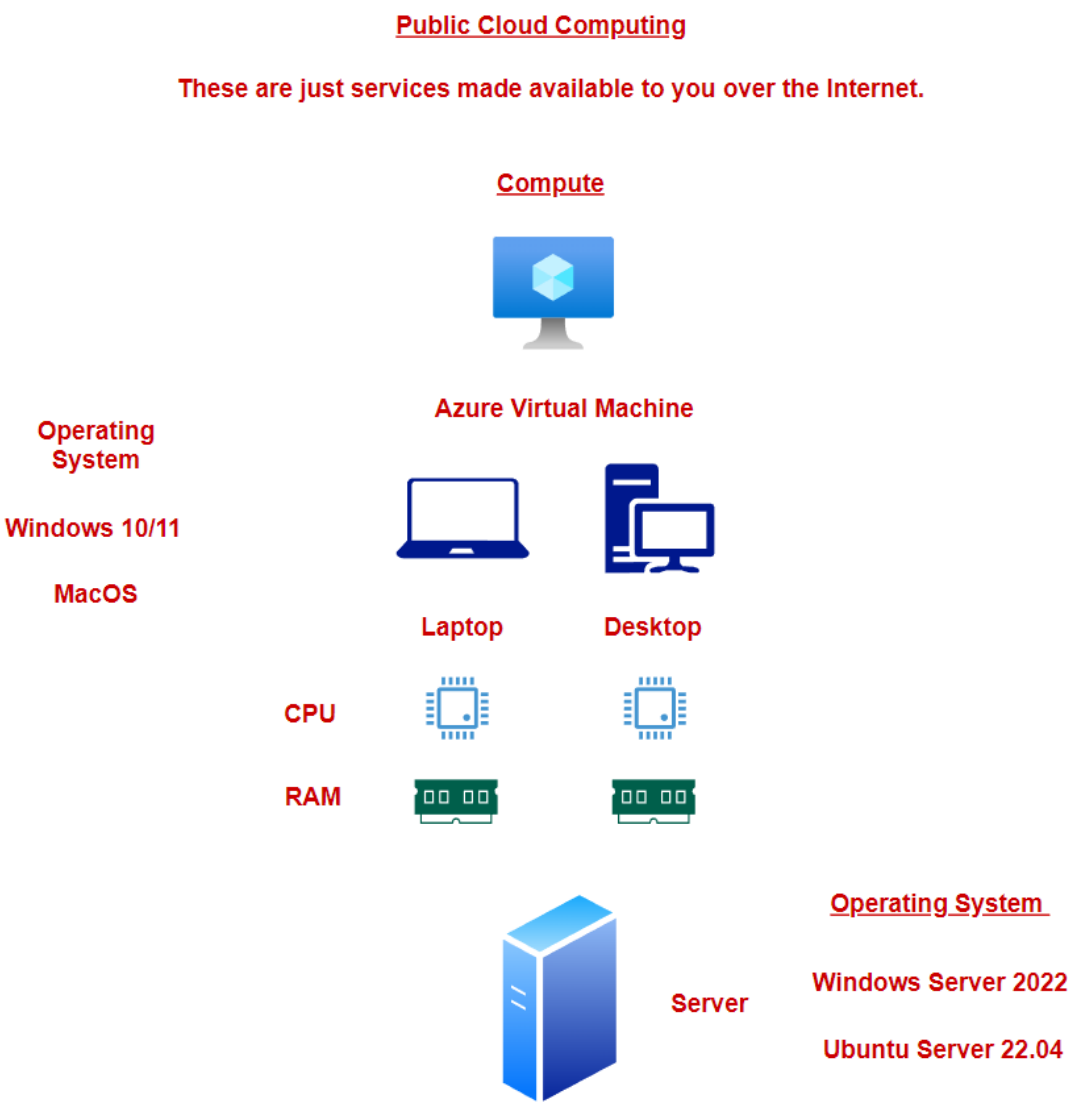
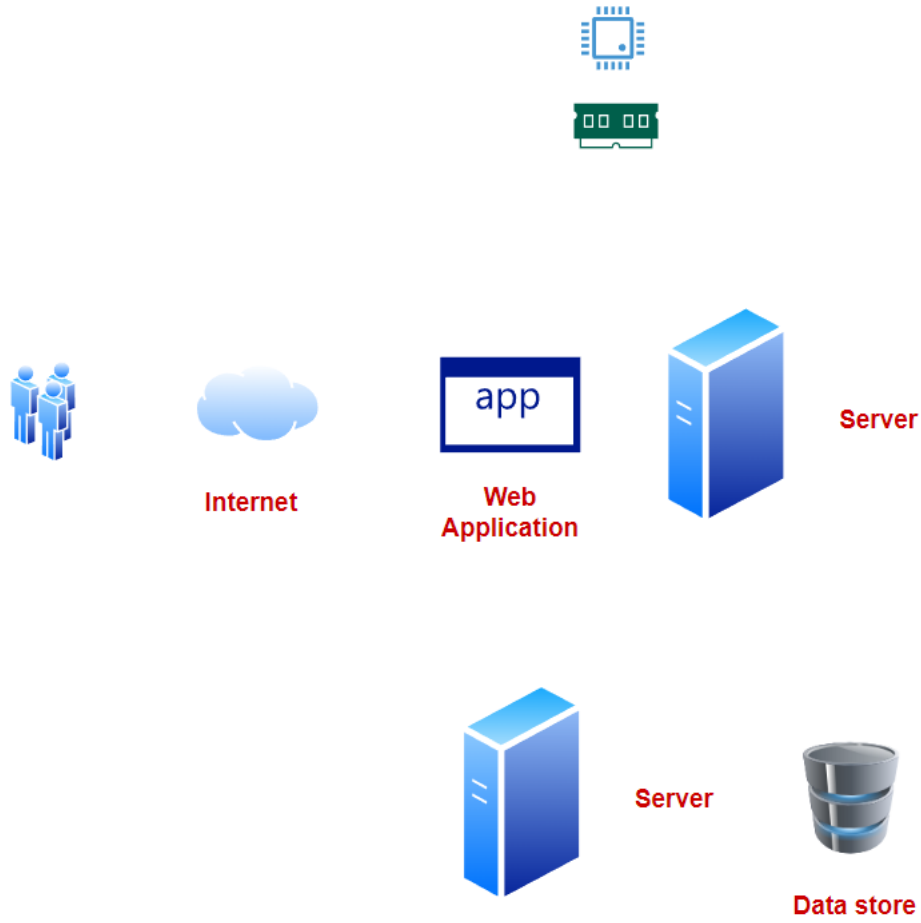
Introduction

Introduction to Cloud Computing





What is Microsoft Azure

This is a cloud platform

They make services available to you over the Internet

You can use this service to build, run and manage applications

Design and implement data storage – Basics

Understanding data

Data Engineering - Wikipedia reference

Building of systems to enable collection and usage of data

The data is normally used to enable analysis and data science and can also involve machine learning

Working with large data sets

Here the data arrives in large volumes

The data arrives at a fast rate

Data storage **Data processing** **Visualizing your data**

Classification of data

Structured data - Tabular data that is represented by rows and columns in a database

	ProductID	Name	ProductNumber	Color	StandardCost	ListPrice	Size	Weight	ProductCategoryID	ProductModelID
1	680	HL Road Frame - Black, 58	FR-R92B-58	Black	1059.31	1431.50	58	1016.04	18	6
2	706	HL Road Frame - Red, 58	FR-R92R-58	Red	1059.31	1431.50	58	1016.04	18	6
3	707	Sport-100 Helmet, Red	HL-U509-R	Red	13.0863	34.99	NULL	NULL	35	33
4	708	Sport-100 Helmet, Black	HL-U509	Black	13.0863	34.99	NULL	NULL	35	33
5	709	Mountain Bike Socks, M	SO-B909-M	White	3.3963	9.50	M	NULL	27	18
6	710	Mountain Bike Socks, L	SO-B909-L	White	3.3963	9.50	L	NULL	27	18
7	711	Sport-100 Helmet, Blue	HL-U509-B	Blue	13.0863	34.99	NULL	NULL	35	33
8	712	AWC Logo Cap	CA-1098	Multi	6.9223	8.99	NULL	NULL	23	2
9	713	Long-Sleeve Logo Jersey, S	LJ-0192-S	Multi	38.4923	49.99	S	NULL	25	11
10	714	Long-Sleeve Logo Jersey, M	LJ-0192-M	Multi	38.4923	49.99	M	NULL	25	11
11	715	Long-Sleeve Logo Jersey, L	LJ-0192-L	Multi	38.4923	49.99	L	NULL	25	11
12	716	Long-Sleeve Logo Jersey, XL	LJ-0192-X	Multi	38.4923	49.99	XL	NULL	25	11
13	717	HL Road Frame - Red, 62	FR-R92R-62	Red	868.6342	1431.50	62	1043.26	18	6
14	718	HL Road Frame - Red, 44	FR-R92R-44	Red	868.6342	1431.50	44	961.61	18	6
15	719	HL Road Frame - Red, 48	FR-R92R-48	Red	868.6342	1431.50	48	979.75	18	6
16	720	HL Road Frame - Red, 52	FR-R92R-52	Red	868.6342	1431.50	52	997.90	18	6
17	721	HL Road Frame - Red, 56	FR-R92R-56	Red	868.6342	1431.50	56	1016.04	18	6
18	722	LL Road Frame - Black, 58	FR-R38B-58	Black	204.6251	337.22	58	1115.83	18	9

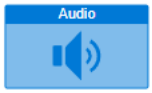
Semi-structed data

This is data that resides in other formats and not in a database as such

Common example - JSON - JavaScript Object Notation

```
{
  "customerid": 1,
  "customername" : "John",
  "city" : "Miami"
}
```

unstructured data



These are all binary objects

It's time to use the cloud

Azure storage account

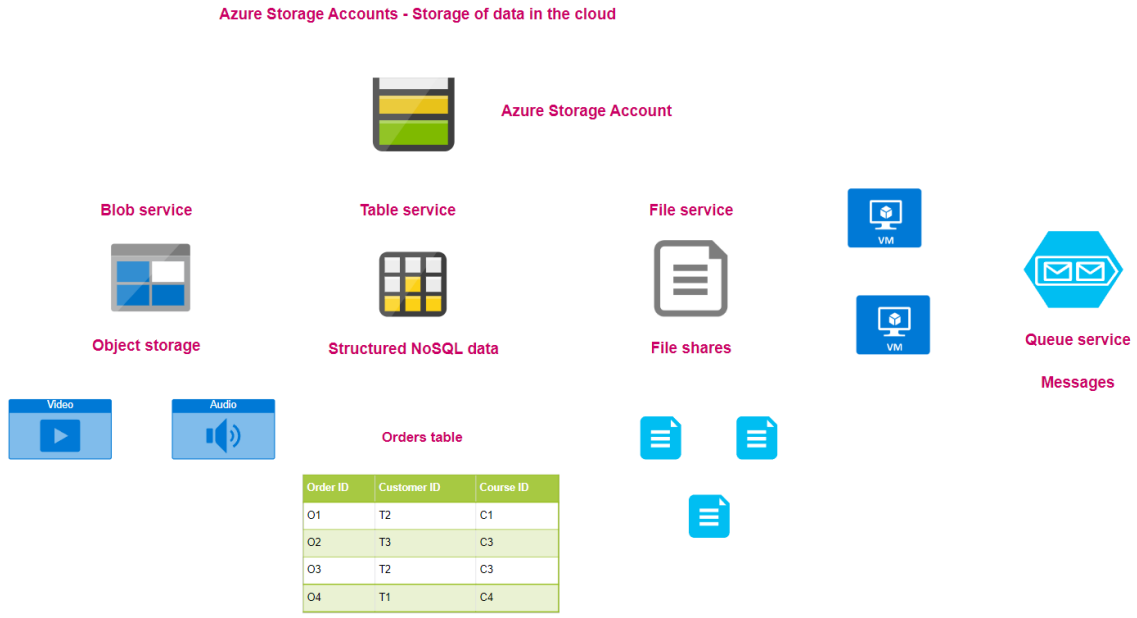
Storage of images

.Net Core application

Azure SQL database

Storage of application data

Lab- Azure Storage accounts



Lab- Azure SQL databases

Hosting data in the form of tables

	ProductID	Name	ProductNumber	Color	StandardCost	ListPrice	Size	Weight	ProductCategoryID	ProductModelID
1	680	HL Road Frame - Black, 58	FR-R92B-58	Black	1059.31	1431.50	58	1016.04	18	6
2	706	HL Road Frame - Red, 58	FR-R92R-58	Red	1059.31	1431.50	58	1016.04	18	6
3	707	Sport-100 Helmet, Red	HL-U509-R	Red	13.0863	34.99	NULL	NULL	35	33
4	708	Sport-100 Helmet, Black	HL-U509	Black	13.0863	34.99	NULL	NULL	35	33
5	709	Mountain Bike Socks, M	SO-B909-M	White	3.3963	9.50	M	NULL	27	18
6	710	Mountain Bike Socks, L	SO-B909-L	White	3.3963	9.50	L	NULL	27	18
7	711	Sport-100 Helmet, Blue	HL-U509-B	Blue	13.0863	34.99	NULL	NULL	35	33
8	712	AWC Logo Cap	CA-1098	Multi	6.9223	8.99	NULL	NULL	23	2
9	713	Long-Sleeve Logo Jersey, S	LJ-0192-S	Multi	38.4923	49.99	S	NULL	25	11
10	714	Long-Sleeve Logo Jersey, M	LJ-0192-M	Multi	38.4923	49.99	M	NULL	25	11
11	715	Long-Sleeve Logo Jersey, L	LJ-0192-L	Multi	38.4923	49.99	L	NULL	25	11
12	716	Long-Sleeve Logo Jersey, XL	LJ-0192-X	Multi	38.4923	49.99	XL	NULL	25	11
13	717	HL Road Frame - Red, 62	FR-R92R-62	Red	868.6342	1431.50	62	1043.26	18	6
14	718	HL Road Frame - Red, 44	FR-R92R-44	Red	868.6342	1431.50	44	961.61	18	6
15	719	HL Road Frame - Red, 48	FR-R92R-48	Red	868.6342	1431.50	48	979.75	18	6
16	720	HL Road Frame - Red, 52	FR-R92R-52	Red	868.6342	1431.50	52	997.90	18	6
17	721	HL Road Frame - Red, 56	FR-R92R-56	Red	868.6342	1431.50	56	1016.04	18	6
18	722	LL Road Frame - Black, 58	FR-R38B-58	Black	204.6251	337.22	58	1115.83	18	9

We make use of a relational database management system

View our data in the form of tables

Server Machine

Install Microsoft SQL Server

Storage such as disks

Host a SQL database

1. Create a virtual machine
2. Install the database software
3. Create your database , tables and store your data
4. Administrative tasks - Backup, High Availability

Azure SQL database

This is a platform as a service

Here the infrastructure is managed for you

Azure Data Lake Gen-2 storage accounts

Working with large data sets

Here the data arrives in large volumes

The data arrives at a fast rate

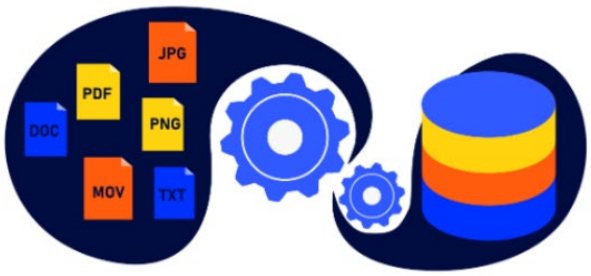
Azure Data Lake Storage Gen2

This service is built on top of Azure Blob storage

Gives the ability to host an enterprise data lake on Azure

You also get the feature of a hierarchical namespace on top of Azure Blob storage

Helps to organize objects/files into a hierarchy of directories for efficient data access



A data lake is used to store large amounts of data in its native, raw format

Data lakes are optimized for storing terabytes and petabytes of data

The data could come from a variety of data sources

The data itself could be in various formats - Structured, semi-structured and unstructured data

Different file formats

JSON

JavaScript Object Notation


```
{
  "count": 1,
  "total": 0,
  "minimum": 0,
  "maximum": 0,
  "average": 0,
  "resourceId": "/SUBSCRIPTIONS/E5250E15-0516-48F0-889B-DAE6C15B6529
    /RESOURCEGROUPS/PRODUCTIONGRP/PROVIDERS/MICROSOFT.DBFORMYSQL
    /SERVERS/WORDPRESS-SERVER2020",
  "time": "2021-02-16T17:36:00.0000000Z",
  "metricName": "cpu_percent",
  "timeGrain": "PT1M"
}
```


The JSON contents are enclosed in curly brackets. It is a JSON document

Each document consists of fields. Each field has a name and value

Avro

This is a row-based format file

Each record in the file contains a header that describes the structure of the data in the record

The data iself is stored in binary format

This format is ideal for compressing data

Results in less storage

Requires less bandwith requirements


```
1  Obj□□□avro.codec□null□avro.schema□□{"type":"record","name":"HybridDelivery.ClientLibraryJob","fields":[{"name":"count","type":
   ["int","null"]},{"name":"total","type":["double","null"]},{"name":"minimum","type":["double","null"]},{"name":"maximum","type":
   ["double","null"]},{"name":"average","type":["double","null"]},{"name":"resourceId","type":["string","null"]},{"name":"time",
   "type":["string","null"]},{"name":"metricName","type":["string","null"]},{"name":"timeGrain","type":["string","null"]}]}��
   [��tQE�n�?�"□���□��□□□
```

Parquet data format

This is a columnar data format

It was created by Cloudera and Twitter

Data for each column is stored together in something known as a row group

This data format support compression and different encoding schemes

Design and implement data storage- Overview on Transact-SQL

The internals of a database engine

SQL database

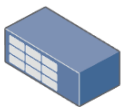
The database is a collection of tables , view etc

The tables store structured data

With a SQL database you can run SQL queries against your database tables

If you want to perform an operation such as getting data from a table, you will fire the SQL query

SELECT * FROM Customer WHERE CustomerID=2



The database engine will process the operation and return the result to the user

CustomerID	CustomerName	Address
1	CustomerA	AddressA
2	CustomerB	AddressB
3	CustomerC	AddressC
4	CustomerD	AddressD

Design and implement data storage- Azure Synapse Analytics

What have we seen so far

CSV

Semi-structured data

Azure Storage Account

Azure Data Lake Gen2

Data Lake

One place for all data

It can then be analyzed at a later point in time

Unstructured data

Cloud implementation of Microsoft SQL Server

Azure SQL Database

OLTP system

Online Transactional Processing

Used to record the day to day operations

Tables

Azure Storage Account

Web Application

Azure SQL Database

Student

Course

Purchase

StudentID	StudentName	StudentAddress
S1	StudentA	AddressA
S2	StudentB	AddressB

CourseID	CourseName	Price
C0	CourseA	20
C1	CourseB	30

OrderID	StudentID	CourseID
C0	S1	C0
C1	S2	C0

A data warehouse

Data Warehousing

Centralized repository of data taken from different data sources

Normally used by business for analyzing data

Web Application

Azure SQL Database

Student

Course

Purchase

StudentID	StudentName	StudentAddress
S1	StudentA	AddressA
S2	StudentB	AddressB

OrderID	StudentID	CourseID
C0	S1	C0
C1	S2	C0

CourseID	CourseName	Price
C0	CourseA	20
C1	CourseB	30

Let's say senior management wants to know

1. Purchases done per month per region

2. Most popular courses for the month per region

3. Times during the day where purchases are made the most

Azure SQL Database

SQL Data Warehouse

1. SQL Data Warehouse would again be powered by some database engine

2. Your data would be structured in the form of tables

3. But the data warehouse would store larger amounts of data

4. Data in the data warehouse is then analyzed accordingly

5. You need a lot of processor power to analyze the data

6. Just like a SQL database, you can fire SQL queries against a SQL data warehouse

7. Business users can also use reporting tools to visualize the data in the warehouse

So now would you just copy the data from a SQL database to a SQL data warehouse for analysis

NO

Azure SQL Database

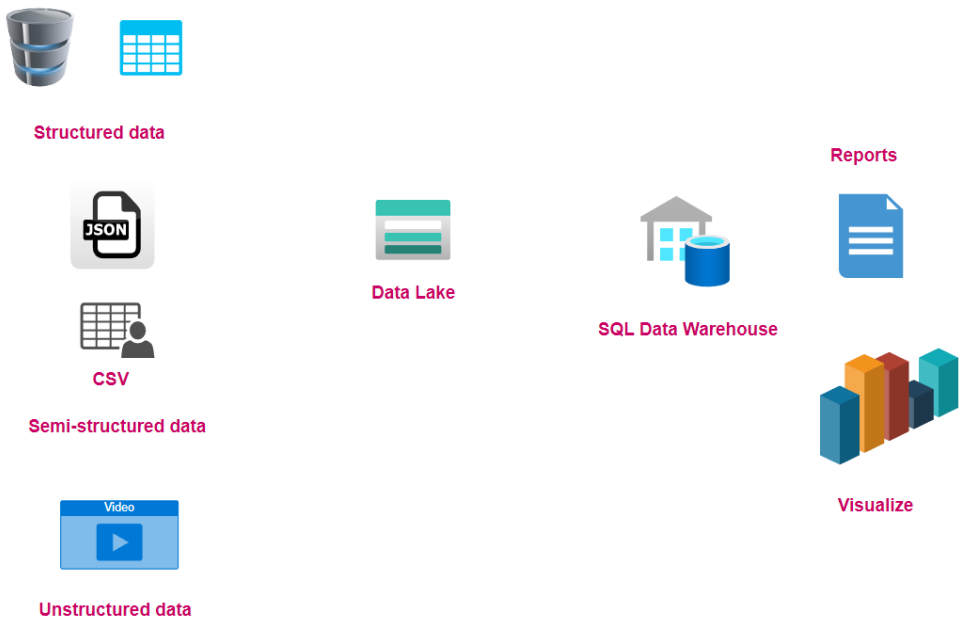
TRANSFORM

SQL Data Warehouse

Might need to clean the data

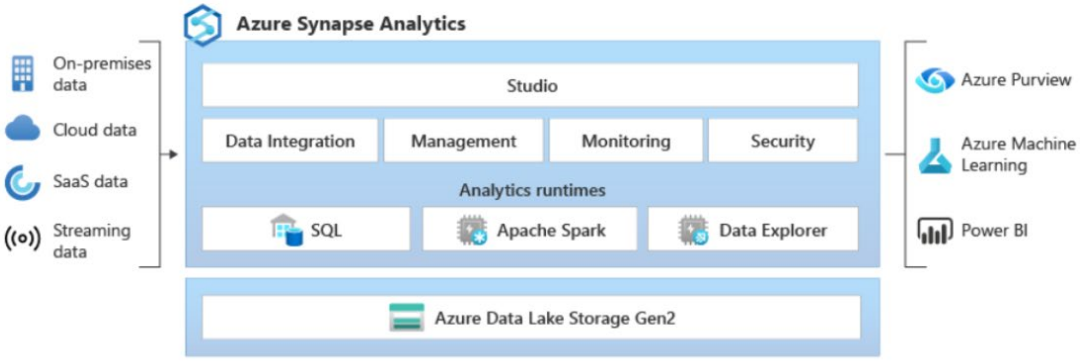
Only transfer some columns of data

Transform the data stored



Moden data warehouse architecture

Enter Azure Synapse



Reference - https://learn.microsoft.com/en-us/azure/synapse-analytics/overview-what-is

Compute options

There are 2 compute options we are going to look at

Serverless SQL pool

You can use this option to perform quick adhoc analysis of data

You can use T-SQL to work with your data

Here you are charged based on how much you use

SQL Pool

This is used to build your data warehouse

If you need to persist your data

Here compute nodes will be used to process the data

Here you are charged based on a metric known as DWU

More expensive

But the data is already structured and ready to be queried and visualized by business users

Dedicated compute power

Loading data into a SQL data warehouse

So far we have dealing with external tables

We have not built our data warehouse

By having the SQL Pool in place we can now build our data warehouse

Structured data

CSV

Semi-structured data

Data Lake

SQL Data Warehouse

Normally for a SQL database, you would use the INSERT command to insert just a row. Or insert multiple rows based on a transaction.

The SQL data warehouse is similar to a SQL database where you define tables via the CREATE TABLE command

But when it comes to writing data, you would normally load data in chunks

You would first have an initial load of your data and then delta loads

For a data warehouse, what's important is how you read your data

Designing a data warehouse

SQL Data Warehouse

Fact Tables

These are measurements or metrics that correspond to facts

For example - Sales Table - This records all the sales that have been made

The sales data are facts that sales have actually been made

Dimension tables

This helps to provide some sort of context to the facts that are presented

For example - What are the products that were sold

Who are the customers who bought the products

Building your fact table

Building facts around the sales data

The sales data will keep on getting added to the sales fact table

OLTP
Transactional data

OLAP
Analysis

Orders/Sales data

This is your fact table

It contains facts about the sales made for your courses

Order ID	Course ID	quantity
O1	C1	10
O2	C2	20
O3	C3	30
O4	C3	40

Sales table

Transactions for 2018

Transactions for 2019

Transactions for 2020

Transactions for 2021

Transactions for 2022

Your Dimension Table

OLTP
Transactional data

OLAP
Analysis

Courses
AZ-900
DP-203
AZ-104

Customer
UserA
UserB
UserC

Instructor
InstructorA
InstructorB
InstructorC

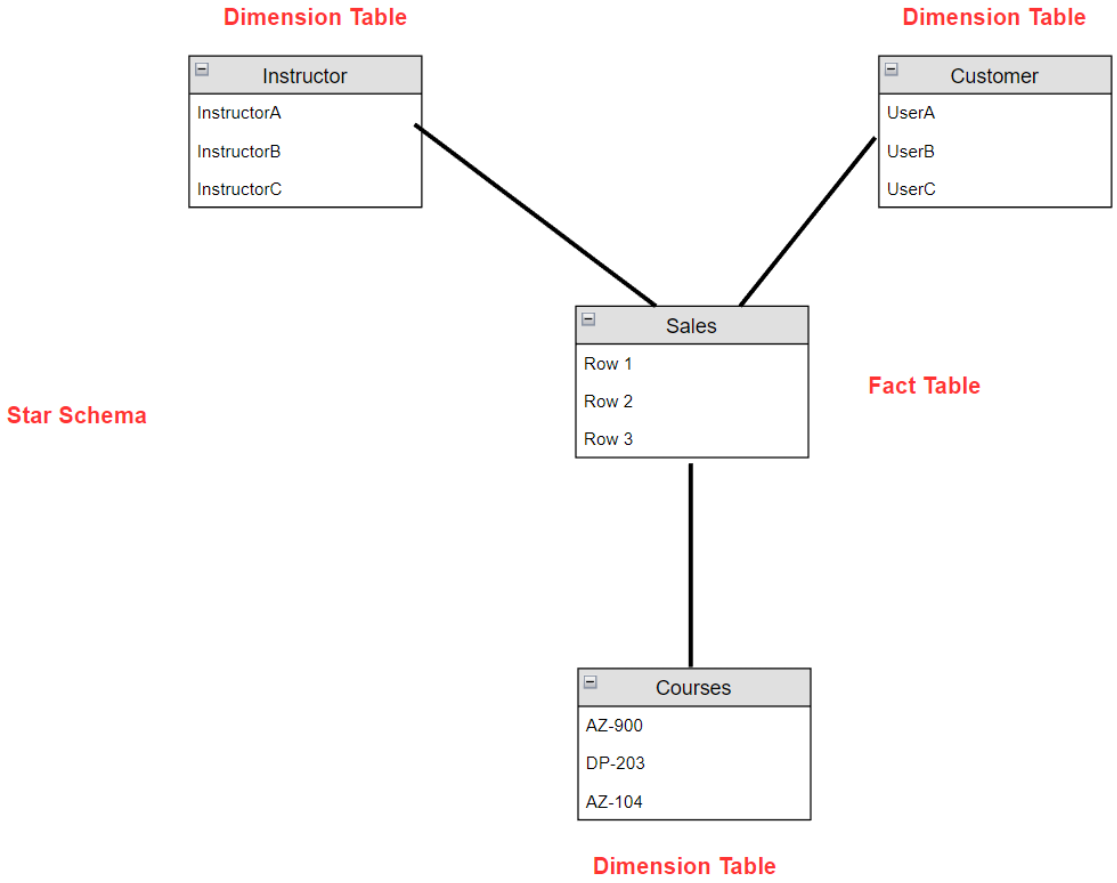
Sales
Row 1
Row 2
Row 3

Sales Fact table

Which courses sold the most in the year?

Which regions had the most customers?

Who are the most popular instructors?


Dimension Table
Dimension Table
Instructor
InstructorA
InstructorB
InstructorC
Customer
UserA
UserB
UserC
Sales
Row 1
Row 2
Row 3
Fact Table
Star Schema
Courses
AZ-900
DP-203
AZ-104
Dimension Table

More on Dimension tables

More on Dimension Tables

Products

Customers

This data might be in a different source system

Could be in csv format

We then need to transfer the data to a dimension table in the data warehouse

Sales data

Fact table

This might be in a database

Source System A

ProductID	ProductName	ProductPrice
1	AZ-204	10.99
2	DP-203	11.99
3	AZ-104	12.99

Source System B

ProductID	ProductName	ProductPrice
1	BookA	10.99
2	BookB	11.99
3	BookC	12.99

One Product Dimension Table

ItemKey	ProductID	ProductName	ProductPrice
1	1	AZ-204	10.99
2	2	DP-203	11.99
3	3	AZ-104	12.99
4	1	BookA	10.99
5	2	BookB	11.99
6	3	BookC	12.99

Here the ItemKey is a surrogate key

Helps to uniquely identify each row in the table

You can use the Identity column feature in Azure Synapse for tables to generate the unique ID

Ideal practice

Don't have NULL values for properties in the dimension table. Won't give desired results when using reporting tools.

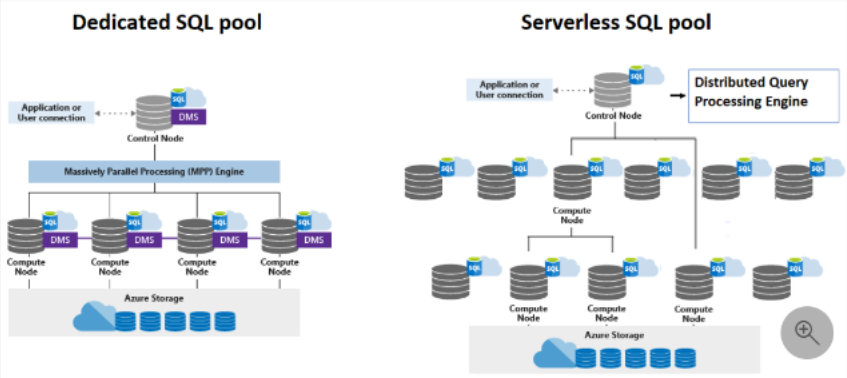
Try to replace the NULL value with some default value

Understanding the Azure Synapse Architecture

Azure Synapse Architecture

Reference

- https://learn.microsoft.com/en-us/azure/synapse-analytics/sql/overview-architecture



Here the compute and storage are seperate components

Here the compute and storage can scale independently

With the dedicated SQL pool, you can pause the compute

You still pay for the storage

For the dedicated SQL pool , the compute power is charged in terms of DWU.

Data Warehousing unit - This is a combination of compute, memory and IO

When it comes to storage, your data is actually sharded or split into distributions.

This is used to optimize on performance

There are 60 distributions in the dedicated SQL pool

```
DBCC PDW_SHOWSPACEUSED('[dbo].[FactSales]')
```

Tables types in Azure Synapse

Different types of tables

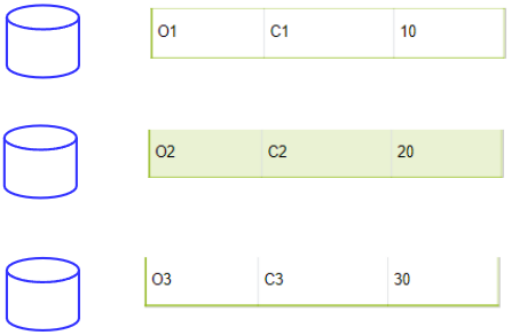
Hash-distributed tables **Round-Robin distributed tables** **Replicated tables**

Hash-distributed tables

Here the SQL pool uses a hash function to decide how to assign a row of data to a distribution.

This is good when you have large tables, the data can be evenly spread across the distributions.

Order ID	Course ID	quantity
O1	C1	10
O2	C2	20
O3	C3	30



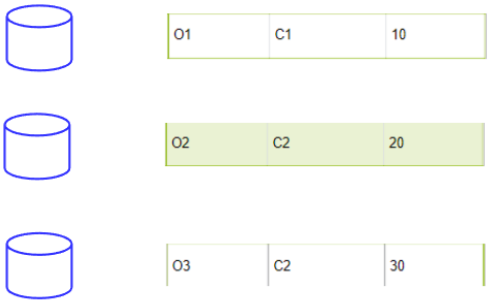
If your column data used for hashing is the Order ID

Round-Robin distributed tables

This type of table provides fast performance when you want to load data into staging tables

Here the data is evenly distributed across the table.

Order ID	Course ID	quantity
O1	C1	10
O2	C2	20
O3	C2	30



In retrospective to hash-distributed tables

If your column data used for hashing is the Couse ID

Order ID	Course ID	quantity
O1	C1	10
O2	C2	20
O3	C2	30

O1	C1	10
O2	C2	20

O3	C2	30

Replicated tables

This provides the fastest performance for small tables

Here each compute node caches a fully copy of the table.

Order ID	Course ID	quantity
O1	C1	10
O2	C2	20
O3	C2	30

Order ID	Course ID	quantity
O1	C1	10
O2	C2	20
O3	C2	30

Order ID	Course ID	quantity
O1	C1	10
O2	C2	20
O3	C2	30

Order ID	Course ID	quantity
O1	C1	10
O2	C2	20
O3	C2	30

About the staging area

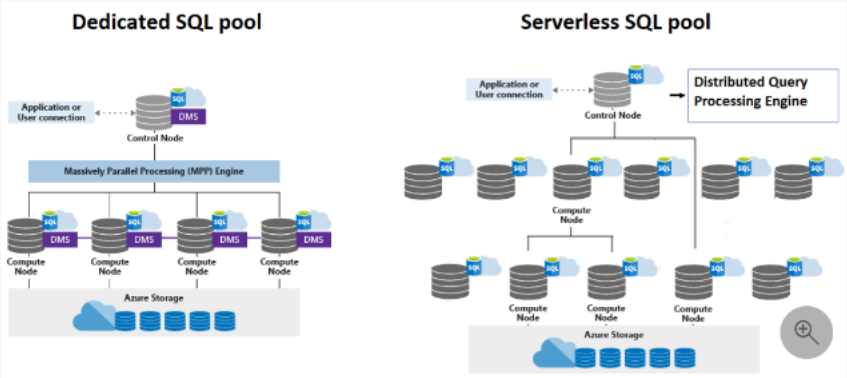
In case of Azure Synapse, you can also get a faster load time of data if you first load data into staging tables. And then copy the data from the staging table data into the data warehouse tables.

Compute to data distribution

Azure Synapse Architecture

Reference

- https://learn.microsoft.com/en-us/azure/synapse-analytics/sql/overview-architecture



Here the compute and storage are seperate components

Here the compute and storage can scale independently

Since your data is split across distributions. Each compute node can work on each distribution independently.

The compute node actually takes the user query, splits the query in such a way that each compute node can work independently.

Designing the tables

Fact Tables

Tend to be large

Hash distributed tables

Decide on the column value for hashing purposes

Dimension tables

Tend to be smaller

Replicated tables

Staging tables

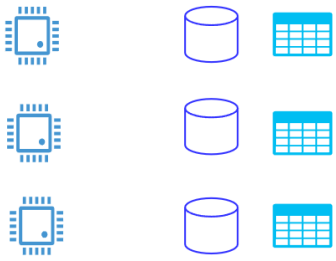
Round-Robin Distribution

Fact Tables - Orders

Portion of hash distributed table based on Course Category ID

Dimension Table - Instructor

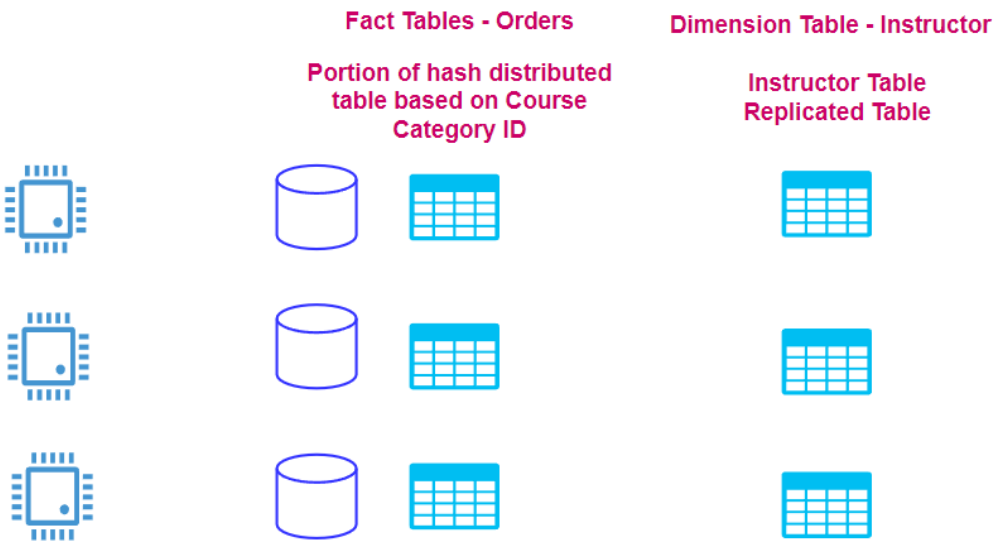
Instructor Table Round-Robin Distribution



Azure Synapse uses the Data Movement Service to facilitate data movement across Compute Nodes.

This can be an expensive operation.

JOIN the Fact Table and Dimension Table to get popular instructors for a particular course category



Lab- Surrogate Keys

Surrogate Keys

Here the ProductID is referred to as the Alternate Key or the Business Key

This refers to the Primary Key in the source system

100 %

Results | Messages

	ProductID	ProductModelID	ProductcategoryID	ProductName	ProductModelName	ProductCategoryName
1	680	6	18	HL Road Frame - Black, 58	HL Road Frame	Road Frames
2	706	6	18	HL Road Frame - Red, 58	HL Road Frame	Road Frames
3	707	33	35	Sport-100 Helmet, Red	Sport-100	Helmets
4	708	33	35	Sport-100 Helmet, Black	Sport-100	Helmets
5	709	18	27	Mountain Bike Socks, M	Mountain Bike Socks	Socks
6	710	18	27	Mountain Bike Socks, L	Mountain Bike Socks	Socks
7	711	33	35	Sport-100 Helmet, Blue	Sport-100	Helmets
8	712	2	23	AWC Logo Cap	Cycling Cap	Caps
9	713	11	25	Long-Sleeve Logo Jersey, S	Long-Sleeve Logo Jersey	Jerseys
10	714	11	25	Long-Sleeve Logo Jersey, M	Long-Sleeve Logo Jersey	Jerseys
11	715	11	25	Long-Sleeve Logo Jersey, L	Long-Sleeve Logo Jersey	Jerseys
12	716	11	25	Long-Sleeve Logo Jersey, ...	Long-Sleeve Logo Jersey	Jerseys
13	717	6	18	HL Road Frame - Red, 62	HL Road Frame	Road Frames

In Dimension tables, you will also want to have a surrogate key

The Surrogate key is also sometimes referred to as the non-business key

This can be simple incrementing integer values

In the SQL pool tables, you can use the IDENTITY column feature

Slowing changing dimensions

DimProduct Table

What happens if the ProductName has a slight change in the source system

Results | Messages

	ProductSK	ProductID	ProductModelID	ProductcategoryID	ProductName	ProductModelName	ProductCategoryName
1	16	680	6	18	HL Road Frame - Black, 58	HL Road Frame	Road Frames
2	76	706	6	18	HL Road Frame - Red, 58	HL Road Frame	Road Frames
3	136	707	33	35	Sport-100 Helmet, Red	Sport-100	Helmets
4	196	708	33	35	Sport-100 Helmet, Black	Sport-100	Helmets
5	256	709	18	27	Mountain Bike Socks, M	Mountain Bike Socks	Socks
6	316	710	18	27	Mountain Bike Socks, L	Mountain Bike Socks	Socks
7	376	711	33	35	Sport-100 Helmet, Blue	Sport-100	Helmets
8	436	712	2	23	AWC Logo Cap	Cycling Cap	Caps
9	496	713	11	25	Long-Sleeve Logo Jersey, S	Long-Sleeve Logo Jersey	Jerseys
10	556	714	11	25	Long-Sleeve Logo Jersey, M	Long-Sleeve Logo Jersey	Jerseys
11	616	715	11	25	Long-Sleeve Logo Jersey, L	Long-Sleeve Logo Jersey	Jerseys
12	676	716	11	25	Long-Sleeve Logo Jersey, XL	Long-Sleeve Logo Jersey	Jerseys

Type 1

Here you just update the changes as they are

Type 2

Here you keep both the OLD and NEW values in the Dimension table

ProductSK	ProductID	ProductName	StartDate	EndDate	IsCurrent
1	1	ProductA	2021-03-20	2021-04-20	False
2	1	ProductAA	2021-04-21	9999-12-31	True

Type 3

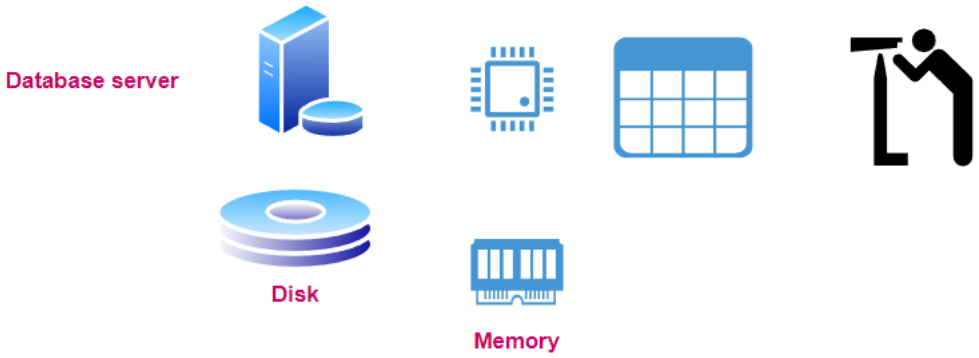
Here instead of having multiple rows to signify changes, we now have additional columns to signify the changes

ProductSK	ProductID	Original Name	Changed Name	EffectiveDate
1	1	ProductA	ProductAA	2021-04-20

Indexes

Indexes

Indexes are normally used in database systems to help query data more efficiently.



Youn might have queries that use the WHERE clause to search for data.

The database server must be able to efficiently search for the data.

In a dedicated SQL Pool, a Clustered Columnstore Index is automatically created.

This is good for large fact tables, it provides better compression of data and better query performance

With a normal Relational Database system, the data is normally stored in a row-wise format.

	ColumnA	ColumnB	ColumnC
Row 1	Value 1	Value 2	Value 3
Row 2	Value 4	Value 5	Value 6

Page 1 in Memory

	ColumnA	ColumnB	ColumnC
Row 3	Value 7	Value 8	Value 9
Row 4	Value 10	Value 11	Value 12

Page 2 in Memory

Wherease normally for a data warehouse, the data is stored in a column-wise format

Page 1 in Memory	Page 2 in Memory	Page 3 in Memory
ColumnA	ColumnB	ColumnC
Value 1	Value 2	Value 3
Value 4	Value 5	Value 6
Value 7	Value 8	Value 9
Value 10	Value 11	Value 12

Your queries against the SQL data warehouse might be targeted towards certain columns and aggregation towards the values in the column.

Where you cannot create clustered columnstore indexes

Consider heap tables or clustered index tables

1. If the table contains columns of the type varchar(max), nvarchar(max) or varbinary(max)

Consider heap tables

2. If you want to create temporary tables

3. If you have a table with less than 60 million rows

Clustered Indexes

This can be defined to improve the performance of queries that have highly selective filters, that really narrow down on a row of data.

Nonclustered Indexes

If you want to improve the filter performance on other columns, then you can create a nonclustered index

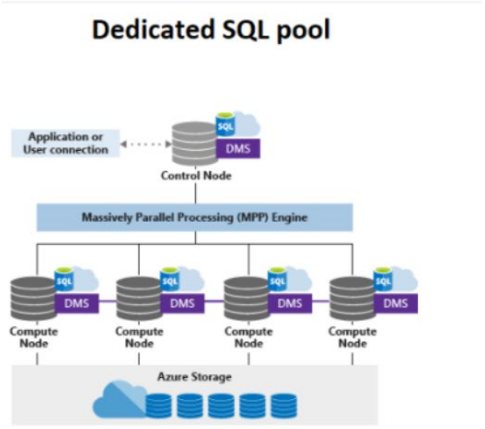
But the non-clustered index needs additional space and processing time for the index itself.

Which Method to choose

Copy command vs Polybase vs Bulk Insert

The Bulk Insert method is slower than the Copy command or Polybase

Polybase gives you better performance when you are loading large amounts of data



With Bulk Insert all of the commands go through the Control Node

You can use the Bulk insert command when you have less data to transfer

With Polybase, your data movement operations go through the compute nodes in parallel. This gives better loading times. And if you have more compute nodes, the data loading process becomes faster.

Polybase also remember allows you to create EXTERNAL tables. And then you can import data into the dedicated SQL pool , creating tables using the CREATE TABLE AS SELECT command.

Even the COPY command has high throughput and goes through the compute nodes in the dedicated SQL pool.

The COPY command is simple to execute, you don't need to define the EXTERNAL DATA SOURCE, EXTERNAL FILE FORMAT and EXTERNAL TABLE.

Partitions

Table Partitions

For your tables in a SQL Pool, your data is already distributed into different distributions

When it comes to a hash distributed table, this can be effective when you perform JOINS on your tables

But you can also split your table data into different partitions

This helps to divide your data into smaller groups of data

This helps when your queries are based on filtering of data , when using the WHERE clause

This helps to eliminate certain partitions when it comes to querying your data

Partitions are normally created on a date column

Supported for all distributions types when it comes to the dedicated SQL pool

Supported for heap tables, clustered columnstore and clustered indexes.

When it comes to a clustered columnstore table, for optimal compression and performance, you should ideally have a minimum of 1 million rows per distribution and partition.

Let's say that you have a table that has 12 monthly partitions.

Remember that a table already has 60 distributions.

So this already means that it should have 60 million rows when it comes to distributions.

And then a total of 60 million * 12 = 720 million rows of data considering the 12 partitions to effectively make use of partitions.

Also it makes operations like inserting and deleting data easier when you have partitions.

Design and Develop Data Processing- Azure Data Factory

Extract, Transform and Load

Application Log Files

Server log files

Sales log

Data comes in from various sources

Process the data

Makes it available in a data store

Data from different sources could be in different formats

Data needs to be changed to align with the format and structure of the destination database

Data needs to be cleaned

Extract, Transform and Load

This is designed as a pipeline to collect data from various sources.

The data is transformed accordingly

The transformed data is then load onto a target data store

Extract, Load and Transform

Here the key difference is that the Transformation happens after the data is loaded.

The Transformation occurs in the target data store.

What is Azure Data Factory

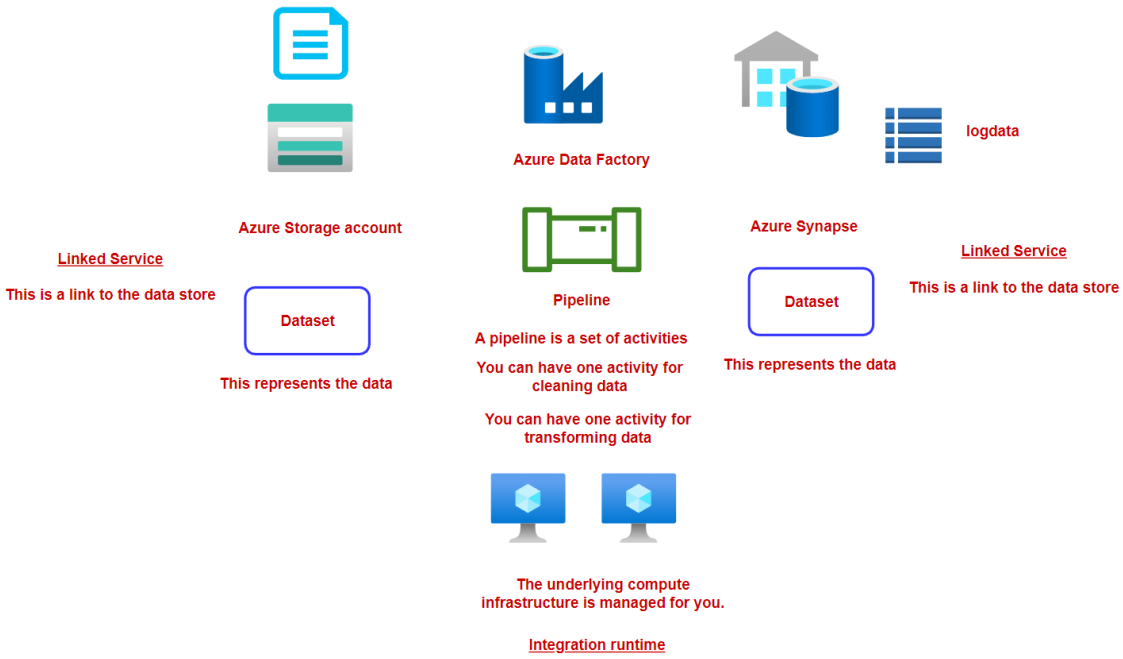
Azure Data Factory

This is a cloud-based ETL and data integration service

You can create data-driven workflows that can be used for orchestrating data movement

It can also be used to transform data at scale

You can connect to a variety of data stores as the source and the destination

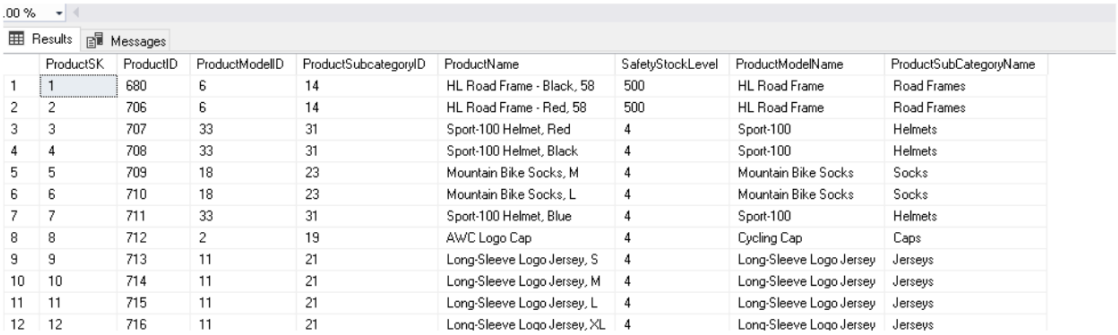


Lab- Cache Sink- The first step

Using Cache Sink and Lookup

The Cache sink is used to write data into the Spark cache and not the data store

The Cache lookup can then be used to reference the data in the Cache sink



.00 %

Results | Messages

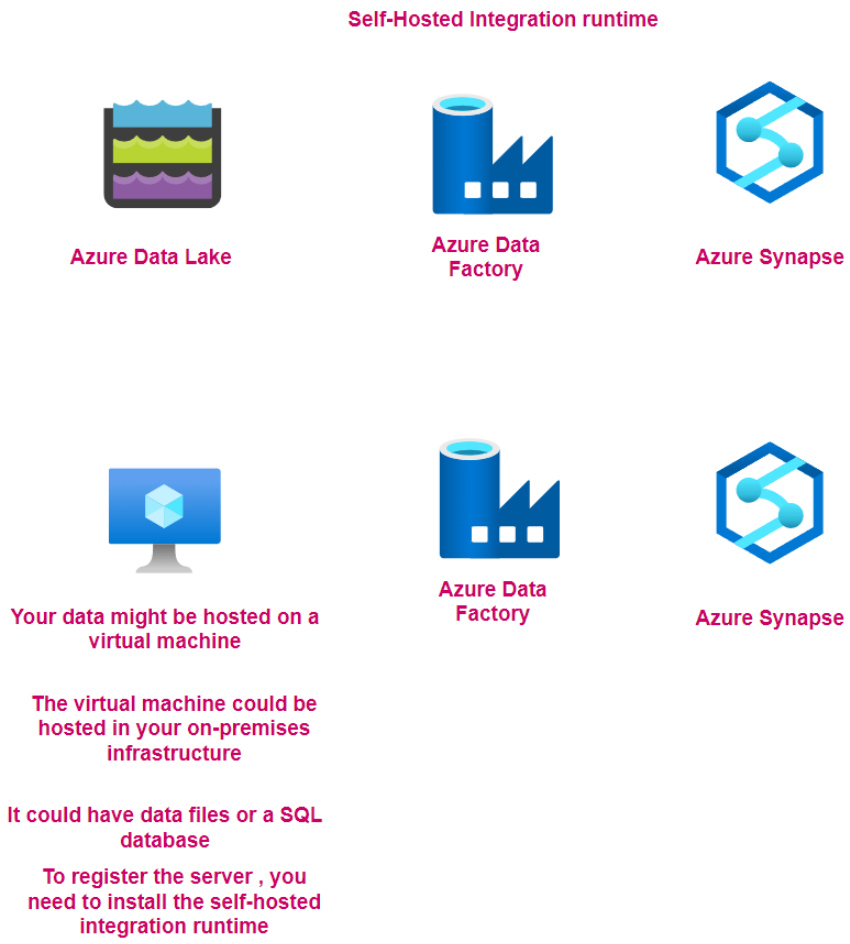
	ProductSK	ProductID	ProductModelID	ProductSubcategoryID	ProductName	SafetyStockLevel	ProductModelName	ProductSubCategoryName
1	1	680	6	14	HL Road Frame - Black, 58	500	HL Road Frame	Road Frames
2	2	706	6	14	HL Road Frame - Red, 58	500	HL Road Frame	Road Frames
3	3	707	33	31	Sport-100 Helmet, Red	4	Sport-100	Helmets
4	4	708	33	31	Sport-100 Helmet, Black	4	Sport-100	Helmets
5	5	709	18	23	Mountain Bike Socks, M	4	Mountain Bike Socks	Socks
6	6	710	18	23	Mountain Bike Socks, L	4	Mountain Bike Socks	Socks
7	7	711	33	31	Sport-100 Helmet, Blue	4	Sport-100	Helmets
8	8	712	2	19	AWC Logo Cap	4	Cycling Cap	Caps
9	9	713	11	21	Long-Sleeve Logo Jersey, S	4	Long-Sleeve Logo Jersey	Jerseys
10	10	714	11	21	Long-Sleeve Logo Jersey, M	4	Long-Sleeve Logo Jersey	Jerseys
11	11	715	11	21	Long-Sleeve Logo Jersey, L	4	Long-Sleeve Logo Jersey	Jerseys
12	12	716	11	21	Long-Sleeve Logo Jersey, XL	4	Long-Sleeve Logo Jersey	Jerseys

Let's say you want to continue the ProductSK key from where it left off

So we can get the maximum value first in the ProductSK column and then store it in the cache

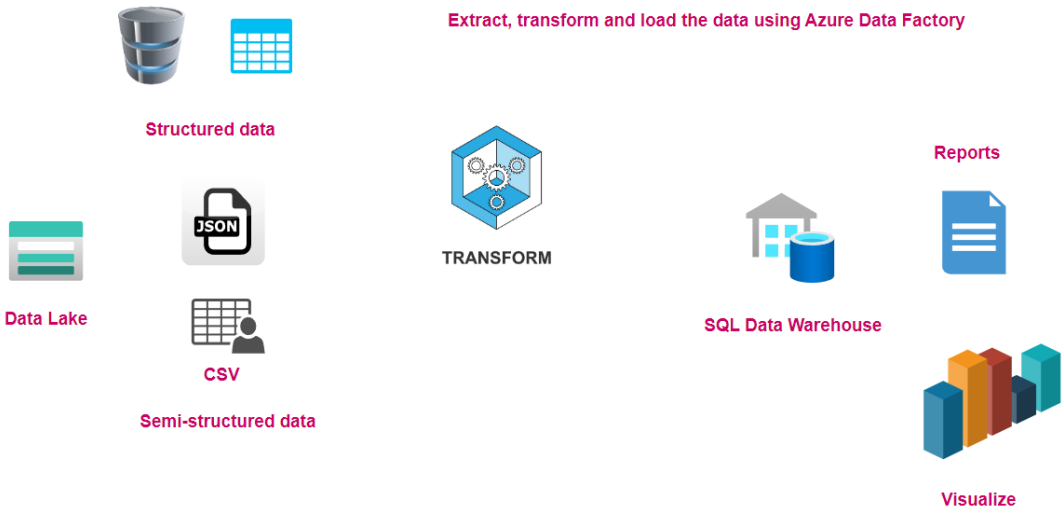
And then reference that value using the cache lookup feature

Self-Hosted Integration Runtime



Design and Develop Data Processing- Azure Event Hubs ,Stream Analytics

Batch and Real-Time Processing



Batch Processing

Here you would consider transfering the rows of data as a batch

Daily Nightly process of taking the delta data

Here the disadvantage is that if the load fails for any reason , you need to carry out the load process again

And depending on the amount of data this could take time

Real time processing

Here the data is processed in real time, at that point itself.

Process streams of data every 5 minutes

The data ingestion service must be able to consume and process the data in real time.

What are Azure Event Hubs

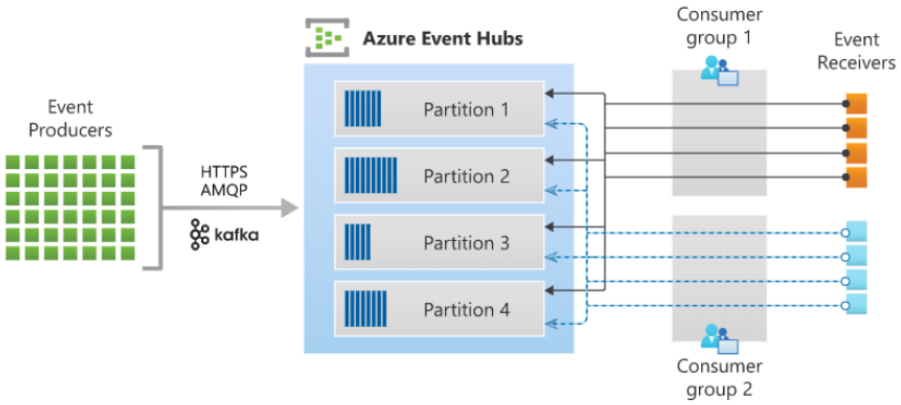
What are Azure Event Hubs

This is a big data streaming platform

This service can receive and process millions of events per second

You can stream log data , telemetry data, any sort of events to Azure Event Hubs

Event Hubs Architecture



https://docs.microsoft.com/en-us/azure/event-hubs/event-hubs-features

The different components

Event producers - This is an entity that sends data to an event hub. The events can be published using the protocols - HTTPS, AMQP, Apache Kafka

Partitions - The data is split across partitions. This allows for better throughput of your data onto Azure Event Hubs

Consumer groups - This is a view (state,position or offset) of an entire event hub

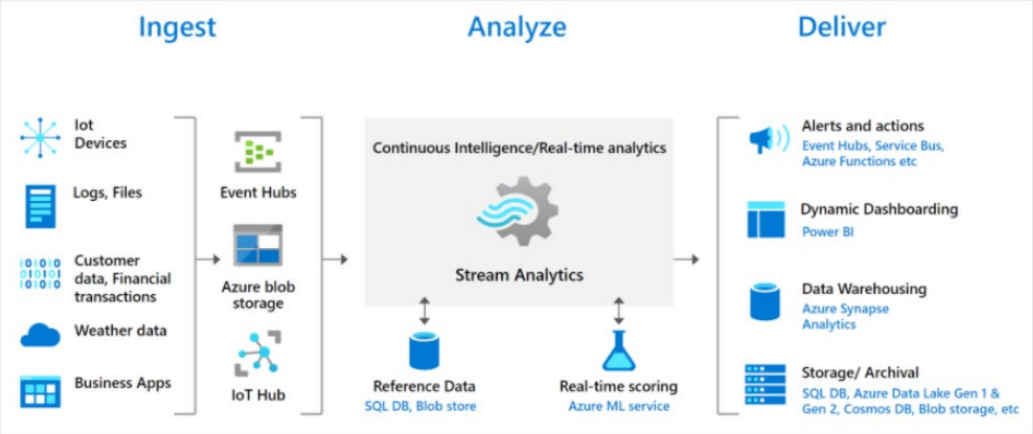
Throughput - This controls the throughput capacity of Event Hubs

Event Receivers - This is an entity that reads event data

What is Azure Stream Analytics

Azure Stream Analytics

This is a real-time analytics and event-processing service



https://docs.microsoft.com/en-us/azure/stream-analytics/stream-analytics-introduction

Design and Develop Data Processing- Scala, Notebooks and Spark

Quick look at Jupyter Notebook

Jupyter Notebook

This is a web application that allows one to create and share documents

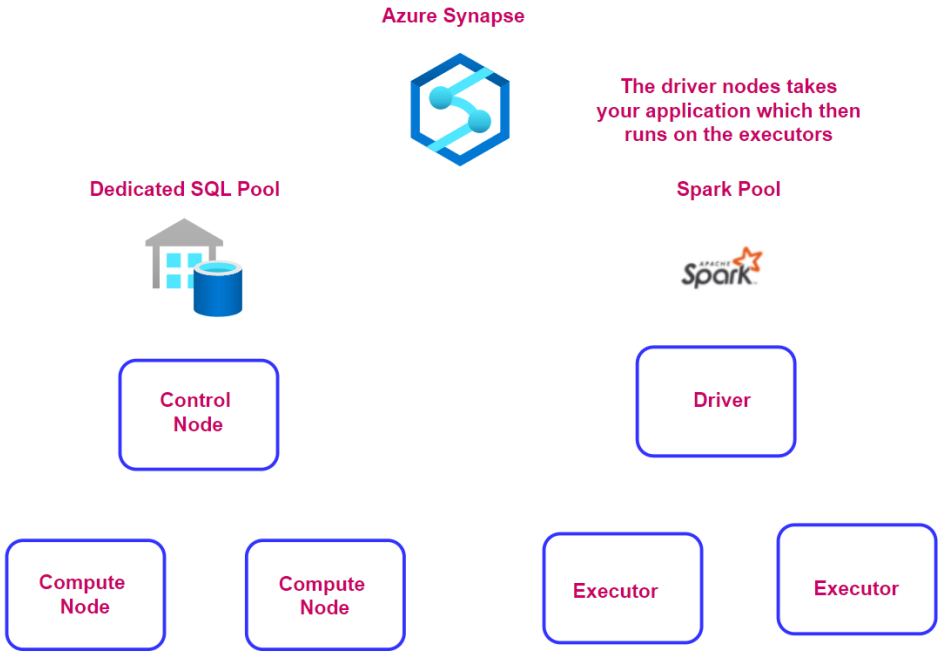
These documents contain live code

The documents can also contain visualizations

The normal use cases for notebooks - Data cleaning and transformation, statistical modeling , machine learning

To get started , you can install Jupyter notebooks on your local machine

Spark Pool- Combined Power



In the Spark Pool, the Spark Instances are created when you connect to the Spark pool, create a session and then run a job

When you submit another job, if there is capacity in the pool and the Spark instance has spare capacity, it will run the second job

Else if the pool has the capacity it will create a new Spark instance to run the second job

Design and Develop Data Processing- Azure Databricks

What is Azure Databricks

Databricks

This is a company that provides a platform for developing and maintaining enterprise-grade data solutions.

The creators of Spark went ahead and created the Databricks platform

Ingest data | Process data | Work with SQL | Data Exploration

Machine Learning | Security and governance | Visualizations

Structured data

SQL Data Warehouse

CSV

Semi-structured data

Data Lake

The data grows at a rapid pace in your data lake

Looking at delta data can be a challenge

Data governance can be an issue

TRANSFORM

SQL Data Warehouse

Different tools and platforms

Security can become an issue

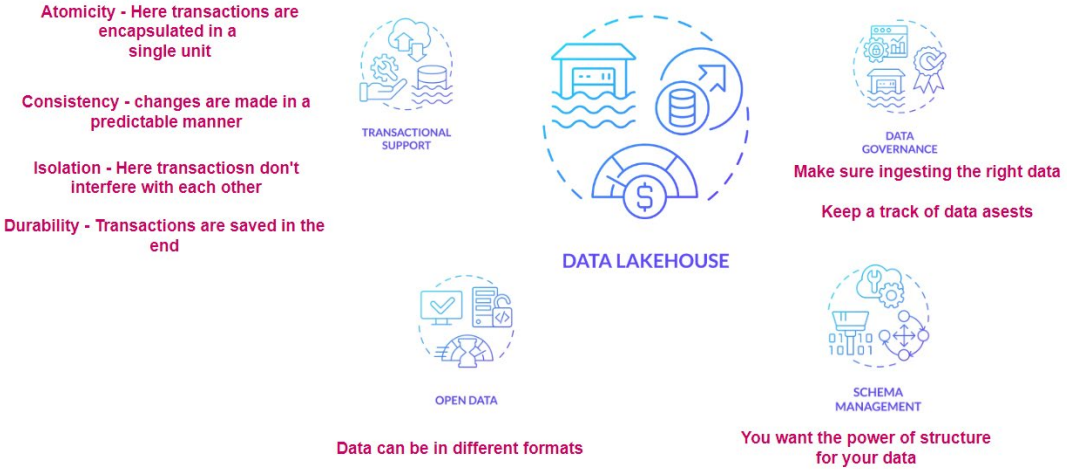
You need your data to be in a structured format before you can start analyzing your data

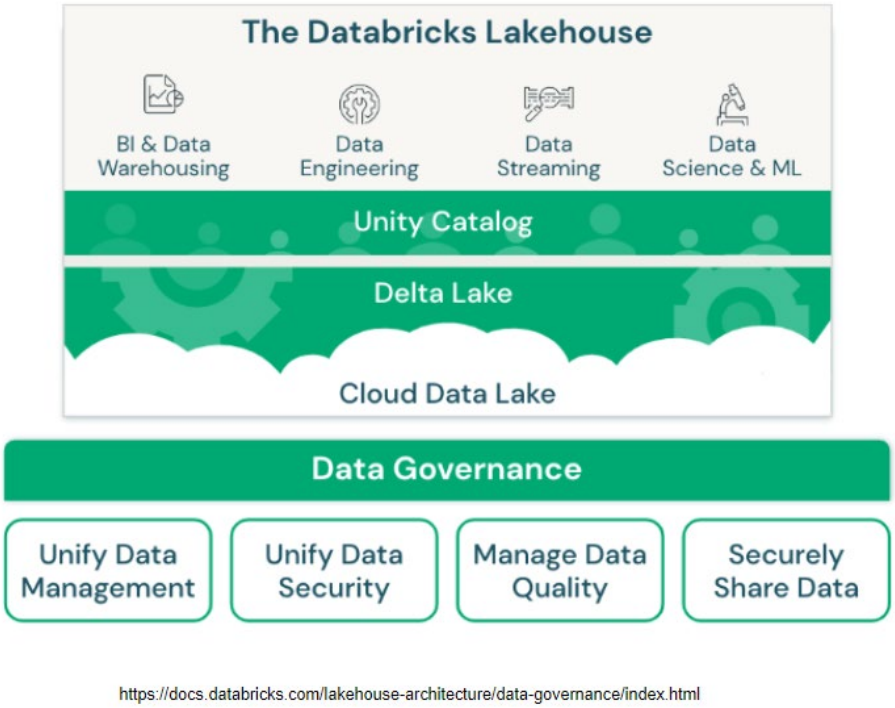
A lot of data comes in semi-structured formats like JSON.

It becomes expensive to maintain a data warehouse

You can make use of EXTERNAL tables in Serverless SQL pools

We can use data pipelines to orchestrate data movement

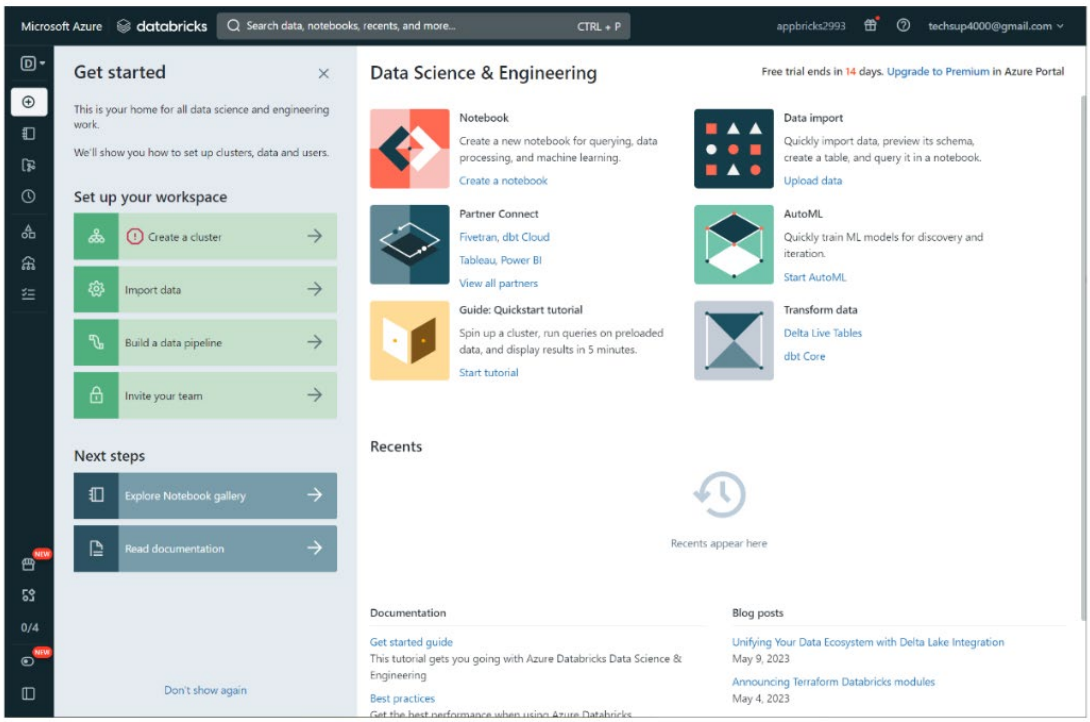




https://docs.databricks.com/lakehouse-architecture/data-governance/index.html

Concepts with Azure Databricks

Create a workspace - This is used for storing all of your Databricks assets.



You need compute resources for processing data - Here you can run your notebooks and jobs.

You can create an all-purpose cluster for your interactive analysis. Here you can manually terminate and restart the cluster.

You can create a job cluster that can be used for running your jobs. The cluster is terminated when the job is complete.

Databricks runtime - This includes the use of Apache Spark.

Databricks File System - This is an abstraction layer on top of blob storage. Here you can store your folders and files.

You can make use of Notebooks when it comes to running your commands and visualizations via a web interface.

You are charged based on Databricks units.

Database - Collection of objects such as tables and views

Table - This is a representation of your structured data

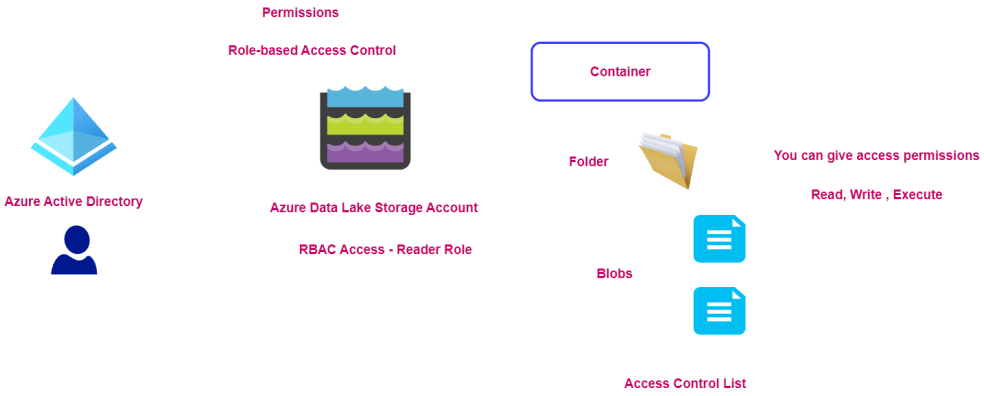
Delta tables

This is the default way tables get created.

Here your data can be stored as files in cloud storage. The table metadata is stored in the metastore. This helps to maintain ACID support for data.

Design and Implement Data Security

Using Azure Active Directory



	File	Directory
Read (R)	Can read the contents of a file	Requires **Read** and **Execute** to list the contents of the directory
Write (W)	Can write or append to a file	Requires **Write** and **Execute** to create child items in a directory
Execute (X)	Does not mean anything in the context of Data Lake Storage Gen2	Required to traverse the child items of a directory

https://learn.microsoft.com/en-us/azure/storage/blobs/data-lake-storage-access-control

Look back at how we were accessing the data lake account

Accessing Azure Data Lake Storage Account from Azure Databricks

Here we were using the Account keys

Cmd 1

```
import org.apache.spark.sql.types._
import org.apache.spark.sql.functions._


spark.conf.set(
    "fs.azure.account.key.datalake244434.dfs.core.windows.net",
    "dilbGv2rof6G4emB0qWgVwAOOexu/bIpvJiUnfal7+klHqCsKLB+JkQzMfRlgu0fm14iUFNHXPeU+AStZZXK2w==")

val file_location = "abfss://csv@datalake244434.dfs.core.windows.net/Log.csv"
val file_type = "csv"

val dataSchema = StructType(Array(
    StructField("Correlationid", StringType, true),
    StructField("Operationname", StringType, true),
    StructField("Status", StringType, true),
    StructField("Eventcategory", StringType, true),
    StructField("Level", StringType, true),
    StructField("Time", TimestampType, true),
    StructField("Subscription", StringType, true),
    StructField("Eventinitiatedby", StringType, true),
    StructField("Resourcetype", StringType, true),
    StructField("Resourcegroup", StringType, true),
    StructField("Resource", StringType, true)))


val df1 = spark.read.format(file_type).
options(Map("header"->"true")).
schema(dataSchema).
load(file_location)

display(df1)
```

Using Role-based access control

```
df = spark.read.load('abfss://parquet@datalake244434.dfs.core.windows.net/Log.parquet', format='parquet')
display(df)
```

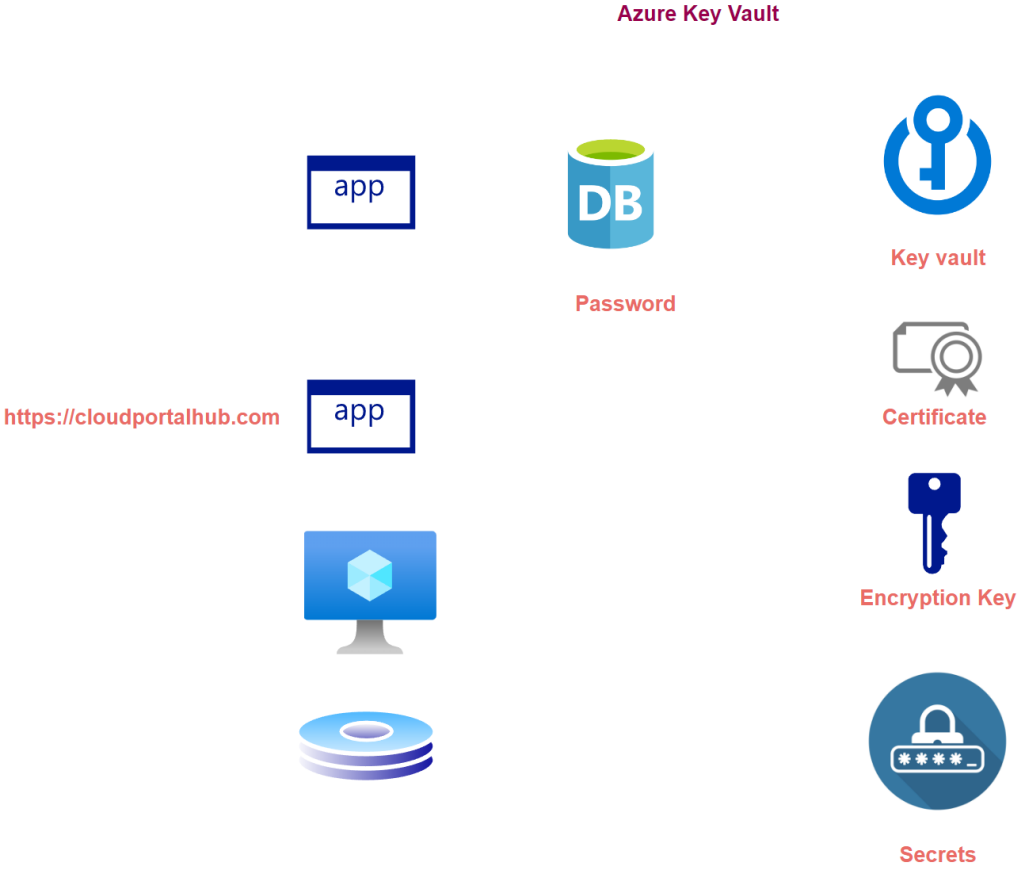
[1] ✓ 3 min 52 sec - Apache Spark session started in 3 min 28 sec 231 ms. Command executed in 23 sec 489 ms by techsup4000 on 8:15:21 PM, 5...

Job execution Succeeded **Spark** 1 executors 4 cores View in monitoring Open Spark UI

View Table Chart Export results

Correlationid	Operationname	Status	Eventcategory
99fe9c3a-e36e-44e0-acd4-58272...	Update SQL database	Succeeded	Administrative
99fe9c3a-e36e-44e0-acd4-58272...	Create Deployment	Started	Administrative
99fe9c3a-e36e-44e0-acd4-58272...	Create Deployment	Accepted	Administrative

Lab- Azure Databricks- Secret Scope- Key Vault


Azure Key Vault
app
DB
Key vault
Password
https://cloudportalhub.com
app
Certificate
Encryption Key
Secrets

About Managed Identities

Azure Active Directory

Principal

More on identities

Give permissions to resources

Role-based access control

Azure data lake storage account

Application object in Azure AD

Give permissions to resources

Role-based access control

Managed Identity

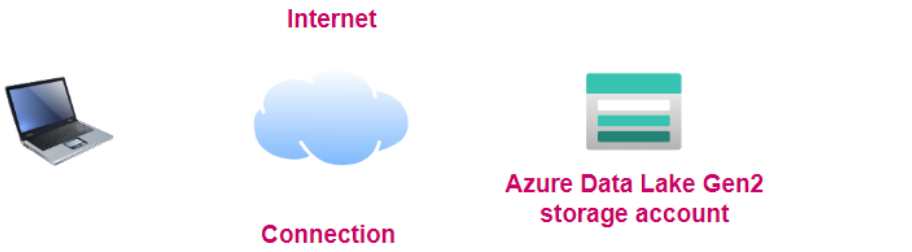
Azure Active Directory

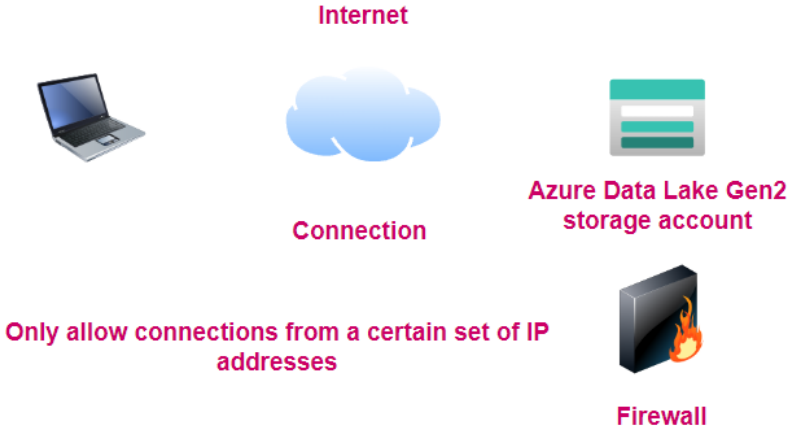
Give permissions to resources

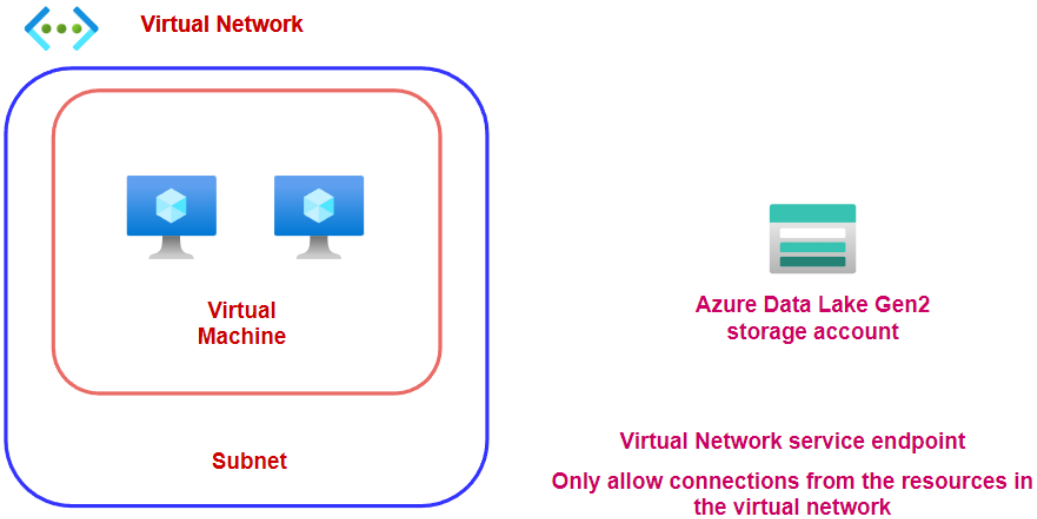
Role-based access control

Azure data lake storage account

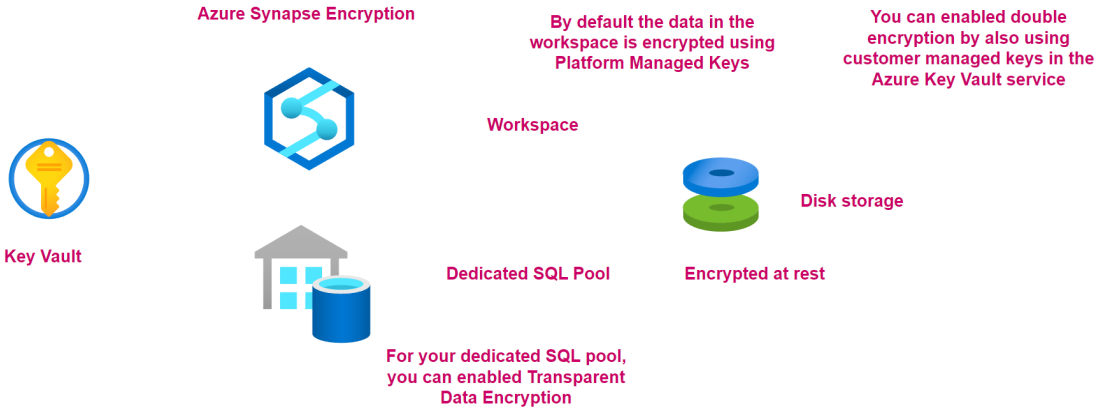
Azure Storage Accounts- Network and Firewall







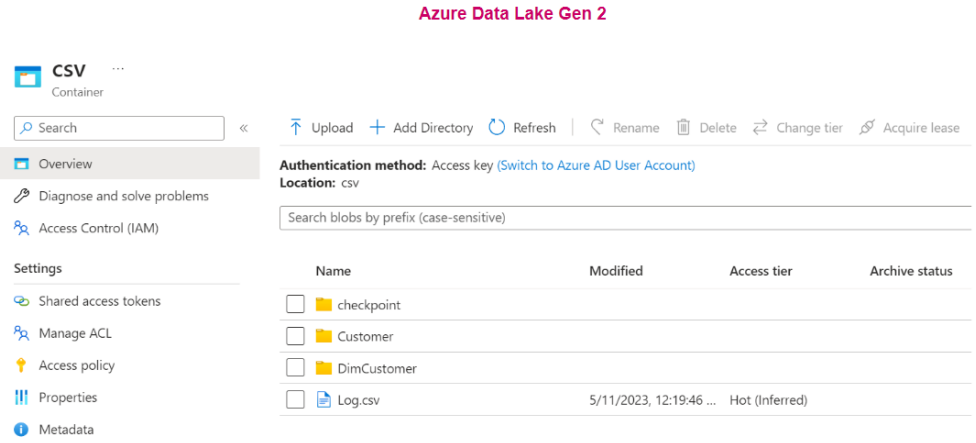
Azure Synapse Encryption



Azure Synapse- Azure AD Authentication

Monitor and optimize data storage and data processing

Best practices for structing files in your data lake



Your data is stored in the form of blobs

You can store data in various file formats - CSV , JSON, Parquet, Avro

The CSV and JSON file formats take up more space

Avro and Parquet are much more efficient in terms of file storage - Compression techniques

Parquet - Column-based file storage

Avro - Row-based file storage

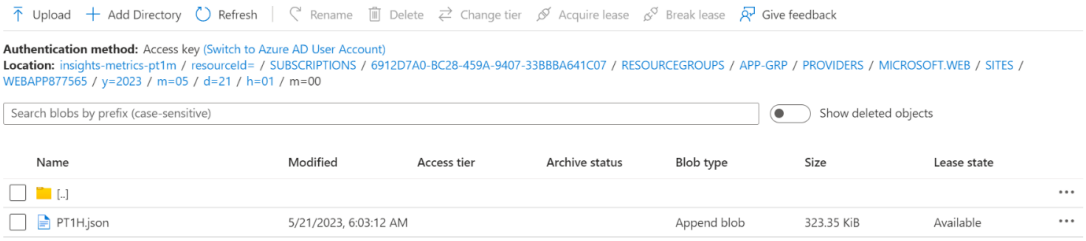
Try to have larger files when it comes to data processing, instead of having many smaller files.

Systems are more efficient in processing larger files.

Data structure

Time-related data

More efficient to load data from multiple time folders



Data in different zones

raw data - This is data ingested from different sources

Data that is then enriched - Here data is cleaned, bound to a schema. Here data engineers can create different data sets. Get valuable insights.

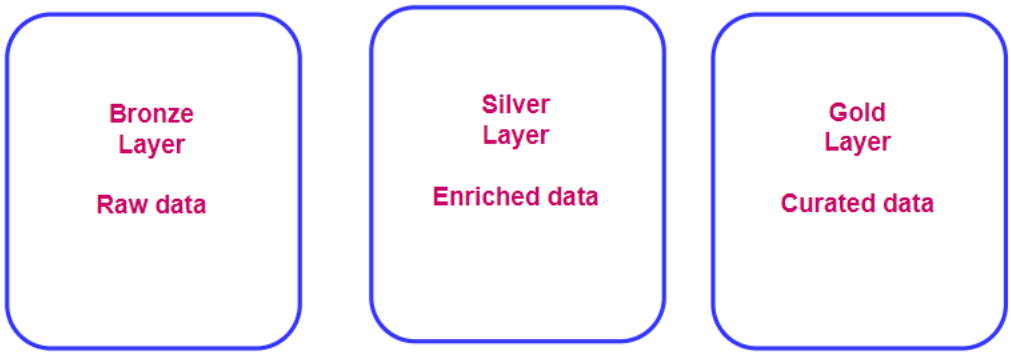
Develop curated data - This can be served to users to create rich insights onto the data.

/raw/department/application

/processed/department/application

/curated/department/application

Medallion lakehouse architecture



Azure Data Lake Gen2- Access tiers

Azure storage access tiers

Objects

There is a cost for storing objects

There is a cost for accessing objects

Companies might store millions of objects in a storage account

Use case - Initial there could be some objects that are accessed quite freqently. Then after some time, maybe a week or two , those objects are accessed less frequently.

Can a company save on costs when it comes to less frequently accessed objects.

An object can be set to a particular tier

Hot Access tier

This is optimized for objects accessed more frequently

Here you have high storage costs and lower access costs

Here the data needs to be stored for at least 30 days

Cool Access tier

This is optimized for objects accessed or modified infrequently

Here you have lower storage costs but higher access costs when compared with the Hot access tier.

This is optimized for objects that are rarely accessed

Here the data needs to be stored for at least 180 days

Archive Access tier

Here you have lower storage costs but higher access costs when compared with the Cool access tier.

Good for long-term backups.

You can set the Hot and the Cool access tier at the storage account level.

You can set the Hot ,Cool and Archive access tier at the blob level.

Azure Stream Analytics- Streaming Units

Azure Stream Analytics job

When you create a Stream Analytics job, you assign a number of Streaming Units

The streaming units determines the computing resources that are allocated to execute the job

To ensure low latency when it comes to stream processing, all of the jobs are performed in memory

If the Streaming Units utilization reaches 100% , then your jobs will start failing

The metric when it comes to the Streaming Units percentage it does not take into account the amount of memory being used for the job

Hence always ensure to monitor the streaming units being consumed for a job

If you Streaming Units utilization is high , like 80%, add more Streaming units to the job.

Buit if the streaming units is low, and you still have backlogged events, it could be that the CPU is not able to keep up with the processing of events. So in this case you should increase the number of streaming units.

Azure Stream Analytics- The importance of time

Azure SQL database

Application time - This is when the application generated the event

Azure Event Hub

Arrival Time - This is when the event reaches the Azure Event Hub

Here in Azure Stream Analytics this is represented as EventEnqueuedUtcTime

Azure Stream Analytics

Then the event reaches Azure Stream Analytics

In Azure Stream Analytics the events are processed by their arrival time

If the application generates the Application time, you can tell Azure Stream Analytics via the query and TIMESTAMP BY to process the events via the Application time.

In order for Azure Stream Analytics to understand the events that are coming in, it creates a watermark just to understand the ordering of events

Why does it need to do this?

Clock Skews

There is no guarantee that all clocks for all systems are sychronized

Network latency

Delays for the events to reach the intended destination

Azure SQL database

Azure Event Hub

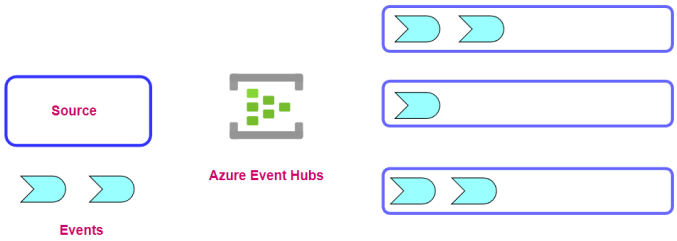
Azure Stream Analytics

Hence you could have late arriving events or early arriving events

Events on the application side are not generated that frequently

The Azure Event Hubs could be having too many partitions that is not required if the incoming data is less.

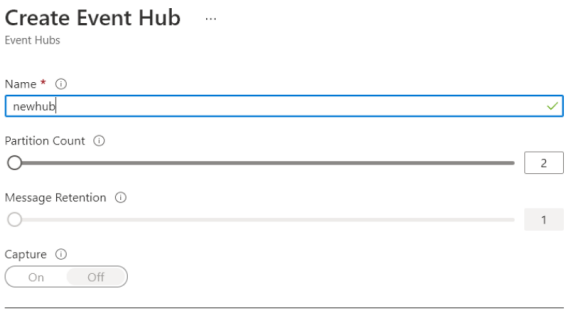
Azure Event Hubs and Stream Analytics – Partitions



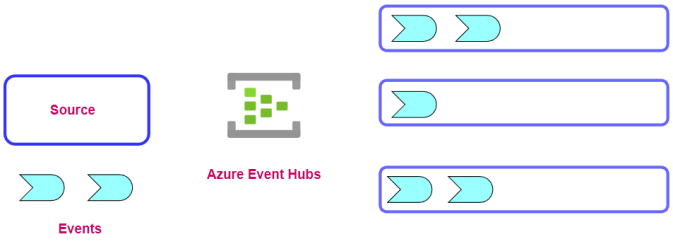
The events can be sent to different partitions

This helps in better throughput , where the ingestion service can take in more events at a time

You can also have multiple instances of a consumer to consume the messages from each partition, increasing the throughput at which you can receive messages



You can mention the number of partitions when creating the event hub



The events can be sent to different partitions

This helps in better throughput , where the ingestion service can take in more events at a time

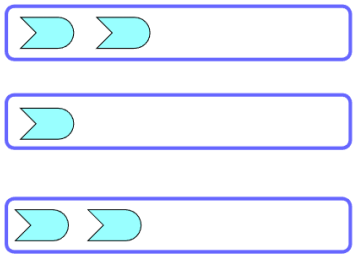
You can also have multiple instances of a consumer to consume the messages from each partition, increasing the throughput at which you can receive messages

When sending events from the source, you can decide which attribute of your data can be used as a partition key

Like a per-device or user identity attribute

Then the events can be split across the multiple partitions

Azure Event Hubs



Partitions

Azure Stream Analytics

The job can also work on the different partitions in parallel

This helps to increase the efficiency of processing

So Azure Stream Analytics can make use the partitions that are available in Azure Event Hubs