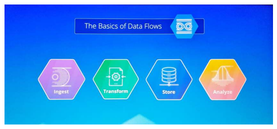
Dataflows in Azure

Thursday, 12 January 2023 5:00 PM

Dataflows basic

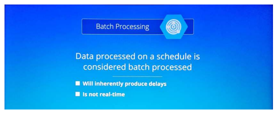
Data flow → encompasses the initial ingestion of data, any required transformations, storage and analysis.
→ Data flow is essentially about what needs to happen with data in order to meet business requirements and how it can be used to answer questions about. the business.



The Basics of Data Flows

Terminology

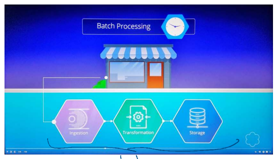
Batch Processing: it's useful in cases where there is a large amount of data quickly moving into a system.
→ Data over course of the day.



Batch Processing

Data processed on a schedule is considered batch processed

- Will inherently produce delays
- Is not real-time



Batch Processing

"End of the day"
"Overnight"

3 states

Stream processing
→ Analysis of the data as it's received.
→ While it may not quite be real-time processing, delays are tipically sub-seconds.
→ As data comes into the pipeline, analysis is performed and results are generally available within seconds, or even milliseconds.
→ It requires technology that can handle the processing continuos of data in-stream. Example Lamba
→ lambda supports (batch and stream processing), incidentlly
por cierto
→ Batch or stream for ETL and ELT.

ETL vs ELT → Both concepts refer to the process of obtaining data, extracting it from its source and transforming it.
→ the transformed data is then stored for analysis.
ETL → transforms the data before loading and storing it.
ELT → loads collected data before transforming it. which usually means data can be handled at greater scale.

Hybrid Processing
→ Data flow scenario, data that's processed, used and stored is generally distributed among cloud and on-system.
→ As such, the dataflow it self. will often travel from on-prem to cloud, and maybe even vice versa.
→ Bandwith. (from on-prem up to the cloud)
→ It's important to assess → the size of the pipe,
→ the latency
→ Cost implications of data transfer. (ingress/egress to and from the cloud)

→ An important consideration is how the connection from the on-prem enviroment to the cloud is constructed.
→ While a site-to-site VPN might be sufficient, latency issues may dictate that an Express Route be considered instead.

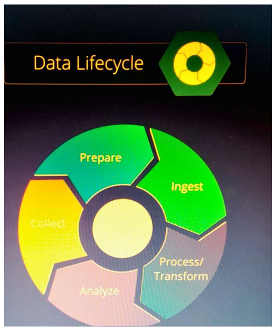
ETL → Schema on write. → because the data is first transformed into some standard format BEFORE

ETL → Schema on write. → because the data is first transformed into some standard format BEFORE it's written to storage.
T→in memory

ELT → Schema on read → there is no schema enforced on the data during initial ingestion. Instead, the data is transformed AFTER it's been stored and while it's being used.

When considering ETL vs ELT, the main driver generally comes down to scale: Using ETL requires data to be transformed before it can be loaded, this means there is lots of computer power needed.
→As such, this can negatively impact the ability to process large amounts of data.
→ELT, instead, separates data ingestion from the transformation process does is allow huge amounts of data to be ingested and loaded BEFORE it is transformed
By breaking the process apart in ELT, it's possible to ingest lots more data than with ETL.
→You can essentially ingest data as fast as it's written.
Ultimately, if scale is a concern, ELT is the preferred strategy over ETL

Data life cycle



The different stages of the lifecycle affect data flow.

Collection: data is acquired from other processes or