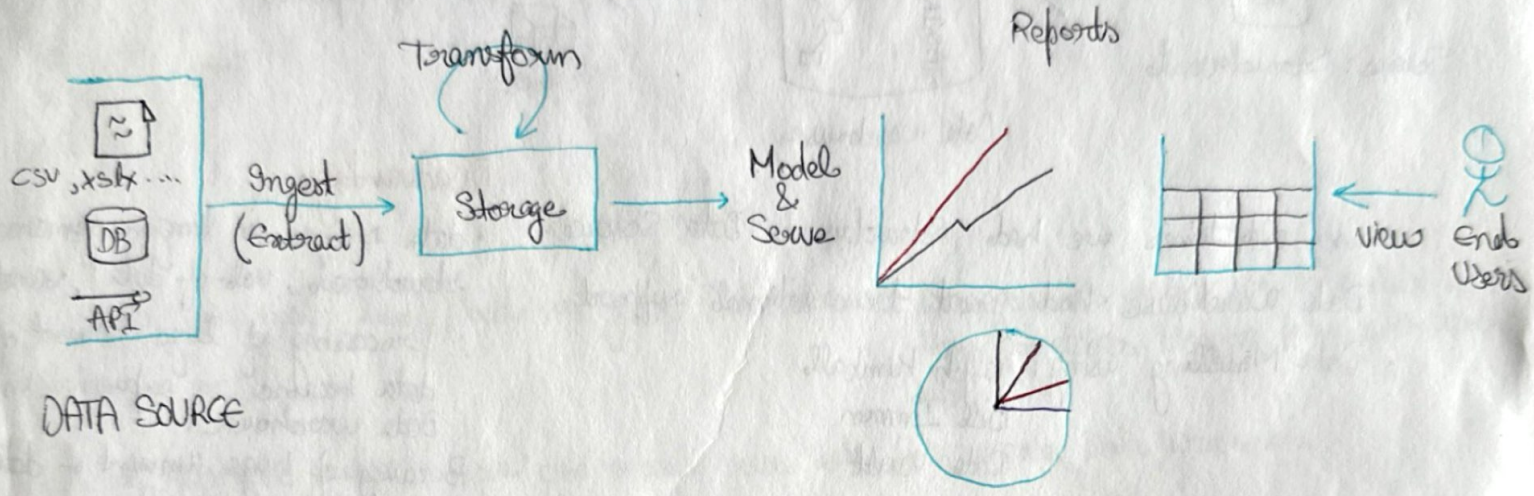
Basic BI Flow

Batch : scheduled pulling of data.

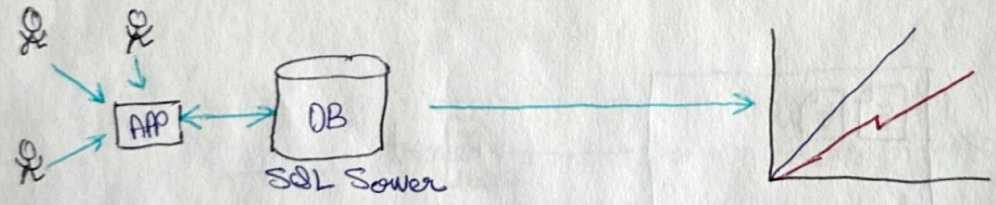
Streaming : constant pulling of data.



DATA SOURCE

|← Data Engineer →|← Data Analyst →|

why not directly connect DB to generate reports? because of mismatch & security reasons.



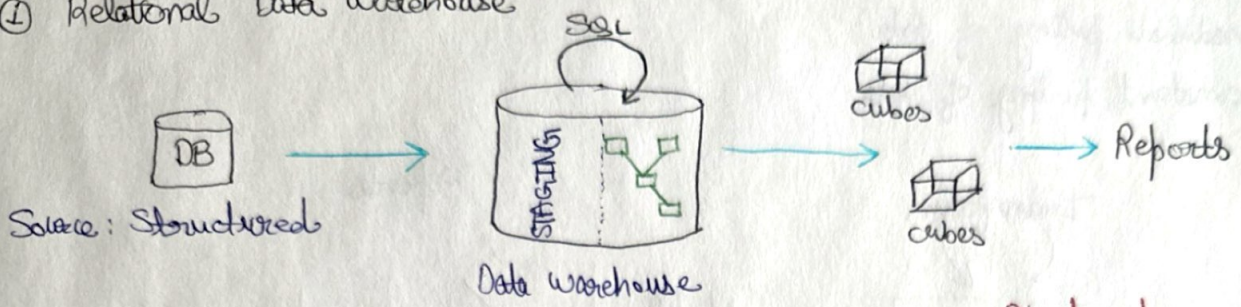
- this side is transactional
- OLTP
- Many requests from many users.
  ∴ should have concurrency.
- data is Normalized (3NF) ⇒ no. of tables ↑
- no. of tables ↑ ⇒ join operations ↑
  ⇒ querying data is expensive.

- this side is analytical
- OLAP
- De-normalized ⇒ no. of tables ↓
- here we are interested for all rows.

This side we de-normalized data using Star schema, Snowflake schema, etc.

Scanned with CamScanner

① Relational Data Warehouse

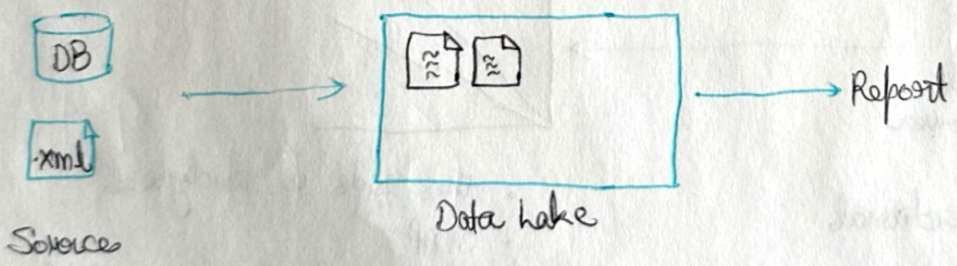


- in old times we had Structured Data Source.
- Data warehouse had great transactional support.
- Data Modelling using: Ralph Kimball
  Bill Inmon
  Data Vault
- Schema-on-write : define schema beforehand

Disadvantage

- data source no longer remained structured, vol. of data ↑, variety
- ∴ processing of large amount of data became so expensive in Data warehouse.
- Because of huge amount of data, DWH became too slow.
- Schema-on-write ⇒ Agility ↓

② Data Lake



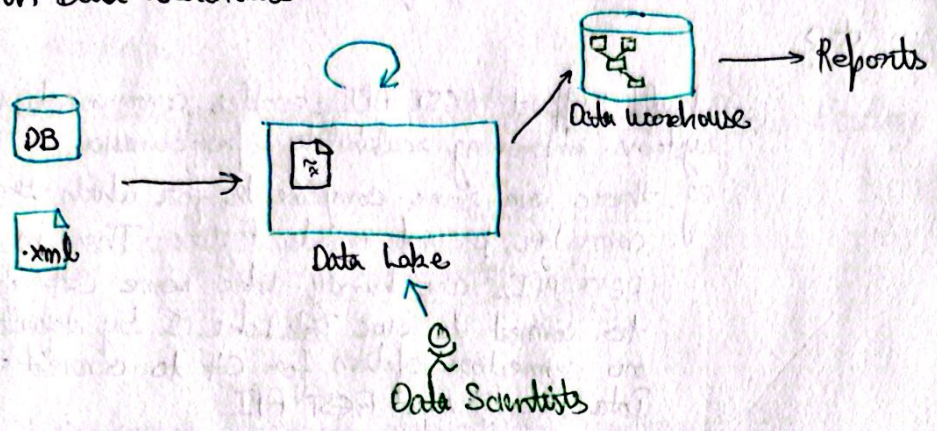
- Schema-on-read, ∴ could support different sources.
- Hadoop compatibility: processes large amount of data ⇒ more compute nodes for processing

Disadvantage:

- Working with Hadoop tools is not so easy
- Data querying is complex in Hadoop environment.
- Lack of concurrency as compared to traditional Data Warehouses.

Scanned with CamScanner

③ Modern Data Warehouse

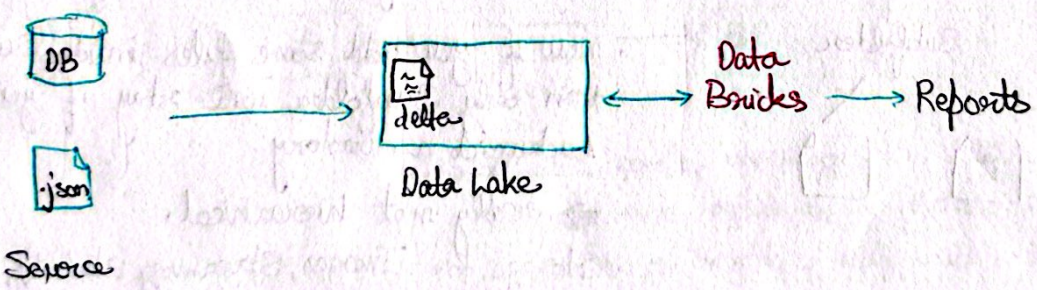


- you can just dump/load in your Data Lake ⇒ Data Scientists might use this data as they sometimes like to play with raw data.
- to generate reports you can model your data in separate Data Warehouse

Disadvantage

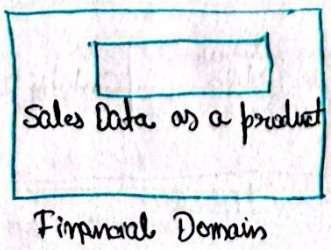
- Data duplication inside Data Lake & Data Warehouse

④ Data Lakehouse



- delta format supports ACID properties.

⑤ Data Mesh (new concept)



- data is divided as per separate domain.
- Data is treated as a product
- each domain needs to have enough training to handle data.

Scanned with CamScanner

Lecture 3

How should our Data Lake look like, what features or characteristics you would like to have in it?

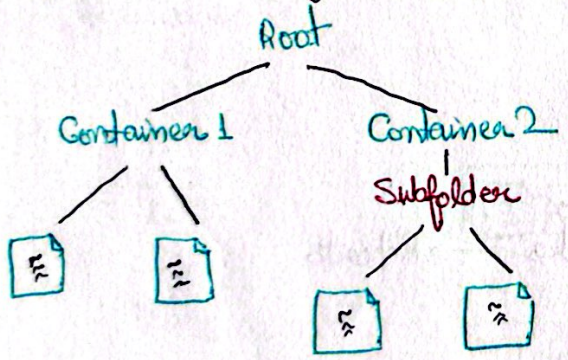
- store files
- should be scalable
- should be cheap
- high performance
- reliable
- hierarchical
- secure
- hadoop compatible

- should support REST API : enables communication b/w different systems, improving scalability & performance
  eg: there are some components for which there is no connector present inside Azure. Then in this case REST APIs are handy. like some C# app. wants to connect to our Data Lake & by default there is no connector option for C# to connect to our Data Lake ⇒ use REST API

what options do we have inside Storage Accounts?

① Blob Storage

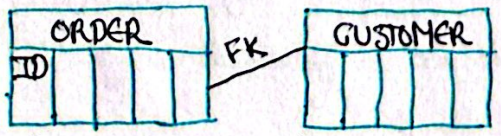
- problem is if you try to implement following:



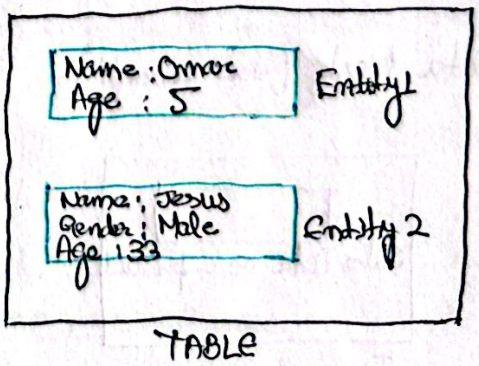
- if you create Subfolder inside Container 2 & don't add any files & if you go back to Container 2 you will see Subfolder is vanish.
- now if you add some files inside Subfolder then only Subfolder will stay if you hop backward a directory.
- ⇒ really not hierarchical.
- Storage for: Images, Streaming vids (old), Log Files

② Table (Cosmos DB)

- ideal for No SQL
- schemaless
- has foreign keys to refer different tables



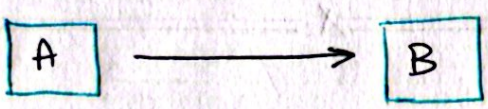
- key/value pair in form of Entities



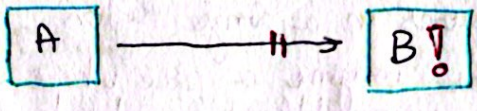
Scanned with CamScanner

③ Queue

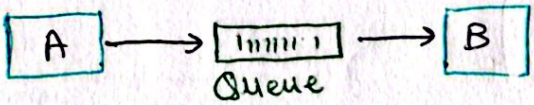
- store messages
- decouple : when you want to decouple components for application



- component A sends request to component B
- component A expects some response from component B
- Suppose A is sending too many requests ⇒ B can result in timeouts.



- in case of some n/w issues the requests from A might not reach B & would eventually lost.
- sol. is to add a Queue.



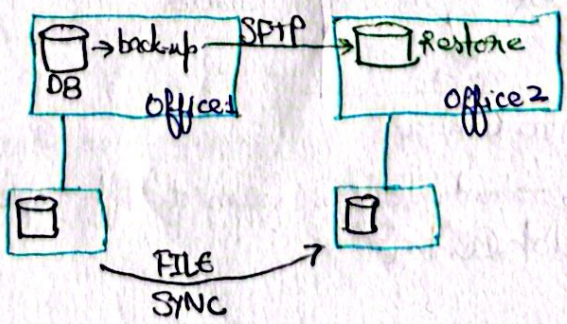
→ A will write messages to Queue
B grabs the messages & process it
B is busy ⇒ message will wait in the Queue

→ Queue doesn't guarantee FIFO.

→ Service-Bus (advance version of Queue)

④ File Share

- shares file over n/w
- replace File Servers
- handy in case of Lift & Shift
- can be mounted.



SFTP - DB backup was huge. Internet conn. not reliable.

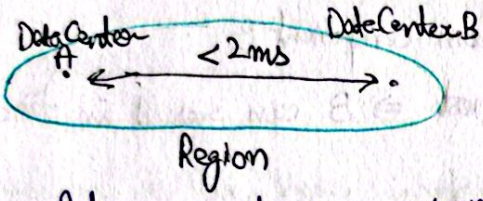
Scanned with CamScanner

⑤ ADLSg2

- just check/enable "Hierarchical Namespace"
- used for Big Data Analytics
- options for granularity ⇒ increase performance
- Hadoop Compatible

Lecture 4

Storage Account Redundancy



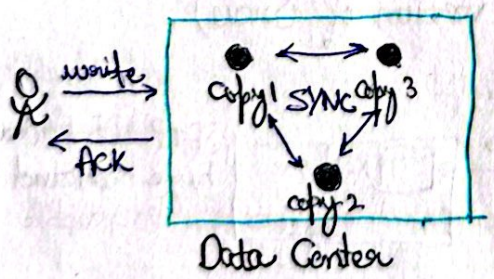
- to lower latency, create your resources in a single Region
- choosing a proper DataCenter matters. imagine a case where your Front-End & Back-End are deployed in Seperate Regions.

	A	B	C	D
A	–	10ms	50ms	100 ms
B	10ms	–		
C	50ms		–	
D	100 ms			–

eg of Latency Table

i. LRS - Locally Redundant Storage

→ within a Data Center 3 copies of data is saved which are sync. together.



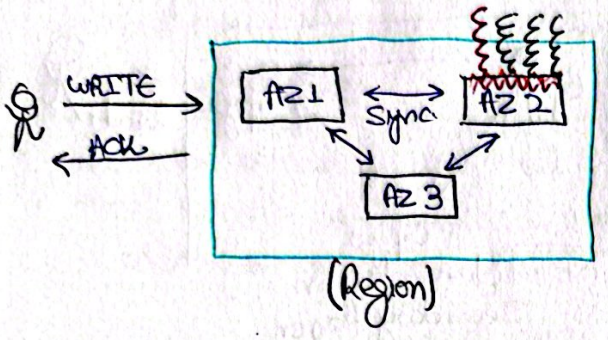
Server/Rack Failures

Same DataCenter
Different Server/Rack

→ data is saved redundantly in form of diff. copies of same file on different servers within same DataCenter.

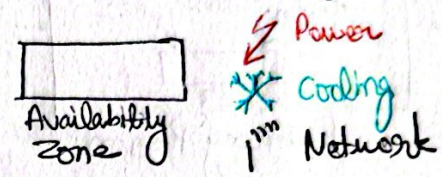
Scanned with CamScanner

2. Zone Redundant Storage
ZRS



→ data is not stored in same datacenter but in different Availability Zones.

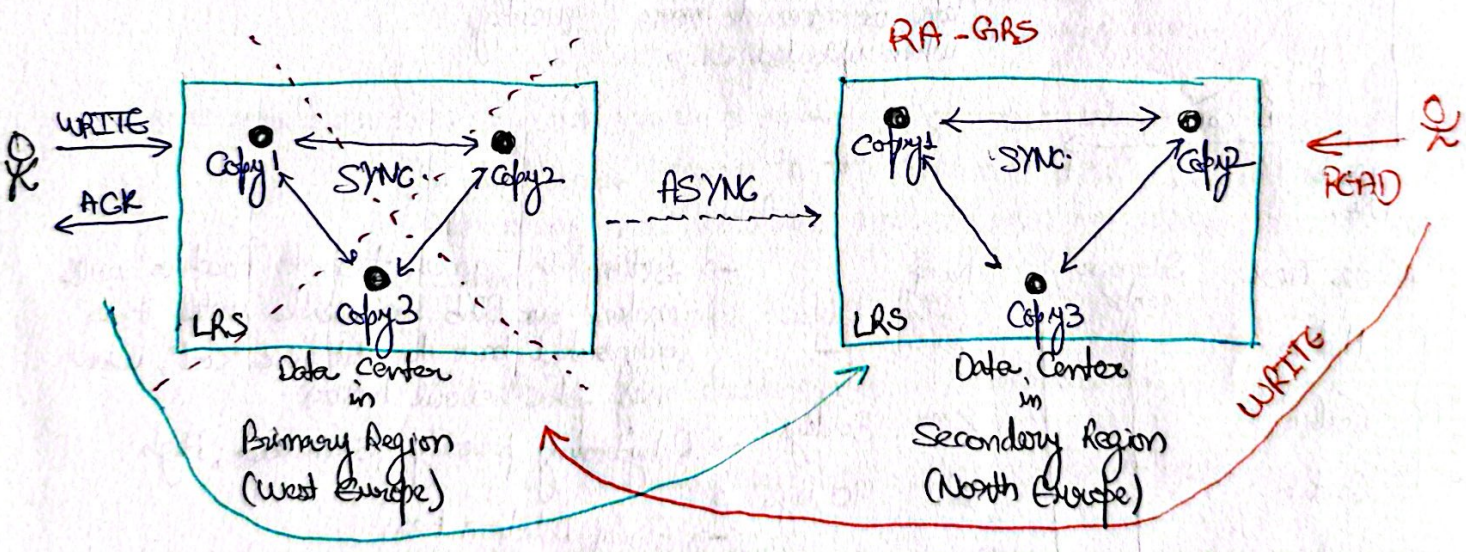
→ diff. AZs in same Region.

→ 

one or more datacenters, physical location within a Region that has separate power, cooling, n/w

→ protects from Data Center Failure.

3. GRS - Geo Redundant Storage



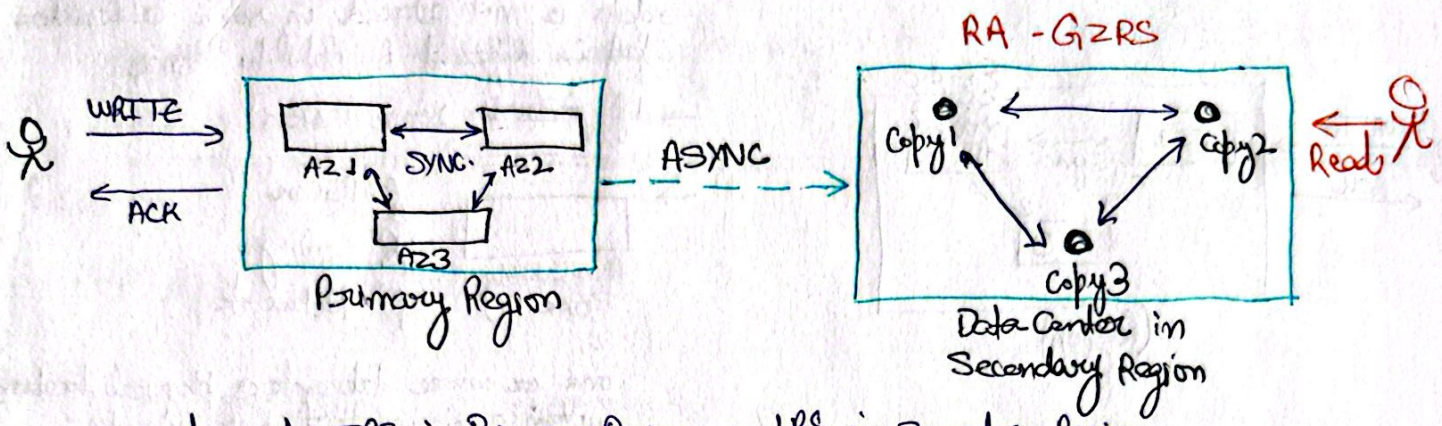
- GRS is combination of LRS in primary region & LRS in Secondary Region.

a) RA GRS = Read Access Geo Redundant Storage

- if users are closer to the location of your Data Center in Secondary Region, you can enable RA GRS. This way users can read data in less time
- write operations will still be happening at Primary Region.

Scanned with CamScanner

b) GZRS - Geo Zone Redundant Storage

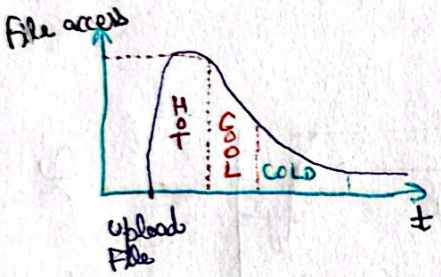


→ combo. of ZRS in Primary Region + LRS in Secondary Region.

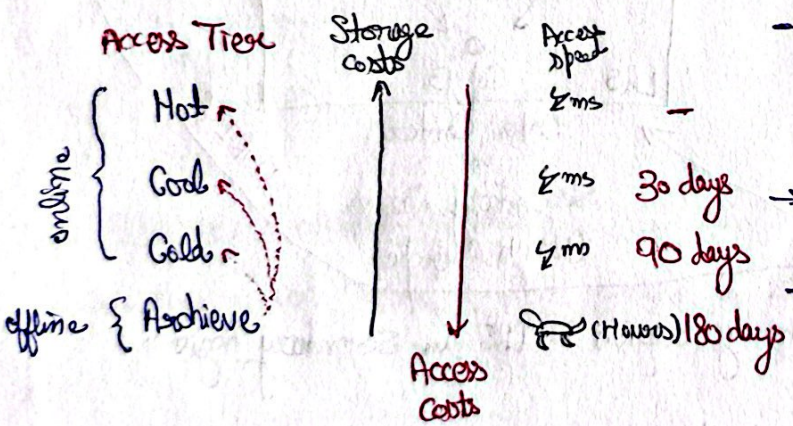
LECTURE : 5

---

* Data Lake Pricing : we pay for : Data Storage & Data Access & operations
  → Read
  → Writes.



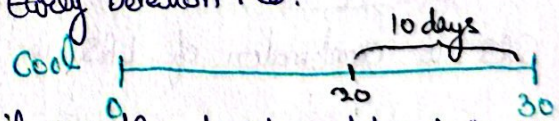
. data is accessed more frequently when uploaded.



→ setting the access tier to "Archive" will make your Blob inaccessible until it is rehydrated back to "Hot" or "Cool", which may take several hours.

→ Rehydrating priority : Standard, High

→ Early Deletion Fee:



if you plan to store data in "cool" tier, by default it should stay there for 30 days. If you move your data to some other tier on lets say 20th day then you'll have to pay 30-20=10 days of fee.

Life Cycle Management Policy :

. Initially, a file after being uploaded is exposed to workload very oftenly & by default we dump it in "Hot" tier. But with time, it becomes less useful & you'd want to change its tier from "Hot" to "Cool" or some other access tier. You do that with Life Cycle Mgmt Policy.

→ Storage Acc. → Lifecycle Mgmt → Add a rule

. moves file automatically b/w access tiers based on your created Rules
. rules are of the form : if - then statement ⇒ "Last modified", "Created"
. check "Enable access tracking" to add "Last accessed"

Scanned with CamScanner

Lecture: 6

Common file types (CSV, XML, JSON)

CSV

→ first row is name of columns
→ structured
→ text files
→ can be opened by any tool
→ no schema enforcement & everything is a string
→ ask for metadata : to avoid ambiguity in date format

which part is month & which part is date { 1950-02-01, 1999-03-10 } which portion is month & which is day

: time zone
: ~~n~~number formate - in case of decimals which delimeter is used to separate nos.
: max length of string - some company uses a separator for thousands
: column delimeter - some uses currency symbol
: text qualifiers

→ text qualifiers : eg if we have a value in a row which is Architect, Analyst & it should belong to the column "Title". But if we are not using text qualifiers Analyst will be assign to the next column.

∴ we use double quotes.

everything b/w double-quotes is treated as a column

\ : escape characters - in case double-quotes in b/w double-quotes
: encoding - UTF8

→ flat files ; no hierarchy

Scanned with CamScanner

XML :→ in form of tags

Extensible Markup Language

```
<root>
    <order Id="1">
        <header>
            <OrderDate>2023-08-12</OrderDate>
            <comments>Super shop!</comments>
        </header>
        <lines>
            <line>
                <productId>1</productId>
                <quantity>3</quantity>
                <unitPrice>10</unitPrice>
            </line>
            <line>
                <productId>2</productId>
                <quantity>3</quantity>
                <unitPrice>100</unitPrice>
            </line>
        </lines>
    </order>
</root>
```

element → order

attribute → Id="1"

value we care about

- it is used to exchange data b/w different applications.
- hierarchical
- support for schemas & data types.
- common complaint is these files are very lengthy; lots of tags
- used by APIs

JSON Java Script Object Notation

- successor of XML
- hierarchical
- easy to understand
- no tags
- used by APIs

Scanned with CamScanner

Lecture : 7

PARQUET

→ binary format & need a tool to work with. (eg. Parquet Viewer)
↓
→ additional functionality to filter rows using SQL like statements in search bar.

→ hybrid storage
  . ideal for OLAP queries (like sum, avg., etc.)
→ used for analytical workloads.

Row Storage ⇒ we write row after row

		A	B	C	D .....
Row Group 1	1	A_1	B_1	C_1	D_1
	2	A_2	B_2	C_2	D_2
RG2	3	A_3	B_3	C_3	D_3
	4	A_4	B_4	C_4	D_4
RG3	5	A_5	B_5	C_5	D_5
	⋮	Country		Sales Amount	PRODUCT

$\boxed{A_1}, B_1, \boxed{C_1}, D_1, \boxed{A_2}, B_2, \boxed{C_2}, D_2, \boxed{A_3}, B_3, \boxed{C_3}, D_3 \ldots$

```
SELECT
    SUM(SALES_AMOUNT)
FROM
    TABLE
GROUP BY
    COUNTRY
```

to execute this query ; program has to go through the whole file eventhough just 2 columns are being used.

∴ great for OLTP not for OLAP

	A	B	C	D
	COLOUMNS			
ROWS 1				
2				
3				
4				
5				

Columnar Storage ⇒ data is stored column by column

- data is stored column by column.
- faster executn : lower reads & writes.

$\boxed{A_1, A_2, A_3, A_4, A_5}$ B_1, B_2, B_3, B_4, B_5, $\boxed{C_1, C_2, C_3, C_4, C_5}$. ...

- OLTP ✗
  OLAP ✓

Projection — meaning selecting specific columns (like WHERE predicate use to filter things on PRODUCT)

Scanned with CamScanner

Hybrid Storage

- data is grouped into row groups.
- where we have predicates in query.
- rows are grouped in row groups
- data is saved row group by row group & within each row group data is saved column by column.

```
SELECT
    SUM(SALES_AMOUNT)
FROM
    TABLE
WHERE PRODUT = XX
GROUP BY
    COUNTRY
```

RG1 A1, A2, B1, B2, C1, C2, D1, D2
RG2 A3, A4, B3, B4, C3, C4, D3, D4
RG3 A5, A6, B5, B6, C5, C6, D5, D6

→ in Parquet file type, we have Meta Data about max & min value in each Row Group based on which a processing engine decides which data to be read & which to skip. Inside Meta Data there's also information about starting & ending location of every column in a Row Group.

→ works well with Projections & Predicates (filtering)

→ Data Compression :

73 MB – csv (10M rows)
224 MB – Parquet (10M rows)

↓

because of Dictionary Encoding: → part of meta data

Country
USA
USA
Poland
Poland
Poland
Germany
Germany
⋮

⇒

Dictionary
USA → 0
Poland → 1
Germany → 2

∴ instead of storing the long strings we store it as coding.

- in Analytical workloads you want denormalized data ⇒ repetition of values ⇒ encoding is handy here.

Scanned with CamScanner

- run-length encoding ; in this we specify no. of occurence as well

Country		
USA	0	0,2
USA	0	
Poland	1	1,3
Poland	1	
Poland	1	
Germany	2	2,3
Germany	2	
Germany	2	

→ schema should be present inside Metadata
- need not to ask data provider again & again. It is already present inside Metadata of Parquet file.

X ═══ X

LECTURE : 8

- **Delta Lake** ≠ Data Lake
  - → open source storage format
  - → default file format used by Databricks, Microsoft Fabric
  - → Parquet + transaction log
    - ↓
    - stored in _delta_log directory in json form (automatically)

—.crc
—.json

- when you perform CRUD operatn in Databricks, new json file is created in logs which tells us what has been done to the old file & what are the new files that need to be considered.
- new parquet files will be created ~~along~~ but we will still be having old parquet files on which the CRUD operatn had taken place.

→ ACID guarantees

Atomicity – either completes or rolls back i.e success or failure

we would have multiple files because of parallelism - kyunki shate-2 task sabko assign kiye honge, they would generate their own parquet files parallely.
→ different nodes

partitioning - Table Folder
- delta_log
- Country=US
  - parquet files
- Country=CN
  - parquet files

• now suppose your table file is partitioned in 100 diff. parquet files. If you want to make update to whole dataset/table by writing a query. Physically, update has to happen to all 100 files ⇒ 100 new pq. files. And lets say our query engine starts to re-write 20 files & after that our cluster dies ⇒ 20 files re-written
80 files failed
∴ data in not consistent state

now if we want to make any changes in our whole table; pcor table toh partitioned hai? to yeh aise hoga ki log file main jab tak commit nahi hota, tab tak new parquet file ignore hongi.

⇒ in case of Delta Files a new commit is written to transaction log if all changes have been written.
toh Delta Lake atomicity guarantee deta hai.

• Consistency
• State 1 $\xrightarrow{\text{migrating to}}$ State 2
• making sure integrity is maintained.

• Isolation
• about concurrency
• pessimistic model - eg. you are writing a row & you think someone can delete that row at same time so you lock the resources.

optimistic concurrency - our transactn will conflict each other so there is no point in locking the resources.
↓
Delta Lake uses this
↓
uses Snapshot reads
• jab koi ek transactn jis pt. of time par shuru hui use usi pt. of time ka jo data hai wahi milega.
based on transactn log every transactn would know how the data looked like when it started during operation.

every developer will have their Snapshot version of data based on when they started their transaction

Scanned with CamScanner

- Durability
  - changes ~~that~~ has been made should be persisted.
  - LRS, ZRS, GRS – makes sure our data is always available in case of any failure

Audit History

DESCRIBE HISTORY _____

- we can see $\underbrace{\text{who}}_{\text{userName}}$ made $\underbrace{\text{what}}_{\text{operation}}$ changes, $\underbrace{\text{at what time}}_{\text{timestamp}}$

Time Travel

- we can $\underbrace{\text{peek at}}_{\text{using timestamp or version commands}}$ ~~into~~ the old data before modification.

Vaccum Command

- removes uncommitted files
- removes no longer needed files (default 7 days of retention)
  - jab kabhi transactn complete nhi hoti, tab entry log transaction mein enter nhi hoti but parquet file ban jati hein. Parquet files ki space hoti hai. Aur hum storage ke paise dete hein. Par jo hume parquet file nhi chahiye jo uncompleted transactn se bani thi usko remove kaise karein?
  - impact time travel

Schema Enforcement

eg: Table 1 column : first_name, age → data set 1
Table 2 column : first_name, colour → data set 2

} combining data set with different schemas

if you join both tables you could get: first_name, colour
maybe age column gayab ho jaaye.

Sometimes schema is not enforced by Parquet.

- but if you save your dataset as Delta Lake & try to do same operation it will show schema mismatch error.

Scanned with CamScanner

Checks & Constraints

- you can define ~~a~~ constraints
- if you try to do CRUD operation ~~that~~ does not match with constraints you will get an error.

Schema Evolution

- what if data is changing in your source & you want to add more column when Schema Enforcement is enabled.
- tell Delta explicitly that you are merging ~~data~~ schemas
  mergeSchema → true

Merge

New Data $\xrightarrow{\text{merge}}$ Existing Data

if id matches = update
if not then add/insert entry

Optimize & Z-order

- in practical, it is faster to work on bigger files
- grab all small files & consolidate in one bigger file in Delta Lake
- when we do ~~that~~ we can optionally enable Z-ordering
  - way to instruct databricks or Delta to collocate data based on some columns. eg order rows in our files based on country column

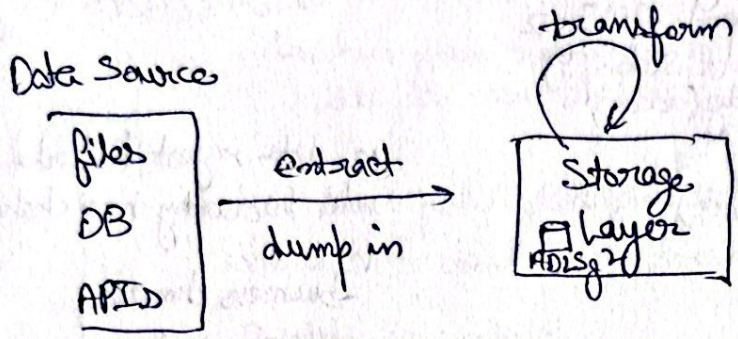
Unified Batch & Streaming

- we can use same table (Delta) for batch as well as streaming processing
- ∴ allows us to reuse the code.

Scanned with CamScanner

Lecture : 9

Data Lake Structure



- we do not want Data Swamp. When you allow everyone to dump data into Data Lake without governance. You do not know which data is present where & in what state. In case you start a new project, data might be present inside your databases but because of Data Swamp you would end up pulling data again.
- on which state of data we need to perform Transformation.
- To avoid Data Swamp we split data into layers/zones which store data in form of different state, quality etc.

Raw/Staging/Landing/Bronze Layer

What?
- we want one to one copy of source
- no transformations
- immutable - we do not apply any transformation here
- retained forever
- no one should make reports based on this data. ∴ give limited access to this layer.
  - → Data Engineers
  - → Data Scientists

Why to have Raw Layer
- limit stress on source system: meaning once you connect to Source Sys. from your Database you just grab the data & dump it.
- simplifies development: whole processing doesn't happen in a single go. So you be re-running process in diff. phases. So you don't go to Source Sys every time you run new process
- we could re-run whole process from scratch.
- because of bugs, when value is wrong you backtrack logic through each phase/stage (what was actually received) & verify at which step you made a mess. You could go to a pt. & end up in Raw layer data

Scanned with CamScanner

How to apply?

- use separate container in ADLSg2
- use different Data Lake or shared ADLSg2
- define hierarchy (to avoid Data Swamp & for security (access))

Hierarchy

Business Domain
Source System
Dataset / Table
Load type
Ingestion Date

\RAW ← layer (container)
└ SALES ← domain
  └ SAP ← source system
    └ SALES ORDERS ← Dataset
      └ DELTA ← Load type
        └ 2023 (year)
          └ 10 (month)
            ├ 15 (day)
            │ └ [ ] [ ]
            └ 16
              └ [ ] [ ]

partitioning based on date

} file formate : Native (original)

jo data ingest kar rahe ho uski Hierarchy honi chahiye

- business domain
- source system
- dataset
- Load type
  - full
  - delta : in case of huge tables
- ingestion date (when given data source was extracted & saved to a data lake)

- formate of files :

Source	Raw Layer
CSV	CSV
XML	XML
DB	Parquet / Delta

⇒ this Hierarchy in our RAW Layer is important to avoid :
- Data Swamp
- Security issues / access issues

- define Lifecycle Management Policy : to migrate data from Hot → Cool, Cool → Cold, Cold → Archieve

- watch out for PII data (Personal Identifiable Information) : if you receive this, if there is law on how to process it consult legal team.
  - ~~anon~~ anonomize it.

Scanned with CamScanner

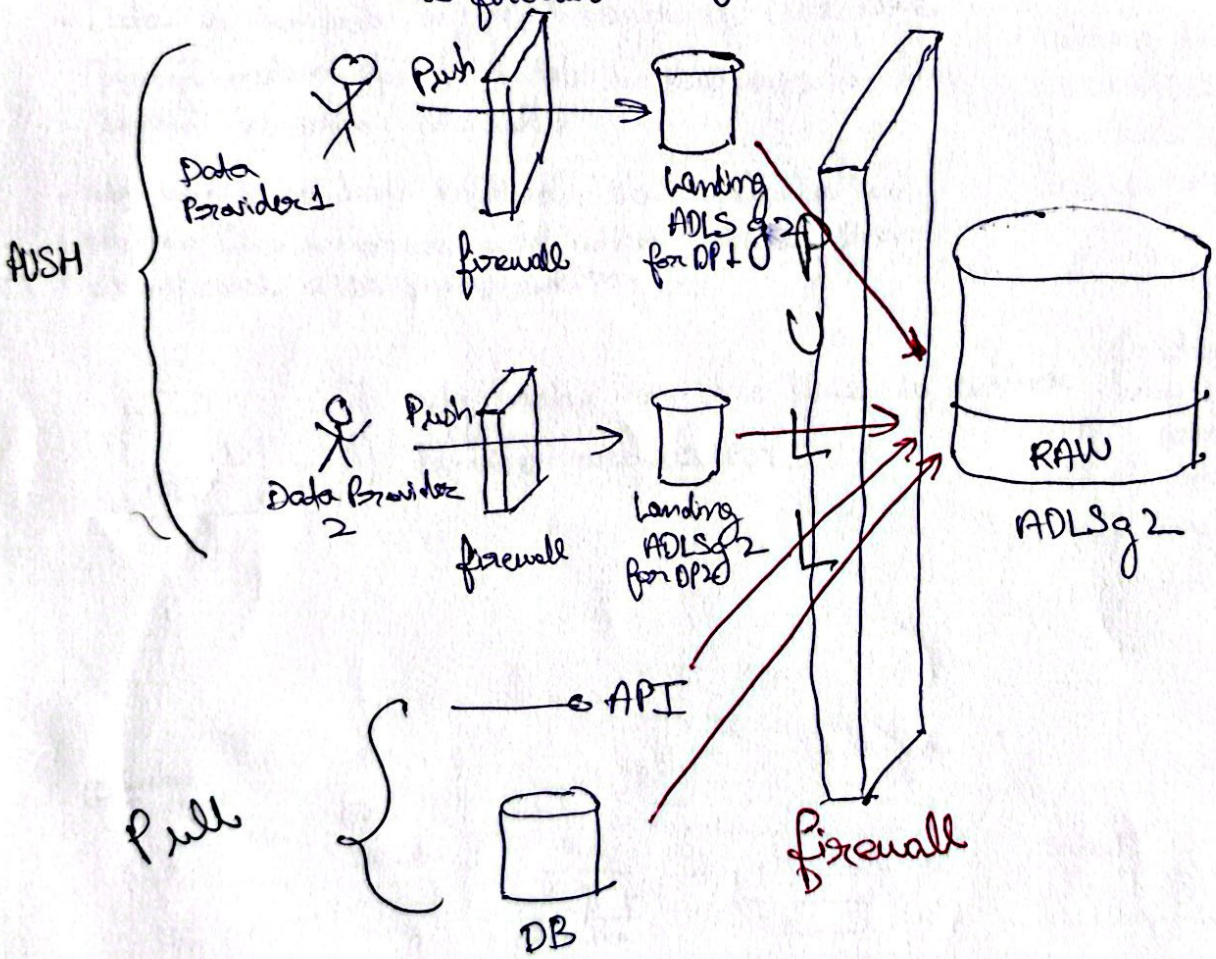
- Security, Networking
  we may receive data in form of
  PUSH vs PULL

  ↳ when your resources is connected to some database & grabs data to our datalake

  ↳ when other party (data provider) connects to our Data Lake & uploads the data to it; meaning - they need some kind of access

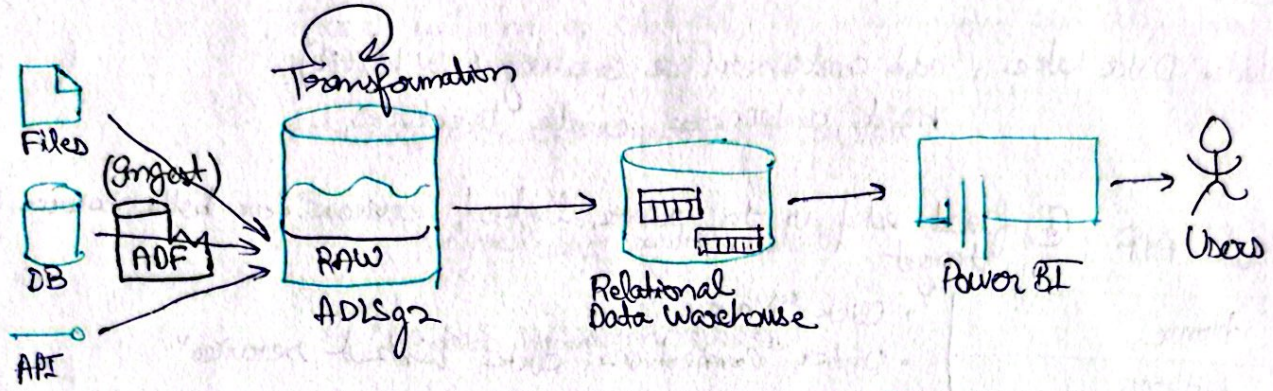
  In this scenario:

  - create small landing Data Lakes for different Data Providers & use firewalls



LECTURE - 10

ADF place in BI flow

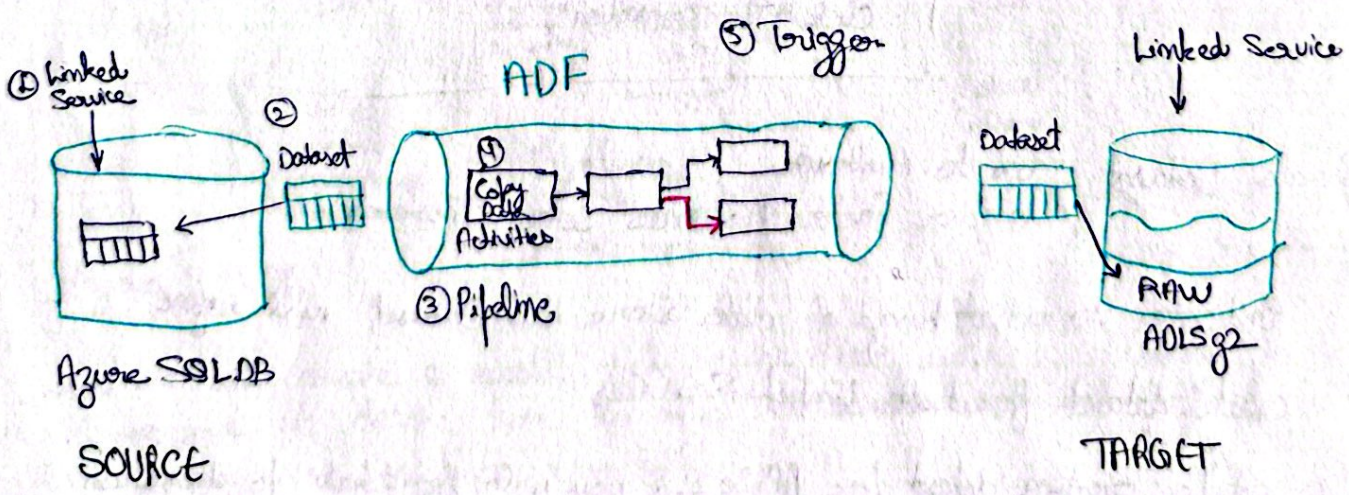


- Data is dumped in RAW Layer of Data Lake. Transformation applied on data in Raw Layer & stored in same Data Lake
- We will also have Relational Data Warehouse in modern data warehouse into which data is then transferred after transformation.

RAW Layer
- nothing but a container inside ADLSg2
- you add directories into this container

i) copy data from one place to another
ii) provides orchestration

Orchestration
- blue arrows show flow of data; is some kind of procedure/step & putting these steps in appropriate order.



Scanned with CamScanner

- Create Azure SQL Database
- Create Data Lake (Storage Account)
- Create ADF
- Inside Data Lake ; add container (for creating RAW LAYER)
  inside container ; create directories.

- Inside ADF : ① first step is to create linked services for both source & target

Home
Author
Monitor
Manage
Learning Center

on left side

- Click "Manage"
- Under "Connections" click "Linked services"

  2 options : Data store Compute

Linked Service for SQL DB (source)

- Name new Linked Service
- Select Server Name & Database you want to connect with
- Select Authentication type
- Add User name & Password
- Click "Test connection"

Linked Service for ADLSg2 (Destination

- Select datalake
- Name new LS
- Select Authentication type
- Select Azure Subscription
- Select Storage Account name
- Click "Test connection"

② Create Dataset : Go to Author
  Under Factory Resources select "Datasets"

Indicate to what kind of data store this Dataset will refer

- create dataset for both Linked Services

Target Side } - while creating dataset for ADLSg2 : you will be asked to choose an option from different file formats. And also for the path where you want data to be stored.

Scanned with CamScanner

③ Create Pipeline:

Author → Pipelines → New Pipeline

There will be a canvas to which you can drag & drop various activities

④ Activities → Move & Transform → Copy data

Source:

Define/Select source dataset

Sink:

Select Destination dataset

"Publish all" to save changes

⑤ Add "Trigger":

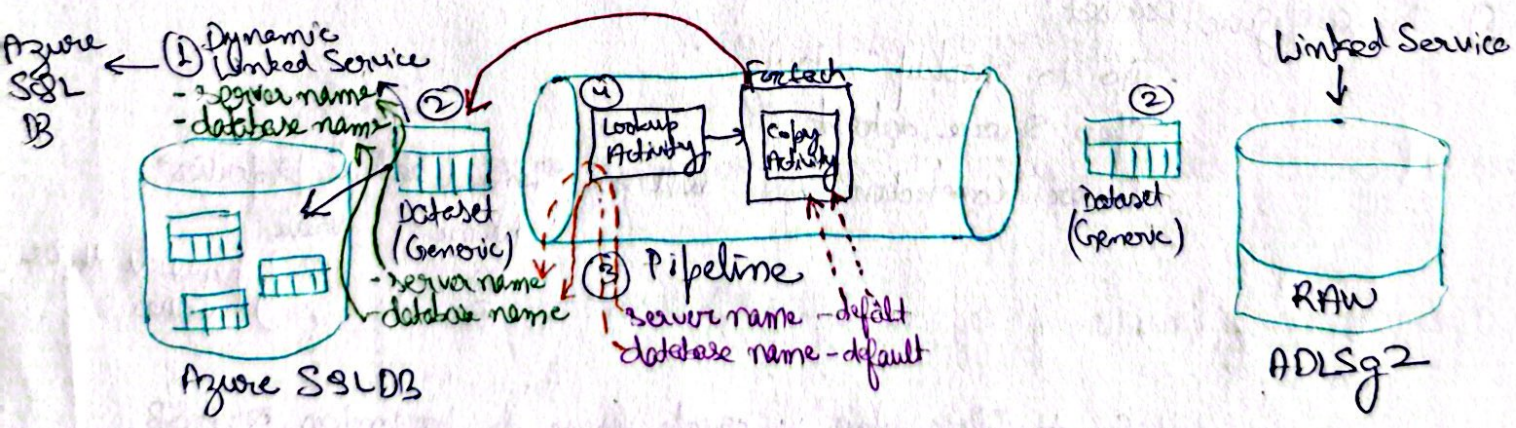
when your Pipeline should do its job.

X ═══ X ═══ X

LECTURE 11

Dynamic Pipelines

- If we have to copy lots of table, we do not want to create dataset separately all the time. So instead of creating multiple pipelines for different tables we will consider following architecture.



- This time we will create a create linked service that would be pointing to any SQL DB because at design time we might not know what database to connect to because we know that at run-time

- At target we are using same linked service - Not indicated ADLSg2

Scanned with CamScanner

② we will configure our Dataset to represent data read from SQL database at source side. And at target side our Dataset should be dynamic as well. Because we are using dynamic Dataset at source side ∴ we want Dataset at target side that could process all the tables coming from source side.

③ Create dynamic Pipeline

④ Choose 'Lookup' under "Activities".

- Under 'Lookup' in 'Settings' → Create New "Source Dataset" → then create New "Linked service". Now we will configure our "Linked service" by adding New "Parameters" eg: Server Name, Database Name } Created at Linked Service level

- Under "Account selection method" select "Enter manually" → "Add dynamic content".
- This will take you to "Pipeline expression builder" from which you get some expressions for evaluating server name & database name to be used under: Fully qualified domain name
  Database name

⑤ Authentication type : SQL authentication

⑥ To configure Dataset:
- Go to "Lookup" activity
- Open Source dataset
- Under "Connection" you will see "Linked service properties"

Name	Value
A	?
B	?

} provide these values

- Go to "Parameters" : create those to parameters eg: A&B
- Use A &B to pass values to linked service parameter
- Go to "Connection" → Add dynamic content
  This will take you to "Pipeline expression builder".
  click on Parameters to get corresponding expression to be used under:

Name	Value
A	
B	

Scanned with CamScanner

⑦ At Pipeline level:

add 2 parameters eg: serverName, databaseName and give them default values

now pass them to our Dataset parameters using Lookup Activity

⑧ Go to Lookup:

Settings → Dataset properties → Name Value (Add dynamic content)

→ Pipeline expression builder → Parameters → Copy expression

if you see just one parameter click on { } (code part) of pipeline & play with json file.

After all this it is better to run a query to fetch the different table names & their schema.

⇓

Use query

↓

Query

↓

Preview data (to verify)

Uncheck "First row only"; Preview data (to verify) & click "Publish" to save changes

"Debug" shows what kind of output we get from our "Lookup Activity"

Go to "Output"

→ [→

i/ps to our activity

o/ps of lookup activity

if you click on this you will see info. in Value element & this info is what you asked in query.

Scanned with CamScanner

- Now you know what data you are expecting. You will then have list of tables you want to process. Then you start copying every table by iterating over the list of elements (tables in this case). Inside each loop we will have copy activity.

Activities → Iteration & conditionals → ForEach → Settings → Items (configure this)

↓

Add dynamic content → Pipeline expression builder → Activity outputs

↓

actually you have to give the values of Lookup o/p here. One great thing about ADF is you can see o/p of the previous activity.

Select appropriate option & take the evaluated expression & paste it in Items.

to create copy activity inside "ForEach":

- click on "edit"
- opens a canvas
- drag "Copy Data" & drop.
- define Source & Sink

↓

from options select the generic dataset that you created earlier.

↓

give appropriate value to parameters defined at pipeline level

So in copy activity, we used Dataset Generic as source & we used value of server name & database name that was gotten as parameters for pipeline

Here connection is established now next part is querying.

Scanned with CamScanner

- Query:

create a dynamic query because after every iteration we will refer to different table

Add dynamic content → Pipeline expression builder → forEach iterator

↓

Iterate over all tables

↓

at this point we know which data to read or define your own query

At Sink side:

we have to decide / define where we want to store our data. And we want to make it dynamic

- Create a new generic dataset that could store data on Data Lake

↓

Azure Data Lake Storage Gen2

↓

Select format (CSV)

↓

Set properties

Name

Linked Service - select one you already created

File Path

↓

Go back to "Sink"

↓

Sink dataset

↓

open

↓

File path

↓

Make it dynamic : Add dynamic content

↓

Parameters ⇒ depends on how you want to structure your data

↓

Create

+ serverName
+ databaseName
+ schemaName
+ tableName

this we get from pipeline level

this we can take from the loop →

RAW

└ Server +

└ Database +

└ Y=?

└ M=?

└ D=?

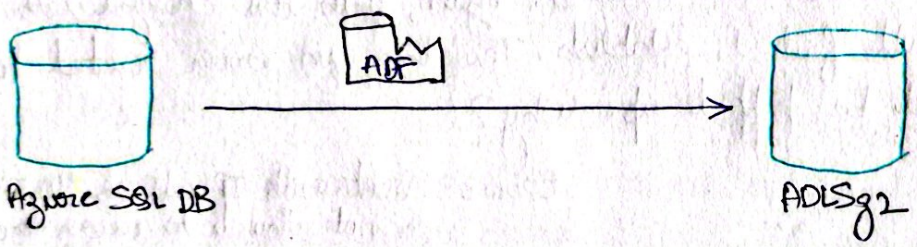
└ [~] + schema + table

Lecture 12

Azure Data Factory Integration runtimes

- so far we were ingoring/not taking into consideration networking part.
- in real applications source & sink side are protected by firewalls containing various rules. So ADF integration runtimes play important role in helping this

Connecting securly to a database



First create a Linked Service

- Open ADF & go to Manage tab
- Connections → Linked Services → New → Azure SQL Database

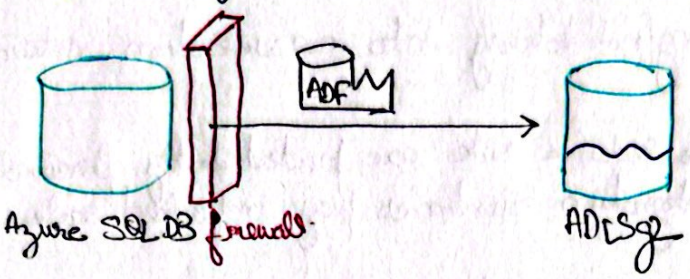
In same window

↓ Pickup a Subscription
↓ Pickup Server name
↓ Database name
↓ Use Authentication type
↓ User name & Password
↓ Test connection

you will see this connection would fail.

Scanned with CamScanner

because our database is by default protected by a firewall & our ADF has to go through this firewall



In order to fix this if you go to your server & under "Security" you will see under "Networking" that "Public network access" is "Disable". Now if you attempt to enable it that would mean you are giving permission/access that it can be accessed over internet. Again, with this changed settings the connection would fail to establish. This time an error related to IP address would be popped up. Why?

Error: Client with IP address 40.74.27.172 is not allowed to access the server

Easy Solution:

Configure "Firewall rules" & enable "Allow Azure services & resources to access this server.

By enabling this option we allowed any Azure service & any bad customer with this IP ~~access~~ address to access our data by bypassing our firewall.

* Networking stuff are set at Server Level.

Solution is

Integration Runtimes : ADF is serverless, we just drag & drop activities. But take ex: you are copying some stuff, that would need some compute power. So, ADF uses compute infrastructure which is called Integration Runtimes. Just like VM helps to execute our activities.

a) Auto-Resolve IR

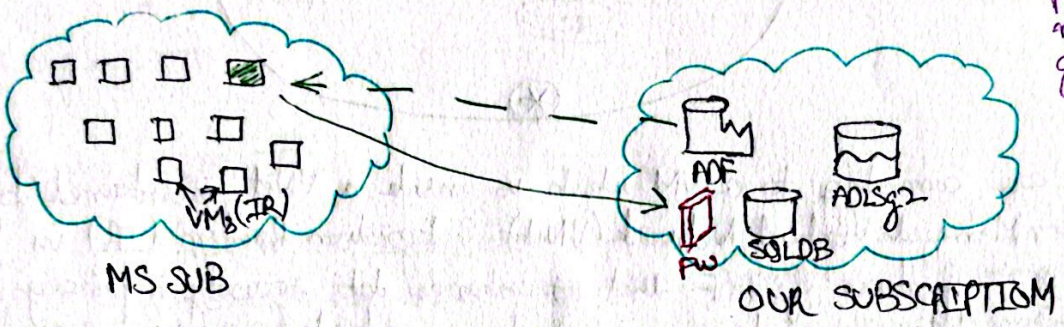
Scanned with CamScanner

a) Auto-Resolve IR

Suppose we have some Azure services running in our Subscription like ADF, ADLSg2, SQL DB. Along with our subscription what is also included is Microsoft Subscription to which we do not have access to. But our ADF will be using some objects from this Microsoft Subscription.

We will install IR software on those VMs present inside MS SUB.

Ideally suitable for public available data with no firewalls like FB API, google API etc.



- These VMs would have some IPs assigned to them. The error related to IP that occurred before because of some VM ~~try~~ with some IP tries to access our SQLDB
- All these VMs managed by Microsoft
- It uses Public Endpoints.
- As these VMs are not ours maybe after use same VM will get assigned to some other customer which could by pass our firewall

① • How to implement : Go to your SQL Server + Add a firewall rule & add IP address. Save your settings & run test connection in ADF

Cons:-

② by hardcoding this particular IP address imagine if next day you would like to run same pipeline. This time you are not guaranteed to get same VM allotted to you previously. So in result you could see IP related error again.

- for example we cannot install any kind of ODBC driver (to connect to a DB) because that VM is not ours.

Scanned with CamScanner

NOTE!!!

In case of AutoResolveIR, you can bypass the firewall of the Server you are connecting to. You do that at Server level, under Networking by simply providing IP of the VM that is trying to connect to the Server, by defining a firewall ~~rule~~ rule.

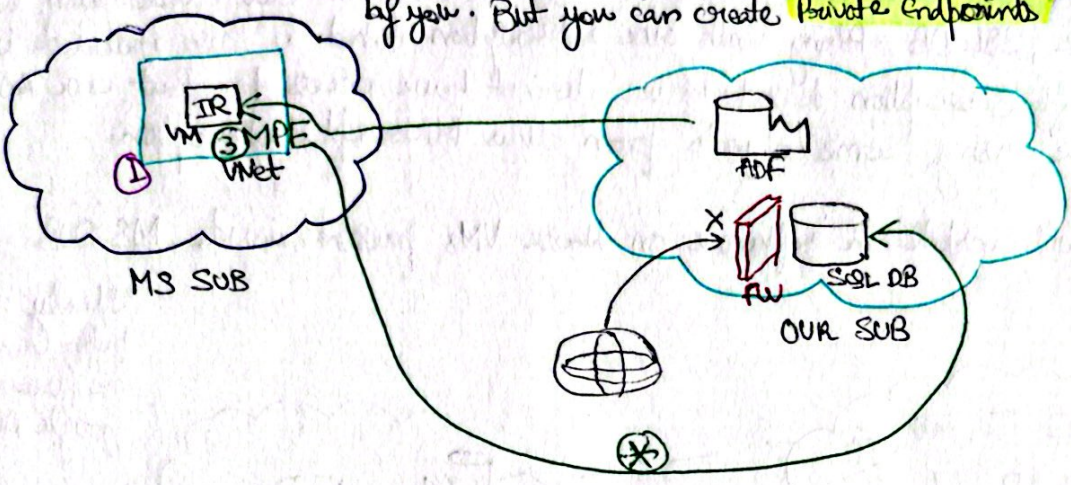
Based on your location, Microsoft have some certain set of IPs that can be used by ADF to connect to a Server. In case you have concerns that maybe next day a different IP would be assigned to AutoResolveIR that could lead to connection failure then one solution is to just bypass firewall for all the possible IPs dedicated for VMs ~~that~~ is utilized by ADF, by adding firewall rule for all the IPs.

But this is not best approach.

Scanned with CamScanner

b) Managed VNet IR

this VNet is still managed by Microsoft. You still can't install any ODBC driver. You just put VM inside a VNet that can only be accessed by you. But you can create Private Endpoints



- Here we are connecting to a VM which is inside a VNet maintained by Microsoft. Managed Virtual Network (VNet) Integration Runtime (IR) in Azure is a feature of ADF that ~~et~~ enhances data security & privacy by enabling data integration services to connect to resources securely within a virtual network.

- Managed VNet IR supports private endpoints, allowing secure access to Azure services such as Azure Storage, Azure SQL Database, & others over a private IP within the VNet.

① This creates Managed VNet

- ADF → Manage → Integration Runtime" → New → choose "Azure" for runtime env. → then go to "Virtual Network" & "Enable" Virtual Network Configuration & create

② Create Linked Service

New → Azure SQL DB → Connect Via integration runtime → Azure SUB → Server Name

→ DB name → Managed private endpoint (Managed VNet IR)

not created yet

③ Create Managed Private Endpoints

ADF → Manage → Managed private endpoints → New → Azure SQL DB → Azure SUB → Server name → Create

② so in theory you deploy Private Endpoints inside the earlier created Managed VNet and inside that you actually chose to which server our Private Endpoint should point to.

Now, so far we created an endpoint pointing to some Azure resource. But to start using it, this endpoint has to be approved by someone who has access to the resource to which we want to connect to.

⊛. Go to Azure SQL Server → Manage approvals in Azure portal → Networking → Private access → Select the newly created private endpoint

Also disable Public access

- But what about the case if we want to install something on those IR for ex some client certificate

c) Azure - SSIS IR

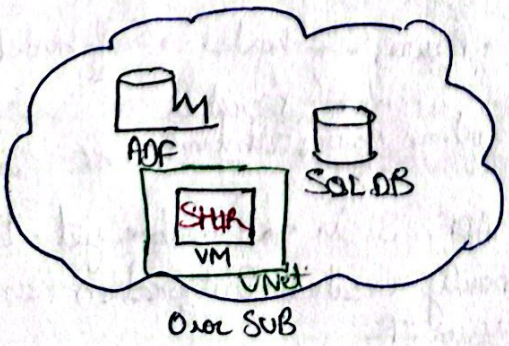
SQL Server Integration Services - It is a platform provided by Microsoft for data integration & workflow applications.

- component of Microsoft SQL Server database s/w that can be used to perform a broad range of data migration tasks.
- An ETL tool

for example you are using SSIS packages on premesis & now you want to migrate to Azure & you want them to be executable inside our ADF.

d) Self-Hosted IR (SHIR)

- Here it is our job to create this runtime
- you provision your VM & install SHIR on it
- but while creating a VM you make sure it should be inside some virtual network. In Managed VNet IR it is taken care by Microsoft.



- in Managed VNet, virtual n/w was taken care by Microsoft you just had to create Private Endpoints
- But in Self-Hosted IR you create your virtual network.

Scanned with CamScanner

Create a VM

- Select "Subscription" & "Resource group"
- Give name under "Virtual machine name"
- Select "Region"
- Select appropriate "Image"
- Under Administrator account choose:
  - Username
  - Password
- In "Networking" you choose "Virtual Network"
  - ↓
  - Create New
  - ↓
  - with default Address range & Subnet name
  - this will be taken care by Networking team

Check "Delete public IP & NIC when VM is deleted

So far VM has been created but it could not communicate with our ADF. Therefore, we install SHIR on the VM.

Connecting to VM : you go into the recently created VM

Go to resource → Connect → Native RDP → Download RDP file → Connect
Select
↓
then provide the credentials you used on creation of VM

After few moment Server Manager will pop-up
↓
Local Server
↓
IE Enhanced Security Conf. (disable this)

Installing SHIR

- Go to ADF → Manage → IR → New → Azure, Self-hosted → Self Hosted → Create

you will see 2 keys that is use to register your integration runtime

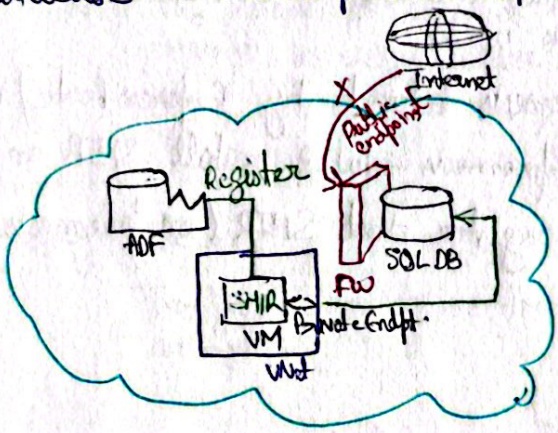
So far you created your runtime in ADF. It is not usable yet. Because you didn't install s/w on your VM. Your newly created Integration Runtime is still in ADF. So you can download your IR

Scanned with CamScanner

- Download and install integration runtime
- Copy the URL
- Open your VM → Open any web browser → Paste the link & download it

Next step is to register in IR
- Copy one of the keys from ADF portal
- In the setup installatn of IR paste it & then hit register

Now you are registered with ADF but it would not yet connected with our SQL Database. So created private endpoint that it will get to our SQL DB



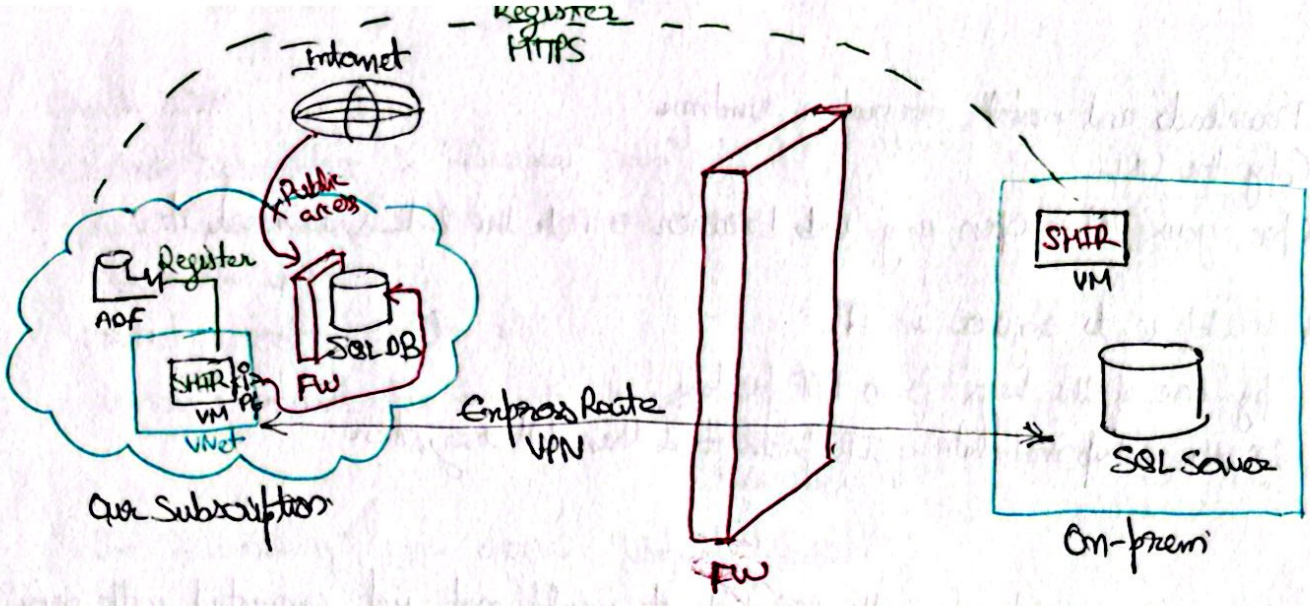
- Go to SQL DB → Private access → Create a private endpoint → Instance details (fill this)

→ Resources → Target sub-resource (Select the option where you want your endpt to point to)
→ Virtual Network → Virtual network → Subnet
for now choose this same as we did for VM

Finally: Go to ADF → New linked service → Azure SQL DB → Connect via integration runtime SHIR that we created
& now if you hit on "Test connection" it will work.

Therefore in this approach we are able to ~~to~~ connect of ADF to SQL DB securely through an infrastructure that is managed by us.

Scanned with CamScanner



In practice, we may have some sources that are not on Azure. In this our ADF can connect to that on-premesis as well.

- Connect our VNet to the on-premesis network by Express Route / VPN
- Deploy a VM/regular m/c ~~on~~ in on-premesis n/w & install SHIR on that m/c

Then by using keys from ADF register that SHIR (on-premesis).

Selection Criteria:

→ Integration Runtime

Azure IR / Auto-Resolve IR : if the source we are trying to connect is available publicly without network restrictions, without any firewall.

Azure IR with Managed VNet : if resource is not publicly available so you gotta choose b/w Azure IR with Managed VNet or Self Hosted IR. If you do not bother about infrastructure.

Self-Hosted IR : when you want full control like to deploy, provision & maintain Virtual m/c.

Scanned with CamScanner

(14) Lecture 14

Error handling in Azure Data Factory

ADF provide several mechanisms for error handling, including retries, custom error handling using activities, & monitoring tools.

Retry & Retry Interval in Copy Activity

The Copy Activity in ADF is used to copy data from a source to a destination. It includes built-in error handling mechanisms through the use of Retry & Retry Interval settings.

Retry :

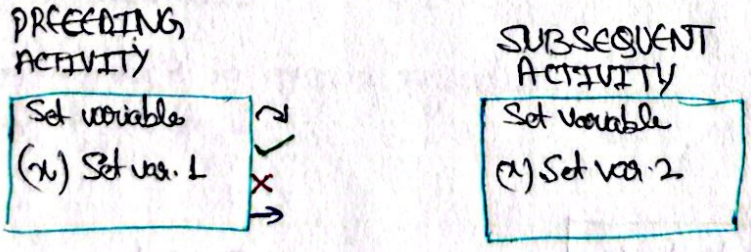
This parameter specifies the no. of times the Copy Activity should attempt to retry after encountering an error before it fails.
eg: if Retry is set to 3, the activity will attempt to execute three additional times after the initial failure.

Retry Interval :

This parameter defines the time interval (in secs) between each retry attempt.

Type of connectors available to connect two adjacent activities

ADF provides various connectors to link activities & manage the flow of data.



↷ Skipped : The subsequent activity runs only if the preceding activity is skipped

✓ Succes : " " " " " " " " succeeds

→ Completion : The subsequent activity runs regardless of the preceding activity's outcome.

× Failure : The subsequent activity runs only if preceding activity fails

Scanned with CamScanner

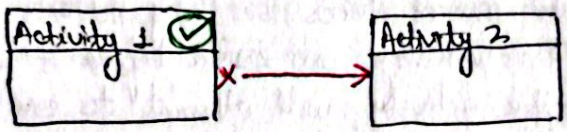
Determining pipeline status

- if status of leaf activities is succeeded the Pipeline status is Succeeded
- if the leaf activity does not run then the status will be in accordance with the parent activity of that leaf activity.

Conditional Paths

Patterns

a) Try Catch

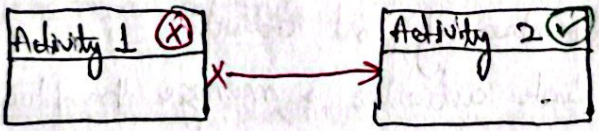


Pipeline status

Succeded

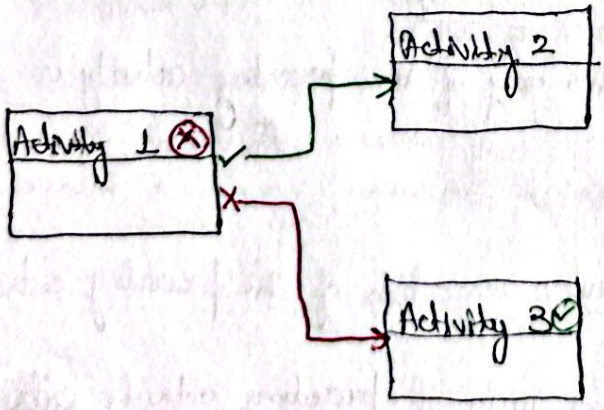
(since the leaf activity was skipped data factory will check status of its parent)

In this case business logic in activity 1 succeeded & error handling part in act.2 is skipped



Succeeded

b) Do if else



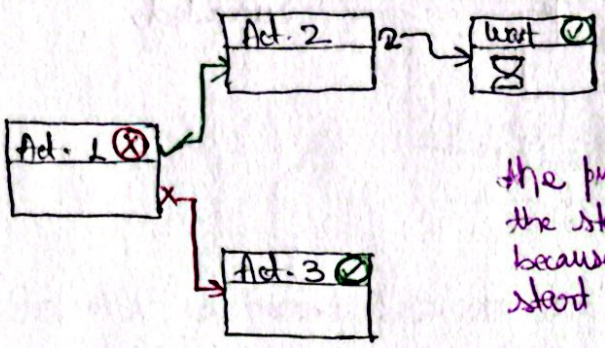
Failed

here we have got 2 leaf activities so ADF will check status of each leaf activity. Since, activity 2 was skipped, status of its parent activity (activity 1) will then be checked.

Scanned with CamScanner

c) Do If Skip Else -
we add some dummy activity

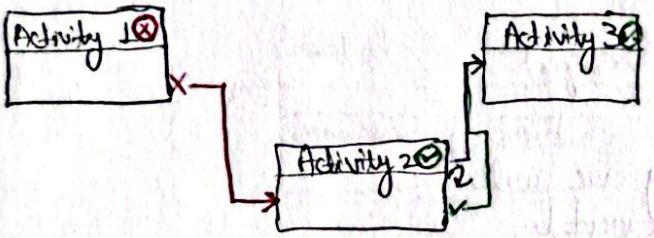
Pipeline Status



Succeeded

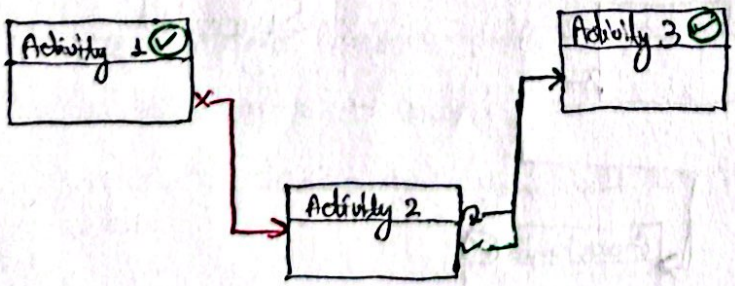
the purpose of that dummy logic is just to change the status of Pipeline. Because Activity2 was skipped because of failure of Act. 1, the dummy activity will start since it is connected through on-skip connector.

d) Try Catch
Proceed



Succeeded

here we want activity 3 to execute regardless of the status of activity 1



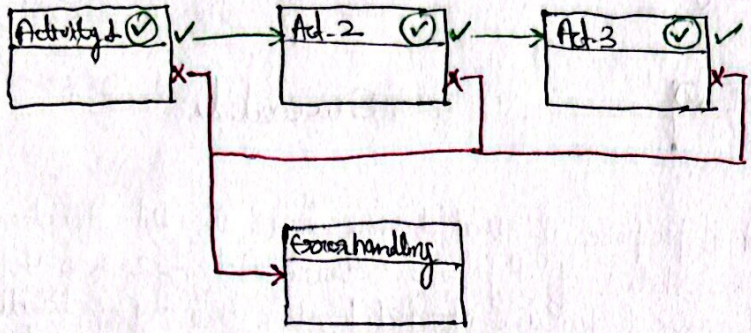
Succeeded

Scanned with CamScanner

e) Generic Error Handling

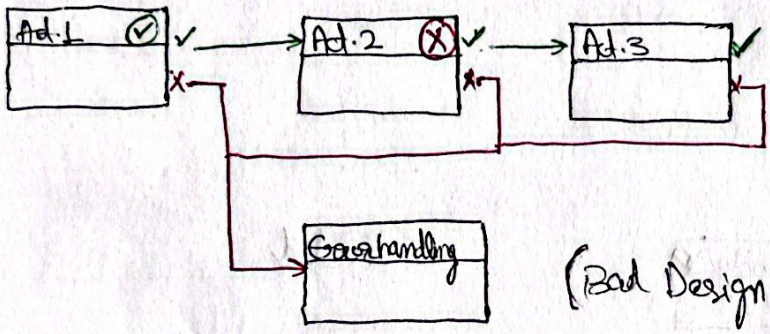
Pipeline Status

On Success scenario



Succeeded

failure scenario

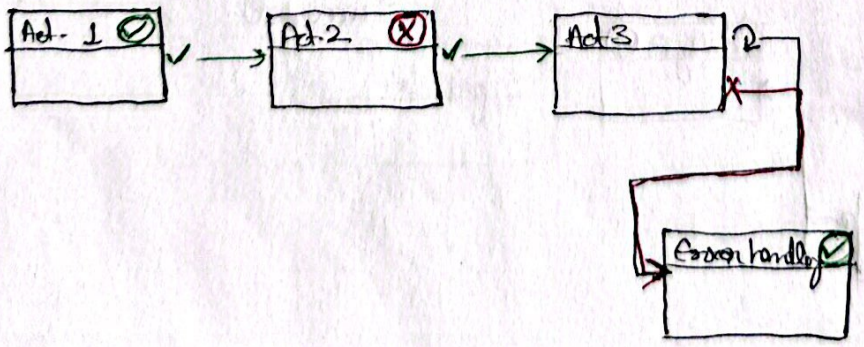


Failed

(Bad Design)

if there is any error in any activity we will expect error handling activity to get executed. But error handling will not work. Because, here the way you make connections, error handling will only trigger if all ~~activity~~ activities fail (AND Logic)

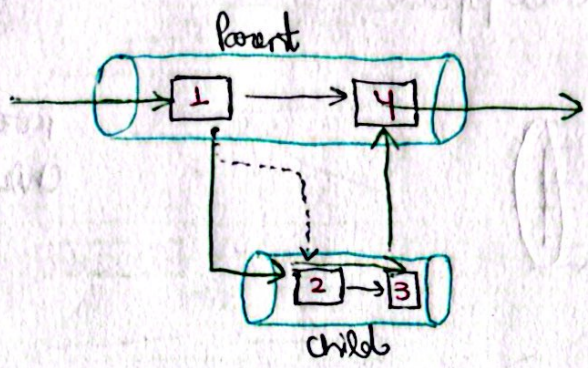
Good Design: connect last activity in our chain using on fail [X] & on skip [⊘] connectors.



Succeeded

Scanned with CamScanner

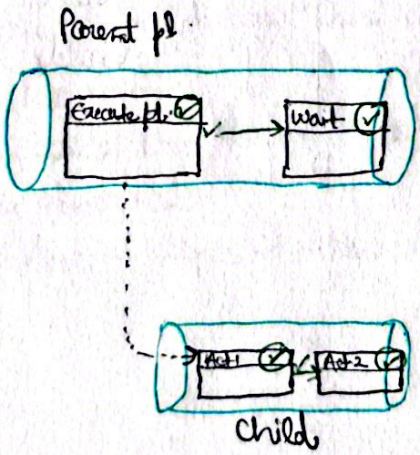
Parent-child scenarios



you want activity 1 in Parent pipeline to execute first, which in-turn triggers child pipeline. So, all the activities in child pipeline will be executed before activity 4 in parent pipeline runs.

- In ADF, a parent-child scenario refers to the orchestration where one pipeline (the parent) triggers the execution of another pipeline (the child).
- This setup modularize & organize complex workflows by breaking them down into smaller, manageable pipelines.

a) Scenario A
when all activities succeeded



Pipeline Status

→ in Parent pipeline, use the Execute Pipeline activity to call the child pipeline.
configure this activity to specify which child pipeline to run & pass any necessary parameters

→ define activities & logic that should be performed by the child pipeline. This could include data transformation, data movement & other data processing tasks

Parent - Succeeded
Child - Succeeded

- Go to Parent Pipeline → Add trigger
- Go to Monitor to check pipeline status
- Under "Pipeline runs" you shall see the status

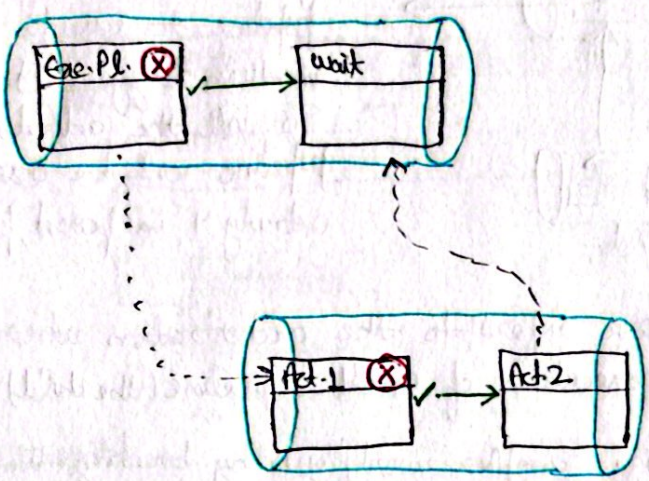
Scanned with CamScanner

b) Scenario B

Bug in first activity in child pipeline

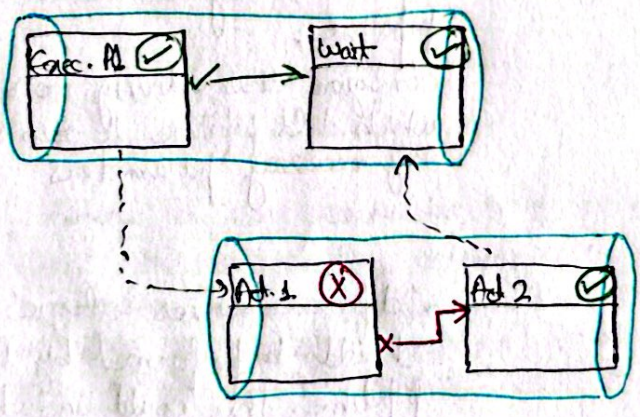
Pipeline Status

Parent - Failed
Child - Failed



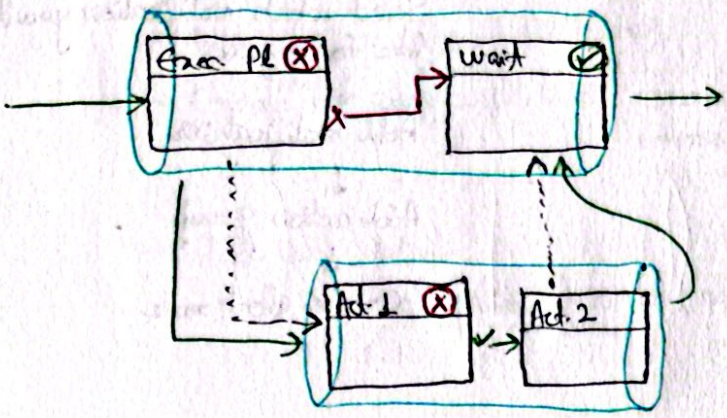
c) Scenario C : Suppose now you handle error in Child Pipeline by using appropriate connector.

Parent - Succeeded
Child - Succeeded



d) Scenario D : Instead of handling error in Child Pipeline you attempt to handle error in Parent Pipeline.

Scanned with CamScanner



Parent - Succeeded
Child - Failed

ALERTS

- Go to "Monitor" & under "Notifications" you will see ⚠ Alerts & metrics

Basically all the resources on Azure generate metrics & logs & based on those metrics we can set-up alerts

Azure Portal → Search for "Data Factory" → select your "Azure Data Factory" instance

Monitor Section

Once inside Data Factory instance, go to the Monitor section.

Create New Alert Rule

Monitor → Alerts → + New alert rule → Alert rule name
Severity (set the severity level)
Target criteria
↓
Failed pipeline run metrics
↓
Define Dimensions: for example, you might want to filter by specific pipeline names. And by Failure Type
↓
Alert logic: condition
Time aggregation
Threshold count
↓
Evaluate based on: Period
Frequency

Till here you have just created Alert. Next step is Notification setup.

Scanned with CamScanner

Configure notification → Action group name → Short name → Add notifications
Select which notification you'd like to receive
↓
Add notifications
↓
Add action group
↓
Create alert rule

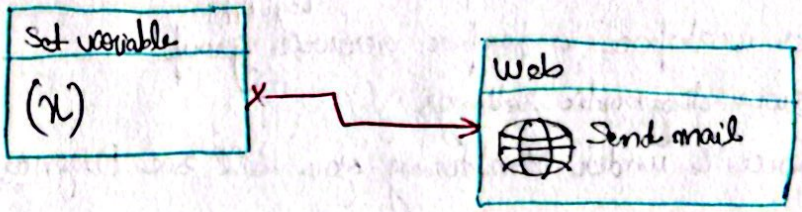
Logic App

- create a Logic App
- Azure Portal → Create a resource → Logic App
  Fill in the required details:
  Name
  Resource Group
  Location.
  ↓
  Review + create

- Design the Logic App:
  Go to resource → Logic App Designer → Choose a trigger → means this should trigger our logic app
  when a HTTP request is received
  ↓
  New step
  ↓
  Search for & select the Outlook.com
  ↓
  Send an email action
  ↓
  Sign in
  ↓
  Configure the email action:
  - To: Enter the recipient's email address
  - Subject: Enter a subject (e.g. "ADF Pipeline Failure Alert)
  - Body: you can use dynamic content to include details from the HTTP request
  ↓
  Save the Logic App
  ↓
  Get the HTTP Post URL
  (this URL will be used to call the Logic App from

- Configure ADF to call the Logic App.
  - → Navigate to your Data Factory
  - → Create or edit a Pipeline : Create a new pipeline or edit an existing one where you want to handle errors.

  - → Add Web Activity
    in the Pipeline designer, add a web activity to the canvas.



Configure the web activity

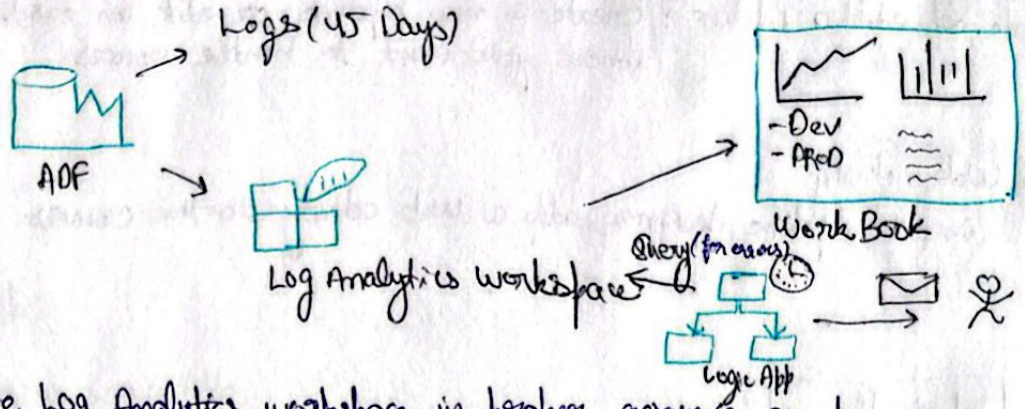
Settings → URL (Paste the HTTP POST URL from your Logic App.)
Method : Set to 'Post'
Body : construct the JSON payload that the Logic App expects

SUMMARY:

1. Create a Logic App with an HTTP trigger & an action to send an email
2. Design the Logic App workflow to handle the incoming request & send notifications.
3. Configure ADF to call the Logic App using Web activity triggers upon activity failure.

Scanned with CamScanner

Diagnostic Settings



- Create Log Analytics workspace in proper resource group
- Go to ADF & configure diagnostic settings
- Go to ADF resource & under Monitoring you will see Diagnostic settings
  ↓
  Add diagnostic setting
  ↓
  Diagnostic setting name
  ↓
  define what kind of Logs you want to be saved in Log Analytic workspace
  eg: allLogs
  Destination details
  Send to LAW
  ↓
  Save

By default ADF stores logs for 45 days. In order to get more logs, you create seperate Log Analytics workspace. So, ADF sends logs to Log Analytics Workspace.

∴ ADF will be sending logs to that LAW

Once in LAW, under Usage and estimated costs you can define "Data Retention"

Basically, LAW will handle logs from various Azure services & those logs can be queried using KQL.

Under Monitoring
↓
workbooks
↓
Default Template (here you define your KQL query)

Scanned with CamScanner

Summary

1. Create a Log Analytics Workspace : Set up a centralized repository for collecting log & telemetry data.
2. Link ADF to Log Analytics : Configure diagnostic settings in ADF to send logs and metrics to the workspace.
3. Query & Analyze Data : Use KQL to write queries & analyze data in the Log Analytics workspace.
4. Create Dashboard : Visualize the data & monitor it using custom dashboards.
5. Set up Alerts : Create alert rules to get notified of important events & issues.

Lecture 15

Data Ingestion :

- What is Source Type?
  - DB
  - File type
  - API (REST, GRAPH, etc.)
- Where is Source Located? this impacts the choice of Integration Runtime
  - cloud
  - on-premesis

  like for on-prem source you can go ahead with SHIR or Azure Managed VNet IR with load balancer.
- Are you Pulling or Pushing data?
- Data is coming in form of Streams or Batch?
- What is the Ingest Schedule?

  make sure you do not disrupt resources during Business hours.
- Is the data Source under development?

  in some scenarios when your source is constantly under development, you may require to be in touch with the source team to have some kind of roadmap regarding timelines about certain changes to be expected. So that then you can accordingly make changes in Apl. code.

  Same phenomenon could be seen if source is API & is constantly changing.

Scanned with CamScanner

- How to connect?
  - you may require to go through documentation
  - you may need to contact someone who knows the source system.
  - in case of dbs: ip, port, protocol, server name, db name
  - in case of APIs: url

it may happen when some companies do not want their data to be exposed on internet so in this case even if you know all the details of db like ip, port, server name, db name, password etc. you won't be able to connect.

Solution is to check if there is any 'Line of Sight' b/w the tool we are using to connect to the source & source itself.

eg: ADF to Database.

obviously, there would not be direct line of sight. There will lot of obstacles in between like firewalls etc. So, you contact the networking team & ask them to whitelist some IP addresses on Firewalls.

But this is not so easy, you would have to provide with some documentation explaining why this thing needs to be opened. So will be asked for security related stuff they you should be able to provide. This is k/a Security Audit.

Whitlist Firewall → ADF IP
→ IR Choice

- Security
  - Authentication
    - user/password
    - access token: they have some validity period
      usually used in case of APIs

      sometimes your ingestion process takes longer than the validity of the tokens. So make plans about how to get a new token.
  - Authorization
- POC (Proof of Concept) to verify connectivity
  - quick & dirty
    eg. you may use Postman tool to test connectivity in case when source is an API.

- Are there any Rate Limits or Throttling?
  - preventing DoS in case of APIs they have a limit that defines how many request are acceptable
    like: no. of requests/min.

Scanned with CamScanner

this would also mean how much data you can grab per minute.
In case the data that you are planning to pull is so huge that it exceeds the Rate Limits, so you intend to introduce some kind of artificial breaks (delays) to make sure your requests are not killed.

- Is the data source Synchronous or Asynchronous?

  in case of APIs, if you make a Synchronous request in form of an SQL query intending to grab some data, then in response you get some result as output.
  But in case of Asynchronous approach you send a request but as a result you don't get the data back immediatly but just an acknowledgement. In other words you periodically ask source is the data ready? & finally when it is ready you pull the data.

- Pagination

  in case of APIs usually data is not sent whole in once. The response will be split into multiple responses. Plan on how to iterate over different responses/pages.

- Is it supported by ADF?
- How much data?

  in some cases where data source is a db then if there is a big table that you want to import, you would not import the full table again & again

  - enable CDC
  - modified date
  - incremental id
  - date range (in APIs)

  } to detect cdc

- Initial load?

  In case of incremental load, you should have some data pulled before starting incremental load.

- Can we query the source?
- what Data is it?

  master data
  transactional data

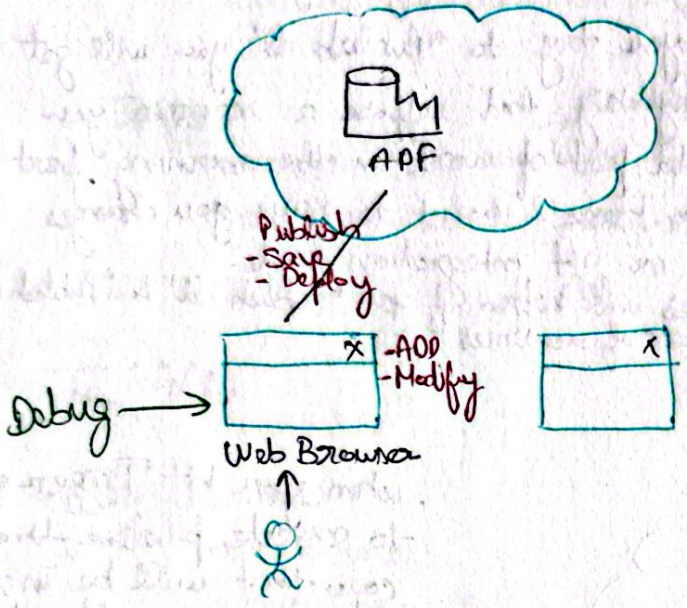
Scanned with CamScanner

- Is it special?
  PII
  Highly confidential
- Historical data?
  does it differ?
- Data Specification?
  encoding, column delimitor....
- Get sample data
- Contact Person
  you may have someone at other side who knows about the source data.
- Is it already loaded?
- What Frameworks the company is using?

Scanned with CamScanner

Lecture 16 : Executing ADF pipelines

- trigger mode execute the code that is persisted in ADF service (meaning only after you "Publish All")



Debug

- Suppose you deploy ADF in Azure & way you interact with DF is through web browser
- A developer make new changes, add, modify new pipelines via web browser
- Sometimes we want to test our pipeline locally. So debug does that for us. By debugging it means the version of code that is visible to us on web browser is getting executed

Create a New pipeline
↓
Add a Wait activity
↓
Press Debug

- if you copy some data & hit debug the data would be copied. Similarly, if you remove some data & debug, the data would be removed

a) No git integration mode

Go to Data Factory at top left corner → Set up code repository

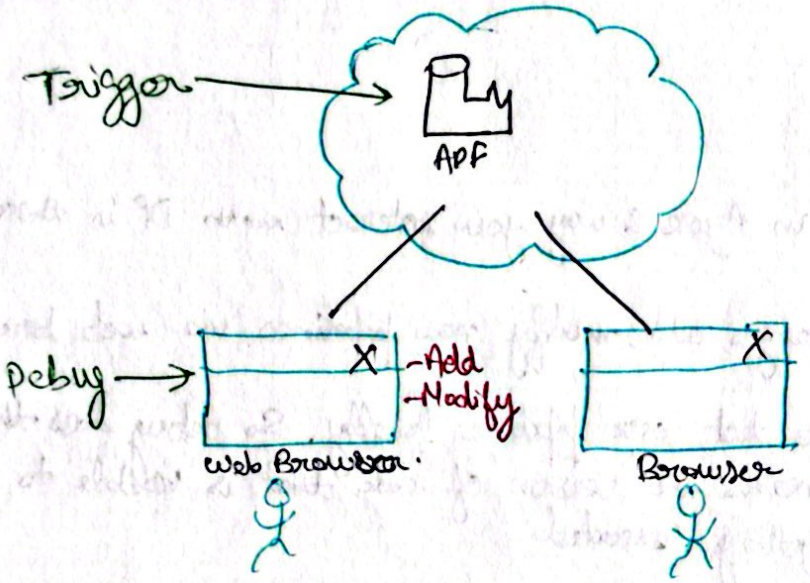
Manage → Source control → Git configuration

Scanned with CamScanner

- to save changes we hit "Publish all" which simply means deploy. With no git integration mode this is the only way to save changes.
- if you hit refresh you will notice your wait activity is gone
- without configuring any activity if you try to "Publish all" you will get error because you need to fill everything. And suppose a scenario you are working on some issue, you did part of work in the morning, part in afternoon & now you want to go home & want to save you changes but you cannot hit "Publish all" in no git integration mode.

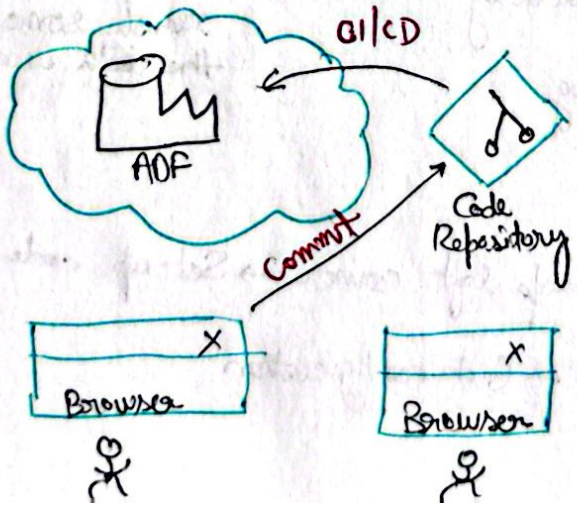
with no git integration only way to save changes will happen if you "Publish all" & "Publish all" will only work if you have done all the configuration of resources.

Trigger



- when you hit Trigger mode to execute pipeline, the code that will be in the data factory would get executed

b) with git integration



- every changes made by the developer is stored in form of commit in the code repository
- you can even Push an incomplete code & later pull it & continue your work

Scanned with CamScanner

∴ **Debug**

- for Local / Dev test
- runs manually
- executes the code that is loaded on UI

Trigger

- executes code loaded (published) on Data Factory service
- you first need to publish before hitting Trigger.

TRIGGER

a) Trigger now

- on demand
- runs manually

- Create a dummy activity → Publish it → Add trigger
  Trigger now

- it still requires someone to go to UI & hit button.
- not ideal for scheduling.

b) Schedule trigger - wall-clock schedule
- Many to many

Add trigger → New / Edit → New trigger

- here you can define Type
- start date
- Recurrence
- you can specify on which day, month etc.

↓

Ok

↓

Publish

you can see all your triggers in Manage, under Author option.

Scanned with CamScanner

if you look at the code { } you will find the list of pipelines on which this trigger is applied

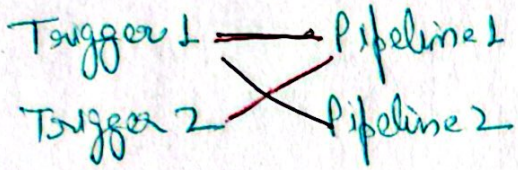
- you can use same trigger for multiple pipeline

Create New pipeline → create a dummy activity → Add trigger
Select old trigger

now if you again peek the code { } you will see another pipeline.

↓

you can add another trigger for this pipeline



c) Tumbling Window

- used in case of streaming
- we split timeline in equal intervals
- fires at a periodic time intervals from a specified start time, while retaining state.
- has a one-to-one relationship with a pipeline & can only reference a singular pipeline.

Go to pipeline → Add triggers → New/Edit

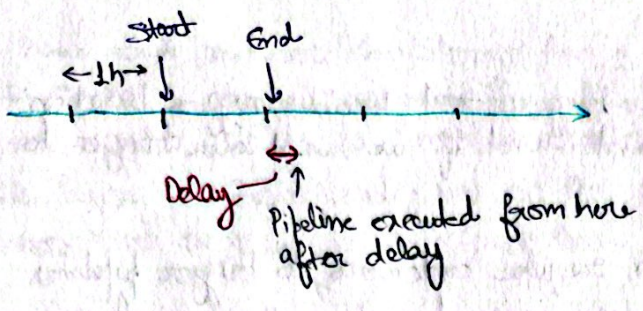
Name
Type : Tumbling Window
Start Date
Recurrence :
Every Minutes

Suppose we trigger our trigger every hour but there will be a small delay in the source system to store data so we want to wait for some time before we start our processing. ← Delay

So it means we are telling data factory to start pipeline every hour but also we are telling to wait for sometime = Delay

Backfill
Concurrency
Retry

Scanned with CamScanner



startTime
endTime

indicate what was the previous window that has just finished, that caused our ~~window~~ pipeline to executed after the small delay

WindowStart and WindowEnd

you can use windowStart & WindowEnd system variables of the tumbling window trigger in your pipeline definition.

Go to pipeline level
↓
Parameters
Name
[windowStart
windowEnd]

then pass the system variables as parameters to your pipeline in the trigger definition.

Go to trigger → New/Edit → Add trigger
↓
New
Name
Type - Tum. Win.
Start Date
Recurrence
↓
Ok

The moment you click ok, the trigger will detect our pipeline has 2 parameters and set them accordingly.

@trigger().outputs.windowStartTime
@trigger().outputs.windowEndTime

In Monitor → you can see Trigger run & Pipeline ~~and~~ for status & in Pipeline Run you can see our parameters are properly assigned & you can query data for ex. query only that data which was present during that window.

- One special feature about tumbling window is we can Backfill our data. i.e. we can configure it to run for past data. To make this backfilling process faster we can define maxConcurrency option, which will tell how many instances of this trigger can be executed at the same time. ex maxCon. = 24 means at same time 24 instances of our trigger can be executed to backfill the data.
- Retry policy - retry pipeline if it fails this option is not available in Schedule trigger

Scanned with CamScanner

d) Storage Event

- So far we studied about the triggers that would trigger based on time not occurence of some event. What if we want our trigger to run dynamically.
- Data integration scenarios often require customers to trigger pipelines based on events happening in storage account, such as the arrival or deletion of a file in Azure Blob Storage account.

- Event driven Architecture
- Uses Event grid
- con: it is able to process events that are in Storage account only.

Go to Azure Data Factory → Edit tab → New pipeline (dummy)

Parameters
Name
FileName

eg we want to execute this trigger when new file is uploaded/modified in storage account.
name of file that has been uploaded/modified

↓
Add trigger
↓
New
Name
Type: Storage events
Azure sub
Storage account name
Container nam: you want the event to occur on.
Blob path begin with
Blob path ~~begins~~ end with: .csv
Event
↓
Continue

you will be asked to enter value for parameter

@triggerBody().fileName

Scanned with CamScanner

We have event producer like Storage Acc. e fires various events. For ex: when file is created & maybe someone is ~~not~~ interested in this information because someone would like to do smth in response. So, the event is sent to middle layer i.e Event Grid. And on the other hand we have various event consumers that register to the Event Grid to listen to the event type.

Event Producer (Storage Account) → Event Grid → Event Consumer ADF

e) Custom Event Trigger

Event-driven architecture (EDA) is a common data integration pattern that involves production, detection, consumption, and reaction to events. Data integration scenarios often requires Azure Data Factory customers to trigger pipelines when certain events occur.

You send events to an event grid topic. Data Factory subscribes to the topic, listens, & then triggers pipelines accordingly.

Scanned with CamScanner

Lecture 17

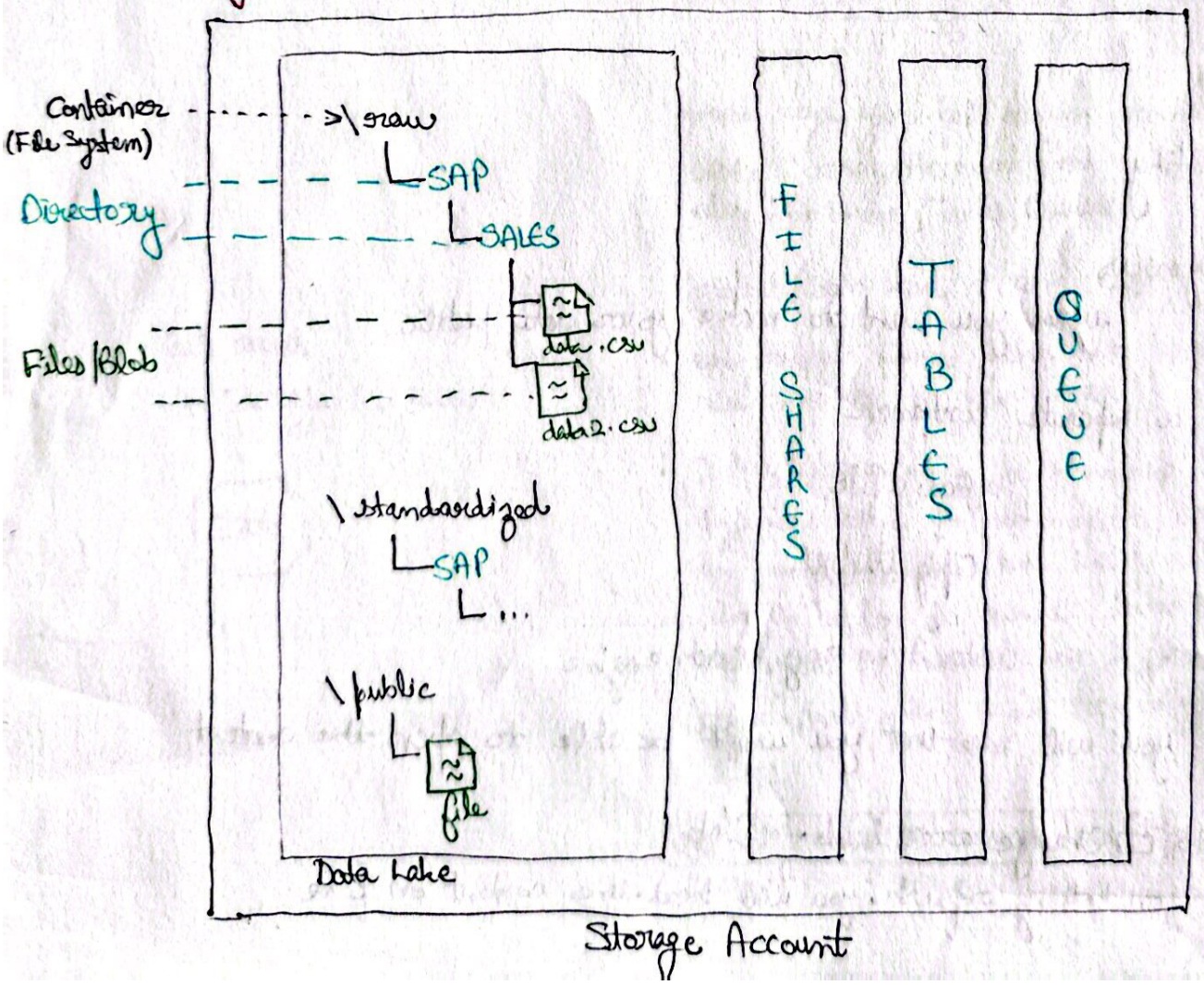
Azure Data Lake Security

Access

- Authentication
  - I know
  - I have
  - I am

  when you use more than 2 of these three = Multi Factor Authentication
- Authorization

Storage Account Architecture



Scanned with CamScanner

- Create Storage Account (a Data Lake)
- Make containers & directories as shown in fig.

Access to Data Lake

A) No Identity
- we do not know who is connecting
- we do know IP address of the connecting person

B) With Identity
- we know exactly who is trying to access
- " " " which service is trying to access

A) No Identity

a) Anonymous
- when you want to make your data public

Make a separate container
↓
upload a file
↓
Copy its URL
↓
Paste it in any search engine

you will see that you won't be able to open the content

Go to Change access level → Blob
on hitting refresh, you will read the content on file.

Scanned with CamScanner

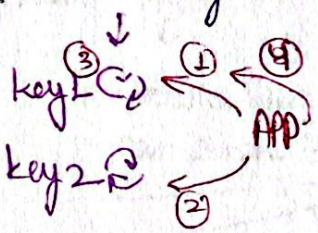
- Read-only
- container level

To allow or disallow anonymous access for storage account, set the account's AllowBlobPublicAccess property. Globally turn-off the access level in Storage Account itself.

Go to Storage Account → Settings → Configuration → Allow Blob anonymous access
○ Disabled ○ Enabled

b) Access Key

Storage Acc. → Security + Networking → Access keys



- these keys give full admin permissions of whole Storage Account (not only Data Lake but also File Share, Tables, Queue)
- protect both keys at any cost.
- we do not know who the person is who used the key to gain access
- 2 keys because in a scenario where some App uses key1 to authenticate, to connect & we suspected this key has been compromised. So in order to avoid down time of APP, configure APP to use key2, meanwhile rotate key1. And then again re-configure APP to use key1.
- (1) Set rotation reminders option is also present

- full access
- at Storage Acc. level

STOP

Scanned with CamScanner

Go to ADF → New Linked Service → ADLg2

Authentication type

Account key (by default)

• Account key & Access key

↙ In ADF

↓ In Storage Acc.

To allow or disallow "Access key" for storage account:

Go to Storage Account → Settings → Configuration → Allow storage account key access

o Disabled o Enabled

Scanned with CamScanner

Lecture 18

Azure data lake security - Azure key Vault & managed identity

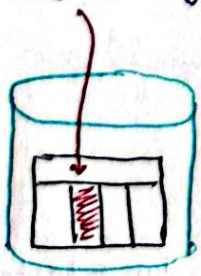
- Whenever we have to create a linked service, we specify the authentication type. So far we saw in lecture 17 that we hard-coded the access key.
- if you open the linked service code, you could see the credentials.
- Imagine a scenario where you created 5 ls & each needs access key for authentication. But later on you come to know if some of keys has been compromised, you will need to rotate key & again go to the linked service & paste the updated access key in all 5 linked services. Therefore, it is not a good practice.

We solve this problem using Azure Key Vault

- service that stores secrets
- can store 3 types of things

Keys

- suppose we have a database & it contain a table. And if one of columns in the table is super secrete like credit card no. So we encrypt this column using an encryption key



Certificate

- like certificate that encrypt web traffic
- SSL/TLS
- code sign

Secrets (most common)

- storing

Azure Key Vault is not a replacement for databases.

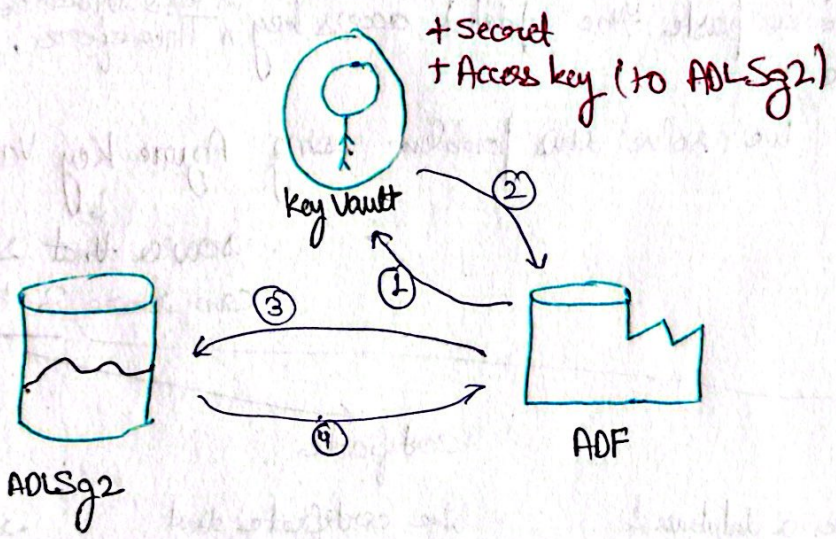
- When you create an AKV it got replicated for high availibility. (AKV2 - read only replica)

Scanned with CamScanner

Creating a key vault

- Go to Azure Portal → key vault → choose desired specifications (Resource group, subs., Name etc)

- we will add a secret to a key vault that will store our access key which gives full access rights to our storage account
- we will configure our linked service in ADF to connect to our key vault & retrieve the access keys.



By default AKV provides 2 access configurations (to connect to it)

- Azure role-based access controls (recommended)
- Vault access policy

- in this lecture we select vault access policy

- creating secrets: to store access keys as secrets

Objects → Secrets → Generate/Import → Name
Sec. Value (here you copy & paste access keys) from ADLSg2 storage acc
↓
Create

- but the thing is how ADF will connect to key vault ?

So we have something kla Managed Identity

- users identities are stored in AzureAD. Likewise resources in Azure has some identity associated with them, managed by Azure which they can use to connect with other resources.
- we do not have to provide any passwords as it is managed by Azure

a) System Assigned MI

- for every instance of service, it will have its own unique identity (1:1) lifecycle is
- Highly coupled means if you have an ADF & when you remove it, its identity will also get removed.

Go to data factory service → Settings → Managed identities → System assigned / User assigned

* name of identity will be same as the name of the resource.
  or Object(principal) ID : 869d3c89.....-c407-4b...-ec...de

b) User assigned MI

- we manually create identities & then assign them to resources. Identity bano ⇒ resources ko dedo
- we could use same identity & assign it to multiple resources. (1:N)
- own lifecycle - for eg if you remove ADF then maybe identity remove na ho kyunki hoskta hai koi doosra resource uss identity ko use kar rha ho.

Scanned with CamScanner

So, to grant access ~~to~~ for Key Vault to ADF, we will configure linked service inside data factory to a key vault to use managed identity. And then inside a key vault we will grant permissions to read secrets.

How to grant access to Data Factory Managed Identity:

Go to key Vault → Access policies → + Create

- you create a Key-Vault
- then create Policy
- inside Policy you give permissions to Principals (resources)
- you can also make a group put all Principals in it & in Policy, give permission to the group.

Key permissions Secret permissions Certificate perm

✓ Get
✓ List

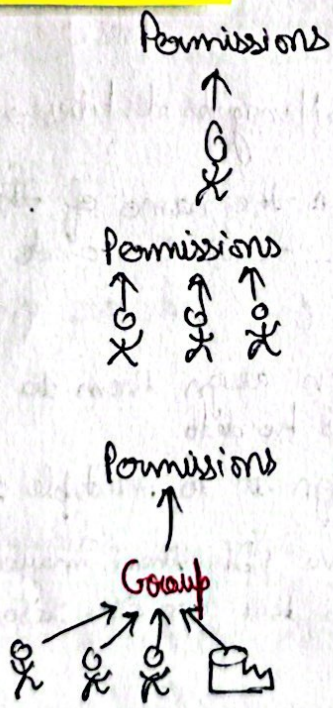
↓

Next

↓

Principals (choose to whom you want to grant access)

- granting access to some user/identity manually is not best practice from management pov.
- later on you will face a scenario where if you grant permission to an employee, same permission will be requested by another employees. So putting all the values in Principals would be hectic. So instead of granting permissions directly to a user we can create a group inside Azure AD then add users/principal to that group as a whole.



- To create group inside Azure AD / Microsoft Entra ID

Go to Microsoft Entra ID → Manage → Groups

Group type
Group name
Members: here add managed identity for ADF
Owner
→ Create

Scanned with CamScanner

Now again go back to Azure key Vault and set Principal = Group you just created in Azure AD (containing MI of our ADF)

Verification:

Go to ADF → Linked Service → Storage account key | Azure key vault

↓

AKV linked service

+New

Name

Azure subscription

Azure key vault name

Authentication method

✓ Sym. Ass. MI

User Ass. MI

linked service to key vault

↓

Secret name

(Secrets stored in key-vault

↓

Secret version

Latest version

* still a bad design so-far because you are still using Access keys.
* if you rotate keys, make sure to go back to Key Vault → Secrets

↓

Update (+ New Version

* secret is just a string so we do not know who is trying to connect because it doesn't contain any identity.
* give full access to whole Storage Account.

Scanned with CamScanner

Lecture 19

Azure data lake security - Shared Access Signature (SAS)

- In case of Access keys, it was full access i.e full access to Storage Account (Blob, Table, Queue, File Shares)
- SAS is more granular

There are 3 types of SAS token:

a) Account level:

- Go to Storage Account
- from left under Security + networking choose "Shared access signature".

here we can be much more specific as compared to access key

Allowed services
Blob File Queue Table

Allowed resource types
Service Container object

Allowed permissions
Read Write Delete List Add Create Update

* SAS token validity - best if you keep it low.
* you gotta choose one of the access key to generate SAS token / to authorize SAS token
* if access key authentication is disabled at account level, then you won't be able to generate SAS token since it requires one of the keys to authorize SAS token

- Generate SAS & connection string after choosing signing key

Scanned with CamScanner

you will see SAS token is just a string

How to use SAS token in Data Factory to connect to our Data Lake

- copy "Blob service SAS URL" (from your source) → 2 parts : path to data lake, SAS token

- Go to ADF → Create new LS → Azure Data Lake Gen2
  - Name
  - Authentication type - SAS URI
  - SAS URL ← blob to dfs.
  - ↓
  - Test connection

Inside Data Lake Storage Account you will see different end-points for Blob, Queue, File & Table service.

* This will be failed. Why? Data Lake is built on top of Blob Storage but it uses different end-point

Go to Storage Acc.
↓
Endpoints
↓
Data Lake Storage Primary endpoint

So, get the endpoint for Data Lake Storage

⇒ you need Data Lake endpoint in SAS token URL

* Features :
  - can be set for one or more services
  - granular

* Suppose if you realise that SAS token has been compromised & you want that it should not work. You can simply do that by invalidating the key you used to sign/generate your SAS token; by rotating the key

* Suppose you are interested to grant permissions at container level or maybe to ~~the~~ some specific directory or file, you cannot do that using SAS token.

Scanned with CamScanner

b) Service Level

- single service ∴ more granular
- can grant permissions even to specific files/blobs
- one token to one service

How to do that?

Go to container → go to any directory → go to a file

Then on the right hand side of file tap on ... → Generate SAS

↓

Signing method
o Account key o User delegation key

Signing key
Stored access policy
Permissions

↓

Generate SAS token & URL

your SAS token will just be a string which would be composed by taking all these things →

⇓

all this is in form of an URL

→ if you cut just the SAS token part & append that token to some other file in same directory then you won't be able to open the file.

→ you can actually decipher each component as per documentation.

→ based on all the components, there would be a unique signature value at the end part of our SAS token.

→ if token is compromised you could rotatate access keys but it is not an idle option because maybe some other SAS token is using the access key.

So we create Stored Access Policy

Go to desired container → Access policy → Add policy

Identifier Permissions
Start time Expiry time

Save your changes

Scanned with CamScanner

· so far you just defined a policy. Now come back again to #

... → Generate SAS

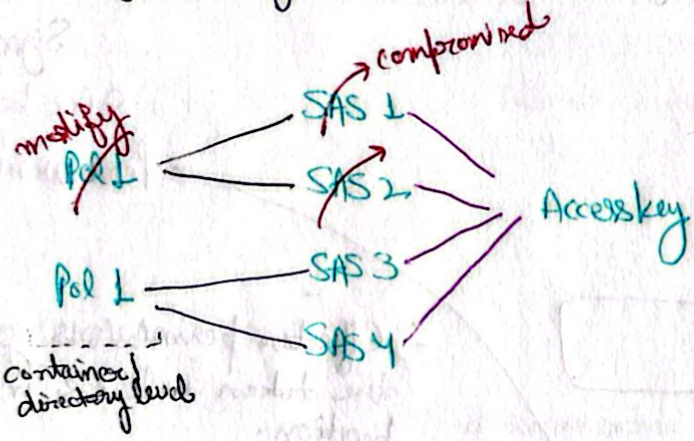
Signing method

Stored access policy (choose your recently created policy here)

∴ Now it is easy to invalidate compromised SAS tokens that are created on the policy

⇒ You creat "Access Policy" & then pass it while creating Service SAS token.

→ We can use same Policy with multiple SAS token & an SAS token can only use exactly 1 Policy



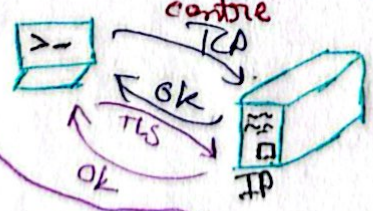
if we did not use Policies we would have to rotate Access key & inturn would invalidate all tokens. So better modify Policy

* Suppose you want to get access to some file using SAS token. So you provide URL like:

https://datalakename.blob.file.....\ ?SAS_token_part

this part has to be converted/translated into some IP. DNS Server is used. token part is not send for translation becoz only host name is important

Ex: the IP points to some m/c at Microsoft Data centre



So your browser sends connectn to this IP address to establish TCP connectn. SAS token not sent yet.

Not step TLS 1.2 connectn. After this token is now sent.

Since your data is being secure connection

∴ This is secure

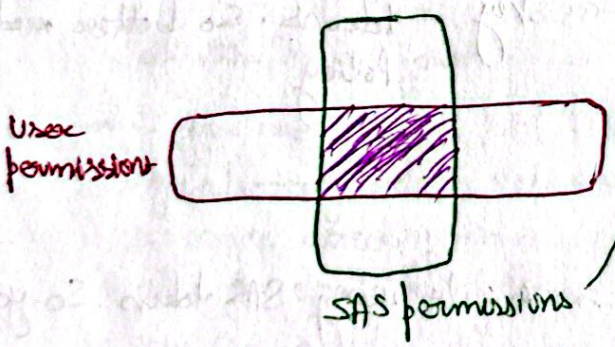
• this is still flawed because of the use of Access keys

c) User delegated

- similar to service SAS & created at same place
- Stored access policy not supported
- no access keys
- can be used only in case of Blob & ADLS services.

How to do that?

Go to container → go to any directory → go to a file → ... → Generate SAS
↓
Signing method
○ Acc key ○ User del. key



→ Permissions.

∴ Effective permissions granted by the token will be intersection portion.

→ if SAS token is compromised as you do not have access keys you cannot rotate keys. So, either disable User permissions, invalidate User del. key

→ no option in UI so use an API

Lecture 20: Azure data lake security - Role-Based Access Control (RBAC)

B) With Identity

Roles: Consists of set of some permissions. Perm1, Perm2, Perm3.....

Based on permissions we have mainly 3 roles

Owner - can grant permissions
Contributor - can do everything that owner can do but cannot assign permissions
Reader - just access

→ Go-to Home → Access control (IAM) → Roles

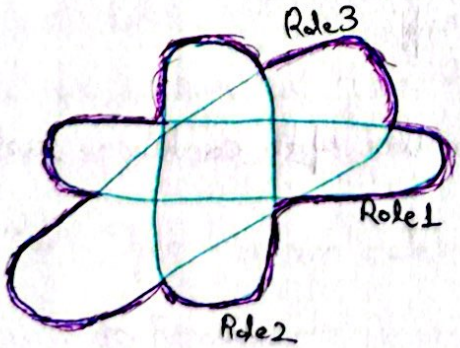
you will see list of roles.

Name Description Type Category Details

View
↓
JSON

here you can see list of actions one can perform

But sometimes you want your own customized role, inside which you could choose your own set of permissions. In this case you can create json file & define action & notaction



- In this scenario if you are assigned Role1, Role2 & Role3, your effective permissions will be the union of all the permissions unlike where in User delegated method where effective permissions will be intersection of User permission & SAS token permission.

Scanned with CamScanner

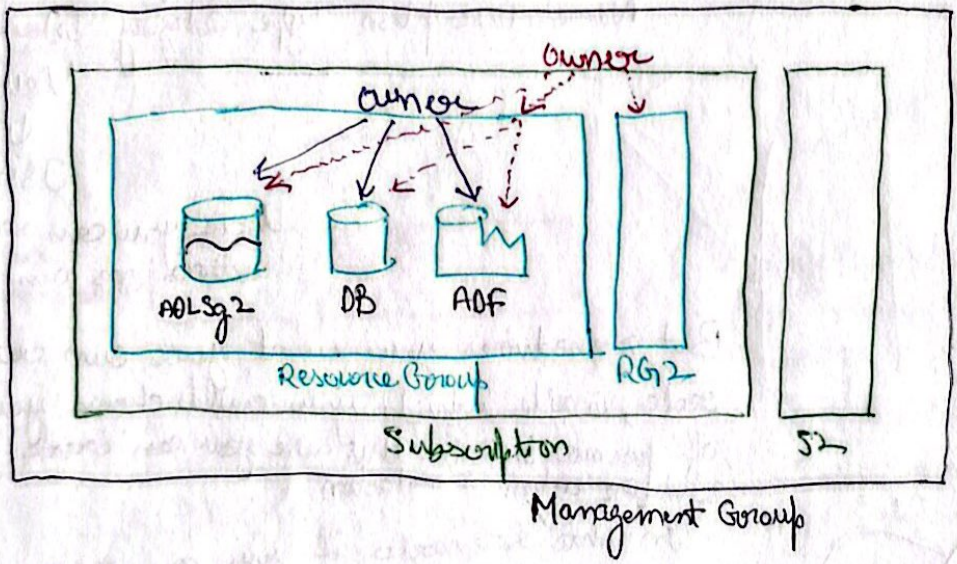
* Role vs Security Principals vs Scope → on which resource to grant access

Role → what we have to grant

Security Principals → to who we have to grant ~~permissions~~ role

Could be MS EntraID user
Could be group
Could be Managed Identity — this is basically identity of a service on Azure.
Could be Service principal → identity for an Application

Scope
- hierarchy of resources in Azure
- ADLSg2, DB, ADF → in logic container (Resource Group)
- Multiple RG in Subs.
- we often organize subs in Management Group.
- permissions are inherited (RBAC)



- in case of Roles (RBAC) only granular upto container level. (not specific files.

How you assign Roles:

Home → at Resource Group level → Access control (IAM) → Role assignments
↓
+Add
↓
Job function roles | Privileged administrator roles

Role	Read Blob	Write Blob
Owner		
Contributor		
Reader		

- suppose you assign a Contributor role (inside Privileged administrator roles) to a Resource Group (Scope) & you also specify to who you grant access, you can add members

↓

Now suppose ab wo member storage acc. me jata hai jisko contributor role mila hai. Wo container mein content jo bhi hai wo dekh sakta hai

↓

But agar aap Authen-type dekhoge wo = Access key hoga. Matlab RBAC use hua hi nhi.

↓

Par in reality mein na aap data read karpaoge na hi modify.

KYU?

Control Plane vs Data Plane	
• to manage & config. our resources. ex in case of Data Lake, do we want to allow anonomous access, access key use ~~ko~~ ~~rahi~~ karni hai ya nhi	• to access data

Purpose of this split is DBA is responsible for conf. of resources making sure it is accessible ~~&~~ maybe we donot want to give access to the actual data.

And maybe hum kisi Data Engg. ko data ka access de de par ~~nai~~ config. modify karne ki (Storage Acc.) permission na de.

∴ By default Owner, Contributor & Reader do not have permission to read data. These role are in context of Control Plane.

Yeh Split kahan dekhein?

Go to Resource → Access control (IAM) → Roles

↓

Contributor (View)

↓

○ Actions ○ Data Actions

↓ ↓

Control Plane Data Plane

Scanned with CamScanner

* Roles for Data Plane :

Storage Blob Data Owner
" " " Contributor
" " " Reader

Owner is so powerful that it can assign itself this role

Roles	Blob Reader	Blob Writer
Owner	X	X
Contributor	X	X
Reader	X	X
S.B.D Owner		
" Contributor	✓	✓
" Reader	✓	X

preferrable in read & write data

RBAC for Datalake

· Go to containers → desired container → change Authentication type at top to (Switch to Microsoft Entra user account)

Group : Azure Entra ID → Groups → New group
↓
add Owners
add Members (add MI of ADF)

RBAC for ADF : we know that ADF has its own managed identity that can get access to data lake.
So assign S.B.D Contributor to mi or create a group in Azure AD, then add MI in Azure AD group

Finally : Go to Storage Acc. → Access Control (IAM) → Add → Add role assignment
↓
Job func. roles → Stor.B.D Cont. → Members
(Add the created group)

Scanned with CamScanner

from here now if you create a Linked Service to our Data lake

ADF → Linked Service → ADLSg2
↓
Authentication type : here now switch to Sys. Assigned MI

Storage acc. name : here you can indicate which data lake you want to connect.

∴ here you do not need to use Key vault, no tension of tokens, no tension of expired secrets.

So far, so good but..... what about granularity?.... grant access to a single directory? single file?

Lecture 21

Access Control Lists (ACL)

- with no identity, you can grant granular level access in case of SAS.
- ACL is similar ⇒ but do not uses token
  we know who is going to connect
- in case of Blob, ACL does not work. Because directories that you create in Blob is imaginary

Files

Read (R) : if you want to grant read access i.e someone can read a file

Write (W): if you want to write on file

Execute (X) : Does not mean anything in the context of Data Lake Storage Gen2

Scanned with CamScanner

Note: always include Execute permission (X)

Directories

Read (R) : Requires Read & Execute

Write (W) : Requires Write & Execute

Role of Execute (X) - Required to traverse the child items of a directory

Containers

directory

raw X

└ SAP X

└ SALES X

files | Blob

data.csv ← R

d.csv ...

Suppose you want to grant Read ACL. But this file inside Data Lake has some hierarchy. Therefore, the user would have to traverse the whole hierarchy so we grant Execute permission to make this process happen.

∴ Execute permission to SALES, SAP & raw

File level

How to get ACL:

... → Manage ACL → Access permissions

+ Add principal + Add mask

add desired user

Directory level

SAP → ... → Manage ACL → Access permissions | Default permissions

└ SALES

this applies to the current scope i.e SAP directory

- files in a directory may have different ACLs & their parent directory may have different set of ACLs & the parent of directory i.e container may have different set of ACLs. And those ACLs may not be inherited by its children.

* every objects has ACLs & they are not inherited by default. So it makes managment bit difficult. But you may want that your children inherits same permissions as parent. So you define this in Default permission. (set of ACLs that you want to be granted automatically to every new child of the parent)

Scanned with CamScanner

after adding a user using Default permissions in eg. SAP if you go to SALES you would see that the user will not have any access

SAP
- SALES
- HR → this time ACL would be applied

Meaning: Under Manage ACL you will see 2 options viz:
Access permissions | Default permission
↓
you configure this part & the permissions you set here will automatically pass on to the new children

*Note: Default permission is applied only to new objects

In reality you will have many files in Data Lake that are already present. So, how to apply ACLs when you know that Default permissions work for newly created files.

Microsoft Azure Storage Explorer (faster)
you can propogate ACLs using it.

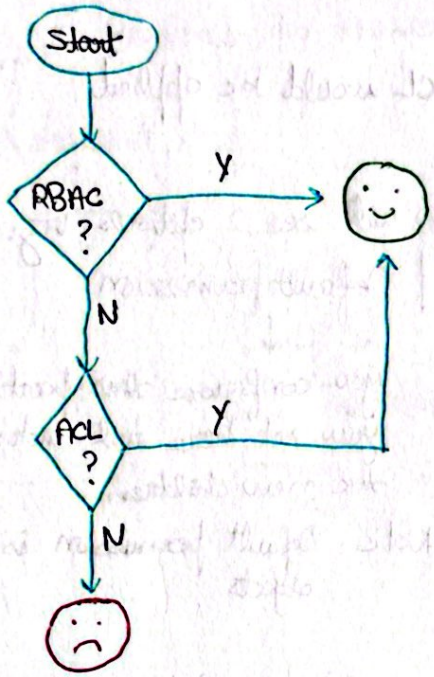
- from tab at left go to your container → Right click → Manage Access Control list

you will see Access & Default option will be on same tab.
↓
configure desired settings

when this is done again go back to your container → Right click
↓
Propogate Access Control list

Scanned with CamScanner

ACL coexist with RBAC



Flow:

when someone tries to enter or ask for permissions, we see if they have permissions using RBAC if yes then access is granted to him.

If no then it is double-check if he has sufficient permissions via ACLs if yes then he is granted access if not then we will be denied to get access

SCENARIOs

Case	Anonymous	Access keys	Account SAS	Service SAS	User SAS	RBAC	ACL
R/W to a specific Container only	X	X	X	✓	✓	✓ (preferable)	✓
R/W to a specific Directory only	X	X	X	✓	✓	X	✓ (preferable)

∵ give power to whole Storage Acc.

Combination of RBAC & ACL is always preferred for Data Lake security

Scanned with CamScanner

Lec. 22

ARM-based CI/CD for Azure Data Factory
(Azure Resource Manager)

There are 2 approaches : either you have Git configured
or Git not configured

Git configured - prefer in Dev. Env.
Not configured - " " Prod. Env.

Go to Manage → Source control → Git configuration → Git repository

Ideally it is preferrable to have Git repository configured, because you can then see different changes, different versions of code & if you want to revert your changes, you can go back to older version of codes. Also, you can collaborate with different teams while working on a similar piece of task.

Dev. instance of ADF — Git configured } Best practice
Prod. " " " — Not configured.

Branch : where developers push their codes that can be trusted.

Azure DevOps : used as code repository.

Pull Request : Request made by a developer when he is done debugging & testing his code in separate branch, to get merged with main branch, so that his code is utilized & deployed at various environment

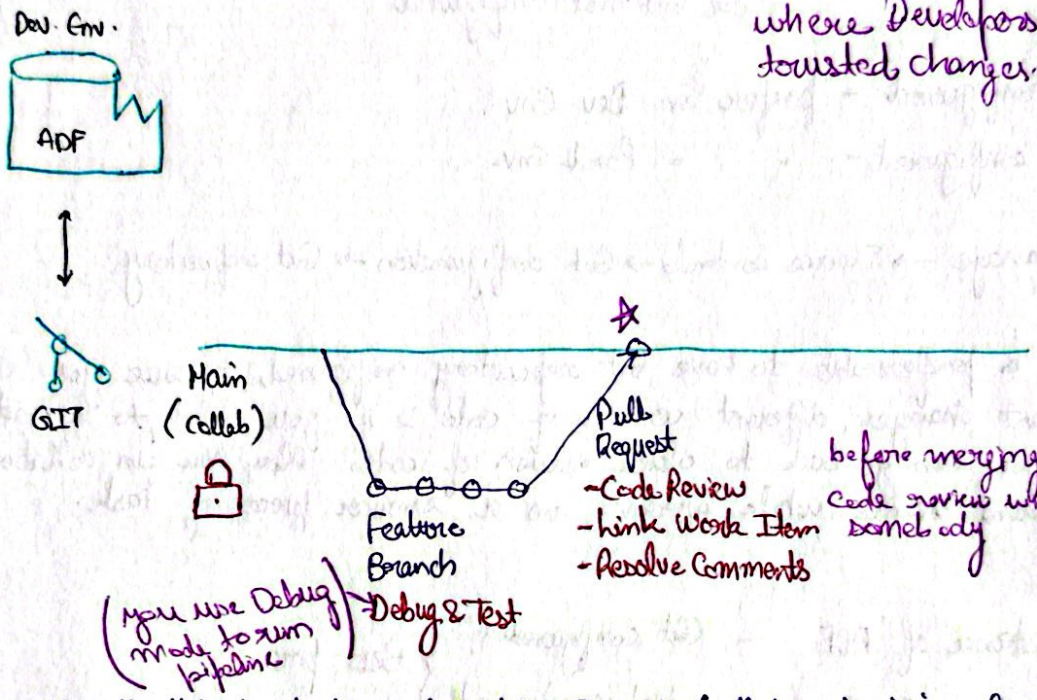
Code Review : It is done during Pull req. Here we ask other developers to check what changes have been made at a branch (feature branch in our case) & make sure all patterns, properties & specifications are met.

Link Work Item : Azure DevOps can be used. We could force linking every pull request to some work item. ADev. can also be used for project tracking. Here we can link pull req. to some task or story.

Scanned with CamScanner

Resolve comments: all comments inside Pull req. must be resolved. Based on policy defined on our repository a merge will not happen if we do not resolve comments.

Collaboration Branch is where Developers push their tested changes.



before merging to Main Branch code review will be done by somebody

Along with all this, protection of Main Branch (collaboration) is also very important

- Once Pull Req. is approved & branch it merged to Main Branch, CI/CD process gets triggered that we define in our repository. ⇒ deployment package is built; means our pipeline will grab the code that existed in Main Branch at the time Pull Req. was completed & it will create a deployment package (ARM template is case of Azure in form of json with some parameters) ⇒ this package is then deployed to Live Dev ADF.

when you select Git integration mode in ADF you will have 2 versions: ① shows code repository
② live version shows the stuff actually deployed in ADF

∴ every change that has been merged to the main branch will be automatically deployed to Dev. env.

means Dev. env. is in syn. with Main (collab) branch

Scanned with CamScanner

- you can also define Base on option, here you chose Main Branch which means when you create a New Feature Branch, on UI you will see all contents of Main Branch.

⇓

you create a New pipeline → you add activities

when you hit "Save" it translates to a commit ⇒ commit ID

⇓

Debug (to verify if everything is fine)

⇓

Then you create a "Pull Req."

⇓

After this step go to "Main Branch" & "Switch to live mode" to verify that your pipeline is deployed & is available.

→ Redirect to Azure DevOps

you can add Reviewers
work items to link
Tags

↓

Create

You can see changes made in form of json inside "Files"

↓

You Approve the Pull Req.

↓

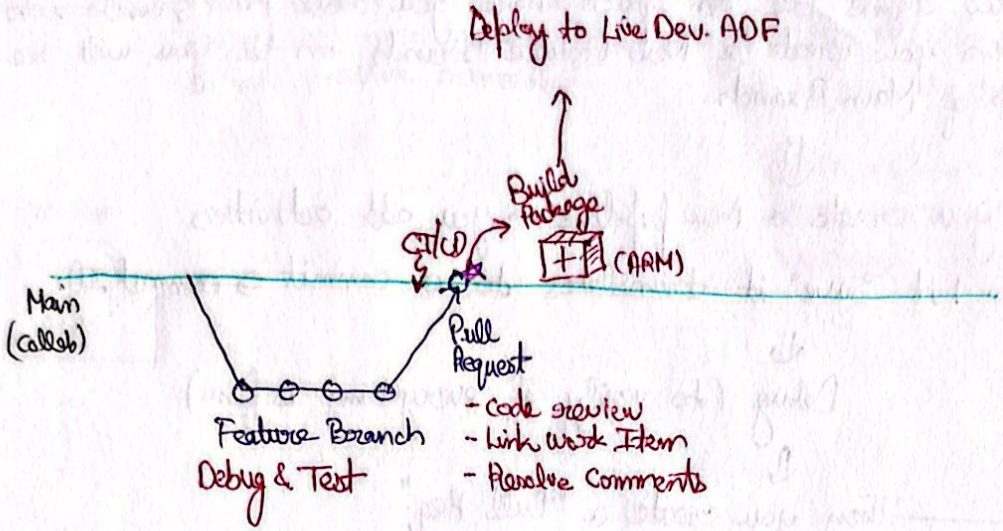
You go to 'Set auto-complete' & hit complete & then Override branch policies.

* After this you are at ☆ stage. After this CI/CD process should be initiated

You can view the artifacts in "Pipelines" in Azure DevOps.

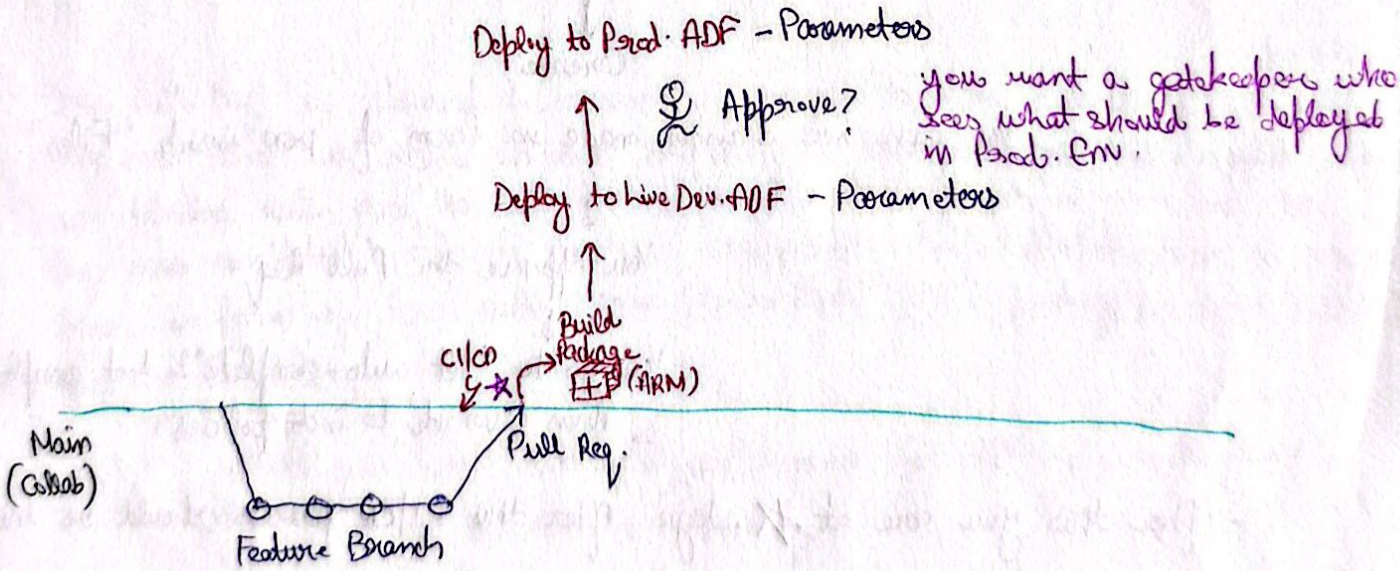
Now, if you go and check Live Mode you will see your Pipeline.

Scanned with CamScanner



Once changes are done, then grab same package in Dev. Env. and deploy to different env. (eg. Prod). In this process we are making sure we are deploying same code that was deployed previously.

Parameterization of packages must be done: eg. linked Services in Dev. env. will point to different Datalake than on Prod. So, by parameterization we make sure these differences are valid. This is the only change done b/w the environments.



Basically you create a Feature Branch first by going into Dev. instance of ADF. Change to Azure DevOps Env. → +New branch

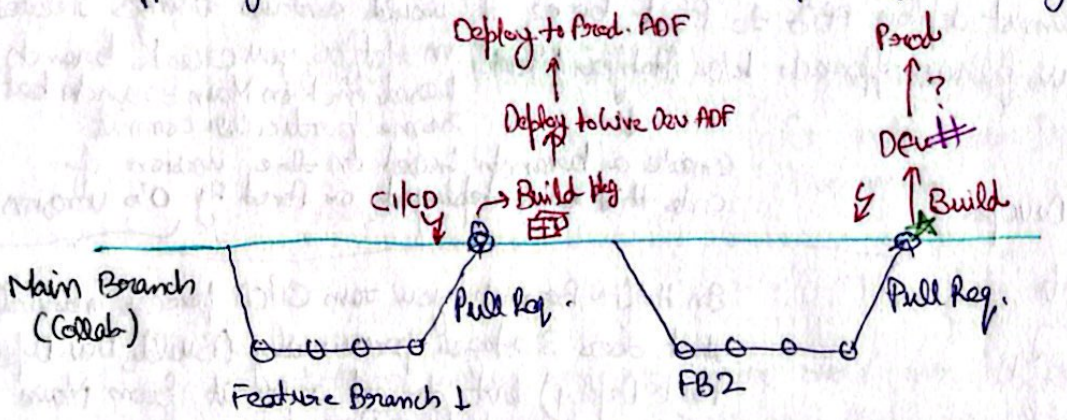
So far all the changes are at Dev. env. but now you want to deploy it to Prod. env.

Go to Azure DevOps → Pipeline → Stages → Review → Reject/Approve
Build Dev. Prod.

After this go back to your Prod. Env. to verify if your changes are reflected there or not.

Feature 2

Suppose you want to add some more options/functionality.



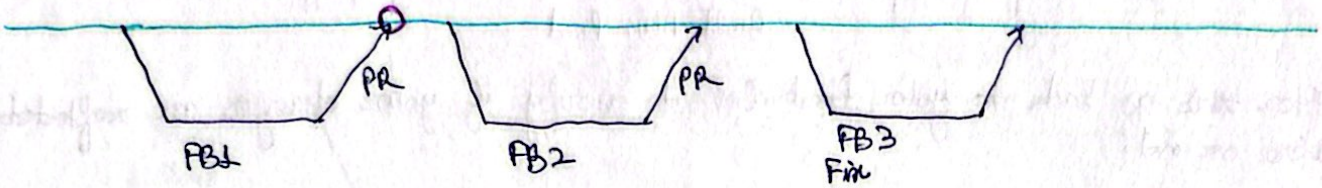
Go to Dev instance of DataFactory → Switch to DevOps → Create a New Branch based on Main Branch ∵ it is the version that contains changes for us from various Developers

Suppose you add some additional functionality in FB2 branch by editing FB1 you then create a Pull Req., Review the code, resolve some comments if necessary, approve & complete the req. Therefore, you merge to the Main Branch ☆ you are now at ☆ position.

Now suppose you are at # therefore you have not yet deploy changes to Prod. So, in Prod we will only be having version of FB1. And if you have bug in Prod somewhere in FB1 branch you attempt to fix it by creating new FB (only if bug is not critical.)

Scanned with CamScanner

Same process that you were doing : creating a new branch based on the Main Branch.



If bug is critical :

you cannot create a new branch based on Main Branch.

Therefore you cannot deploy FB3 to Prod. becoz it would contain things related to FB2. So you follow approach k/a Hotfix Flow

in Hotfix you create branch based not on Main Branch but some particular commit.

↓

Create a branch based on the version of code that was deployed on Prod eg. 0's version

Go to Azure DevOps

↓

Environments

Dev | Prod

↓

Deployments

↓

Click on Pipeline that deployed Feature 1

↓

Copy commit ID

In Hotfix Branch you run CI/CD process manually that does 3 stages previously (Build, Dev. Deploy, Prod. Deploy) but do not run it from Main Branch but Hotfix Branch.

↓

After that merge to Main Branch

Hotfix Branch : ADF → Azure DevOps Git → New Branch

(in case of ADF UI there is not any option where you can paste commit ID)

∴ to create branch based on a specific commit, you use Azure DevOps

↓

Branches

↓

New branch

↓

Based ON (Paste commit ID)

↓

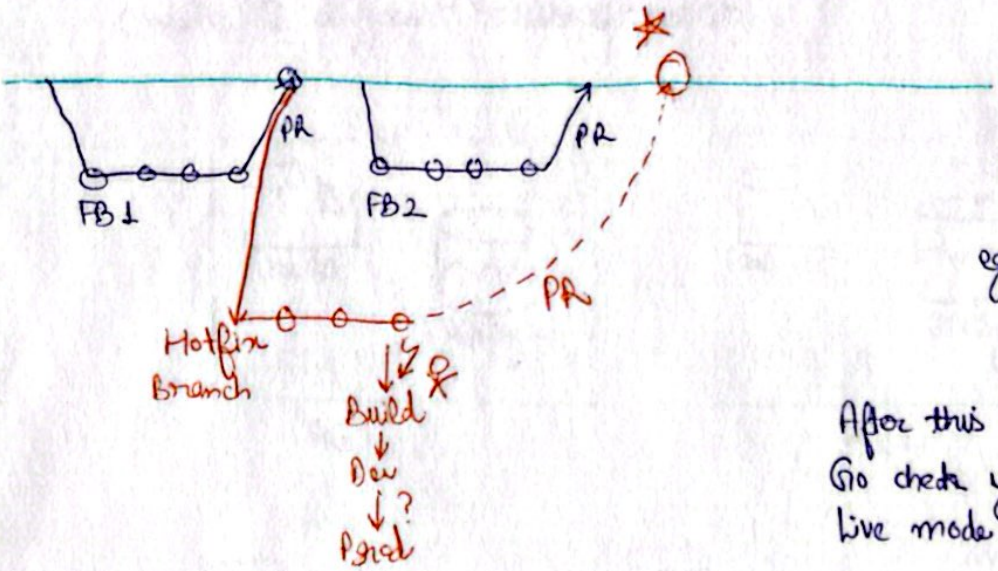
Create

Scanned with CamScanner

Uss pipeline main deploy hi sirf FB1 ka content hoga, FB2 ka content nhi hoga

Hotfix karlo ab par do not create Pull Req. now. Instead manually run CI/CD process

↓



Go to Azure DevOps
↓
Pipelines
↓
Run pipeline
↓
Indicate Branch (kahan se run honi chahiye, kis code version se)
eg: Hotfix
↓
Run

After this Hotfix will deploy to Dev. Env.
Go check your Dev. env by going to Live mode
↓
Then push to Prod. env.

Next thing is we have to make sure our Hotfix is not overwritten in future deployment ∴ create pull req. to Main Branch

Go to Dev instance of ADF
↓
Hotfix Branch
↓
Create a Pull Req.

★ will have feature 1, feature 2 & hotfix changes.

Parameters are different for Linked Services in Dev. & Prod env.

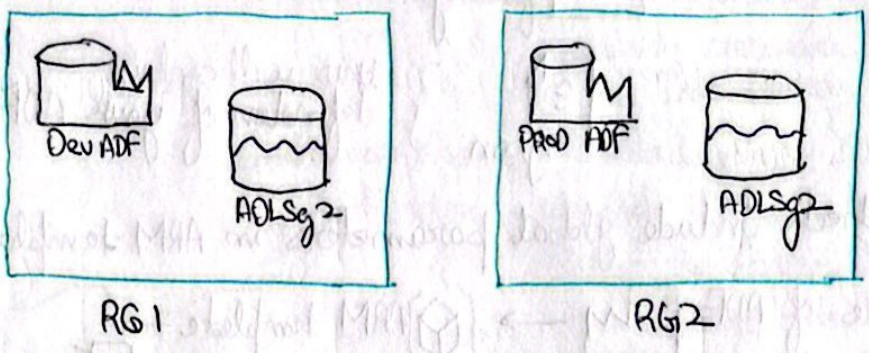
How to parameterized our Deployments?

Scanned with CamScanner

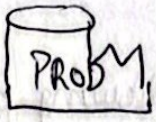
Lecture 23 : ARM-based CI/CD for Azure Data Factory

- parameterization

- create 2 different Resource Groups



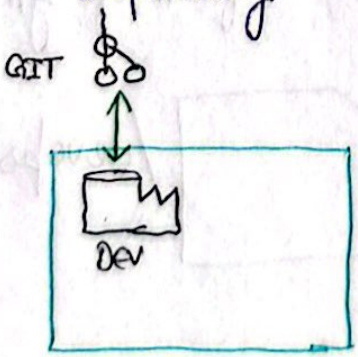
Manage → Git configuration
you will find it empty



Manage → Git configuration
this would be empty as well.
Leave it as it is.

Azure DevOps
- create a new repository ☆
at this moment this will be empty, no ~~pipe~~ pipelines, no environments etc.

Step 1 : Link your Git Repository with Dev. Env.
such that whatever changes you make will be saved in the repository



Go to Dev instance of ADF → Manage → Git configuration
↓
configure
↓
Repository type* ☆
Azure DevOps Git
GitHub

Azure Active Directory
↓
Continue
↑
organization name
↑
Project name
↑
Repo. name (that you created inside Azure DevOps) ☆
↑
Root folder (/data-factory/)

After this you will see on Dev instance of ADF's UI information about Git Repository.

Inside Azure DevOps:

Repos → Newly created repo. → data-factory

factory

.json } this will contain defination of your ADF

Step2: Make sure to check Include global parameters in ARM template

Go to Dev. instance of ADF → ARM template → ☑ Include global parameters in ARM Template

Step3: In Azure DevOps create following files:

Go to Azure DevOps → Repos → Newly created repo → ⋮ → +New

package.json

adf-build-job.yml

/devops/

this files contain variables, jobs, task etc.

Step4: Create a new pipeline in AzureDevOps in form of yml file

in this file, replace temp variables with your env variables

/devops/adf-azure-pipelines.yml

/devops/

Scanned with CamScanner

- provide path to your adf-build-job.yml file in template
  - template : adf-build-job.yml
    - subscriptionId : <subscription_id>
      - your subscription where you want the deployment to happen
      - you will find it inside your Resource Group on Azure
    - resourceGroupName : <resource_group_name>
      - which resource group contains Dev. instance of Data Factory
    - dataFactoryName : <data_factory_name>
    - repoRootFolder : <absolute_path_to_datafactory_folder>
      - folder on Azure DevOps, under newly created Repo., that contain data-factory objects. (defination of data-factory)

    - packageJsonFolder : <absolute_path_to_package.json>

package.json

After this create a Pipeline on Azure DevOps:

Azure DevOps → Pipelines → Create Pipeline

Where is your code?

→ choose "Azure Repos Git" option

Select a repository

→ select the repo. you created specifically

Configure your pipeline

→ choose "Existing Azure Pipelines YAML file"

↓

Select an existing YAML file

→ here provide path to your pipeline file

/devops/adf-azure-pipelines.yml

↓

Continue

Scanned with CamScanner

Step5: Run & test pipeline

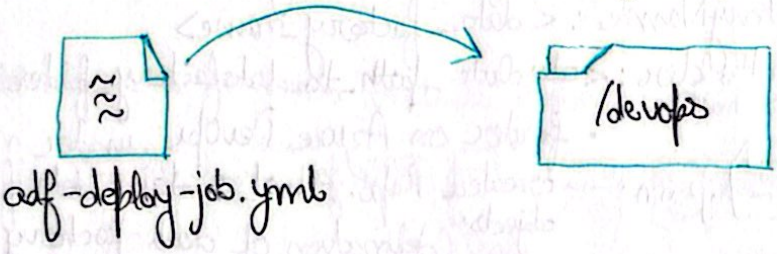
Azure DevOps → Pipeline → Run (this will build package)

So for built process is done
Next part is Deployment

2ND PHASE: (Deployement)

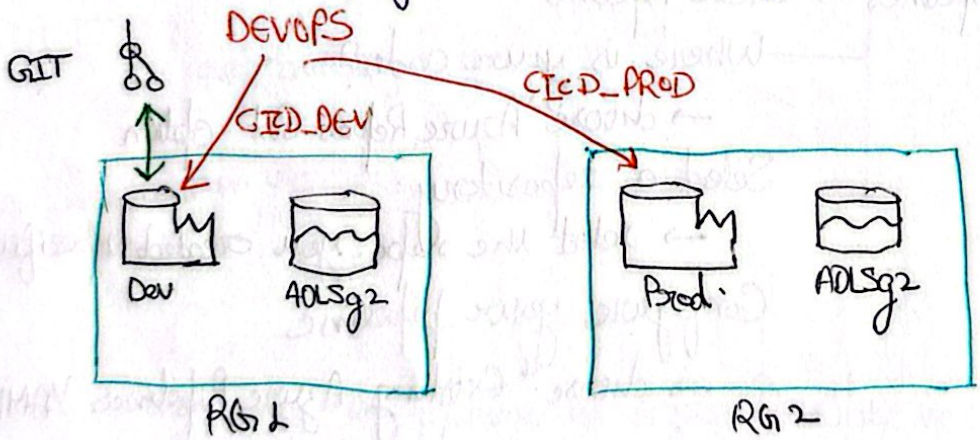
Step6:

- In /devops/ folder create adf-deploy-job.yml



Step7: Add Service connections

- Mostly you ask Administrator for name of Service connection that you may use
- it is something that Azure DevOps will use to connect to ADF in Dev. & Prod. env. to deploy various objects required for Datafactory.



- we will create 2 connections: CICD_DEV
  CICD_PROD

Scanned with CamScanner

- Go to Microsoft Entra ID

Basically we are creating a Service Principal

- Go to App registration
- + New registration

Register an application
Name CICD_DEV

Register

till this Service Principal is being created.

- Go to the recently created Service Principal ⇒ CICD_DEV
  - ↓ Certificates & secrets
  - ↓ New client secret
  - ↓ Add a client secret
    - Description :
    - Expires :

once it is created, copy the value.

- Likewise create Service Principal for Prod as well with the name CICD_PROD

Now assign both service principal role: Contributor

Go to RG → Access control (IAM) → + Add
- ↓ Add role assignment
- ↓ Privileged administrator roles
- ↓ Contributor
- ↓ Next
- ↓ + Select member
  - CICD_DEV

Likewise, do similar with CICD_PROD Principal by going to the RG where Prod. ADF is deployed

Scanned with CamScanner

Inside AzureDevOps : tell it which service principal to use.

AzureDevOps → Project settings → Service connections → Create service connection
↓
choose "Azure Resource Manager"
↓
Authentication Method : Service principal (manual)
↓
Environment
Subscription ID : give ID of Dev. env RG

Subscription Name :

Authentication
Service principal Id : Application ID
CICD_DEV

Service principal key : value you copied before

Tenant ID :
Directory (tenant) ID

Give name to it

Do similar thing for Prod. env.

Step8: Create Azure DevOps Environments

Create 2 separate environments

Azure DevOps → Pipelines → Environments → Create environment
New environment
Name :
↓
create

• Now go inside PROD environment → Approvals & checks → Approvals → Approvers
add any member
↓
create

Scanned with CamScanner

- Assign Storage Blob Data Contributor role to Dev ADF & Prod. ADF on Dev. Datalake & Prod. Datalake

Step 9: Create Variable Groups

Azure DevOps → Pipelines → Library → + Variable group

Variable group name : DEV

Description:

+ Add

Name	value
ADLSg2_LS_properties_typeProperties_url	URL from your Dev. instance

do similar for Prod

this will help in parameterize deployements between Dev. & Prod.

Step 10: Add deployment Stage

edit your adf-azure-pipelines.yml

adf-azure-pipelines.yml

```
- stage : <stage_name> = DEV
  variables:
  - group : <variable_group_name> = DEV
  jobs
  - template : <path_to_adf-deploy-job.yml_file> = adf-deploy-job.yml
    parameters
      environmentName : <environment_name> = DEV
      subscriptionId : <subscription_id>
      resourceGroupName : <resource_group_name>
      dataFactoryName : <data_factory_name>
      serviceConnectionName : <service_connection_name>
      overrideParameters : >
          -<param_1> <val_1>
```

from DEV instance of Subs. & RG

→ CICD_DEV (that you created in Step 9)

→ $(that you defined in DEV variable group)

- go to your running pipeline
- open Artifacts
- open "ARMTemplateParametersForFactory.json"

Scanned with CamScanner

- to same thing for PROD env.
  create separate stage within adf-azure-pipelines.yml
- ~~If~~ for the first time you have to Approve usage of following things:
  allow Pipeline to use DEV env.
  allow Pipeline to use Service Connection for CICD_DEV
  " " " " Variable Group for DEV

- to verify if the Pipeline deployed to Dev. Env.
  Go to Live Dev. mode

- for the first time you have to Approve usage of same things for PROD env. as well

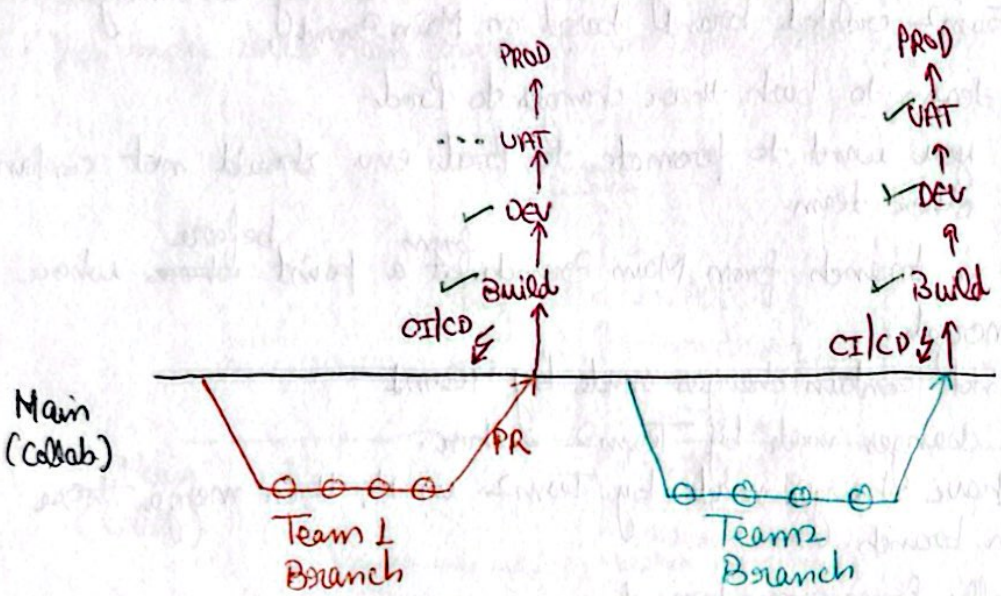
  Review → Approve
  to verify if the Pipeline was deployed to Prod. Env.
  Switch to Prod. Env.

Disadvantage:
- This approach deployes everything
- What if you want to deploy just subset of ADF like just few pipelines?

Scanned with CamScanner

Lecture: 24

Selective CI/CD for Azure Data Factory with ADF Tools



- Suppose in case there are more than one team working on a project and Team1 is working on some features and after they are done with debugging, they make Pull Req.. It will start CICD process which envolves building of package & then changes are deployed to Dev. env.
  (ARM templet containing defination of Data Factory)
- Let's say after approvals the changes are deployed to UAT Env. where business users will test those new functions & if everything went well the changes will be finally to Prod.
- Suppose at UAT Env., the changes that you made were too big that it will take considerable amount of time for Business users to check, verify, test changes & hence the flow stops there & waited for approvals.
- Meantime if Team2 wish to use same instance of ADF to add some set of features. If they create their new feature branch from Main branch & finally after testing by developers it would be merge to Main Branch. After merge CICD will happen, it will build ARM template, then being deployed to Dev → then to UAT → finally to Prod.
  Changes made by Team2 were easier to verify & test.
  ∴ Business team finish their approval faster.

Scanned with CamScanner

But the question is can you approve it?

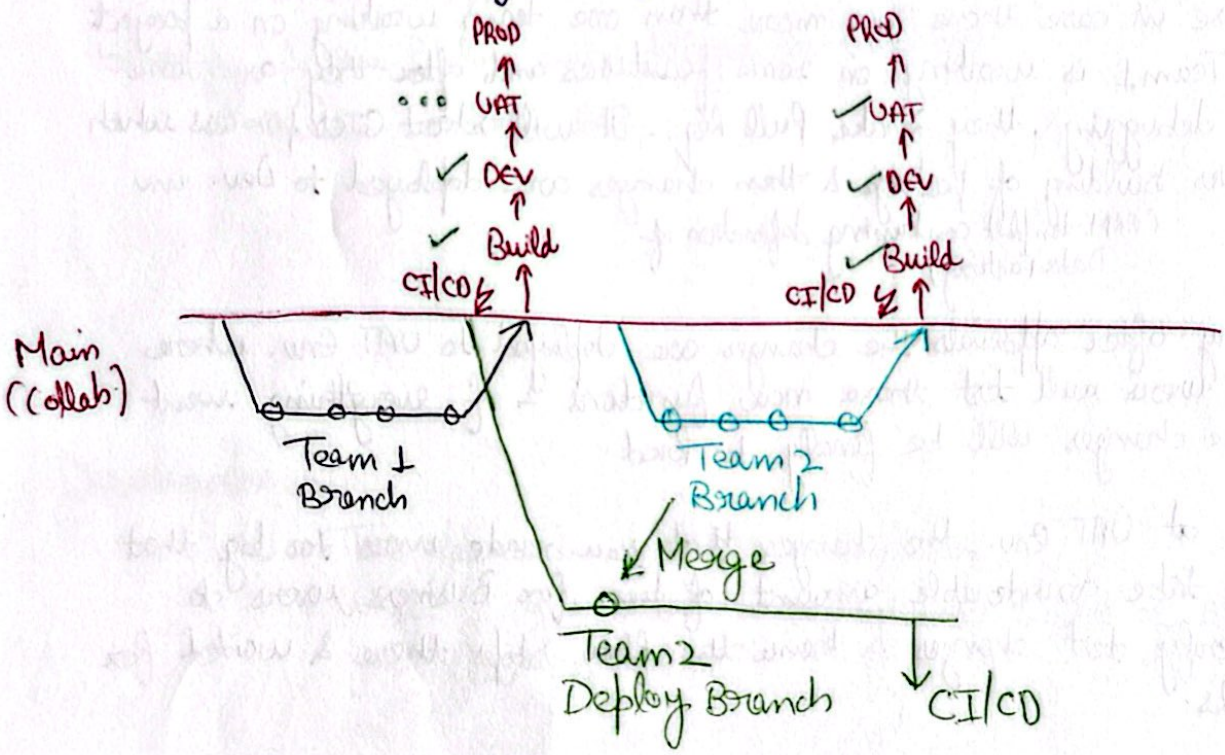
No, because the build ARM template that was create as result of pull request made by Team2 contains all those changes made by Team 1 because they also merged with main branch and Team2 created branch based on Main Branch.

∴ Team 1 is blocking all teams to push their changes to Prod.

∴ Make sure that changes you want to promote to Prod. env. should not contain changes made by any other team.

⇒ One option is to create a branch from Main Branch ~~at~~ from a point ~~where~~ before where Team 1 changes were merged.

- In this way it would not contain changes made by Team1
- Also it doesn't contain changes made by Team2 either.
- If you would like to have changes made by Team2 in it, then merge those changes in your green branch.
- Then run CI/CD manually from green branch

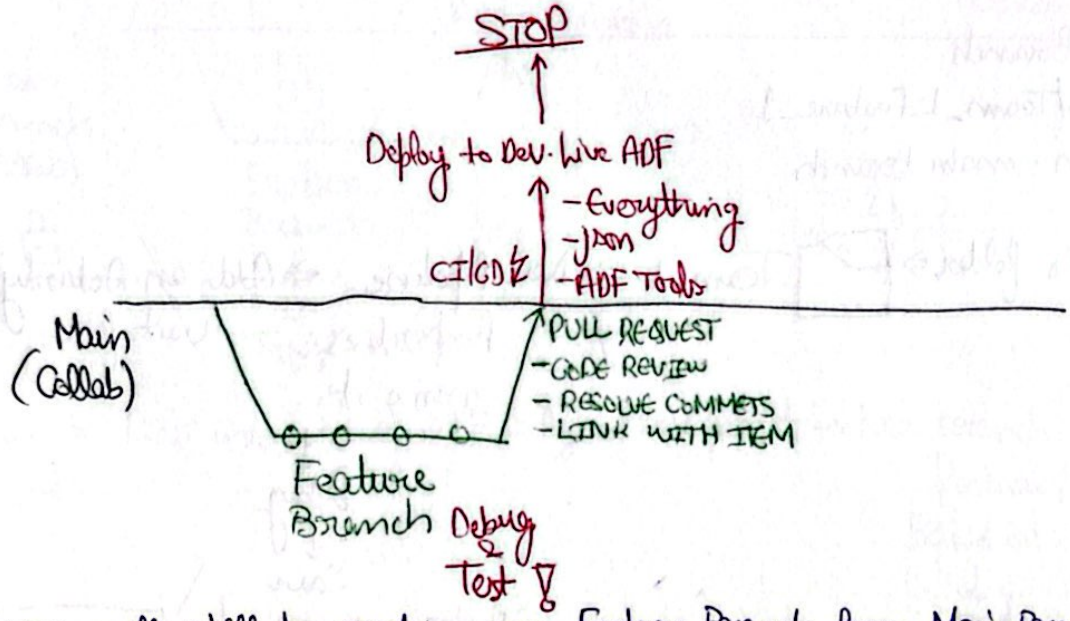


- ok for small team
- ok for small enterprises.

very complicated
TBH!

ADF Tools:

- In this process there is no process of building packages
- You make sure Main Branch is in sync. with Live Dev. Env.



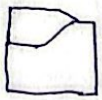
- you will still be creating your Feature Branch from Main Branch.
- when you are done with making changes, you debug & test your pipeline
- you then make a "Pull Request"
- after code review, resolving comments, you finally merge with main branch
- after this CICD will happen
- Securing Main Branch means before merging to Main Branch, code review, resolve of comments should happen first then only you can merge to Main Branch.
- In this our CICD will use ADF tools, that doesn't use ARM templets.
- As every objects in ADF stores as a json file, ADF tools will grab those json files & use API etc. to deploy them.
- you need not to worry about all dependencies all is taken care by ADF tools.
- pipeline directly deploy code to Dev. Live ADF & deploys everything (json)
- you deploy everything to Live Dev. Env. because you want your Main Branch to be sync. with it.

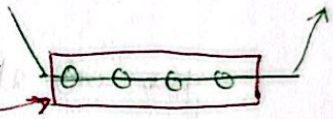
Scanned with CamScanner

Implementation:

- Go to Dev. ADF ⇒ you will see different objects (pipeline, datasets etc.) inside separate folders for different Teams
  ⇒ well organization of objects in dedicated folders for dedicated Teams is recommended to make Selective CI|CD work better.

- Create a New Branch
  feature/Team_1_Feature_1
  Based on : main branch

- Go to your team's folder ⇒  Team_1 ⇒ New Pipeline ⇒ Add an Activity
  Properties : name it
  wait for eg.
  ↓
  Debug
  ↓
  Save
  So far we are at this point

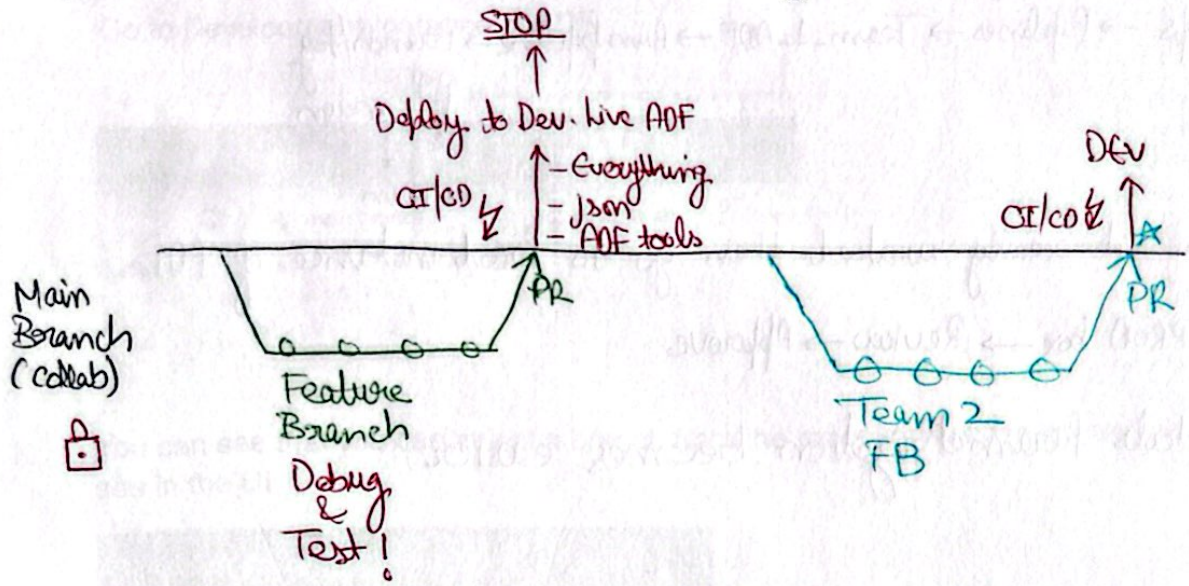


- After this create a Pull Request
  create → Approve → Set auto-complete
  ↓
  Complete
  ↓
  override branch policies and enable merge
  ↓
  Override and complete

till this step your code is deployed to Dev. Env.
also to check if your Dev. Env. is in sync with your Main Branch, simply go to Live Dev. Env. & check if you could see the changes that you just made.

Scanned with CamScanner

Suppose Team2 comes into picture & they wanna make some changes.



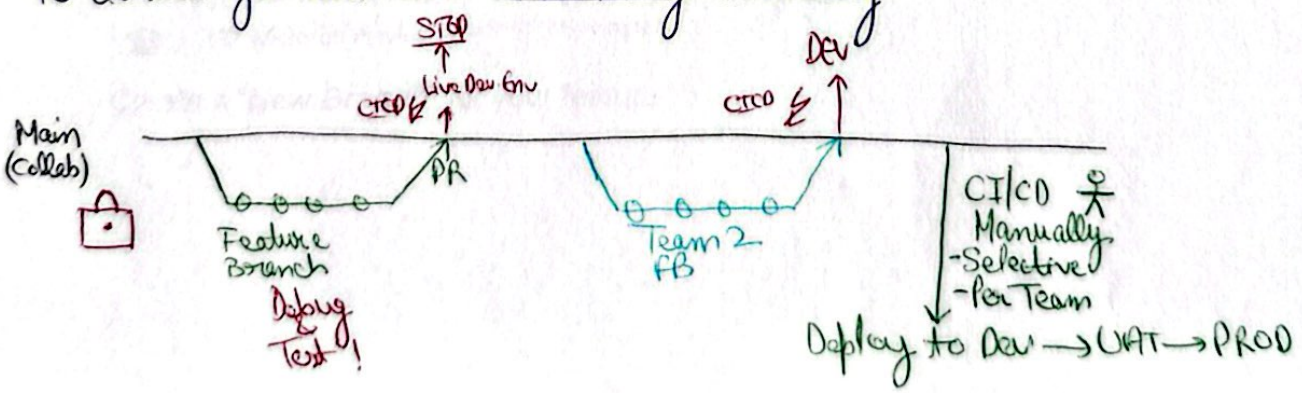
- Go to Dev. instance of ADF → Azure DevOps Git → New branch
  feature/Team_2_branch
  Based on: main branch
  ↓
  create

  you will see changes made by Team 1 in their folder as well. But you don't have to do anything about it.

- Add a New pipeline → Add some Activity → Debug → Save → Make a Pull Request
  Approve & Override the pull request. So you reached at ☆

After this automatic CICD will happen & it will reflect automatically to your Live Dev. Env.

So far changes made by both teams are in Main Branch also in Live Dev Env.
Now, you want to deploy changes made by Team 1 to UAT & PROD Env.
To do this you run CICD ~~automatically~~ manually.



Scanned with CamScanner

Now push only those changes that are made by Team1:

Azure DevOps → Pipelines → Team_1_ADF → Run pipeline → Branch/tag

main

↓

Run

to see if it really worked then go to Prod instance of ADF

Go to PROD box → Review → Approve

with ADF tools Parametrization becomes easier

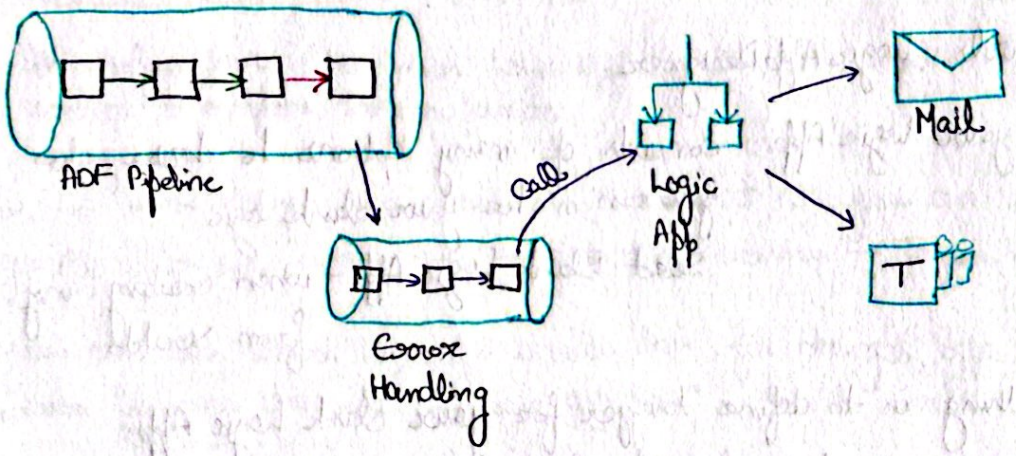
Scanned with CamScanner

Lecture 26

Introduction to Azure Logic Apps

* We know that ADF lacks some functionality like if a pipeline fails to execute, there is no option in ADF to notify developers. So, Logic Apps make our life easier.

We will create a separate pipeline for Error Handling & inside which we will call Logic App, which further would send an alert/notification either via Mail or to the dedicated Teams' channel.



- Logic App has lots of connectors like ADF
- Low-code, no-code.

How to use Logic Apps in Ingestion?

Source : SharePoint

Suppose many teams are keeping an Excel sheet updated. For collaboration, it is beneficial to put this Excel file on SharePoint, instead of sending different updated versions of Excel file.

Scanned with CamScanner

Usually while connecting to a source, you have to configure Linked Service, specially dedicated for the source you are trying to connect to. But sometimes you may not find the appropriate linked Service for the Source. In such cases you can either use APIs; in which case you will have to understand the documentation, authentication etc. or simply use Logic Apps.

Creating Logic App.

→ Create a Logic App in separate RG

→ Plan type : consumption

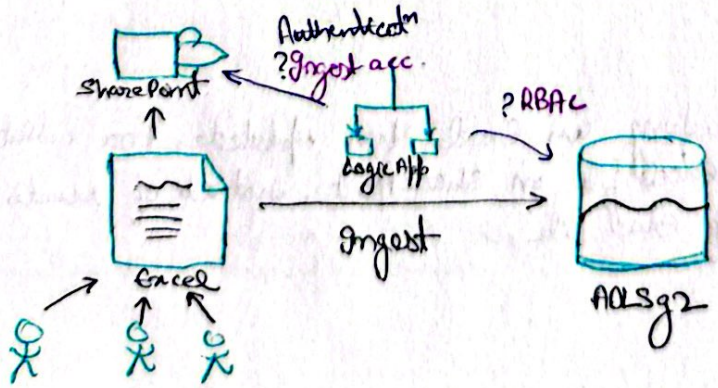
→ Go to Resource

→ Development Tools : Logic App Designer

→ Trigger for your Logic App : consists of many options to choose also you can make your blank logic
Select Blank Logic App - when creating any logic from scratch.

Imp. thing is to define trigger for your Blank Logic App.
So we be searching for Sharepoint triggers
↓
When a file is created or modified

∵ Logic App is different service from Sharepoint so we define how LA will authenticate
↓



Scanned with CamScanner

Step 1: Establishing connections

Logic App – SharePoint

you can use your signin credentials here but this is not a best practice. Instead go to SharePoint and inside "Documents" option, in 'Site access' grant a Read access to a dedicated account.
The moment you sign in using that dedicated acc., your credential will be saved in Azure.

After this configure your trigger so that Logic App would know which Library to connect.

- Copy & paste SharePoint URL inside "Site Address"
- Indicate "Library Name": Documents
- Also define how often your Logic App should peek inside the connected service to initiate trigger process

Now, if some changes are made in SharePoint ~~go~~ you can verify if the Logic App's trigger worked by going to the Overview → Trigger history

* you can see where Logic App should look for changes, o/ps like who changed, ~~whate~~ file was changed in "Logic App Run" in simple words only the meta data

Logic App – ADLSg2

Inside AzureAD create a group → assign this group some roles like contributor and add New member to it as Ingest acc.

After this go to Storage Acc. → Access Control (IAM) → Role assignments
here give access to the group that you created on Azure AD.

Scanned with CamScanner

Step2 : Create New step to grab the file

SharePoint → Trigger | Action → Get file content
↓
Site Address
File Identifier (o/p from trigger)
Dynamic content | Expression
↓
many option: Identifier

Step 3: Copy | Load data to ADLSg2

Create New step to save data → Azure Blob Storage
↓
Connection name
Authentication type Azure AD
↓
Sign in as ingest acc.
↓
Configure the "Create blob (V2)"
→ indicate to which container inside the Storage Acc. the file should be loaded
→ Blob name : dynamic
→ File Content : o/p from "Get file content"

Save changes

Scanned with CamScanner

Lecture 27

Azure Synapse Analytics

Components of Azure Synapse Analytics

→ Pipelines ~ ADF
→ Spark Pool
→ SQL
  ↳ Serverless : data exploration
  ↳ Dedicated : DW for MPP

* Fabric is successor of Azure Synapse Analytics

Creating Azure Synapse Workspace :

- Go to Azure Portal and create Azure Synapse instance
- Select your Resource Group & Subscription ; Also specify Data Lake for storing meta-data (either existing Datalake or New Data Lake)

  Create acc. name
  create file system

- Once ASA workspace is provisioned, go to the resource. You will see it has Azure Synapse Analytic Studio ~ ADF studio ; you enter into it & it will take you to a workspace
- On the left side you will see lot of options viz: Home
  Data
  Develop
  Integrate
  Monitor
  Manage

  you create pipelines here ← Integrate

actually pipeline inside ADF is similar to that in Azure Synapse Analytics.

Scanned with CamScanner

We will ingest data from Lego dataset using Rebrickable API

this is a brand new dataset/source for us.

∴ this scenario is : our source is REST API

Is it cloud or on-prem? Rebrickable API is available on cloud (i.e exposed to internet)

this impact on choice of IR to be used to connect to the source

Since our source here is on cloud ∴ IR provided by Azure would suffice

Pull or Push : here we will be extracting data using API ∴ Pull approach

Streaming or Batch : Batch

Ingest schedule? do we have to worry about not to disrupt the source.
In this case, not really.

Is the source under development: Does API changes?
you ~~the~~ can check this in documentation ~~of~~ under change log

How to connect? Always in API documentation under Authentication.
Identification of your endpoint is very important.
eg: https://rebrickable.com/api/v3/lego/colors/
endpoint

Networking: do we have to care about firewall here - not really ∵ this API is public available.

Security: can anyone access this API. In documentation you will be provided with key to be passed as HTTP header. How to get that key?

Create an account on Rebrickable
↓
Account
↓
Settings
↓
API

Scanned with CamScanner

Create a new pipeline in Synapse workspace

Copy data from Rebrickable API

Dump into our Datalake

· create a linked service
↓
REST
IR
Base URL
Authenticatn type
↓
Anonymous here
∵ we have to use headers & this option is not available by default

①

Synapse workspace

Copy data

· select Copy data activity

· create a container (folder/directory) where you want to dump data

Azure AD
Groups → Data-Lake-Contributor
↓
Members
↓
add Managed identity of Synapse Workspace

Source
①
· Create a Source dataset
↓
REST
· Linked Service
· open the Source dataset
fill Relative URL (usually endpoint)

Sink

Sink dataset
↓
ADLSg2
↓
Format ?
↓
JSON

go to the source website, enter your key & see sample

· Create a Linked Service (for DataLake)
* do not forget to grant access to Synapse workspace for DataLake
· Authentication type → System Assigned MI
↓
select appropriate Subscription & Storage Account

· Specify File Path
(container you created in the beginning in DataLake)

· Go to your Storage Acc. (DataLake)
· Go to Access Control (IAM)
· Check "Storage Blob Data Contributor"

Scanned with CamScanner

* How to provide Authorization Token?

- Check the documentation & identify Authorization Header
- Inside Source configuration in "Copy data" activity add "Additional Headers"

Name	Value
Authorization	make it dynamic

Store this inside a key-vault
- Go to key-vault
- Secrets (to add a new secret)
- Generate/Import
- Secret value: add your key here

to get this secret, our Synapse must get access to key-vault

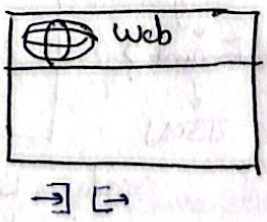
either use RBAC (in Access policies)

- Go to Azure AD
- check for key-vault secret group
- Add MI of Synapse

after doing all this Go back to "Copy data" activity

you will see that under Authorization headers there is no option to access a key-vault

Solution:
- Add a web activity to Get API key from key-vault
- Under Setting - configure it according to documentation



value will be here

add URI of your secret here

* Finally inside "Copy data" activity by referencing the o/p of web activity by carefully selecting the parameters you could get access to Key-Vault

Scanned with CamScanner

Lecture 31

Azure Databricks → a seperate company

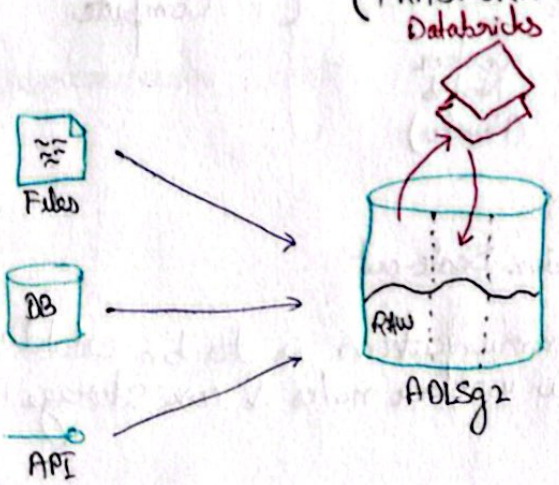
- primarilly for Data Transformation
- best for building Data Lakehouse
- ETL & Data Engg.
- ML, AI, Data Science
- perform analytics
- can be used for Data Governance & Data Sharing
- Streaming Analytics

Connection b/w Apache Spark & Databricks

: initially a group of developers were working on Apache Spark, which is basically for Big Data. It is not easy to work with Apache Spark & thus they came out with some easy solutions, that can be utilized by other developers to work on Big Data easily ∴ comes Databricks

Where does Databricks fits ?

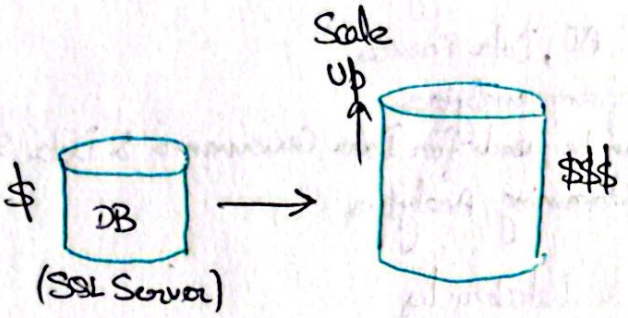
(TRASFORM PHASE)



- you connect to DataLake
- read data from Raw Layer
- transform data
- write back into Data Lake

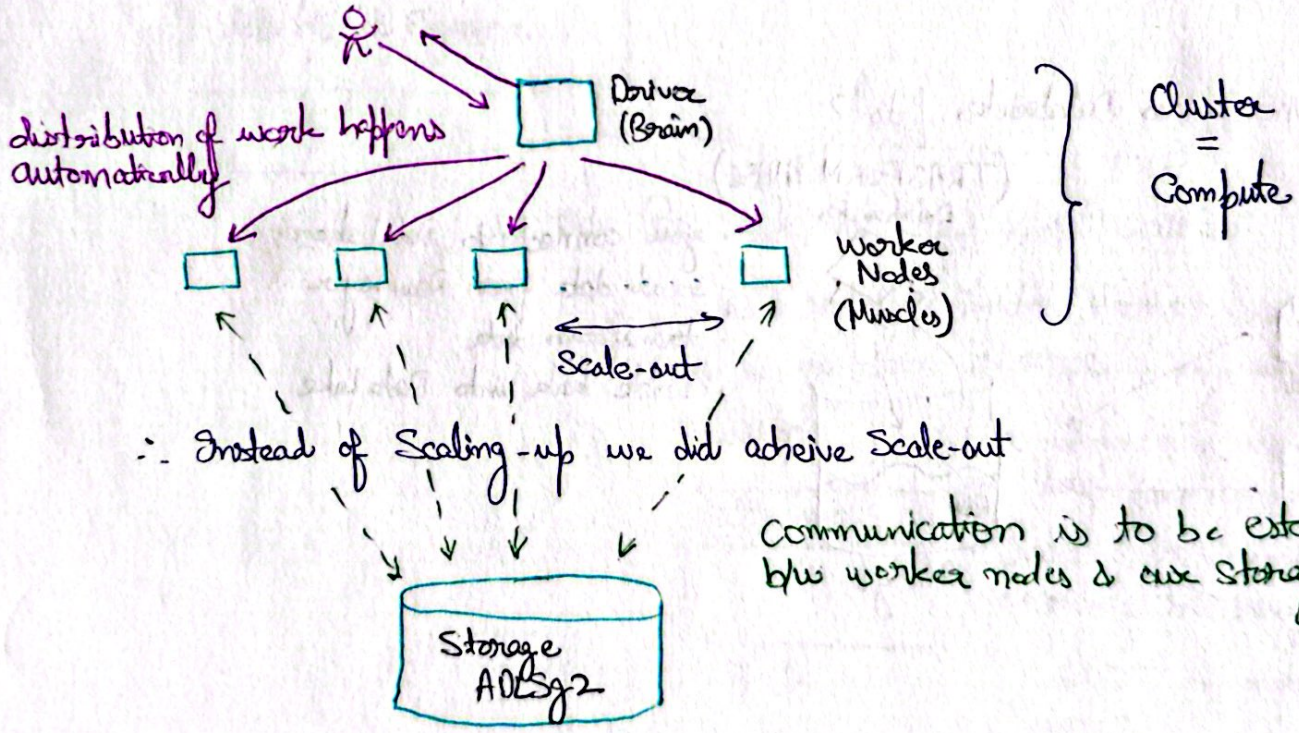
Scanned with CamScanner

Before Big Data, we had on-prem DB like SQL Server & we wrote various store procedures, queries inside those databases for transformation.
With increase in data, it was difficult to process that amount of Big Data.
One solution was to Scale-up (h/w and stuff). Another solution was to distribute loads on multiple nodes(m/cs)



SPARK ARCHITECTURE

consists of Driver Nodes (Brain)
Worker Nodes (Muscles)



∴ Instead of Scaling-up we did acheive Scale-out

Communication is to be established b/w worker nodes & our Storage Acc.

Scanned with CamScanner

How to provision Azure Databricks

- Create Azure Databricks workplace instance inside Microsoft Azure

Azure Portal → + create a resource → Azure Databricks → + create

Choose:
Sub.
RG
Workspace name
Region
Pricing Tier

when you finish creating Databricks Workspace, launch it & you will see an UI

How to create a cluster?

Obviously, you will require some computing services to process your data.
Thus, on left side:

Click on compute → [Create compute]

↓

After that configure your cluster that is in-line with the Business requirement.

↑worker node ↑cost

Policy

o Multinode o Single node

based on your selection here you would ask for additional configurations. e.g in case of Multinode: you would require to adjust computing power of your worker node.

Single-node: you would ask to customize only one node obv ;)

Access mode

→ Single user - for single user
- Supports all languages.
(Standard)

Scanned with CamScanner

→ Shared - SCALA
- for Multiple users
- Isolation
(High Concurrency)

→ No Isolation Shared
- all languages
- no isolation

Scanned with CamScanner

Lecture 34

Common Data Transformations

- we will work with Rebrickable tables.
- the data is in the form of json
- API often return data as json file
- json often is nested

```
"result": [
    {
      "set_num": "fig-000001",
      "name": "Toy Store Employee",
      :
    },
    {
      "set_num": "fig-000002",
      "name": "Customer Kid"
      :
      "last_modified_dt": "2020-05-27T21:47:11.750470Z"
    }
```

- result element is an array of objects. And these objects are data about Mini fig.
- That is the data we care about

⇒ we will create Tabular form of those objects.

⇒ we will try to map those objects in some kind of columns.

e.g. "set_num" ⟶ column

⋮

"last_modified_dt" ⟶ column

⇒ also we will try to investigate about datatype of those objects & try to modify them as per the requirement

⇒ we will also address NULL values, apply some Bussiness logic, address duplicates as well.

Scanned with CamScanner

Step 1 load your data

- Data is ofc present inside your Datalake (Storage Account). Therefore you need to connect Databricks to Storage Account.

  this is already covered : by using Service Principal & secret is being retrived from Azure Key Vault

- use dbutils.fs.ls("abfss://<container>@<storage account>.dfs.core.windows.net") command to test connectivity
- then load your json file : spark.read.json("path = ... ")
- use display method to see the result:

  display(df)

- results ⇒ this is what we have to consider
  ⇒ here you will see an array consisting of several objects

- if you see the schema of df:

  count : long
  next : string
  previous : string
  results : array
  element : struct
  last_modified_dt : string
  name : string
  num_parts : long
  set_ing_rel : string
  set_num : string
  set_rel : string

- no. of rows in a DataFrame = df.count()
  ~ 142 (this is the no. of rows inside each page)

Scanned with CamScanner

Step 2 : Flattening the Array

with explode !!!

from pyspark.sql.functions import explode

- inside the explode pass the column you want to segregate

∴ df2 = df.withColumn("explodedArray", explode(df.results))

this will create a new column as "explodedArray" and its value will be equal to explode(df.results)

explodedArray : instead of objects inside an array, the result will be like each object will be in separate row.

Row 1	∨object : last_modified_dt ⋮ set_val:
Row 2	∨object : last_modified_dt ⋮ set_val:
⋮ Row N	

- remaining columns onto which you did not use explode(), those columns will get duplicated.

Scanned with CamScanner

Step3: Extracting Fields from struct

- to use SQL queries you need to register your dataframe

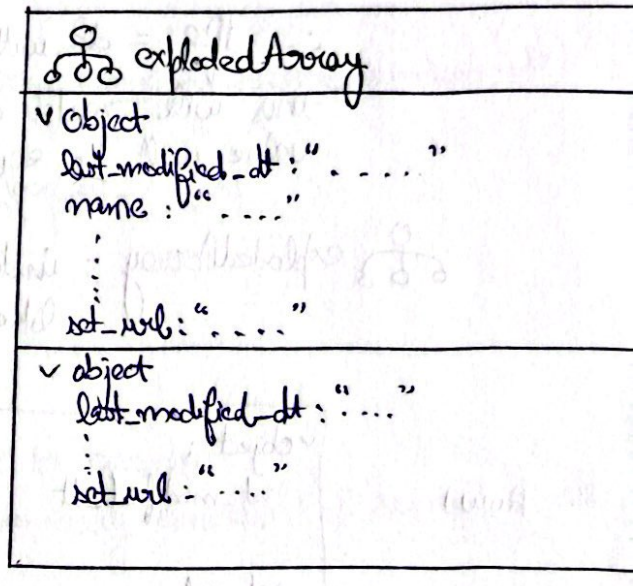
  df.createOrReplaceTempView("data")

⇒ No. of Rows:

```
SELECT Count(*)
FROM mydata
```

⇒

```
SELECT
    explodedArray.last_modified_dt,
    explodedArray.name,
    explodedArray.num_parts,
    explodedArray.set_img_url,
    explodedArray.set_num,
    explodedArray.set_url
*
FROM
    data
```



Output =

last_modified_dt	name	num_parts	.....	set_url

+ columns you dont need anymore

count	next	previous	results

Scanned with CamScanner

Step 4 : Removing unwanted columns

just remove * from your previous query

Step 5: Renaming Columns

Simply us AS

ex:

```
SELECT
    explodedArray.last_modified_dtt AS LastModifiedDatetime,
    :
    explodedArray.set_url AS SetURL
FROM
  data
```

Step 6 : Changing Datatype

if you see schema of your data:

LastModifiedDatetime : string ⇒ we want to change it

Name : string

NumberOfParts : long ⇒ change this as well

ImageURL : string

SetNumber : string

SetURL : string

if you only want to have date portion but not time part

⇒ to_date()

```
explodedArray.last_modified_dtt AS LastModifiedDatetime,
to_date(explodedArray.last_modified_dtt) AS LastModDat,
to_timestamp(explodedArray.last_modified_dtt) AS LastModTime,
cast(explodedArray.num_parts AS int) AS NumberOfParts
```

maybe you will be interested in time as well

cast() is used to change datatype

Scanned with CamScanner

Step 7: Additional conditional logic

maybe you want to add a column based on some other column.

```
CASE
    WHEN .... THEN....
    WHEN .... THEN....
    ELSE ...
END AS MinifigType
```

→ column name

this means if Toy phrase is found anywhere in the name value then it will evaluate to true

⇒

```
CASE
    WHEN explodedArray.name LIKE '%Toy%' THEN 'TOY'
    :
    ELSE 'OTHER'
```

⇒ to make above query case insensitive.

```
CASE
    WHEN upper(explodedArray.name) LIKE '%TOY%'
```

Step 8: Removing Duplicates : using DISTINCT

```
SELECT
    count(DISTINCT explodedArray.name)
FROM
```

finding duplicates: group your data by the column you suspect having duplicates

```
SELECT explodedArray.set_num, COUNT(*)
FROM
GROUP BY explodedArray.set_num
ORDER BY
        COUNT(*) DESC
```

Scanned with CamScanner

Now lets say we have Date-time column & inside the Date-time column value the duplicates will have different dates. Sometimes, Bussiness tell you that keep the one between/among duplicates that has current/latest Date.

But what if the data is too large? You cannot just write SQL query like previous example & explicitly find duplicates.

∴ Windows functions are used in this case.

ROW_NUMBER : outputs the latest version

assigns no. to results

ID	Date
A	2010
B	2010
C	2015
B	2017
B	2018
A	2020
D	2012

⇒ Groupby ID, order by Date

ID	DATE	ROW_NUMBER
A	2020	1
A	2010	2 ✗
B	2018	1
B	2017	2 ✗
B	2010	3 ✗
C	2015	1
D	2012	1

then you can write your logic to filter based on ROW_NUMBER

Scanned with CamScanner

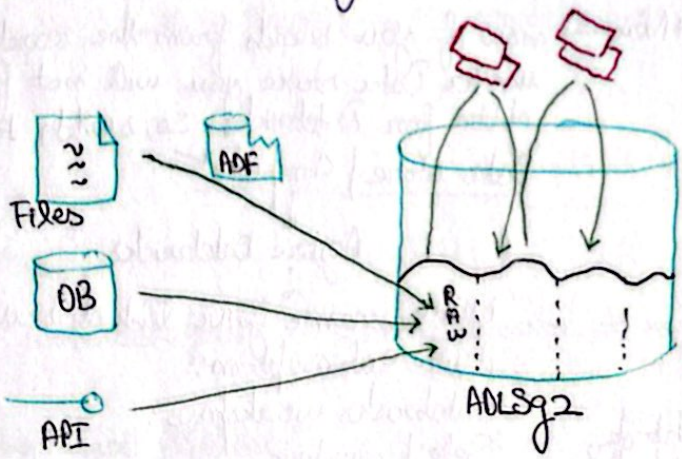
Lecture 37

Orchestrating Databricks Notebooks

So far we executed Databricks notebook manually, cell by cell. But in Production environment this is not the way to run Databricks notebooks. Therefore, we are going to orchestrate all Databricks notebooks

Where does it fit?

- We have been ingesting data from various sources using ADF inside our RAW layer of Datalake.
- Then we were transformations using Databricks either using Autoloader or manually crafted notebooks (depends on what we want to achieve)
- And we were repeating this process depending on number of zones in our Datalake
- agend is how we should our Databricks notebook in our B.I flow.
- So far we have orchestrate ingestion using various pipelines inside ADF. ADF is also good in orchestrating our data.



Collaboration b/w ADF & Databricks:

Create a New Notebook:

databricks → Workspace → Go to your director or create a new one → on the top right corner choose "Create" → Notebook

name your Notebook

Scanned with CamScanner

establish your connection with the cluster, on which you want your Notebook to run

Go to ADF : try to run a pipeline to call your Notebook

Data Factory → Author → Pipelines → New pipeline
name it

Activity 1: suppose this activity ingest the data → Activity 2: this will be used to call our Notebook

Activity → Notebook
Now configure Notebook activity
General | Azure Databricks | Settings ☆ | User properties #
↓
Databricks linked service → +New (create a linked service for Databricks)

Go to Manage → Linked services → +New → now if you search from the search bar under Data store you will not find an option for Databricks. So, switch to "Compute"
Data store | Compute
↓
Azure Databricks
↓
Name : name your linked service
Azure Subscription :
Databricks workspace :
Select cluster
{ ○ New job cluster ✓ Existing interac. cluster
○ Existing instance pool
Authentication type :
a) Access Token
b) Managed service identity
c) User Assigned Managed Identity
↓
Choose from existing clusters
↓
Test connection → create

When a developer starts to work on a notebook, he needs a lot of attempts to make the notebook correct. He would be connecting interactively with notebook again & again.
In case of production environment, when notebook is already tested & verified, at that time there is no need for interactive mode.

Also, interactive cluster are expensive than Job cluster

After creating the linked service for Databricks place it inside configuration #

Scanned with CamScanner

Authentication type

a) Access Token:

where to get this?

Go to Databricks → top right corner click your account → Settings → from the menu on the left side navigate to 'Developer' → Access tokens → Manage

↓

Generate new token

you name it

choose Lifetime

↓

Generate

↓

you will see a connection string

then you can store this token inside Azure Key Vault

• AKV linked service

you must already have linked service to Key Vault that will show ADF path to pick up all the secrets for which it is given access to

Inside Key Vault:

Key Vault → Secrets → +Generate/Import → Name:

Secret Value: paste the token you generated inside Databricks

* Settings

Notebook path* : simply browse to your notebook

• debug your pipeline

b) using Managed Identity (of ADF)

Create a new linked service → Manage → Linked Service → +New → Compute

↓

Azure Databricks

Authentication:

b) Managed service identity

Existing cluster ID:

you have to indicate which cluster you want to connect to.

You will see this option is empty.

Scanned with CamScanner

Go back to Databricks → Compute

↓

if you go to properties of a specific cluster, cluster ID can be seen in many places eg: • it is part of URL

• configuration → JDBC/ODBC under HTTP Path

copy the cluster ID & ~~paste~~ paste it inside the linked service definition

↓

Test connection

this connection will fail with Error 403

this will obviously won't work because you are using Managed Identity of ADF to connect to Databricks. But Databricks does not know anything about Managed Identity of ADF. Therefore, you will have to create an account for ADF inside Databricks.

How to do that?

As per Microsoft's documentation, they say grant Contributor role to Managed Identity of ADF. It will work. Sufficient permission will be granted to ADF.

But the thing is Contributor role is so powerful that it will make Managed Identity an Administrator inside Databricks. We will not want that to happen.

∴ Best way is to make Principal for ADF inside Databricks

Copy Managed Identity name → Go to Microsoft Entra → Search for Enterprise Application by pasting MI name

↓

Properties

Name:

Application ID:

Object ID:

Go to Databricks → Navigate to top right corner → click on User → Settings → → Identity & access → Service principals → Manage → Add service principal → Microsoft Entra ID managed → Microsoft Entra ~~ID~~ application ID

↓

Add

Scanned with CamScanner

- So far you just created a user for ADF inside Databricks but no permissions have been assigned yet to that user (ADF in our case)

Therefore, you want your ADF to connect to a desired cluster, & inside the cluster, ADF can connect to any Notebook. For this, you grant CAN RESTART permission to ADF. In many cases, when cluster is idle, our ADF can restart the cluster if granted CAN RESTART permission

At Cluster level:

Go to Databricks → your cluster → at right side navigate to 3 dots ⋮ → Edit permission
↓
Add ADF MI principal
↓
Can Restart
↓
Add & Save

∴ Now, can connect to this cluster & restart it

At Notebook level:

Go to the Notebook → at right side, navigate to 3 dots ⋮ → Share (Permissions)
↓
Search your ADF MI
↓
Can Run
↓
Add

Now, after creating the Linked Service & establishing the connect if you run the pipeline, the pipeline will fail throwing error:

PERMISSION_DENIED : Single-user check failed
⋮
0322-072734-gpbwtmx, but the single user of this cluster is 'piotr@tybulonazure.pl'

because you set the cluster with single user mode. Only that single user can connect to the cluster, not some other MIs.

∴ Go to cluster settings

Access mode	Single user access
Single user	Piotr (admin)

it worked in Access Token mode because even tho connection was made from ADF, Databricks thought it was you who were connecting because Access Tokes impersonate identities.

- Sol: you can create a New cluster with
  Access mode = Shared

Scanned with CamScanner

Settings → Security → Table access control ⇒ enable this in order to use Shared Access mode

- After that create your cluster with "Multinode" because "Shared Access" doesn't coexist with "Single node" option.
- Set worker nodes as well

Then grant permission to ADF Managed Identity on this newly created cluster.

Select Cluster

Scanned with CamScanner

Lecture 40:

Extra features in Azure Synapse Analytics Spark Pools

Connecting to ADLSg2

- we saw that we can have access to data from Synapse Workspace as well.

Data Tab

from left menu → workspace | linked

↓

if you go to linked tab, you will see all the linked services created from Synapse Workspace.

↓

synapseDatalake

↓

checkpoint
conformed
curated
public
raw

all the containers available inside Data Lake dedicated for Synapse Workspace

↓

Rebrickable

↓

minifigs.json Right click

↓

New notebook

↓

Load to DataFrame

after this you will see that a code will be generated that would help to read file from the Datalake

```
%%pyspark
df = spark.read.load('abfss://raw@tybuldatalakedemo.dfs.core.windows.net/
-------.json', format='json')
display(df.limit(10))
```

Scanned with CamScanner

you will observe after running the code that Spark Pool was able to connect to the DataLake mentioned in the code & can retrieve data as well.

But how Synapse Workspace is connecting to Storage Account?

Interactive Method:

Entra Passthrough

Suppose you grant Storage Blob Data Contributor role to Managed Identity of Synapse workspace on a desired Data Lake (Storage Account)

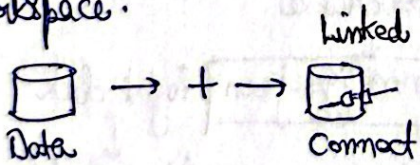
Storage Account → Access Control (IAM) → Role assignments

↓

provide your Synapse Workspace Managed Identity & grant Storage Blob Data Contributor role.

But on the other hand if you are just given "Owner" role on the same Storage Account and you are using your identity to establish connection between Synapse Workspace & the Storage Account you will not be able to read the data.

If you want that Storage Account to be visible under your Synapse Workspace.



→ Azure Data Lake Storage Gen2

New linked service

Name

Connect via integration runtime

Authentication type: System Assigned Managed Identity

Storage account: choose Storage Account onto which you granted role to MI of Synapse Workspace

↓

Test Connection

↓

Create

↓

Publish.

Scanned with CamScanner

- now the content of Storage Account could be visible from Synapse Workspace

Go to recently created linked service for the Storage Account → raw → go to any directory → go to any json

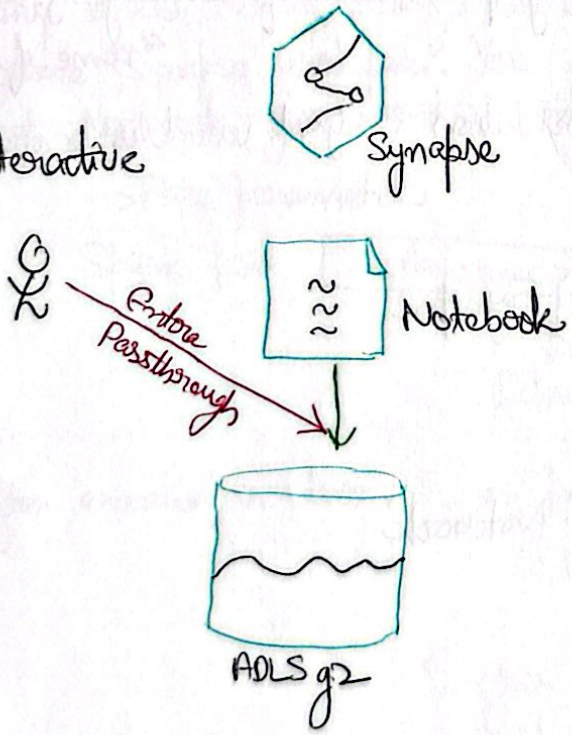
Right click
↓
New notebook
↓
load to DataFrame

```
%%pyspark
df = spark.read.load("abfss://raw@datalake.dfs.core ...", format='json')
display(df.limit(10))
```

this will failed: Access denied 403

∴ if you are running notebooks in interactive way (like right click on the file & load to DataFrame) it will use an identity of user running the notebook. Since you did not have Data Plane permission, therefore you couldn't read the data.

① Interactive



∴ Synapse uses your permissions.

What if some other developer wants to use the same notebook?

In that case you would want Synapse to use its own Managed Identity to run notebooks

You can achieve this by a single setting within Notebook

Scanned with CamScanner

Managed Identity

Go to the Notebook → at top right corner navigate to gear icon ⚙ →

Run as managed identity
⊘Enable ○Disable
↓
Apply

∴ From now on the Notebook will connect as Managed Identity of Synapse Workspace

Linked Service

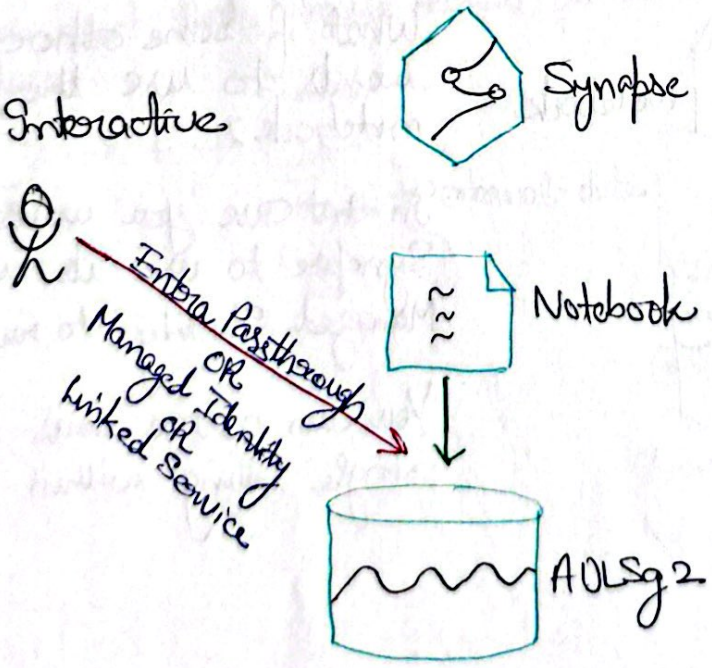
But you already configured the linked service, to connect to the Storage Account & inside that linked service you already defined the way to be connected with Storage Account.

Therefore, you can also use Linked Service to establish connection.

```
%%pyspark
source_full_storage_name = "your_storage_account"
spark.conf.set(f"spark.storage.synapse.{source_full_storag...}.linkedServiceName",
                                    "name_of_your_linkedServic
```

take this code from the documentation & adjust variables accordingly.

1. Interactive



Scanned with CamScanner

② Non-Interactive

integrating execution of Notebooks with pipelines.

Go to Notebook / Create a new Notebook → Set properties → write your code

Attach to [<Pool> ✓]

↓

at top right corner of Notebook navigate to ( ) Add to pipeline

↓

Existing pipeline

[New pipeline]

↓

Set properties

∴ a new activity (Notebook) will get added to Synapse Workspace and under "Settings" below you will see you do not have to provide with any linked service settings because Notebook & Pipeline exists in the same Synapse Workspace. You just have to provide with the name of Notebook you want to execute.

> Base parameters

Spark pool [<Pool_name>]

↓

Debug the pipeline

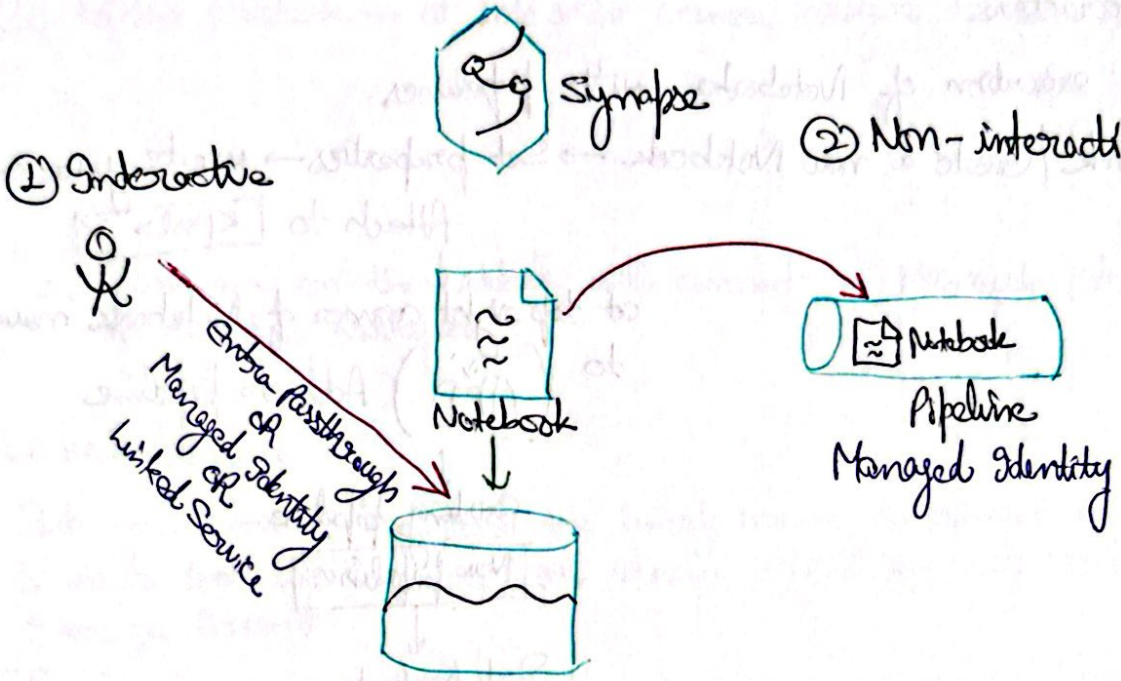
- Now after execution is done, if you peek ( )

↓

you could see the data ∴ data was read

which means if you put your Notebook inside a pipeline then the pipeline will automatically use Managed Identity of Synapse Workspace.

Scanned with CamScanner



Parametorizing Notebooks

- inside Databricks, you set parameters inside Widgets.
- but in Synapse, you create a separate cell at top & define your parameter there.

eg: fileName = "SampleJson"

```
%%pyspark
df = spark.read.load(f'abfss://...../{fileName}.json'...)
```

∴ you are changing the value of a parameter inside your Notebook, & you want your pipeline to run this Notebook even if the value of variable "fileName" is changing.

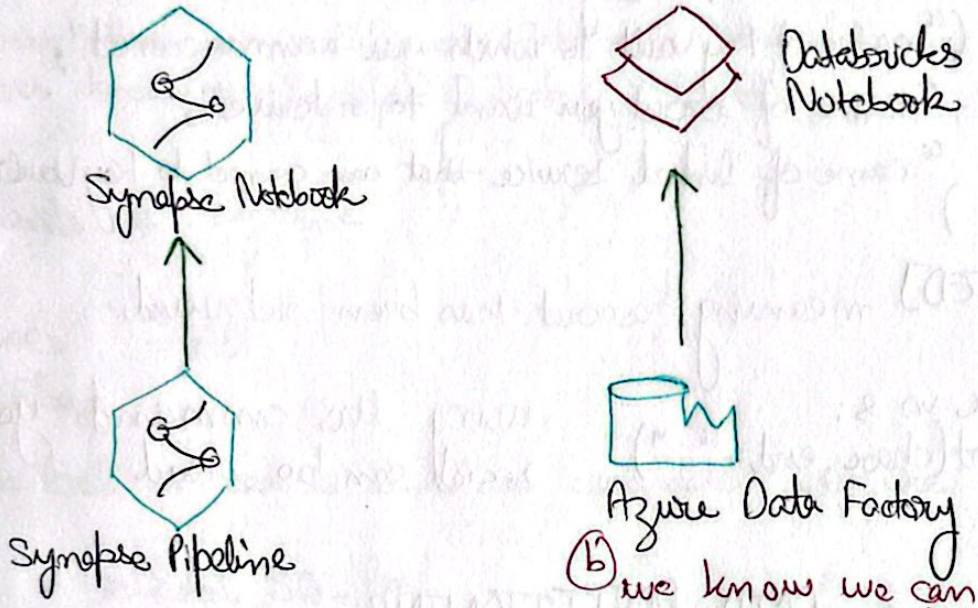
⇒ Navigate to the right side of the cell where you defined your variable
... → [@] Toggle parameter cell → now every variable that you define in that cell will become parameter to that cell.

Scanned with CamScanner

⇒ Open your Synapse Workspace → Notebook activity → at bottom navigate to "Settings" → Base parameters → +New

Name	Type	Value
fileName	String	SampleJson

Synapse vs ADF vs Databricks



(a) you can run Synapse Notebook from Synapse Pipeline

(b) we know we can run Databricks Notebook on Data Factory.

Apart from (a) & (b) you can also call Databricks Notebooks from Synapse pipeline & you can call Synapse ~~pipeline~~ Notebooks from Data Factory.

Integration with Key Vault

- In Databricks to grab secrets from a Key Vault we created Secret Scope. But in Synapse we will just use Managed Identity of Synapse & create a linked service to the Key Vault

Go to Synapse → Linked Services → Key Vault → Test connection → Create → Publish

Name
Azure Key vault = which one
Auth. method = Sys. Assigned MI

Go to Key Vault → Access policies (here you can see what permissions set on a Key Vault)

↓

here you will see an access policy defined that granted get, list secret permissions to a certain Group

add Synapse MI to that group

grab code from documentation to retrive a secret from a Key Vault.

```
s = mssparkutils.credentials.getSecret
    ("name of Key Vault to which you wanna connect",
     "name of secret you want to retrive",
     "name of linked service that can connect to Key Vault"
    )
```

[REDACTED] meaning secret has been retrieved.

```
for char in s:
    print(char, end=" ")
```

using this command you secret can be read.

DATA PARTITIONING

```
from pyspark.sql.functions import year, month, col
dated_df = spark.sql(
    "SELECT * FROM db.table")\
  .withColumn("Year", year(col("LastModifiedDate")))\
  .withColumn("Month", month(col("LastModifiedDate")))
```

Adding 2 column : Year & Month & extracting their values from LastModifiedDate

Now we want to save the data:

```
dated_df\
  .write\
  .partitionBy("Year", "Month")\
  .format("delta")\
  .option("path", "adfss://curated@.....")\
  .saveAsTable("db.table")
```

partition first by month & then by year

save as delta

register table in catalog

save to this directory

Scanned with CamScanner

Lecture 43

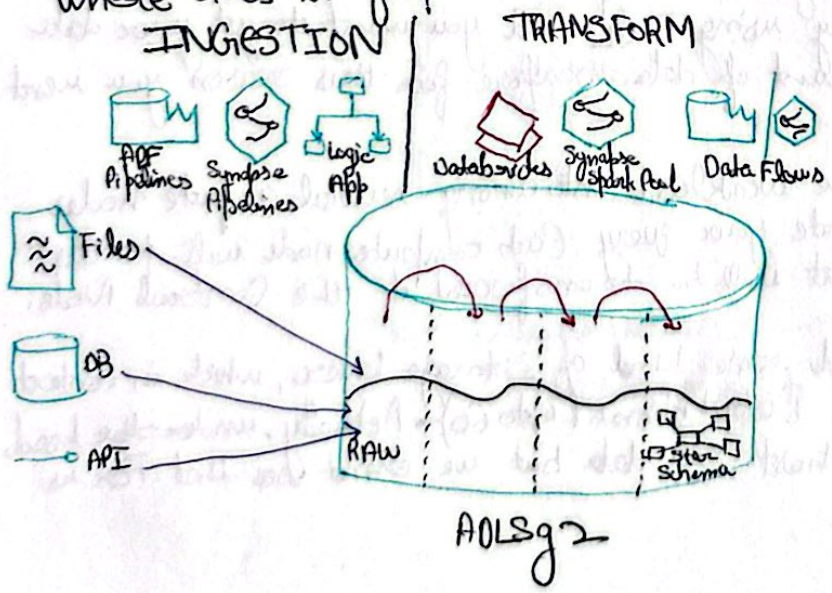
Azure Synapse Dedicated SQL Pool
Architecture and Overview

- Now our data is transformed in form of facts & dimensions inside our Data Lake.
- Next step is to provide this transformed data to the consumers.
- We do not want consumers to have access to our Data Lake
- To acheive this we will create Relational Datawarehouse inside Synapse Analytics known as Synapse Dedicated SQL Pool.

Components of Synapse

- Pipelines = ADF
  (Data Flows
- Spark Pool = used to transform our data, just like Databricks
- SQL : Dedicated SQL Pool

Where does it fit?



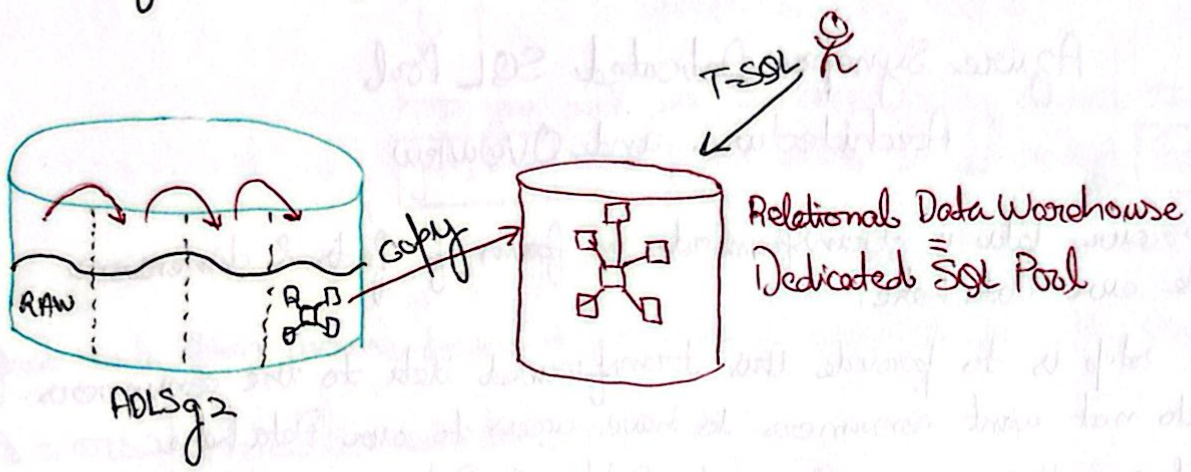
Now the thing is we have the cleaned data in delta format. But how to read that data. There might be some older tools that would like to grab this transformed data and maybe those tools cannot read delta format files.

Or maybe there is a project that has old Data Lake in which they are not storing files as delta but in some other formats.

Therefore, you make your data from Data Lake available by using SQL endpoints by using Dedicated SQL Pools

Scanned with CamScanner

Transform | SERVE



Dedicated SQL Pool uses concept of MPP (Massively Parallel Processing). Therefore, meaning data is getting processed parallely. Best practice is to utilize the dedicated SQL pool only if data is more than 1TB. If data is less consider using a regular Azure SQL Database.

Once data is loaded to dedicated SQL Pool, it is available for end user to connect to it using T-SQL

Suppose, you want to run some SQL query on data stored in that Data Warehouse. You connect to the endpoint exposed by dedicated SQL pool & send your SQL query.

Under the hood:

When you run your SQL query, it goes to the edge node (Control Node), entry point for your whole SQL pool. Meaning you can only interact with the Control Node. Imagine you fire an SELECT query using T-SQL and you want to get your data fast & because there is huge amount of data therefore for this reason you went for dedicated SQL pool.

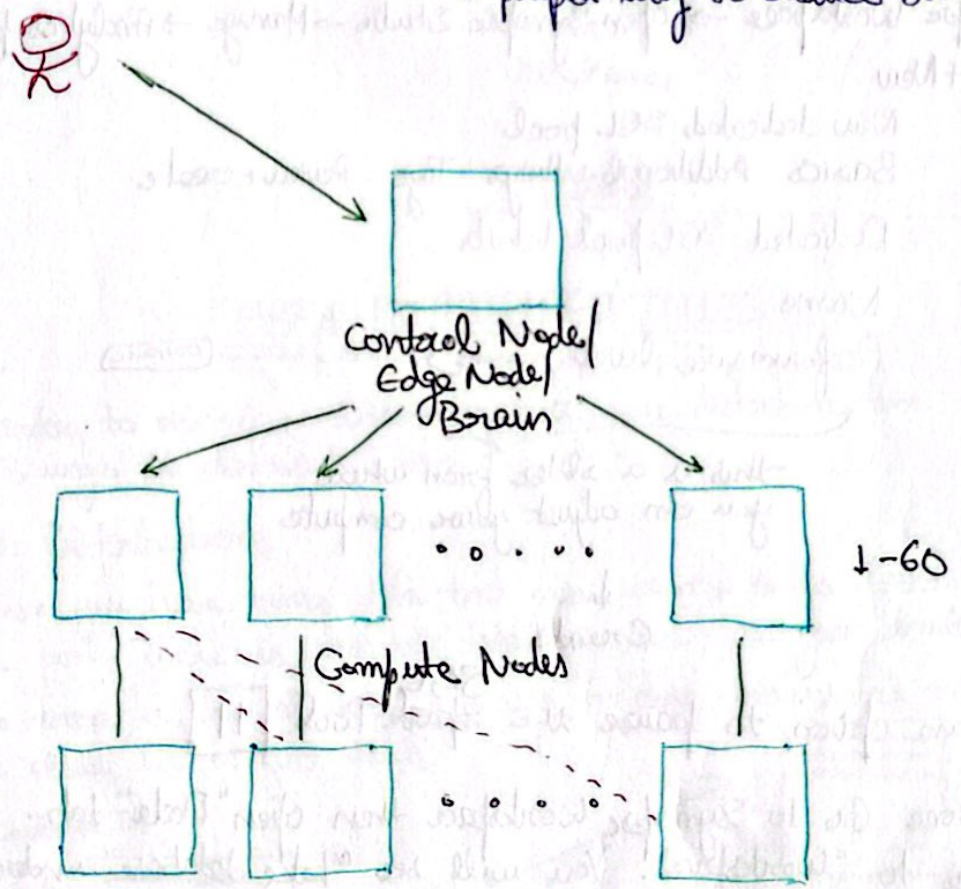
Edge node distributes the workload ~~into~~ among several compute nodes. (between 1 to 60) That further execute your query. Each compute node will process its data & then at end the output will be transferred to the Control Node.

Along with all these we also need some kind of Storage layer, which in context of dedicate SQL Pool is known as Distributions (upto 60). Actually, under the hood Azure Storage Accounts is used to host our data but we cannot see that. For us its just a database.

Scanned with CamScanner

- In case you go for 60 compute nodes, each compute node will have to deal with its own seperate dedicated distribution. Therefore, processing speed would be higher.
- But in cases where you are using few compute nodes, then those compute nodes may be responsible to process data from multiple distributions. Therefore processing speeds may affect.
- As discussed, suppose a scenario where a compute node has to read data from its associated distribution as well as from the distribution that is not associated with it, then the performance of query would be bad. And when data moves from a distribution to a compute node that is not associated with it is kaa Data Shuffle / Data Movement.

make sure to store your data in a proper way to reduce Data Movement



60 Distributions → no more, no less (fixed)
(Azure Storage)

Scanned with CamScanner

Setting while deploying a dedicated SQL Pool:

no. of compute nodes depend on your "Performance Level" selection

Performance Level	Compute Nodes
DW 100c	1 $
⋮	⋮
DW 30000c	60 $$

Where to set this Performance Level:

Go to your Synapse Workspace → open Synapse Studio → Manage → Analytics pools → SQL pools → +New

New dedicated SQL pool

Basics Additional settings Tags Review + create

Dedicated SQL pool details

Name

Performance level ——○- - - - - - - - - - DW10000c

this is a slider from where you can adjust your compute

↓

Create

There is also an option to pause the ~~Spark~~ SQL Pool ||

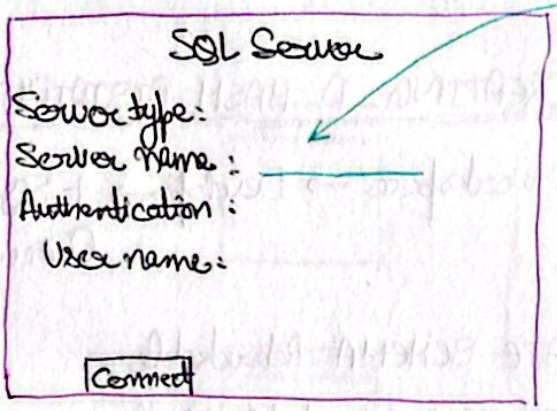
- After setup is done. Go to Synapse Workspace then open "Data" tab. From "Data tab" go to "Workspace". You will see "Lake database" under this the databases that are available are those present in Data Lake. Your recently created Dedicated SQL Pool will be appear under "SQL database".

Scanned with CamScanner

Now if anyone wants to connect to your SQL pool eg. if someone is using SQL Server Management Studio (SSMS) & wants to connect to our SQL pool then:

Go to Synapse Workspace → Dedicated SQL endpoint

This is the place where they would be sending their queries, to the edge node. this endpoint can be used to connect to pool from SSMS.

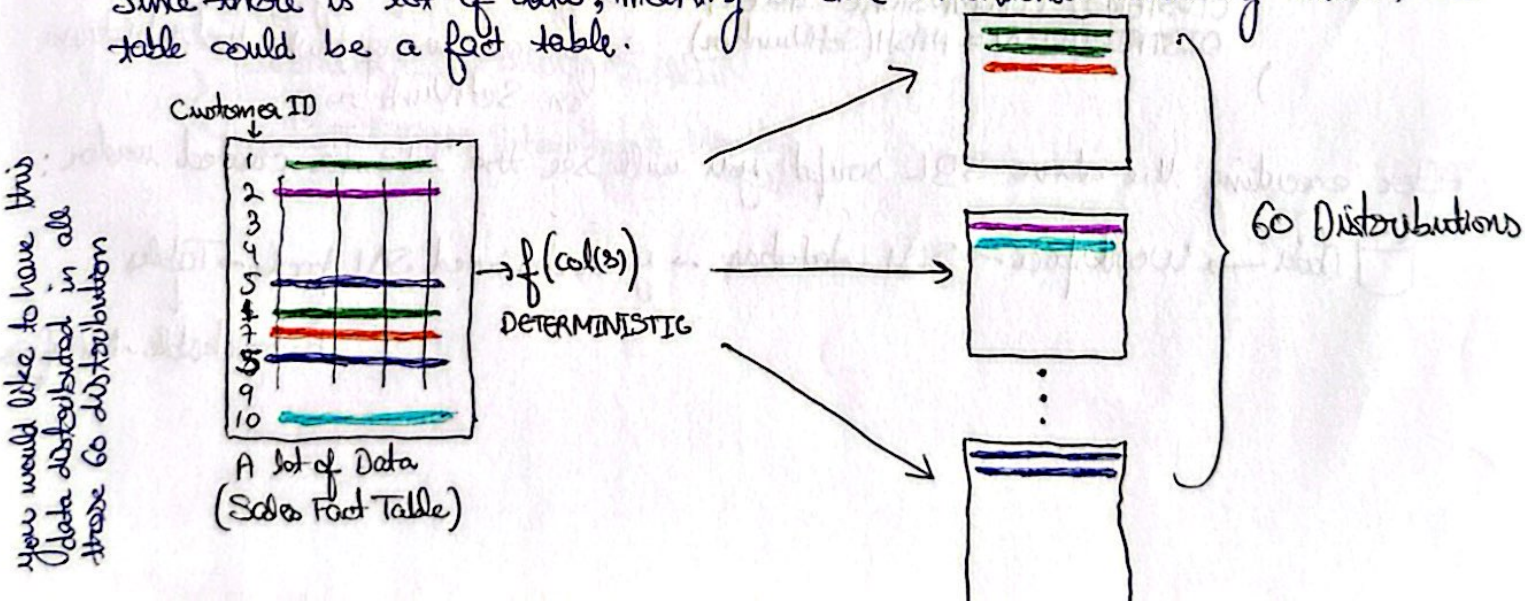


HASH DISTRIBUTED TABLES

In order to minimize data movement, you distribute the data properly. There are 3 ways to distribute data

1. Hash Distribution

Suppose you have some data that you wanna move from Data Lake to dedicated SQL pool. There is lot of data, in this case you should go for Hash. Since there is lot of data, meaning in our dimensional modelling context, this table could be a fact table.



→ hash partitioning is used for Facts
→ >2GB
→ choose those columns for hash partitioning that will be using inside JOIN ~~statement~~ condition or GROUP BY clause
→ do not use Data with date column for hash partitioning
→ best practice is to apply hash partition on the column that is less skew. If the data is skewed, one distribution might end up containing large amount of data than other distributions.
→ use Clustered Columnstore Index if rows are greater than 60 million.

CREATING A HASH DISTRIBUTED TABLE

Go to Synapse Workspace → Develop → + SQL script

Connect to: here choose your dedicated SQL pool

```
CREATE SCHEMA Rebrickable;
CREATE TABLE Rebrickable.Minifigs
(
    SetNumber                VARCHAR(50)     NOT NULL UNIQUE NOT ENFORCED,
    LastModifiedDatetime     DATETIME2(3)    NOT NULL,
    Name                     VARCHAR(255)    NOT NULL,
    NumberOfParts            SMALLINT        NOT NULL,
    MinifigType              VARCHAR(50)     NULL,
)
WITH
(
    CLUSTERED COLUMNSTORE INDEX,
    DISTRIBUTION = HASH(SetNumber)
)
```

DDL

⇒ you are defining index type here

⇒ you are using Hash distribution on SetNumber

After executing the above SQL script you will see the schema created under:

Data → Workspace → SQL database → your dedicated SQL pool → Tables
↓
Rebrickable.Minifigs

2. Round-robin Distribution

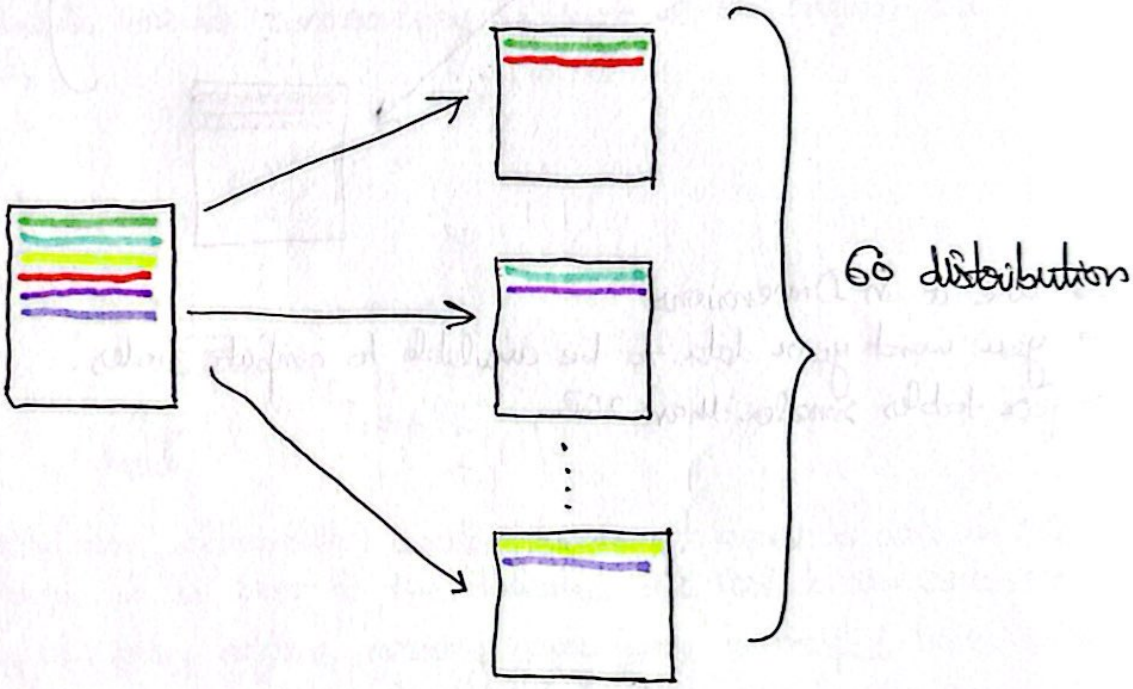
→ fastest method of loading the data as it does not involve computation of hash value

→ used where we care about insert speed like Staging Tables, where we just dump the data

→ 1st row goes to 1st distribution, 2nd row goes to 2nd dist, 3rd goes to 3rd dist. it goes on & on till row 60th goes to 60th distribution & after that 61th row goes again to 1st distribution.

→ this is default if you do not specify any distribution type.

→ no indexes ⇒ HEAP

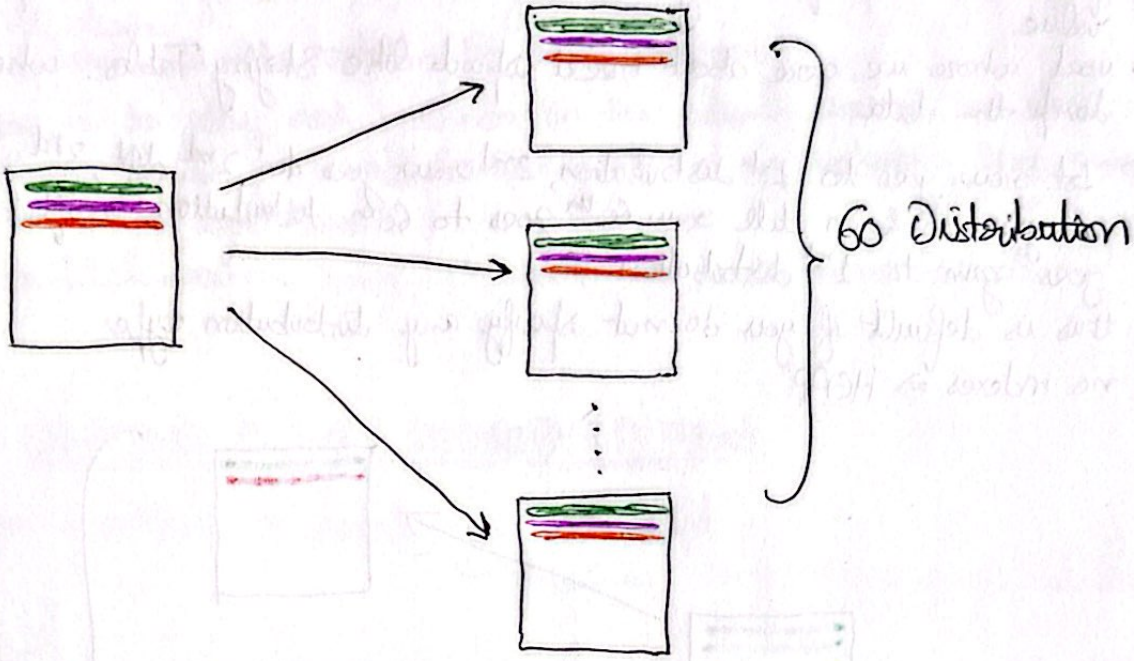



```
CREATE TABLE Rebrickable.MinifigsRoundRobin
WITH
(
    DISTRIBUTION = ROUND_ROBIN
)
AS SELECT * FROM Rebrickable.Minifigs
```

Scanned with CamScanner

3. Replicated Tables

→ creates multiple copies of our data



→ use it on Dimensions

→ you want your data to be available to compute nodes.

→ for tables smaller than 2GB.

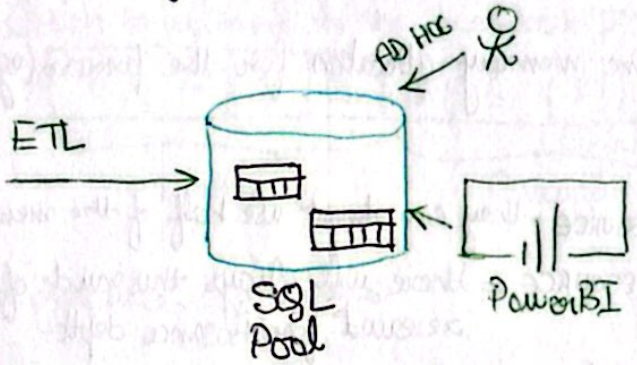
Scanned with CamScanner

Lecture 45

Synapse Dedicated SQL Pool

Workload Management

Imagine you have your dedicated SQL Pool where you have lots of tables. From one side you have your internal queries that would populate data into those tables (Pull & Push that discussed in Lecture 44) & at the same time there would be other users to consume data inside SQL Pool. There may be some reporting solutions that would want to consume data in SQL Pool. Therefore, you should handle concurrency so that all the request are handled properly.



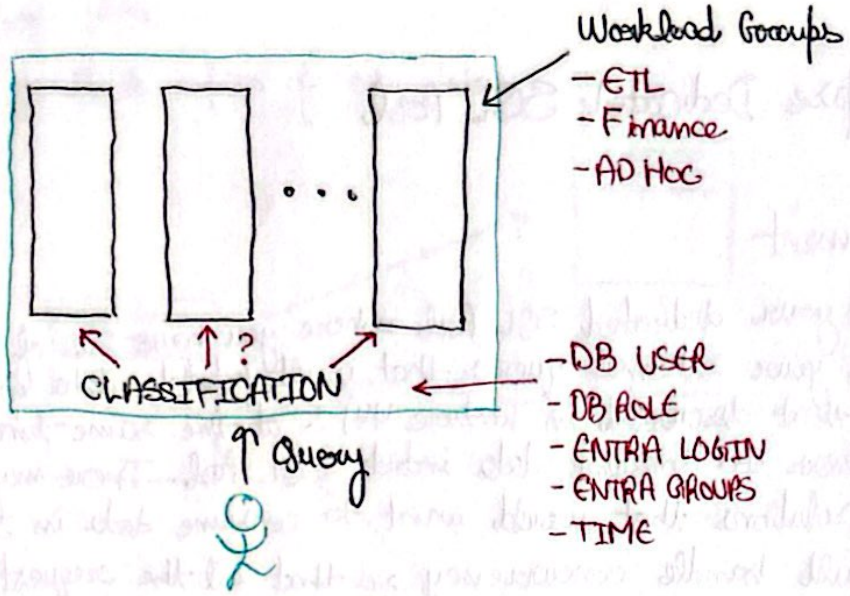
Suppose a process (user, software etc.) would like to get access to data in SQL Pool. When query is fired, it is sent to the dedicated SQL Pool to the Control node. Then classification of query happens, meaning your query is mapped based on some rules to Workload groups, some resources, that further execute the query. Workloads are simply logical groups that are made up of combination of different kind of resources to handle different type of queries.

Therefore you make different Workloads for eg. for Finance department, ETL, for AD HOC queries & design resources accordingly in those Workloads.

You create a Classifier & define some set of rules based on which classification of query takes place. And when you fire some query you specify identity of process that is submitting the query.

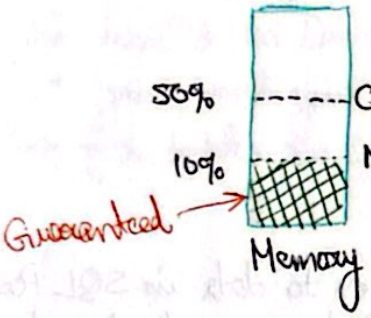
- DB USER
- DB ROLE
- ENTRA LOGIN
- ENTRA GROUPS
- TIME

Scanned with CamScanner



Example: Defining a Workload

- simply to reserve resources
- while defining workload, you define memory allocation for the process (eg. Finance deptt.



eg. you allocate 5% of memory to every request coming from Finance Deptt.

5% = REQUEST_MIN_RESOURCE_GRANT_PERCENT

But this calculation also depends on SQL Pool Service Level. For eg. if you go for DW100c ; Maximum concurrent queries = 4 ; & min % supported for REQUEST_MIN_RESOURCE_GRAT_PERCENT = 25%

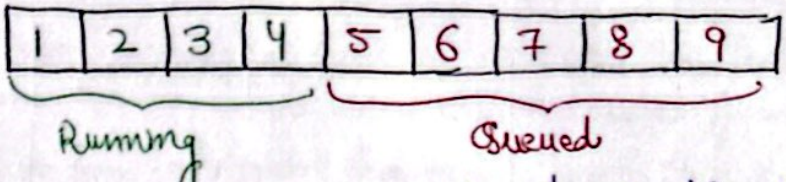
Which means to acheive high concurrency you would have to go for higher Service Level of SQL Pool.

Apart from Classification, workloads also provide workload isolation. There is also facility for ordering of the query to be executed. Meaning ex if Finance deptt, adhoc, etl fires queries at same time then how to decide which query should be executed first.

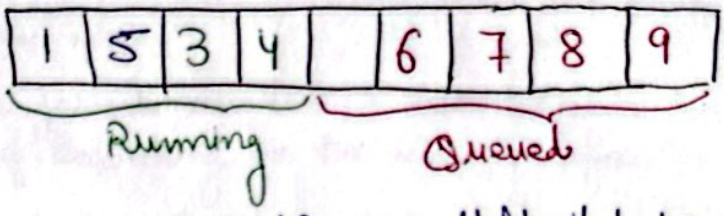
Scanned with CamScanner

there is a Scheduler for SQL Pool that works on FIFO method.

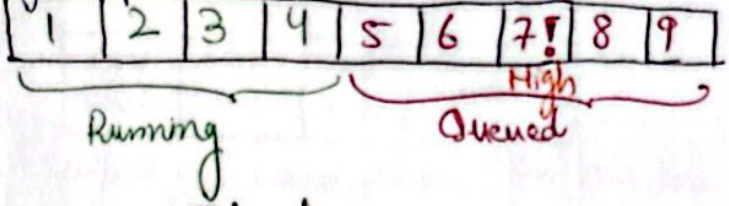
Scheduler FIFO



if query 2 is completed then 5 will take the place of query 2



Suppose query 7 is very important & we would like it to be executed before 5 & 6. You just assign a High importance to the workload group.



Level of importance : 5 levels.

- High
- Above Normal
- Normal
- Below Normal
- Low

∴ Workload Management is all about determining how concurrent query will be executed. First, they are classified by classification classifier, which then assign one of the workload group.

Scanned with CamScanner

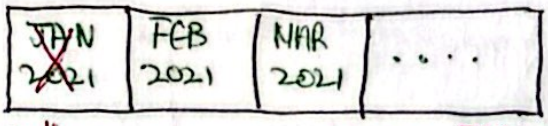
Partitioning

When you are creating a table in dedicated SQL Pool, you might partition your data on some column. End users are abstracted from partitioning.

We usually go for partitioning for maintainence. Suppose you have a big fact table that you already have partitioned & your business logic is that you keep facts for 3 years & older data can be removed. And every month you delete data of oldest month to make space for new data.

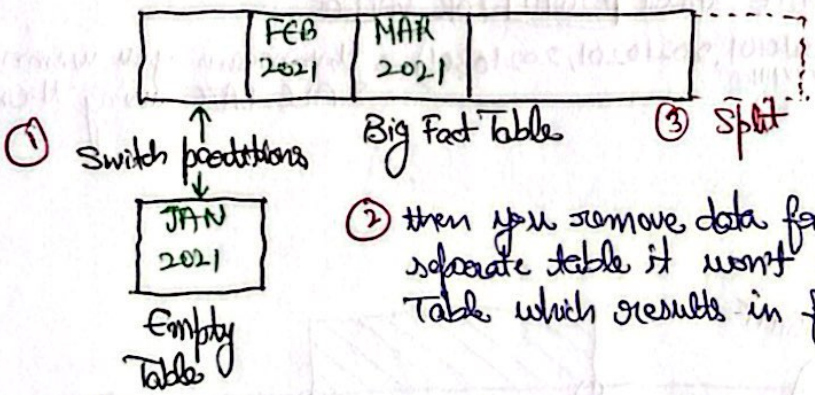
One way could be just to use DELETE clause but since fact table is too big, there may arise scenario of deadlocking on the table. Therefore to avoid this go for partitioning

Suppose you have partitioned data on date (month)



suppose you wanna remove JAN 2021 data

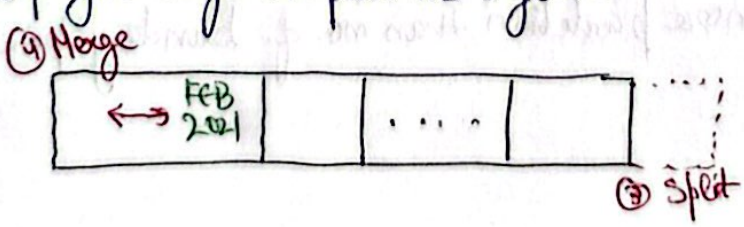
We will create an another empty table whose schema would exactly same as structure of fact table. First operation would be to switch partitions. You switch data from 1st partition to the empty table.



② then you remove data from empty table. Since it is a separate table it won't cause any load to the Big Fact Table which results in faster execution/operation

Then to make space for new partitions you split the new partition & add new range for month.

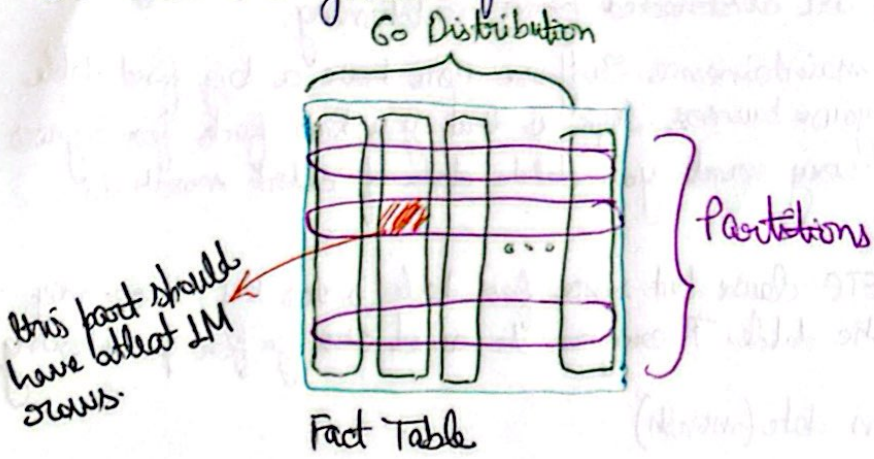
As a last step you merge the partitions together.



Scanned with CamScanner

- best practice is to have partitions less than 100.

In case of big tables you can index them using CLUSTERED COLUMN STORE INDEX
But there is a guideline for that



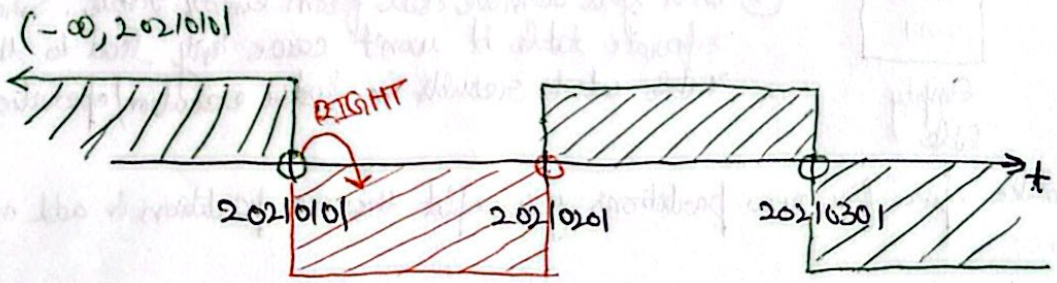
Syntax:

```
CREATE TABLE
  :
  :
WITH(
    DISTRIBUTION = HASH(CUST_ID),
    PARTITION( SALE_DATE RANGE RIGHT FOR VALUES
                (20210101, 20210201, 20210301)
```

YYYYMMDD

these values must be present there

⇒ this means you wanna split the SALE_DATE using these 3 points



There is always 1 more partition than no. of boundary points.

Scanned with CamScanner

PIVOT | UNPIVOT

Type	Parts
A	10
B	2
C	1
A	2

PIVOT ⟹

A	B	C	
12	2	1	Sum(PARTS)

↓ UNPIVOT

IDENTITY

in order to have a Surrogate key that increments automatically

```
ID    INT    IDENTITY(1,1) NOT NULL
```

start with 1 → increment by 1

Detecting Data Skew

```
DBCC PDW_SHOWSPACEUSED("Rebrickable.Minifigs")
```

this will result out in providing number of rows in all the distribution

Lecture 46

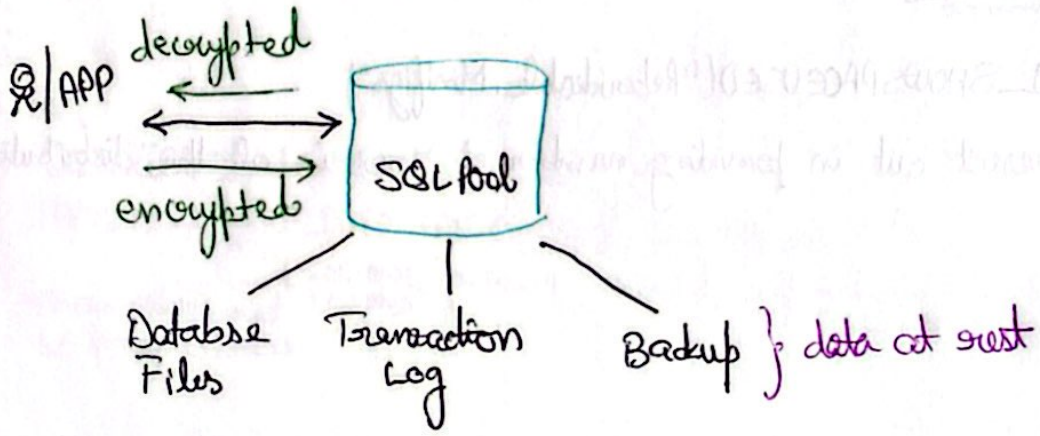
Dedicated SQL Pool Security

1. Firewall (first layer of defence)

→ we protect first layer of security, i.e, networking by enabling firewalls
→ they are set at workspace level
Go to Synapse Workspace → Security → Networking

firewalls restrict access from networking point of view. You define "Rule name" & there you provide range of IPs that could access your SQL Pool. But still it will be using "Public Endpoints" that is not considered to be used in some big companies.
Therefore, an option is to go for "Private endpoint connections" under "Security"

2. Transparent Data Encryption



→ it encrypts data at rest
→ eg. if someone stole your backup, & if if this data is not encrypted the thief can restore this backup. Therefore, to protect the data you enable TDE.
→ transparent means whole encryption part is abstracted from consumers
→ suppose a consumer wants to read the data. But the consumer would want to have decrypted data. This issue is handled by SQL Pool automatically.
∴ when data is read by an application, SQL Pool decrypts the data & when an Application wants to write some data, SQL Pool encrypts the data.

Scanned with CamScanner

- Go to Synapse Workspace
- Since TDE is enabled per SQL Pool so go to specific SQL Pool

  Analytics Pools → SQL Pools → Go to your pool → from the menu at left navigate to Security → Transparent data encryption

  ↓

  Data encryption

  ON OFF

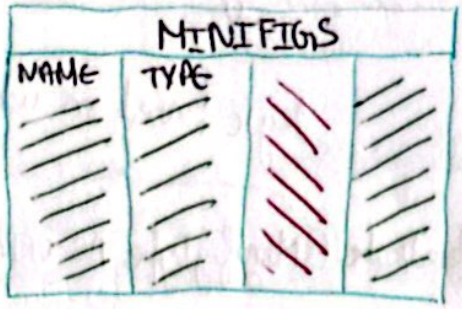
  ↓

  Save

- This method does not restrict the access for some consumer to view some particular columns or rows.

Column Level Security

→ protects data in more granular way.
→ specify set of columns that a consumer can view

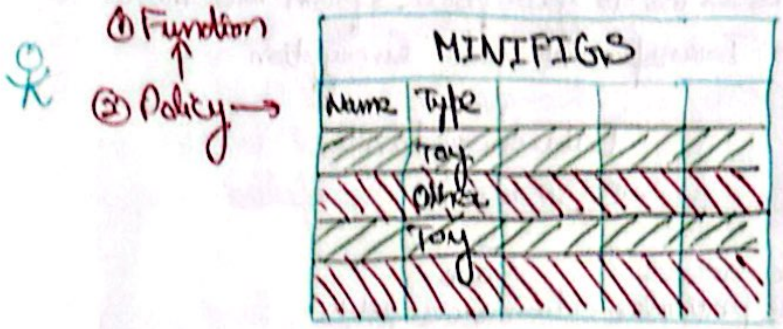


eg: Create 2 logins for 2 users

```
-- On master DB
CREATE LOGIN Student WITH PASSWORD = 'MySecretPa$$1'
CREATE LOGIN Teacher WITH PASSWORD = 'MySecretPa$$1'

-- On DWHTest DB
CREATE USER Student FROM LOGIN Student
CREATE USER Teacher FROM LOGIN Teacher

-- Grant permissions
GRANT SELECT ON Rebrickable.Minifigs (SetNumber, Name) TO Student
GRANT SELECT ON      "      TO Teacher

EXECUTE AS USER = 'Student'
SELECT * FROM Rebrickable.Minifigs
EXECUTE AS USER = 'Student'
SELECT USER_NAME() AS 'User', SetNumber, Name FROM Rebrickable.Minifigs
EXECUTE AS USER = 'Teacher'
SELECT USER_NAME() AS 'User', * FROM Rebrickable.Minifigs
```

Scanned with CamScanner

Row-level Security



- for this you create a function that contains logic. The function ~~ex~~ returns either 1 or 0.
- to make use of this function, you create a Policy that uses the function & is set on the table.

```
CREATE SCHEMA Security
-- create predicate
CREATE FUNCTION Security.securityPredicate(@MinifigType AS VARCHAR(255))
    RETURNS TABLE
WITH SCHEMABINDING
AS
    RETURN SELECT 1 AS securityPredicateResult
    WHERE
        @MinifigType IN('Droid', 'Toy') AND USER_NAME() = 'Teacher'
        OR USER_NAME() = 'DBO'
GO

-- create policy
CREATE SECURITY POLICY MinifigsFilter
ADD FILTER PREDICATE Security.securityPredicate(MinifigType)
ON Rebrickable.Minifigs
WITH (STATE = ON);
-- execute as me
SELECT MinifigType, COUNT(*) FROM Rebrickable.Minifigs GROUP BY MinifigType
-- execute as Teacher
EXECUTE AS USER = 'Teacher'
SELECT MinifigType, COUNT(*) FROM Rebrickable.Minifigs GROUP BY MinifigType
```

logic based on what column

teacher can see 'Droid' & 'Toy'

already created

parameter to func. as column

Scanned with CamScanner

Dynamic Data Masking

→ this is not encryption.

→ if we have sensitive data in a table or you would like to show some dummy data for some specific column whose data is sensitive.

Mask	Real Value	Result
1. Default	Strings Numeric Date	XXXX 0 1900-01-01
2. EMAIL	pioka@MS.com	pXXX@xxx.com
3. Random	(for integer column) 3000	[1000, 10000]

NOTE: Admins can see Real values even if mask is applied

some random value will be assigned to the real value ~~is~~ between from the range b/w [1000, 10000]

```
-- create default mask on LastModifiedDatetime
ALTER TABLE Rebrickable.Minifigs
ALTER COLUMN LastModifiedDatetime ADD MASKED WITH (FUNCTION = 'default()');

-- create random mask on NumberOfParts
ALTER TABLE Rebrickable.Minifigs
ALTER COLUMN NumberOfParts ADD MASKED WITH(FUNCTION = 'random(1,100)');

--note that NumberOfParts changes every time
EXECUTE AS USER = 'Teacher'
SELECT *FROM Rebrickable.Minifigs WHERE SetNumber = 'fig-010440' ORDER BY NumberOfParts DESC

--Guessing the data is possible
EXECUTE AS USER = 'Teacher'
SELECT *FROM Rebrickable.Minifigs WHERE NumberOfParts = 18
```

Scanned with CamScanner

Data Classification

At dedicated SQL Pool level under "Security" there is "Data Discovery & Classification" you can here assign a label to your data for eg. assigning label to sensitive data

Add classification → choose Schema name
Table name
Column name
Information type
Sensitivity label
↓
Save

this might be useful in various type of Security Audits. You can show Security Auditor if you have any sensitive data, what column is sensitive.

This feature along with "Auditing" give answer to the question what type of query is being executed to the sensitive data.

"Auditing" is simply creating log of every query executed on our dedicated SQL Pool.

Enable it

Scanned with CamScanner

Lecture 49

Introduction to Streaming

Event Hubs

Batch Vs Streaming

So far we have done Batch processing that would run our process once or twice a day preferably during night when we do not want to overload our source system as well our process.

But in scenarios where we want our solution to act or react promptly then action has to happen immediatly, we must go for Streaming.

Some use cases when Real-time or Near Real-time processing should happen:

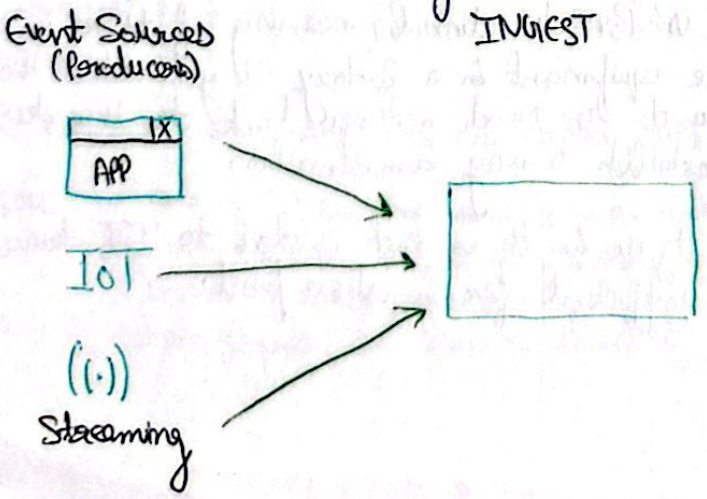
→ In a hospital where some pateints have heart related issue & they have heartbeat sensors attached to them. If their heartbeat drops below certain levels then the concerning Nurse or Doctor should be notified

→ In industry where there are sensors that capture temperature & humidity levels. If there is some weird reading coming from a machine then the m/c should be shut-off to avoid big damage.

* In case of Streaming, streams don't have start or end of data. You split the data into parts & then analyse the data frame by frame/window by window.

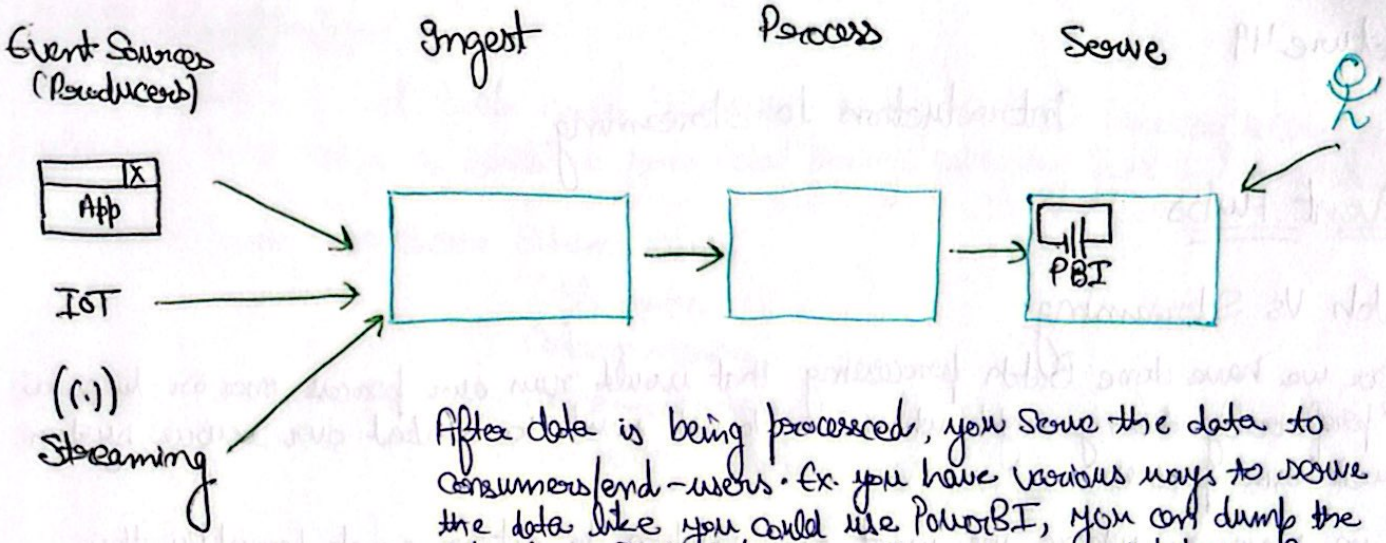
We have to have some source of data & in Streaming we call them Event Producers like some Applications, IoT devices, smart home devices (maybe you wanna detect some movement when there is no one house), various type of Streaming sources.

Those event producers constantly produce data.



But somehow you wanna grab those data in Azure? Here you cannot use ADF.

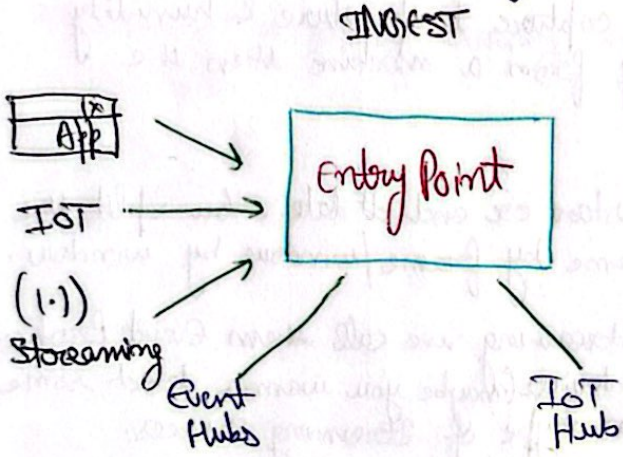
After you have your data inside Azure you would like to process the data



After data is being processed, you serve the data to consumers/end-users. Ex. you have various ways to serve the data like you could use PowerBI, you can dump the data in a Data Lake or store it in a database for some further processing.

Ingest Part

there are 2 services to ingest events



⇒ entry point to Azure

these are the services to which event producers will send the data

- IoT Hub works with IOT devices & is Bi-directional, meaning if there is some sensor measuring temperature of some equipment in a factory. If you detect that temperature is too high, you might want to send message back to the device to shut-down itself. Therefore IoT can establish 2 way communication.
- If you want to use Streaming in Azure & it is not related to IoT devices then mostly likely Event Hubs are sufficient for ingestion part.

Scanned with CamScanner

- Event Hubs are compatible with Apache Kafka
- Event Hubs integrate nicely with Stream Analytics & Databricks

databricks can handle streaming as we saw previously in case of Autoloader, structured streaming was used.

Creating Event Hubs

Go to Azure Portal → Event Hubs → Create → Create Namespace

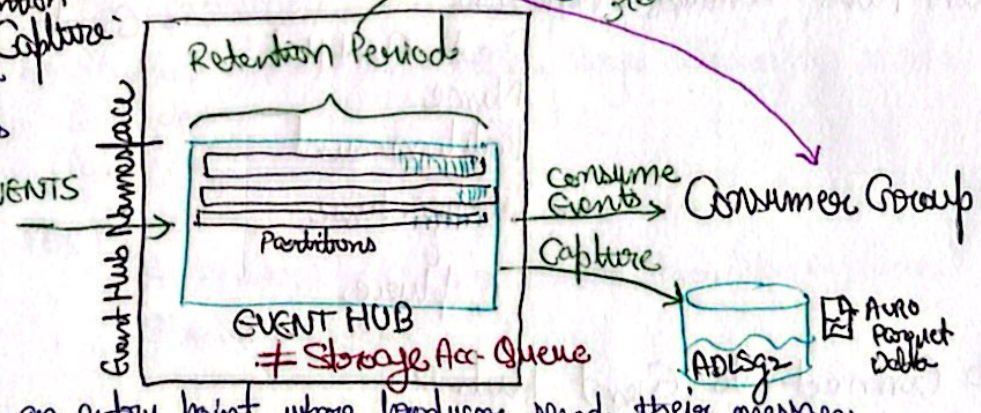
Subscription
└ Resource Group

Namespace name
Location
Pricing tier ← Browse the available plans & their feature

Choose your pricing tier

- Consumer group: → this is important because maybe there are multiple consumers who are connected to Event Hub & each of them wants to read same event.
- Message Retention
- Capture

whenever an event arrives Event Hub & consumer consumes the event, it still stays inside the Event Hub. Therefore, events don't get deleted after a process consumes it. ∴ you can read same event multiple times



- event hub is just an entry point where producer send their messages
- inside Event Hub we have partitions
- whenever Event comes inside an Event Hub, it goes to some partition
- at the o/p part of Event Hubs you would like to consume those events from partitions
- our downstream applications are called Consumer Group

Capture: in case of Streaming maybe you would like to run some batch processing on Events. So you would like to keep the data for longer period than Message Retention period. This feature configure Event Hub to store Events on Storage Account

Scanned with CamScanner

- TU in Event Hubs

TU = Throughput Unit

Ingress: events coming to Event Hub / incoming data.
upto 1MB/s or 1k events/s

Egress: how fast data is read from Event Hubs.
upto 2MB/s or 4k events/s.

Throughput Units ○———— [1]

Enable Auto-Inflate [✓]

ex. you kept it low & if you feel it will not be enough to process your data then you have Auto-Inflate option

this will automatically increase no. of throughput units to accomodate workloads.

- So for you created a Namespace, container for Event Hubs.

Go to Event Hubs → Entities → Event Hubs → Create

Create Event Hub
Name
Partition count
Retention time
↓
Capture

How to Connect to Event Hub?

- how processes should connect to Event Hub
- how we make sure only authorized applications can connect

Go to your Event Hub → Settings → Shared access policies → Add

Add SAS Policy

3 types of permissions
- Manage → gives admin permissions
- Send → who can send events to Event Hub
- Listen → who can consume/read data

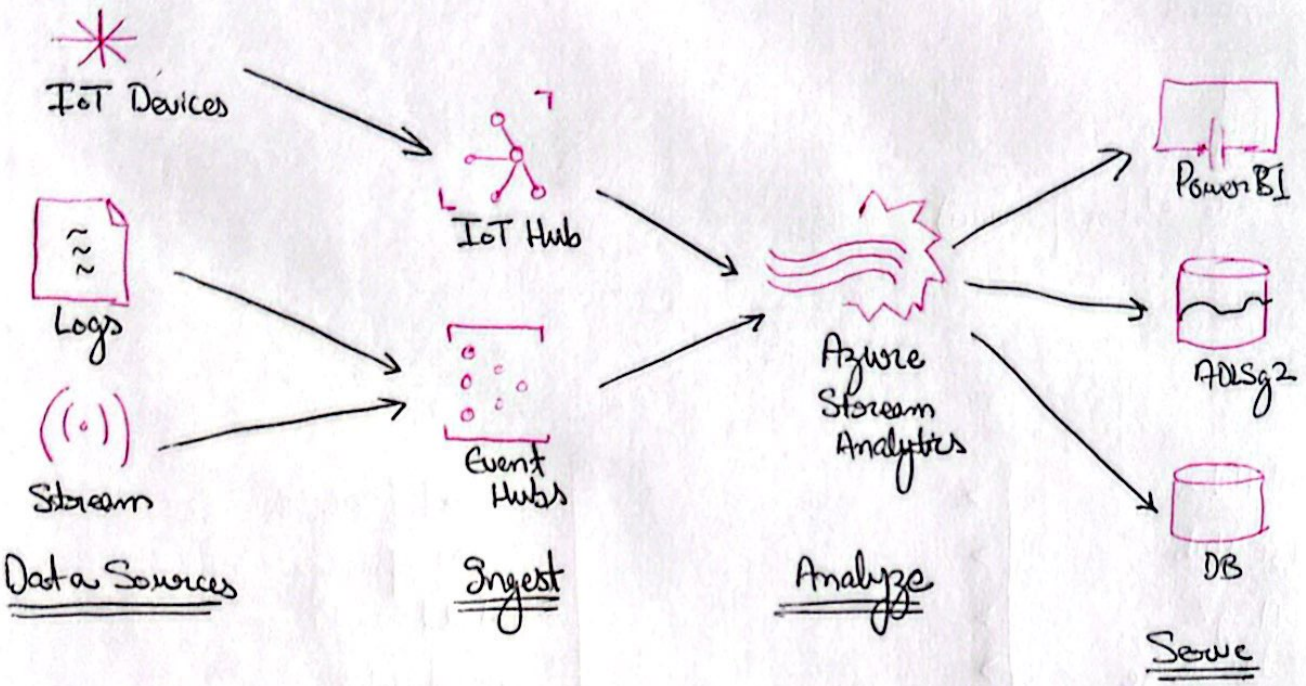
↓
Create

Now if you open your recently defined policy you shall see
- PRIMARY KEY
- SECONDARY KEY
- Connection string-primary key
- Connection string-secondary key

now to make it work you give this connection string to Event producer.

Scanned with CamScanner

Azure Stream Analytics



- Checkpointing is done automatically
- Stream Analytics is designed to analyze & process large volumes of streaming data with sub-millisecond latencies.
- It is fully managed (PaaS) offering on Azure.

Stream Analytics Jobs : A service you create inside Azure to implement Stream Analytics.

Stream Analytics jobs → +Create
- Subscription
  - Resource group
- Instance details
  - Name
  - Region
  - Hosting environment
- Streaming unit details.

Scanned with CamScanner

v Job topology

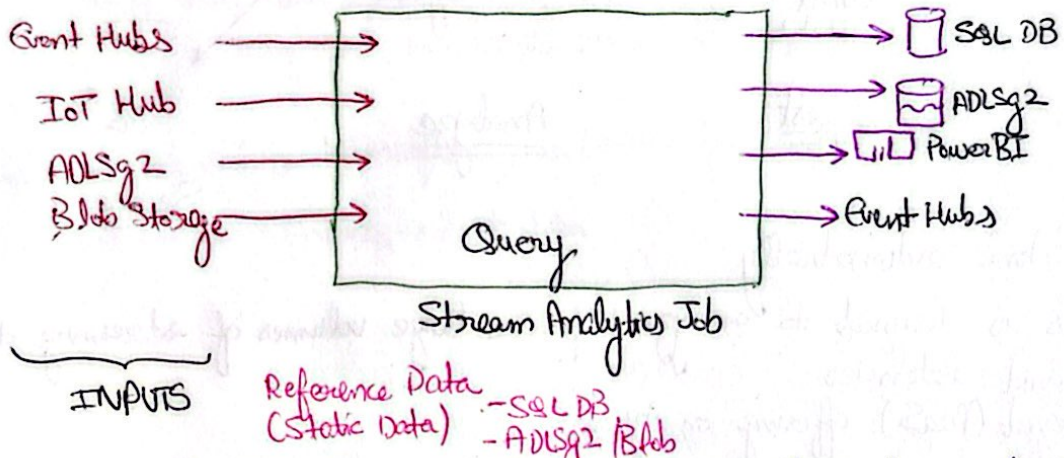
Inputs

Functions

Query

Outputs

No-code editor (preview)



- Query connects data from several i/ps, process it and send the results to downstream (o/ps)
- Queries are written in SQL-like syntax.
- you can write customized functions using JavaScript.
- if we choose Event Hubs as input then you would require to provide Authentication mode from inside your Stream Analytics Job. This defines how Stream Analytics Job should connect to the Event Hub.

  * Grant Azure Event Hubs Data Receiver role to Managed Identity of your Stream Analytics job.

Sample Query:

```
Select
*
INTO
  [Output]
FROM
  [Input]
```

Scanned with CamScanner