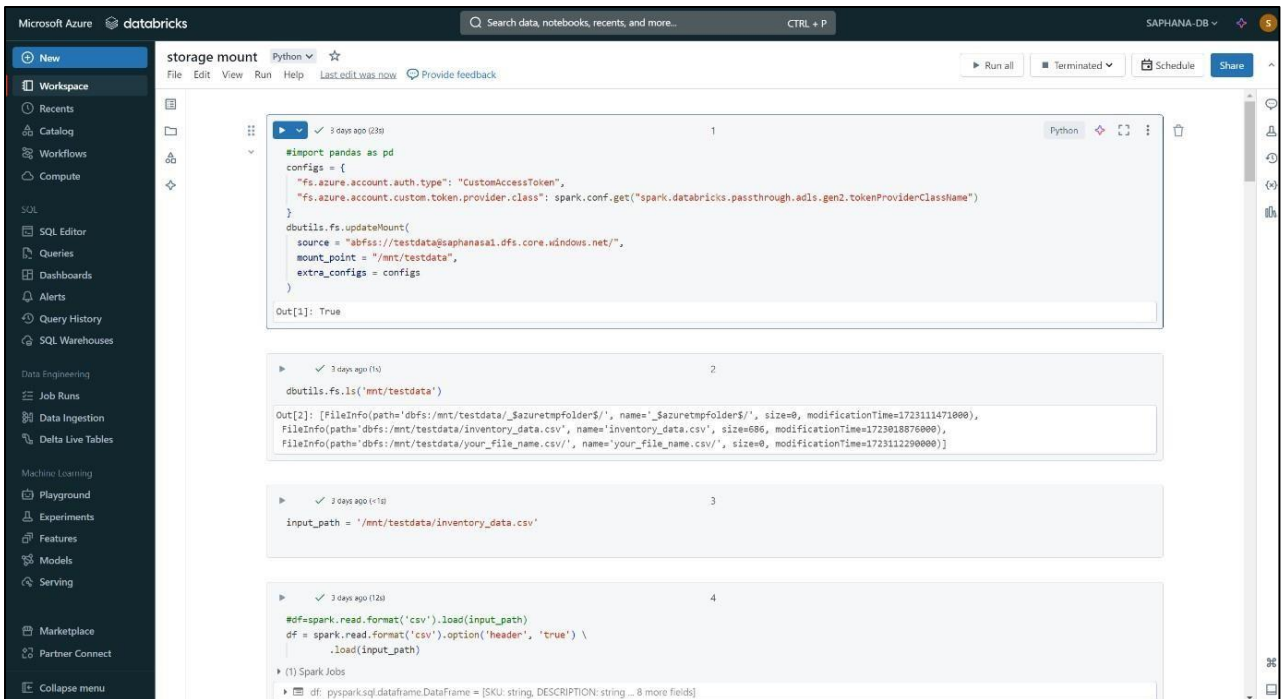


Figure9: Mount the data from test container.

The screenshot shows the "Edit linked service" configuration page for an Azure SQL Database. The configuration includes:

- Name:** AzureSqlDatabase1
- Description:** (empty)
- Connect via integration runtime:** AutoResolveIntegrationRuntime
- Version:** Recommended
- Account selection method:** Enter manually
- Fully qualified domain name:** saphana-sws-ondemand.sql.azure.synapse.net
- Database name:** INVENTORY_DB
- Authentication type:** System Assigned Managed Identity
- Managed identity name:** saphana-sws
- Managed identity object ID:** de96ffa7-2e3e-4e07-a164-09797b322668
- Grant workspace service managed identity access to your Azure SQL Database:** (checked)
- Always encrypted:** (unchecked)
- Encrypt:** Mandatory
- Trust server certificate:** (checked)
- Additional connection properties:** (empty)

Buttons for "Save", "Cancel", and "Test connection" are visible at the bottom.

Figure10: Connect to the serverless SQL Data base

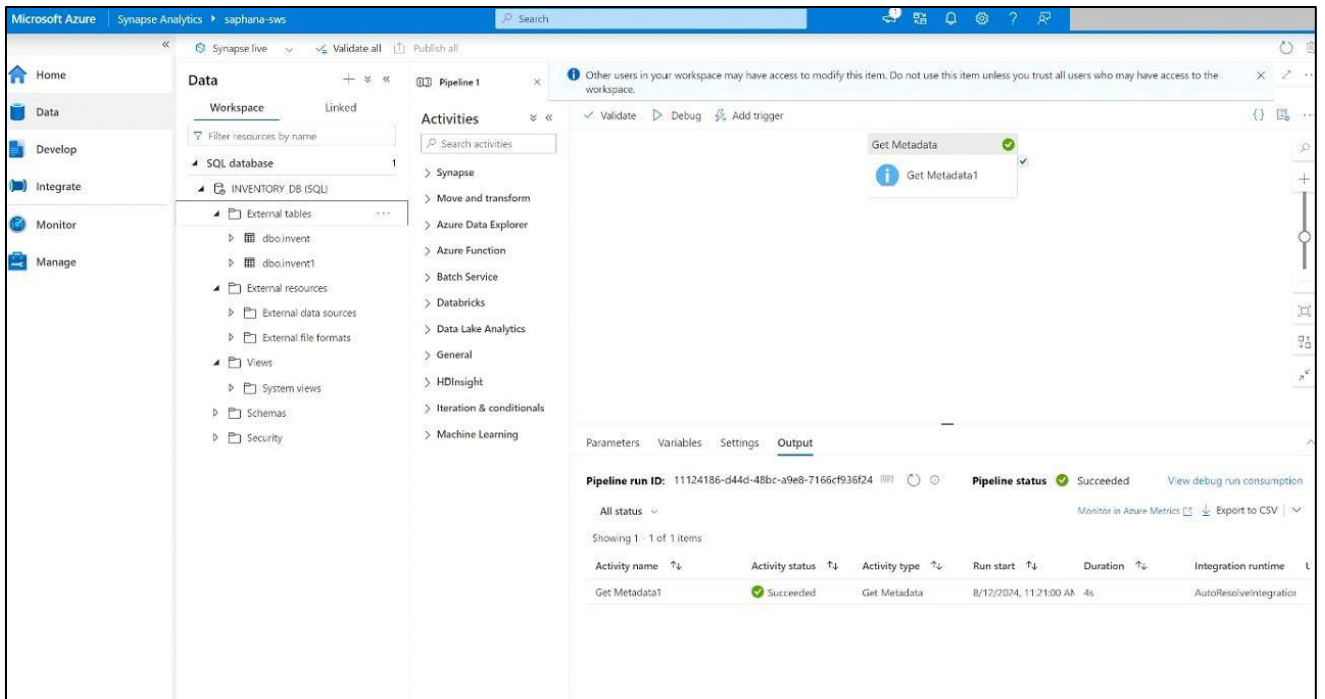


Figure11: Create a pipeline to connect to data bricks

9. TROUBLESHOOTING in Synapse Workspace:

Cannot connect to 'C:\Users\sn020'. Detail Message: The system could not find the environment option that was entered the system could not find the environment option that was entered

C:\Program Files\Microsoft Integration Runtime\5.0\Shared>./dmgcmd - DisableLocalFolderPathValidation '. is not recognized as an internal or external command, operable program or batch file. give a windows command line
The value of the property " is invalid: 'Value does not fall within the expected range.'. Value does not fall within the expected range.

The value of the property " is invalid: 'Access to *.*.*.*.* is denied, resolved IP address is *.*.*.*.*.*, network type is On-premise'. Access to *.*.*.*.*.* is denied, resolved IP address is *.*.*.*.*.*, network type is On-premise

Solution:

```
Administrator: Windows PowerShell
PS C:\> cd 'Program Files'
PS C:\Program Files> cd 'Microsoft Integration Runtime'
PS C:\Program Files\Microsoft Integration Runtime> cd 5.0
PS C:\Program Files\Microsoft Integration Runtime\5.0> cd 'Shared'
PS C:\Program Files\Microsoft Integration Runtime\5.0\Shared> ./dmgcmd.exe -DisableLocalFolderPathValidation
PS C:\Program Files\Microsoft Integration Runtime\5.0\Shared>
```

Error code: 1102 The value of the property " is invalid: 'Value does not fall within the expected range.'. Value does not fall within the expected range.

from pyspark.sql import SparkSession

```

from pyspark.sql.functions import col, create_map, lit
from intercools import chain
# Initialize SparkSession spark
= SparkSession.builder \
  .appName("ModifyColumnExample") \
  .getOrCreate()
# Sample DataFrame data = [
  ("SP7875","Item 1","T345","DEL","Each",20,10,30.0,600.00,"FALSE"),
  ("TR87680","Item 2","T345","DEL","Each",30,15,40.0,1200.00,"FALSE"),
  ("MK676554","Item 3","T5789","DEL","Each",10,5,5.0,50.00,"FALSE"),
  ("YE98767","Item 4","T9876","DEL","Box(10 ct)",40,10,15.0,600.00,"TRUE"),
  ("XR23423","Item 5","T098","CHE","Each",12,10,26.0,312.00,"TRUE"),
  ("PW98762","Item 6","T345","CHE","Each",7,10,50.0,350.00,"FALSE"),
  ("BM87684","Item 7","T349","MUM","Each",10,5,10.0,100.00,"FALSE"),
  ("BH67655","Item 8","T5789","MUM","Each",19,10,3.0,57.00,"FALSE"),
  ("WT98768","Item 9","T9875","KOL","Package(5 ct)",20,30,14.0,280.00,"FALSE"),
  ("TS3456","Item 10","T349","KOL","Each",15,8,60.0,900.00,"FALSE"),
  ("WDG123","Item 11","T349","KOL","Each",25,15,8.0,200.00,"FALSE") ] columns =
["SKU", "DESCRIPTION", "BINNUMBER", "LOCATION", "UNIT", "QUANTITY",
"REORDERQUANTITY", "COST", "INVENTORYVALUE", "REORDER"] df
= spark.createDataFrame(data, schema=columns) df_with_string =
df.withColumn("CURRENCY", lit("US$"))
# Define the correct mapping dictionary
#new_COST = {'30.0': '$30.0', '40.0': '$40.0', '5.0': '$5.0', '15.0': '$15.0', '26.0':
'$26.0', '50.0': '$50.0', '10.0': '$10.0', '3.0': '$3.0', '14.0': '$14.0', '60.0': '$60.0', '8.0': '$8.0' }
# Create a mapping expression using create_map
#mapping_expr = create_map([lit(x) for x in chain(*new_COST.items())])
# Apply the mapping expression to the COST column
#df_modified = df.withColumn("COST", mapping_expr[col("COST")])
# Show the modified DataFrame
#df_modified.show() display(df_with_string

```

10. ODBC for SAP HANA trouble shooting

Communication link failure; -10709 Connection failed

Error message:

HANA ODBC Test Connection Error SQLSTATE: 08S01 NATIVE ERROR: -10709 MESSAGE TEXT: [SAP AG][LIBODBCHEX DLL][HDBODBC] Communication link failure;-10709 Connection failed (RTE:[89006] System call 'connect' failed, rc=10060: A connection attempt failed because the connected party did not properly respond after a period of time, or establish).

Background

Our Objective is to Install ODBC and access SAP HANA on cloud. We faced this error. It took some time to solve. Information available on internet did not help.

System: Windows on-prem

Server: SAP HANA Express Edition Azure VM

Connectivity: ODBC

When trying to configure ODBC on windows. We continuously got above error message.

Solution:

Configure inbound security rule of SAP HANA VM at azure portal.

Check Inbound Port Rules

Ensure there is an inbound security rule that allows traffic on port 30115:(for unknown reasons, inbound security rule config for port 30015 did not help) Source: Any (or specify the IP range of your on-premises network)

Source port ranges: *

Destination: Any

Destination port ranges: 30115

Protocol: TCP **Action:**

Allow

Priority: Ensure it has a lower priority number (e.g., 1000) so it's not overridden by other rules.

Name: A meaningful name like Allow-SAP-HANA-ODBC.

How did we debug this error?

Step 1

On windows command line, when you run the following command

```
./nmap -p 30115 ipaddress(of vm)
```

You should see the output as follows

PORT	STATE	SERVICE
30115/tcp	open	unknown

Instead, if you see the output as follows, then it will not work

PORT	STATE	SERVICE
30115/tcp	close/filtered	unknown

(As mentioned, for unknown reasons, inbound security rule config for port 30015 did not help. The state was always in close state)

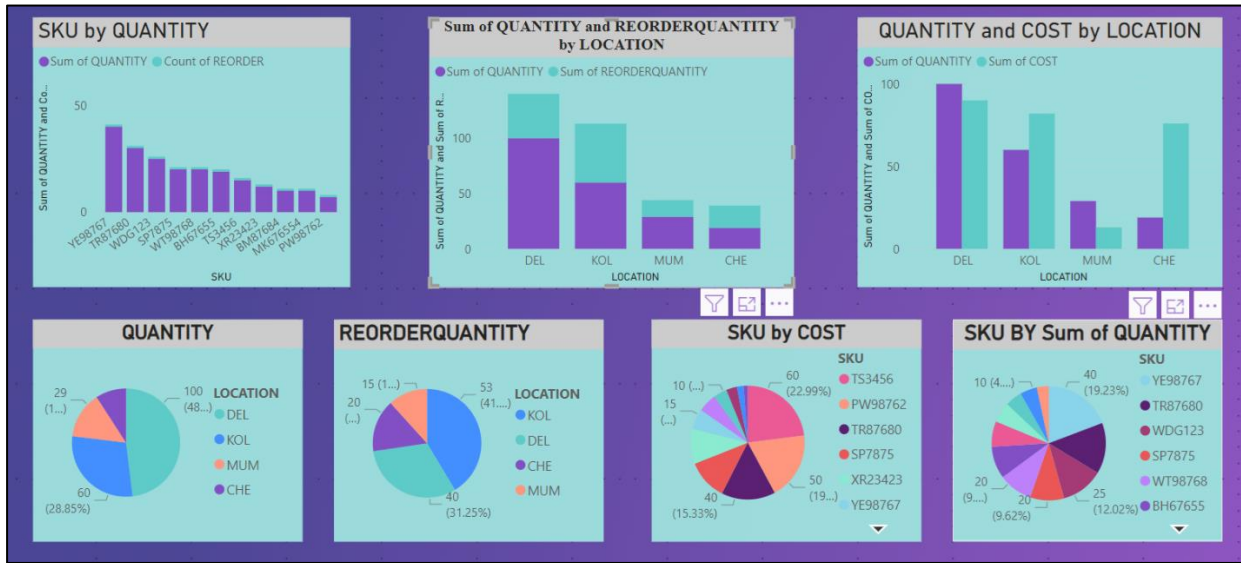
Step 2

Login to terminal of azure VM. Give the following command. You should see the output

as follows `sudo iptables -L -n`

Chain	INPUT (policy ACCEPT)	target	prot	opt	source	destination
ACCEPT	tcp	--	0.0.0.0/0		0.0.0.0/0	tcp dpt:3011

11. Conclusion and Next steps



11.1 Sum of Quantity and Reorder Quantity by Location:

The DEL-DELHI location holds the highest inventory quantity and also shows a significant reorder quantity, suggesting that it might be a central hub or high-demand location.

The CHE-CHENNAI location, although having inventory, shows a lower reorder quantity, potentially indicating less demand or better stock management.

Our next plan is to do end to end Data engineering for Real-Time manufacturing sap hana ah dataset. Such work will be beneficial to many industries

12. References:

1. <https://community.sap.com/t5/technology-blogs-by-sap/c-runtimes-needed-to-runsapexecutables/ba-p/13314763>
2. https://help.sap.com/docs/SAP_HANA_PLATFORM
3. <https://developers.sap.com/tutorials/hana-clients-choose-hana-instance.html>
4. <https://learn.microsoft.com/en-us/azure/data-factory/concepts-integration-runtime>