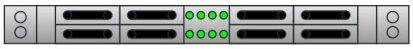
Identify considerations for relational data on Azure

Different options for hosting a database

What would I need to do to have a relational database in place.

RAID Array 

I would need to procure a server, buy storage.

On the server, I would install an Operating System - Red Hat Linux, Ubuntu Server, Microsoft Windows Server.

Then install the database engine of choice - Oracle , Microsoft SQL Server, MySQL.

The database administrator then needs to maintain the database server and the databases itself.

We can then start having our tables of information in the databases.

All of this means that we need to invest initially in hardware.

A cloud platform normally allows us to host workloads without the need of investing in hardware.

The hardware is actually mantained by the cloud provider.

Option 1

We create a virtual machine on Azure.

We can access this virtual machine like any other machine. We can then install a database engine on the machine.

Here we don't need to make any sort of initial investment on the hardware.

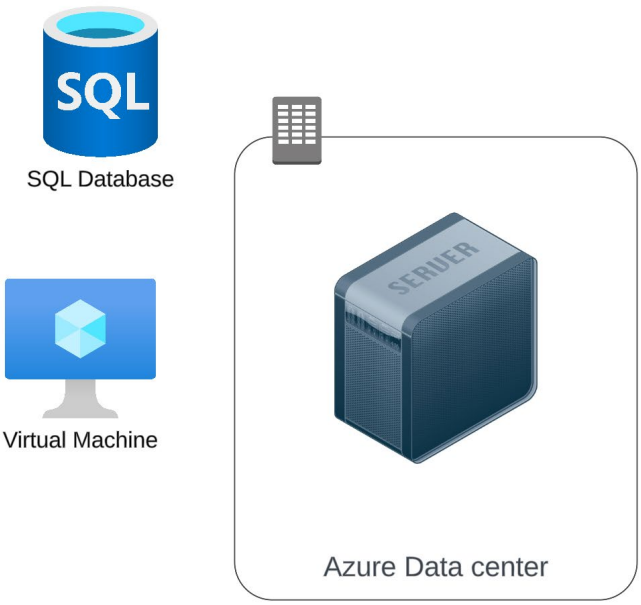
For the virtual machine, we only pay based on how much we use.

At any point in time, we can delete the virtual machine if it's no longer required.

This is an example of Infrastructure as a Service (IaaS)

Option 2

Azure SQL Database



Here even the virtual machine, the compute is managed for you. You don't need to manage the underlying database engine.

This is an example of Platform as a Service (IaaS)

Lab - Creating a virtual machine

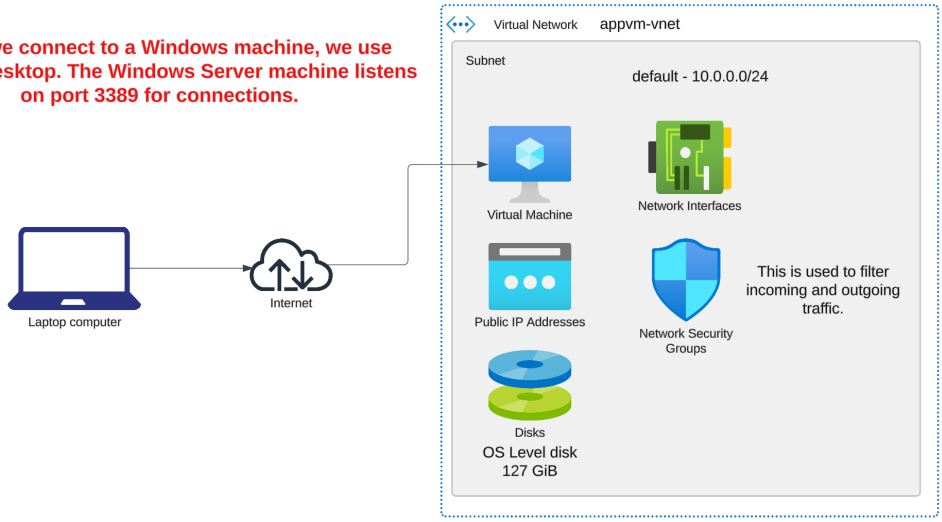
What are we going to do.

Step 1: We are going to build an Azure virtual machine.

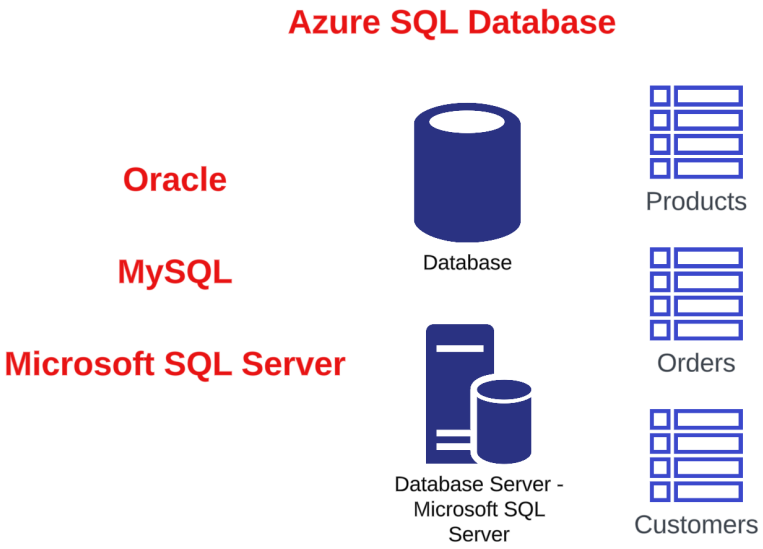
Step 2: We will log into the machine and install the Microsoft SQL Server engine.

Step 3: We will then host a database , a simple table and some data.

When we connect to a Windows machine, we use Remote desktop. The Windows Server machine listens on port 3389 for connections.



Azure SQL Database



We need to install the database software on some sort of compute hardware.

We can install the database engine on a virtual machine.

Database administrator responsibilities

1. Uptime of the database server
2. Database backups and restore
3. Patch installation at the operating system and database engine level.

Azure SQL

If the company does not want the burden of managing the underlying infrastruture, they can opt to use the Azure SQL Database service.


Virtual Machine

SQL Servers

Here the underlying server is managed for you. The database software will be in place. It also has features such as backup/restore and several other features.

SQL Databases

You can simply start hosting your databases. The Azure SQL database is the cloud version of Microsoft SQL Server.

Database Normalization

Why do we even bother with JOINS, why can't we just put everything in one table.

Well if we try to put everything in one table, it can lead to data duplication. Table can become large in terms of number of columns, could become difficult to maintain.

When designing relational tables, we follow the different normal forms.

First Normal Form

Here each column in the table must have a single value.

Columns should not have a set of values.

Student Table

StudentName	CourseName	TotalPrice
StudentA	DP-900, AZ-900	19.99
StudentB	DP-203	12.99
StudentC	DP-900, AZ-900, SC-100	25.99
StudentB	AZ-104	13.99

Here the Course Name has a set of values.

Here we can split the course name.

StudentName	CourseName	Price
StudentA	DP-900	9.99
StudentA	AZ-900	10.00
StudentB	DP-203	12.99
StudentC	DP-900	9.99
StudentC	AZ-900	10.00
StudentC	SC-100	6.00
StudentB	AZ-104	13.99

Now we have repeating rows of information.

The other issue is that , let's say we have a new student on our platform, and that student has not purchased a course as yet.

StudentName	CourseName	Price
StudentA	DP-900	9.99
StudentA	AZ-900	10.00
StudentB	DP-203	12.99
StudentC	DP-900	9.99
StudentC	AZ-900	10.00
StudentC	SC-100	6.00
StudentB	AZ-104	13.99
StudentD	Null	Null

Hence its an ideal scenario to keep the Student and Course Information seperate.

Because let's say that StudentD buys a course but then takes a refund, again we have an issue.

StudentID	StudentName
S1	StudentA
S2	StudentB
S3	StudentC
S4	StudentD

Student Table

CourseID	CourseName	StudentID	Price
C1	DP-900	S1	9.99
C2	AZ-900	S1	10.00
C3	DP-203	S2	12.99
C1	DP-900	S3	9.99
C2	AZ-900	S3	10.00
C4	SC-100	S3	6.00
C5	AZ-104	S2	13.99

Course Table

But here let's note that the price of the course has no dependency on the StudentID.

2nd Normal Form

Every non-prime attributes need to be functionally dependent on the candidate key.

StudentID	StudentName
S1	StudentA
S2	StudentB
S3	StudentC
S4	StudentD

Student Table

CourseID	CourseName	Price
C1	DP-900	9.99
C2	AZ-900	10.00
C3	DP-203	12.99
C4	SC-100	6.00
C5	AZ-104	13.99

Course Table

Course ID	Student ID
C1	S1
C2	S1
C3	S2
C1	S3
C2	S3
C4	S3
C5	S2

Orders Table

Now we can add Student information seperately, Course information seperately and Order information seperately.

Indexes in tables

Indexes

Results | Messages

	CustomerID	NameStyle	Title	FirstName	MiddleName	LastName
1	2	0	Mr.	Keith	NULL	Harris
2	29816	0	Mr.	Keith	NULL	Harris

```
SELECT TOP (1000) [CustomerID]
      ,[NameStyle]
      ,[Title]
      ,[FirstName]
      ,[MiddleName]
      ,[LastName]
  FROM [SalesLT].[Customer] WHERE [FirstName]='Keith'
```

You can create an index, this helps when searching for data in the table

When you create an index, you specify the column for which you are creating the index

The index will contain a copy of the column data in sorted order. The index will then point to the corresponding row in the table

The database system can then use the index when the same column is referenced in the WHERE clause

Index on CourseName

Extra column - Index

CourseName
AZ-104
AZ-900
DP-900

Results | Messages

	CourseID	CourseName	Price
1	D1	AZ-900	99.99
2	D2	DP-900	100.99
3	D3	AZ-104	89.99

Why not then create indexes for each and every column in the table?

Having indexes can also become an expensive operation

It will take up extra space

When you insert, update or delete data in the table - The index also needs to be changed. This can then slow down the insert,update and delete operations

If your table has a lot of inserts, updates and deletes. Then choose your index carefully.

Types of Indexes

Nonclustered Index - Here a seperate structure is created

Clustered Index

Here the data rows in the table are stored based on key values

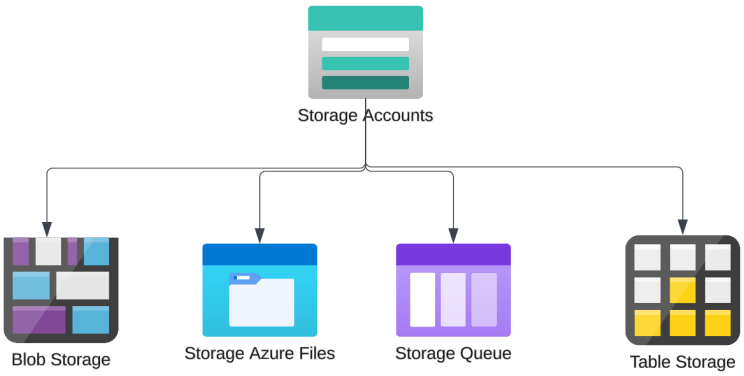
The sorting is done based on the columns defined in the index definition

You can only have one clustered index per table

Describe how to work with non-relational data on Azure

Introduction to Azure Storage Accounts

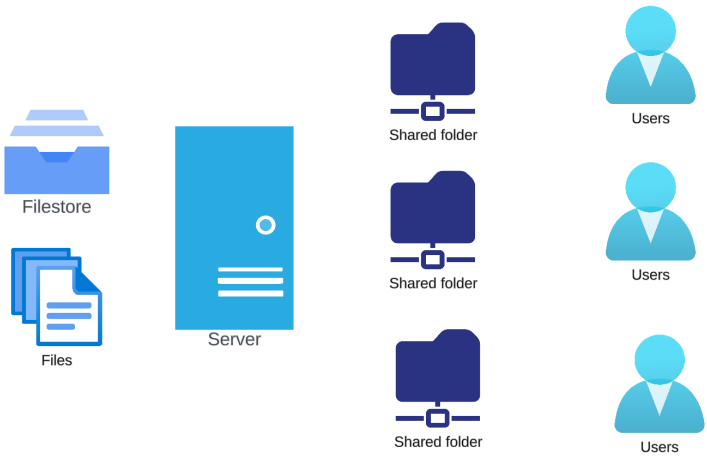
Azure Storage Accounts - This is storage on the Azure cloud for your blob objects, files, queues and tables.



Azure Storage Accounts provides 4 services.

This is used for storing a large amount of unstructured data. Suitable for storing images, documents, video and audio files.

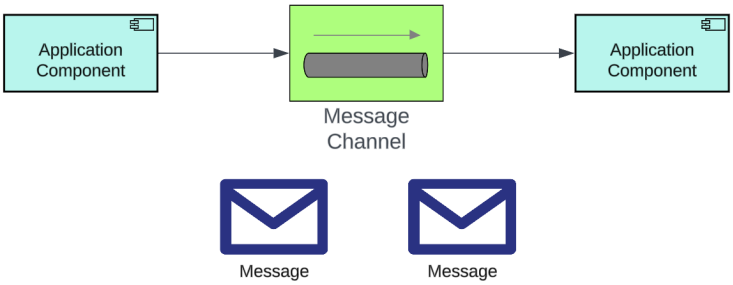
The video and audio files could be stored in an Azure storage account.



Here you need to maintain the file server and ensure enough storage is in place.

Instead you can create file shares using the Azure File share service. Here the storage is managed for you.

If messages need to be shared across multiple application components. Here you need to have the message software and maintain it.



Instead we can make use of the Queue service which provides the basic messaging service.

If an application needs to store data (non-relational structured data), like let's say data about users.

Storage Accounts - Data Redundancy

How does Azure maintain high availability of your data stored in Azure Storage Accounts.

Service Credit – hot blobs in LRS, ZRS, GRS and RA-GRS (write requests) Accounts and blobs in LRS Block Blob Storage Accounts:

Uptime Percentage	Service Credit
< 99.9%	10%
< 99%	25%

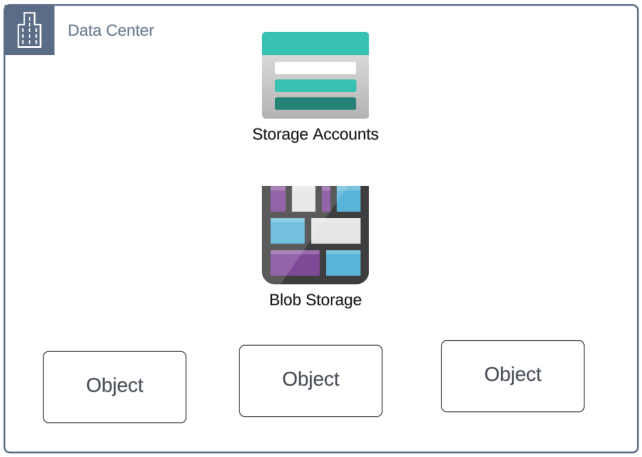
Service Credit – hot blobs in RA-GRS (read requests) Accounts:

Uptime Percentage	Service Credit
< 99.99%	10%
< 99%	25%

There are different data redundancy options in place.

Locally redundant storage

Here three copies of your data are made within a single data center.

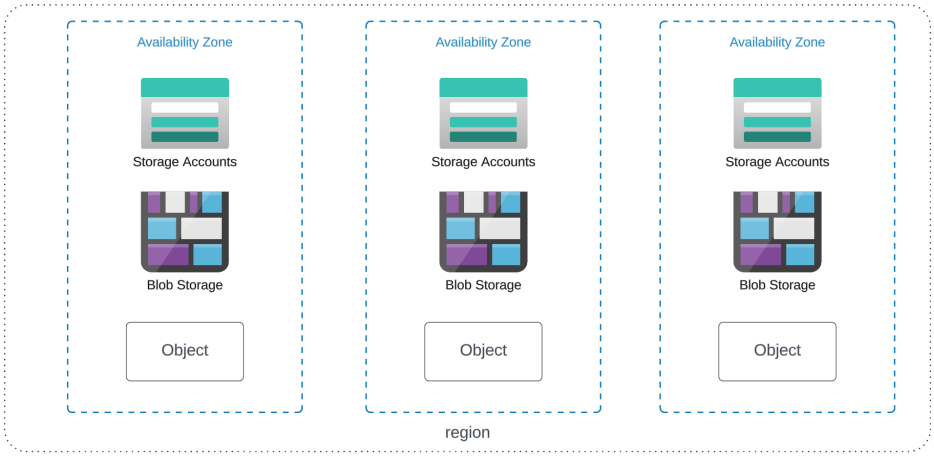


East US

Zone redundant storage

What happens if the data center goes down. Then you don't have access to your objects.

Here the data is replicated synchronously across three Azure Availability zones in the primary region.

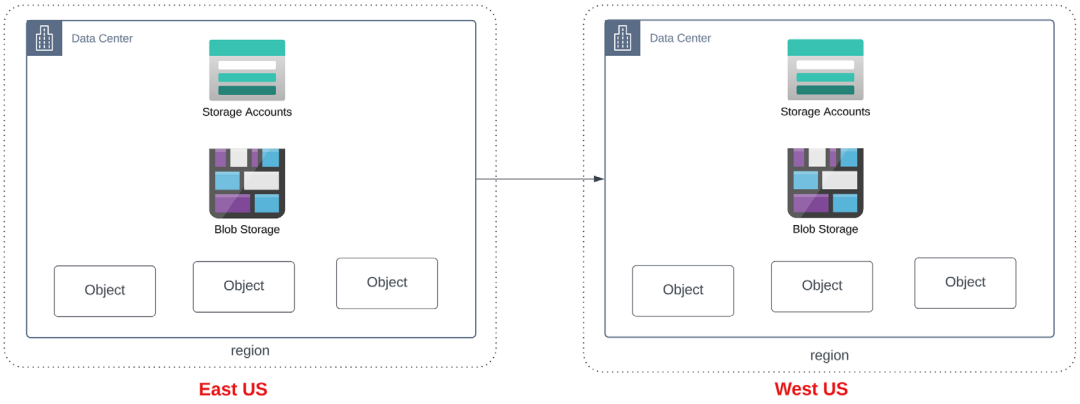


But what happens if the entire region goes down. All of the Availability zones are not available.

Geo-redudant storage

Three copies of your data are made to a single physical location in the primary region using LRS

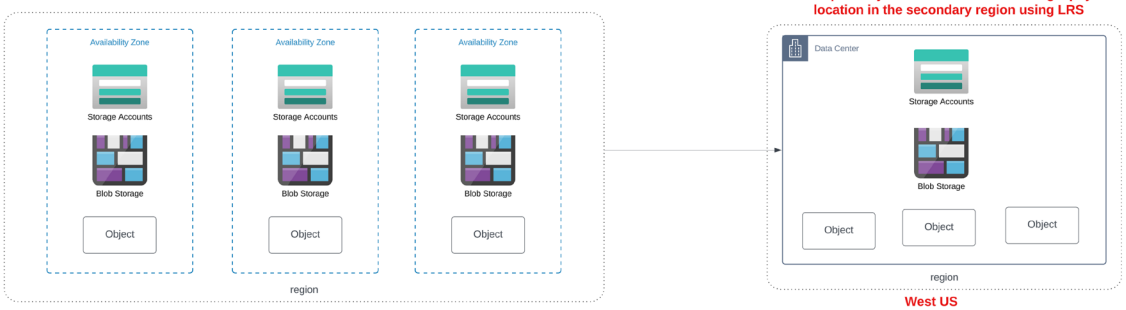
Three copies of your data are made to a single physical location in the secondary region using LRS



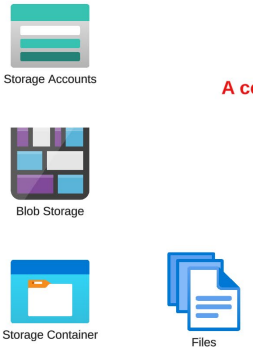
Your data is replicated to a secondary region.

Geo-zone-redudant storage

Here the data is replicated synchronously across three Azure Availability zones in the primary region.



Azure Storage Accounts - Access tiers

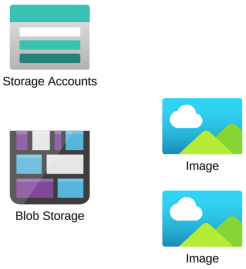


A company can look towards millions of objects in an Azure Storage Account.

Data storage prices pay-as-you-go	Premium	Hot	Cool	Cold	Archive
First 50 terabyte (TB) / month	$0.15 per GB	$0.018 per GB	$0.01 per GB	$0.0036 per GB	$0.00099 per GB
Next 450 TB / month	$0.15 per GB	$0.0173 per GB	$0.01 per GB	$0.0036 per GB	$0.00099 per GB
Over 500 TB / month	$0.15 per GB	$0.0166 per GB	$0.01 per GB	$0.0036 per GB	$0.00099 per GB

A company would want to monitor their storage costs.

An this can especially be the case if objects are not being used.



A thousand images have been uploaded on a particular day. During the first week the images are being used regularly.

But after a week the images are not being accessed. Should be still pay the same when it comes to storage costs.

We can use Access tiers to help in this regard.

Hot	This is the default tier for objects. Here this is optimized for objects that are accessed frequently.
Cool	This is ideal for objects that are infrequently accessed. An object can be set to the Cool Access tier. Here the object needs to be stored for a minimum of 30 days. Here the storage costs are lower when compared with the Hot access tier, but the access costs are higher.
Cold	This is ideal for objects that are rarely accessed or modified, but you still need access to them. An object can be set to the Cool Access tier. Here the object needs to be stored for a minimum of 90 days. Here the storage costs are lower when compared with the Cool access tier, but the access costs are higher.
Archive	This is ideal for objects that are rarely accessed. And if you need to access them, you don't mind waiting for the data to be restored first. Here the data needs to be stored for a minimum of 180 days.

What is Azure Cosmos DB

SQL Databases

ID	Name	Description
C01	AZ-104 Azure Adminsitrator	Azure Administration
C02	AZ-204 Azure Developer	Azure Development

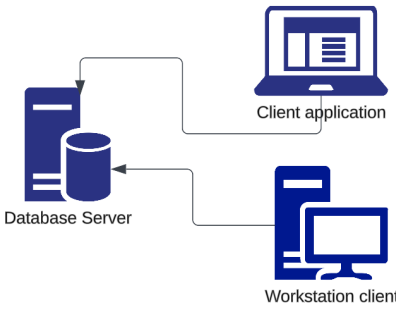
Course

ID	Name	CourseID
S01	Mark	C01
S02	James	C02

Student

What is NoSQL

Familiar with SQL-based database engines - Oracle , MySQL, Microsoft SQL Server



Applications, users connect to the database hosted on the database server.

There were issues when it came to traditional SQL database engines.

Ability to store large amounts of data - These were meant to be transactional systems and not able to manage large amounts of data.

Tables in the databases needed to have a predefined schema. But in today's world , data comes in all sizes, shapes and form.

Needed to have a more flexible way of storing data.

There were many NoSQL-based data stores developed for this purpose - MongoDB , Cassandra.

Azure Cosmos DB is a fully managed NoSQL, relational and vector database.

You get fast access to your data.

Different API's

NoSQL

Data is stored in document format.

You can query for items using Structured Query Language (SQL)

MongoDB

Here documents are stored in BSON

PostgreSQL

Managed open source relational database with better performance.

Apache Cassandra

Here data is stored in a column-oriented schema.

Gremlin

This allows you to store graph-based databases.

Table

Store data in the form of key/value pairs.

Important concepts in Azure Cosmos DB

Azure Cosmos DB

NoSQL

Data is stored in document format.

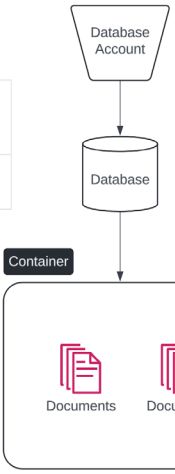
You can query for items using Structured Query Language (SQL)

https://learn.microsoft.com/en-us/azure/cosmos-db/resource-mode

Azure Cosmos DB entity	API for NoSQL	API for Apache Cassandra	API for MongoDB	API for Apache Gremlin	API for Table
Azure Cosmos DB database	Database	Keyspace	Database	Database	Not applicable

https://learn.microsoft.com/en-us/azure/cosmos-db/resource-mode

Azure Cosmos DB entity	API for NoSQL	API for Cassandra	API for MongoDB	API for Gremlin	API for Table
Azure Cosmos DB container	Container	Table	Collection	Graph	Table

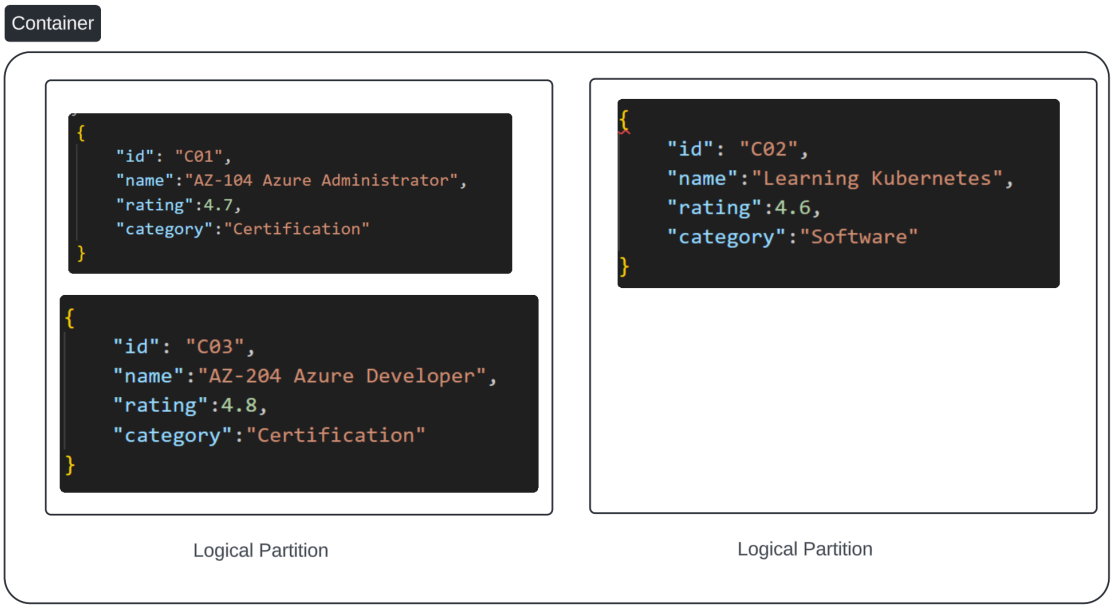


A database is a group of containers.

The data is stored in a container.

The items in a container are split into different logical partitions.

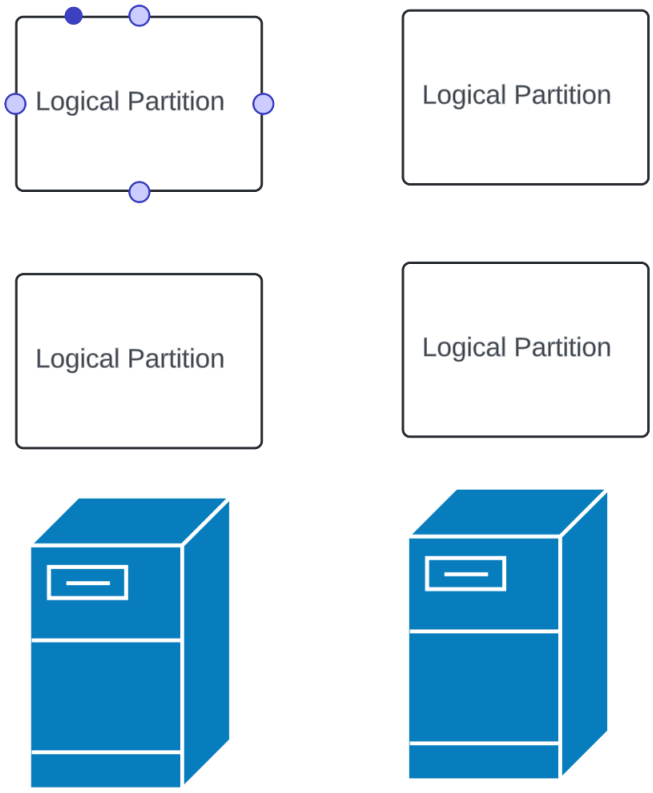
The logical partition for the item depends on the partition key set for the container.



Here the category has been set as the partition key for the container.

Each item also has an id that is unique within a logical partition. The partition key along with the item ID helps to uniquely identify the item in the container.

You can have multiple logical partitions in a container. Each partition can store up to 20 GB of data. It's good to choose a partition key that helps to distribute data across many logical partitions.



Your logical partitions are split across multiple physical partitions.

The number of physical partitions depends on the throughput set for the container.

The cost of all database operations is deemed via the use of Request units.

SQL Database

CPU

Memory

IOPS

In Azure Cosmos DB, all of this is taken as a blended measure known as Request Units.

A read operation of a 1-KB item takes one request unit.

You can set the throughput at the database or container level.

Describe an analytics workload on Azure

Recap on the common data workloads

Files

Blob Storage

Media File

We have seen Blob storage in Azure Storage Accounts for storage of objects.

We are going to look at a variation of this service - Azure Data Lake Gen2.

SQL Database

Table

Then we have relational data that can exists in relational database systems - Azure SQL database.

Azure Cosmos DB

NoSQL and relational data store - Here we have different API's - NoSQL, Gremlin, Table, Cassandra, PostgreSQL, MongoDB.

Data Processing

How do you want to process your data.

Your data could be coming in from multiple data streams.

The data could be in different formats.

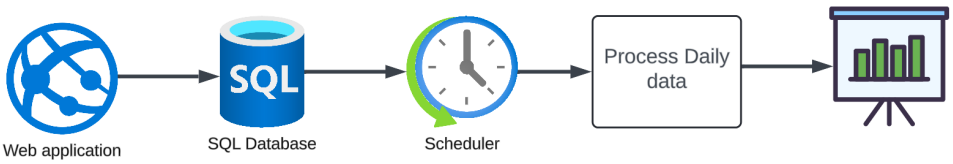
The size of data needs to be considered.

The velocity of data needs to be considered.

There are different ways to process your data

Batch Processing

**Here multiple data records are collected at a time.
They are stored first. Then the records are processed as batches at regular time intervals.**



Sales transactions for an e-commerce site are being recorded on a daily basis to a database.

Job

Daily transaction data needs to be processed so that reports can be generated.

The processing can happen at the end of the day. This is when existing compute infrastructure can be used to process the data.

You have to ensure that your source data is not filled with errors. It could impact the batch process that runs on a daily basis.

If the nightly batch process takes time and if the data causes the batch process to fail. Then you will need to run the process again.

In a batch process , the latency is high. It takes time to see the final results.

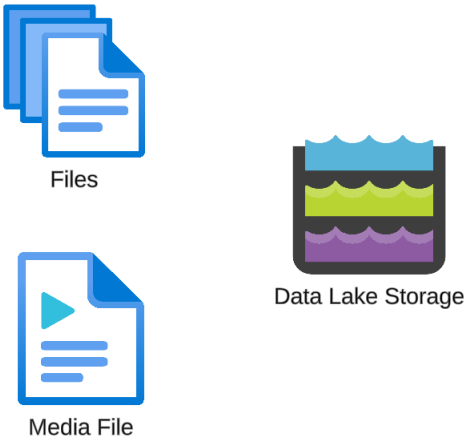
Stream Processing

Here the data is processed in real time. For example for an incoming stream of data , you process the data every 5 minutes.

Here the latency is less because you get results faster.

But you need to ensure that you have a processing system that is capable of ingesting and processing data at that speed.

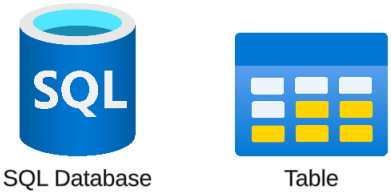
Extract, transform and load



Your data lake would consist of files - semi-structured or unstructured.

Initially all of the data would be in raw format. Especially if you have log files.

In the log files there would probably be a lot of data that you don't need.



Then you probably also have data in your relational databases.

You might want to extract data from here that could be used for analysis purposes.

Files

Media File

Data Lake Storage

SQL Database

Table

Step 1 : Extract the data that is required

The next step is to transform your data.

Step 2 : Transform the data.

Take only the data values that are required.

Clean data - Remove any NULL or incorrect values.

Convert the data into a desired target format.

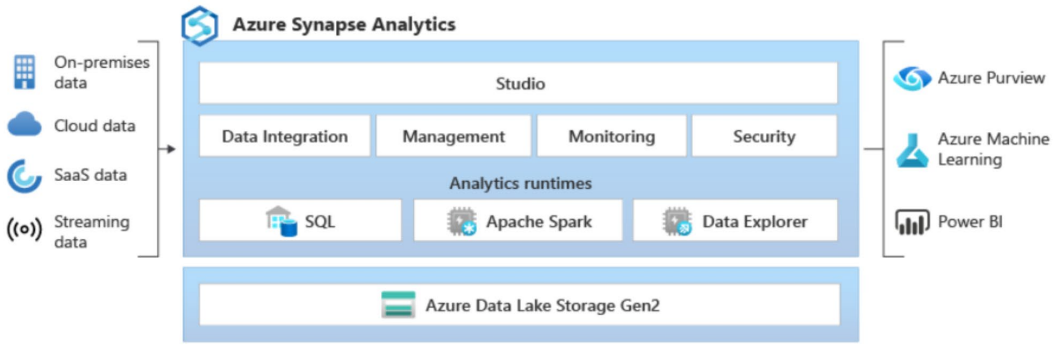
Step 3 : Load the data.

Data Lake Storage

Azure SQL DataWarehouse

Azure Synapse Analytics

Enterprise Analytical service



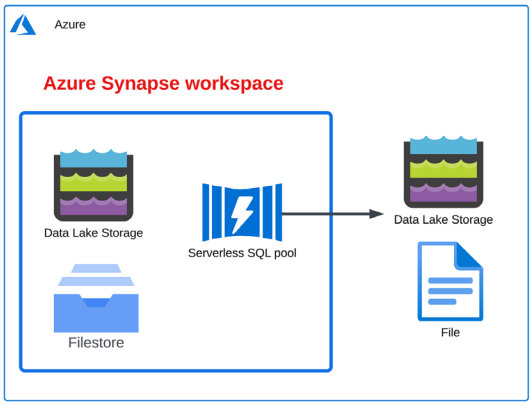
Reference - https://learn.microsoft.com/en-us/azure/synapse-analytics/overview-what-is

Synapse SQL - Here you can host your SQL data warehouse.

Apache Spark for Azure Synapse - You get access to Spark that assists you in the entire data engineering process.

Data Integration - You can Azure Data Factory like features to ingest your data.

Azure Synapse - Serverless Pool Compute option



As part of your Azure account, you could be having data that streaming in onto another Azure data lake gen 2 storage account.

Now let's say you have a CSV-based file and you want to analyze the data in the file.

You can use the built-in Serverless SQL pool to query the data that's in the Azure Data Lake storage account.

Your files - Delimited text, Parquet, Delta Lake.

You can use SQL-like queries to work with the data.

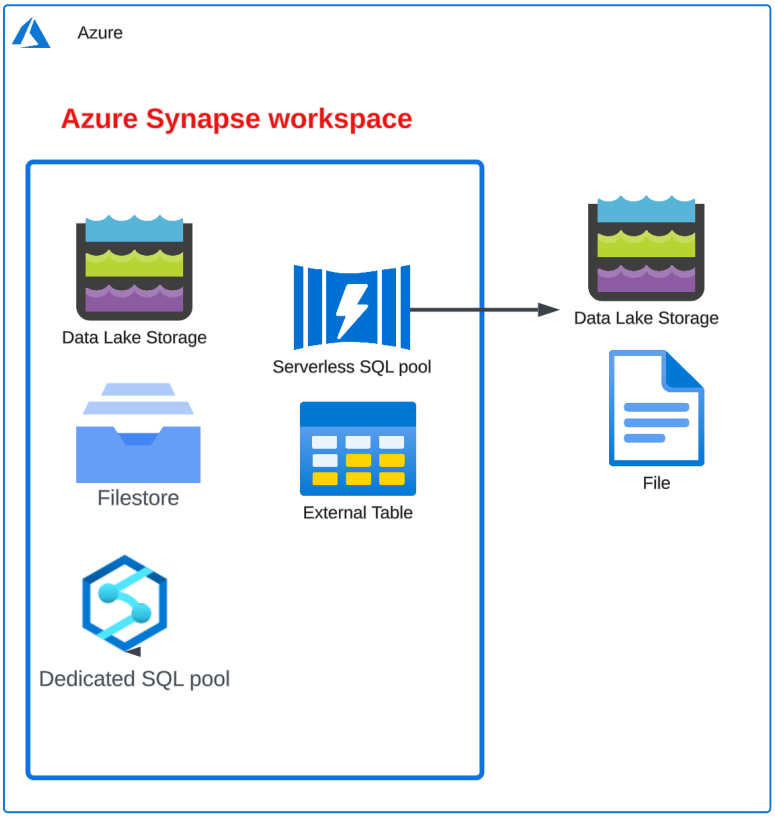
Here the serverless SQL pool has underlying compute that is managed for you.

The compute will manage the queries for you.

You are charged for the data that is processed by the queries.

Azure Synapse - Dedicated SQL pool

Dedicated SQL pool



You can host a SQL data warehouse with the help of the dedicated SQL pool.

With the Serverless SQL pool, you can just define the table schema. The data itself resides in external storage.

But if you need to persist the data in actual tables and query them via SQL , we need to have a SQL data warehouse in place.

The data warehouse gets dedicated compute and storage. The data in the tables are stored in columar format which reduces data storage costs and improves the query performance.

Designing a data warehouse

Just looking back as the usage of a SQL database as a backend for an application.

Now normally the application would add data to a table via the use of the INSERT SQL statement.

Here a row of data would be added to a table in the database.

But in a data warehouse things work a little differently.

Remember here data is used for analytical purposes.

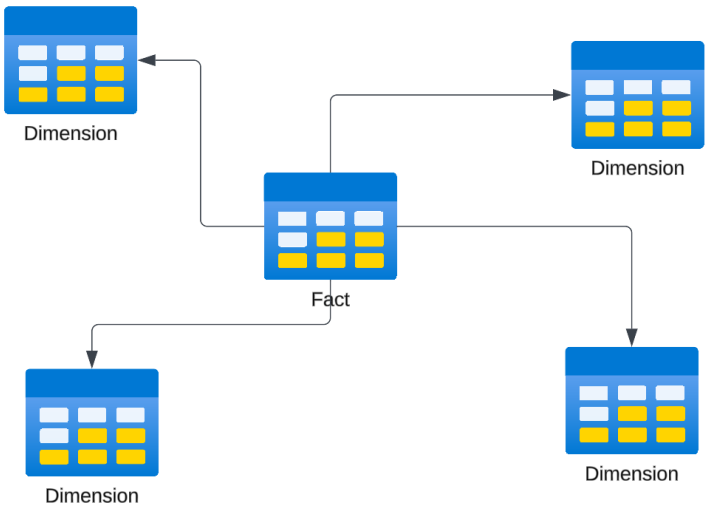
Normally you don't insert rows of data one by one.

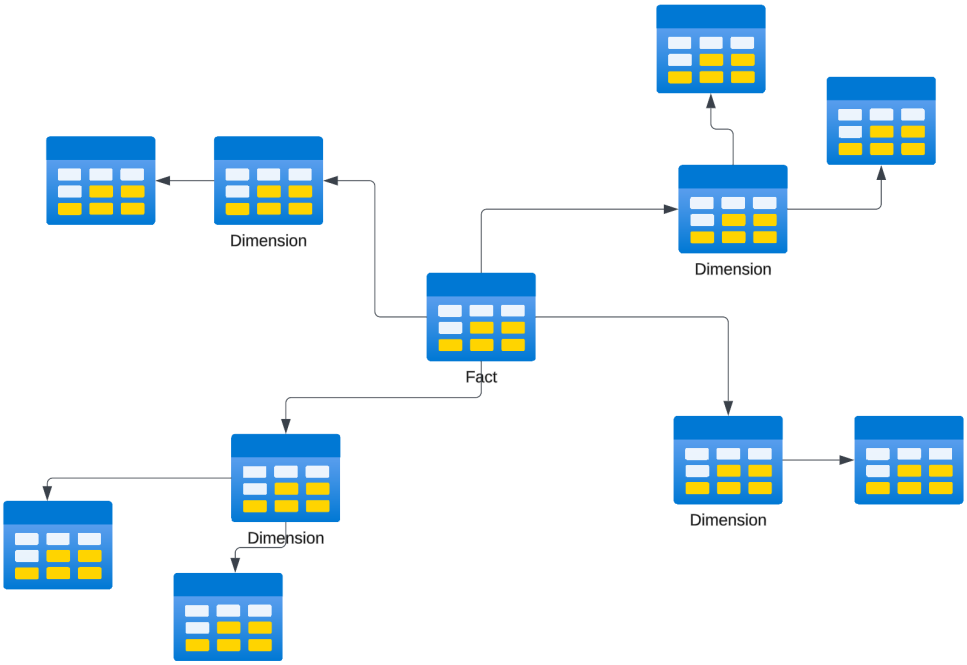
Instead you add and delete rows of data in bulk. This is because of the huge amount of data that is stored in the data warehouse.

Next is the design of the tables

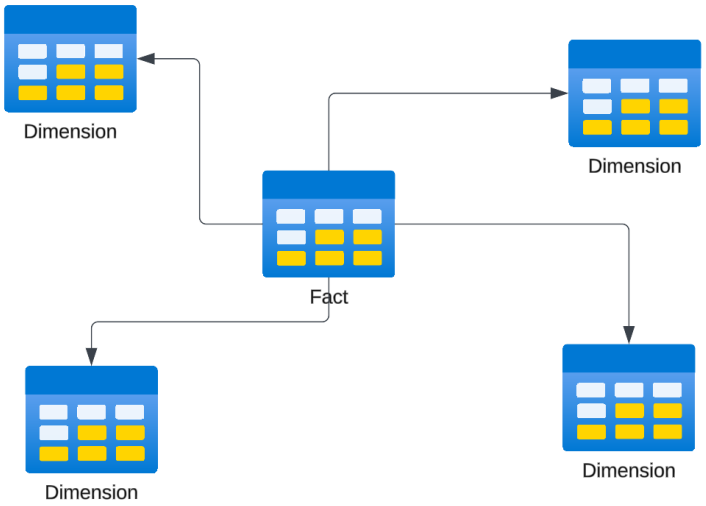
The tables in a data warehouse are split into Fact and Dimension tables.

These tables either conform onto a Star or Snowflake schema.

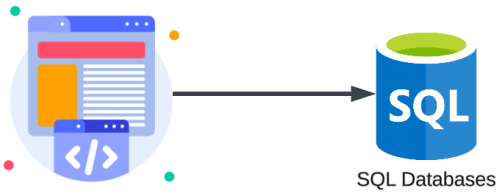




Fact and Dimension tables



A Fact table is meant to store quantitative data, that is data that can be measured.



So let's say that users are making purchases via an ecommerce platform.

The sales data for various products are being recorded in the OLTP SQL database.

Now the sales being recorded are quantitative in nature. You can take the data over time and store in a Fact-based table in the data warehouse.

Dimension tables are used to used to present some context to the facts.

For example, based on the sales being made, you want to analyze what are the best selling products - Hence the product-related information would become a dimension.

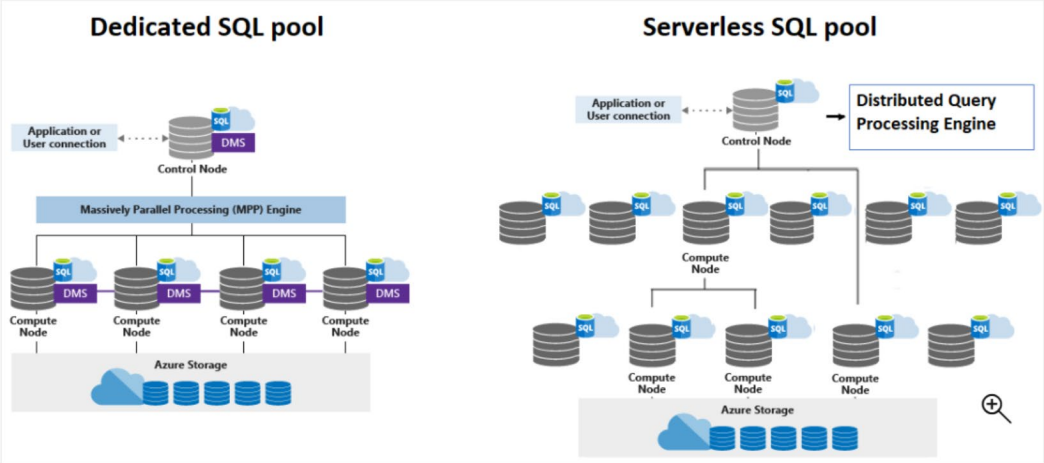
This is because the product information is giving you a view or insight into the sales data.

Or you want to look at the top regions where sales are being made based on the customer's location. So the customer's information can be another dimension to give some more context to the sales.

In this way , we can construct our Fact and Dimension tables.

Understanding Azure Synapse Architecture

Synapse SQL architecture



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

In Synapse SQL , the compute and storage are seperate so that each can be scaled seperately.

In the dedicated SQL pool , the compute power allocated to the pool is determined by a unit known as the data warehouse unit.

All queries are targeted towards the Control Node. And then the Control Node distributes the query for parallel processing across the compute nodes.

Performance level	Compute nodes	Distributions per Compute node	Memory per data warehouse (GB)
DW100c	1	60	60
DW200c	1	60	120
DW300c	1	60	180
DW400c	1	60	240
DW500c	1	60	300
DW1000c	2	30	600
DW1500c	3	20	900
DW2000c	4	15	1200
DW2500c	5	12	1500
DW3000c	6	10	1800
DW5000c	10	6	3000
DW6000c	12	5	3600
DW7500c	15	4	4500
DW10000c	20	3	6000
DW15000c	30	2	9000
DW30000c	60	1	18000

Reference - https://learn.microsoft.com/en-us/azure/synapse-analytics/sql-data-warehouse/memory-concurrency-limits

So if you have 60 compute nodes , then you query will be split into 60 small queries and run in parallel.

All user data in Synapse SQL is stored in Azure Storage.

Azure Data Factory

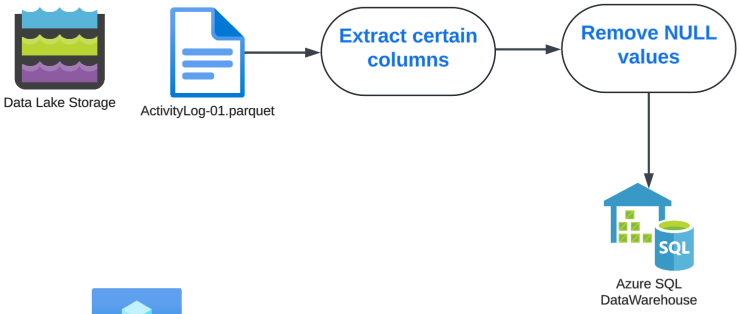
Azure Data Factory

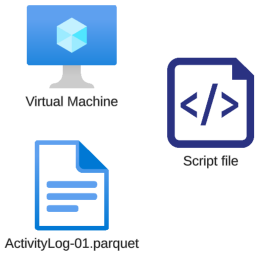
This is a cloud-based ETL and data integration service.

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

You can also transform data at scale.

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

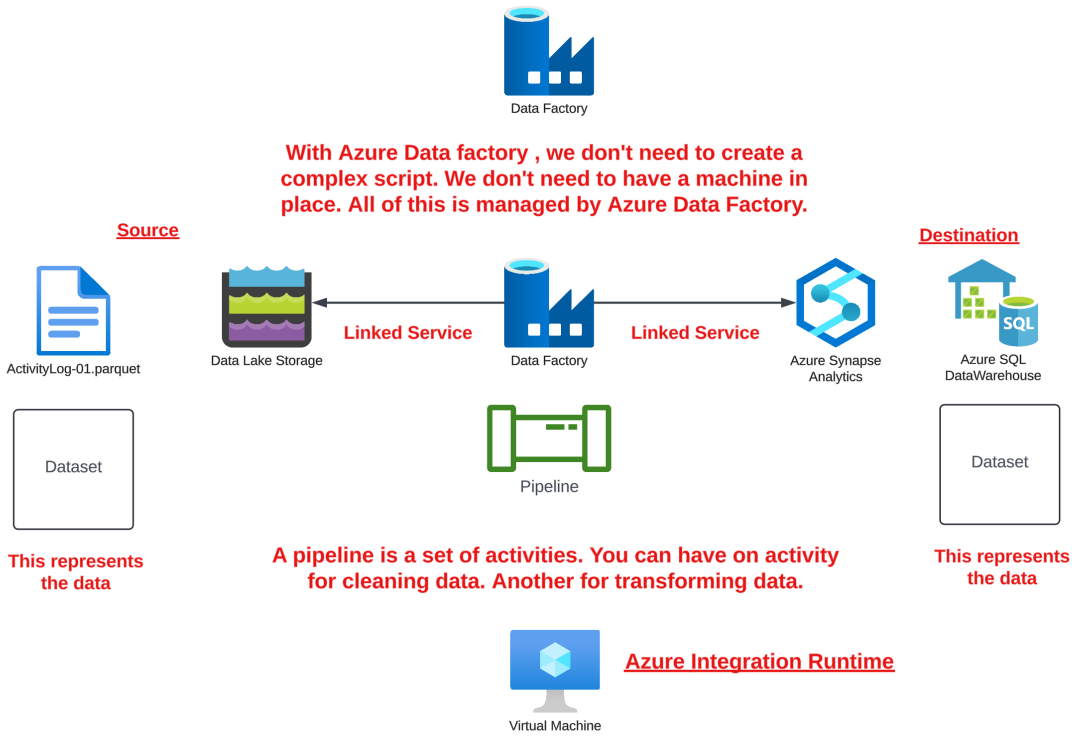




We would need to have a script file that would run on a machine.

The script file would first load the entire data set on the machine.

The script would then perform the required transformations and then load the data onto the target data warehouse.

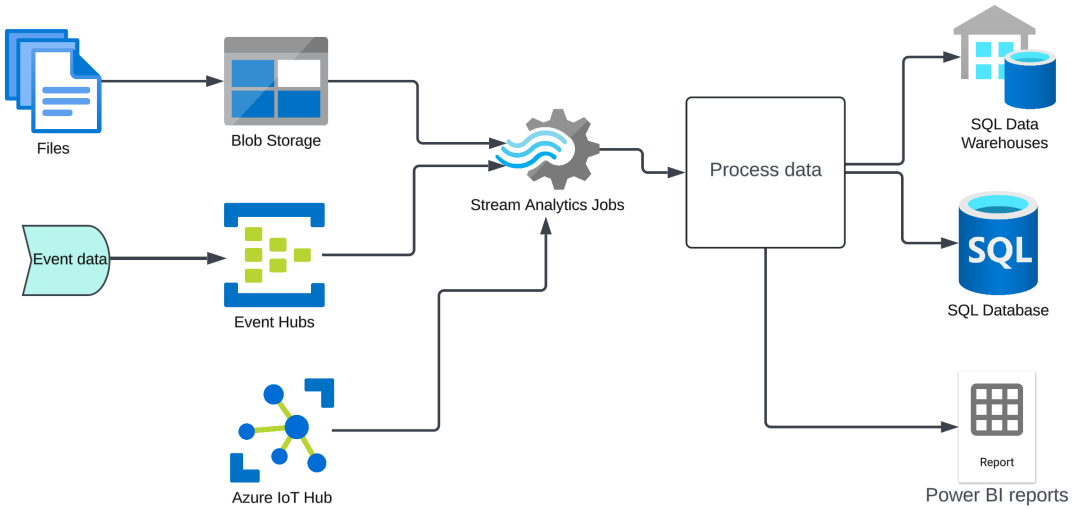


Azure Stream Analytics

Azure Stream Analytics

This is a fully managed stream processing engine.

You can use this service to process and analyze large amounts of data in real time.



Tasks when working with Azure Stream Analytics

1. **Create an Azure Stream Analytics job.**
2. **Define inputs that can be used to take in data.**
3. **Define a query that can be used to process data.**
4. **Define outputs to place the processed data in a target store.**

What is Azure Databricks

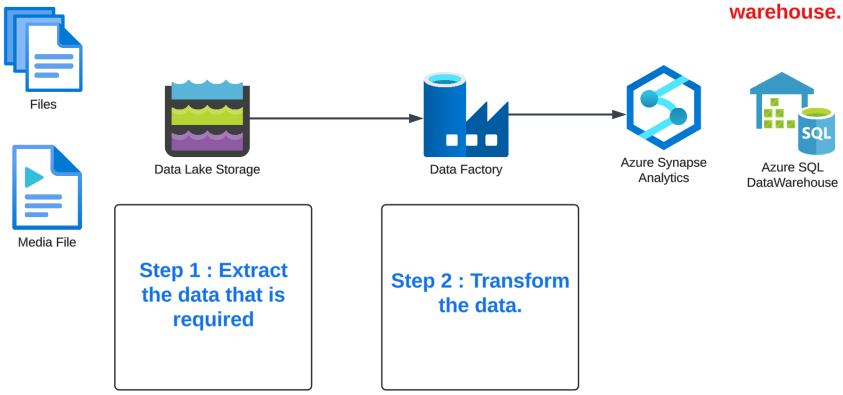
A lot of the data comes in the format of semi-structured files.

Data comes in at a rapid rate. There are a lot of delta changes to consider.

Maintaining data governance can be quite challenging.

Your data needs to be in a particular format because you can start analyzing it.

Can become expensive to maintain a data warehouse.

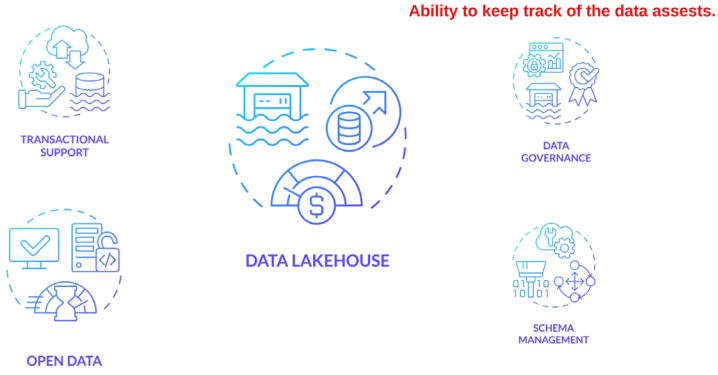


Atomicity - Here transactions are isolated in a single unit.

Consistency - Changes are made in a predictable manner.

Isolation - Transactions don't interfer with each other.

Durability - Transactions are present even if the entire system fails.



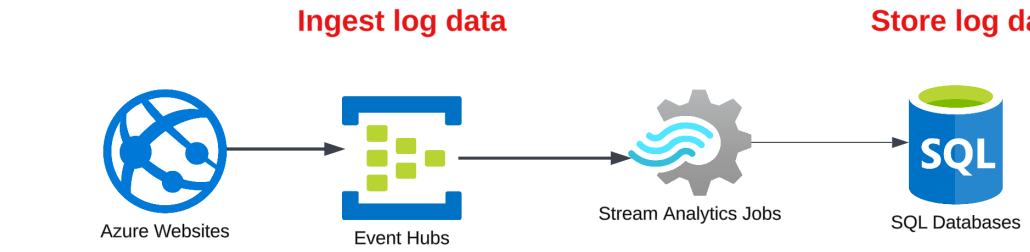
Ability to keep track of the data assests.

Your data can be in different formats.

You want to be able to structure your data.

Mini-Project

What are we going to implement

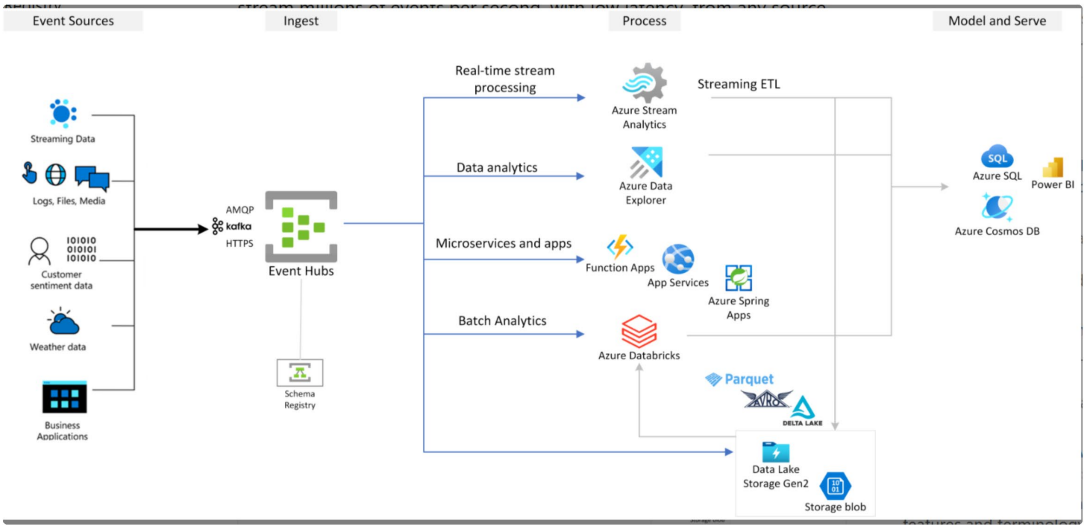


Azure Event Hubs

Azure Event Hubs

This is a data streaming service that can stream millions of events per second.

This can be from any source or destination.



Reference -
https://learn.microsoft.com/en-us/azure/event-hubs/event-hubs-about