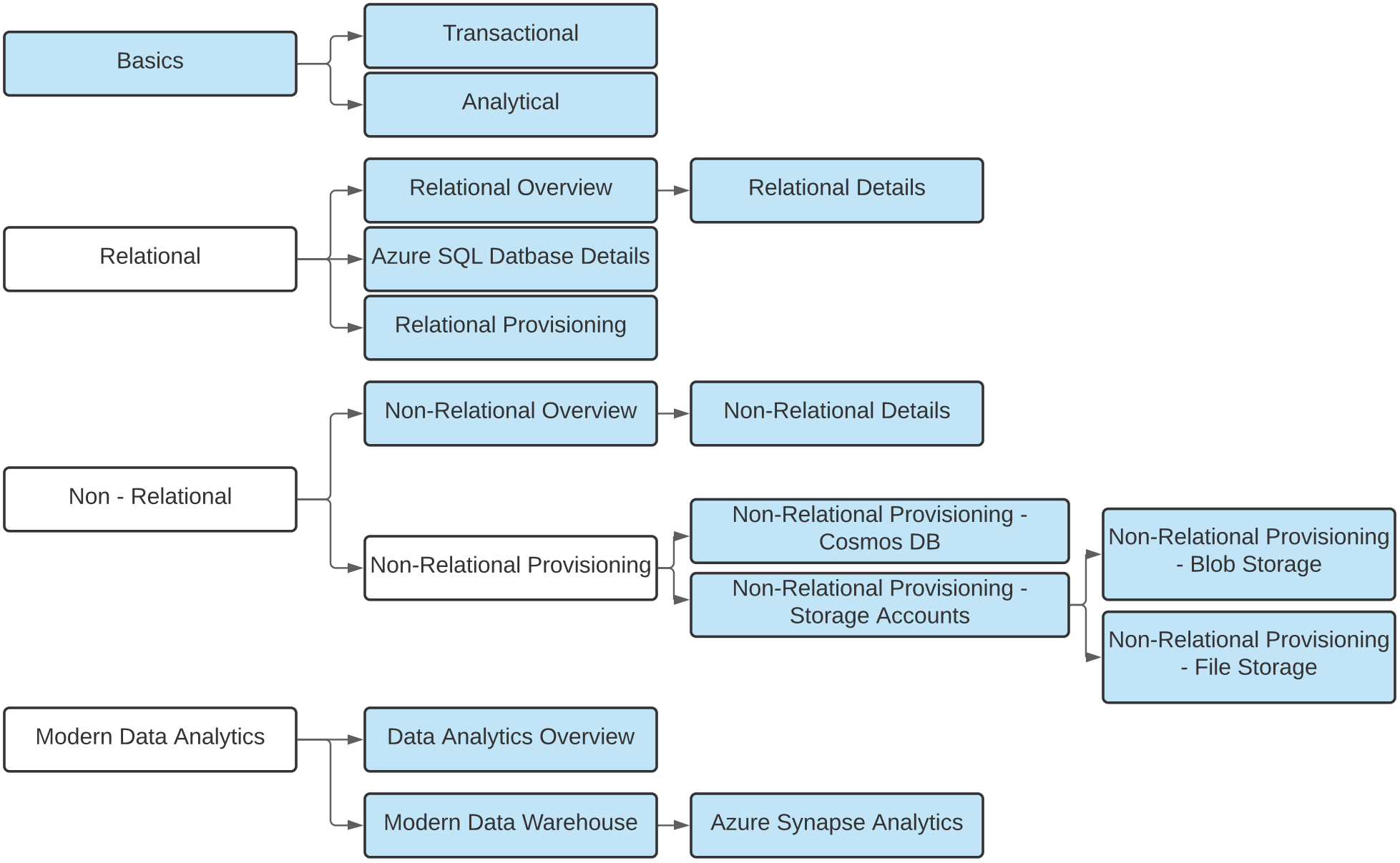
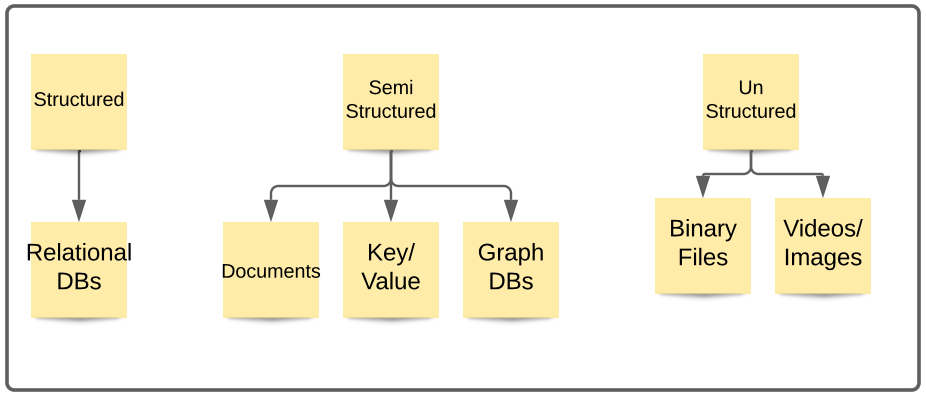
AZ900 Study Guide

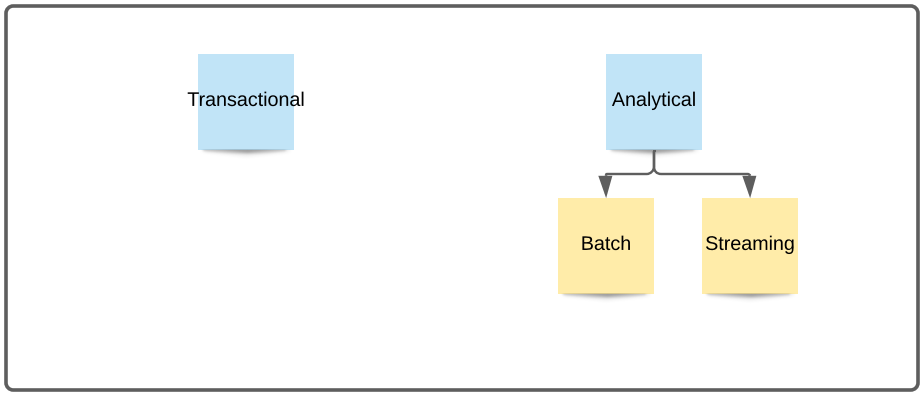


Basics

Data



Data Processing



Data Roles

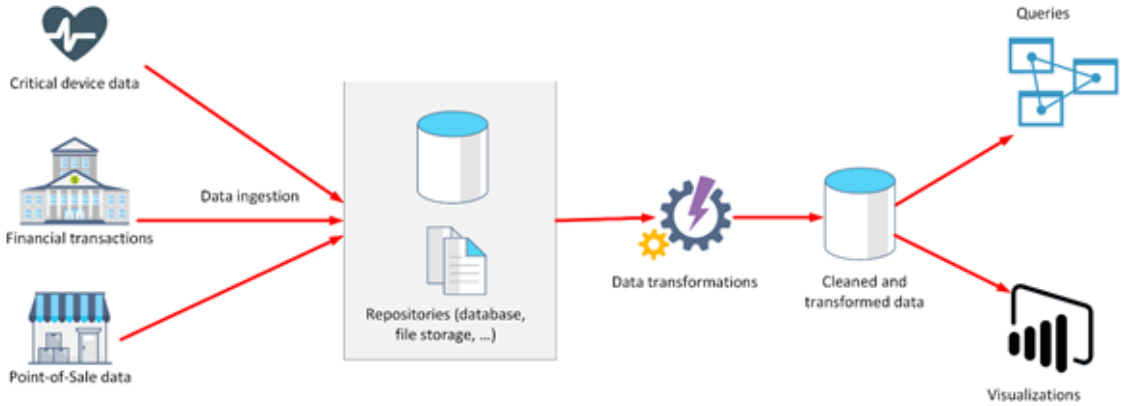
Database Administrator (DBA)	An Azure database administrator is responsible for the design, implementation, maintenance, and operational aspects of on-premises and cloud-based database solutions built on Azure data services and SQL Server
Data Engineer (DE)	Vital in working with data, applying data cleaning routines, identifying business rules, and turning data into useful information. Implement data-related assets that include data ingestion pipelines, cleansing and transformation activities, and data stores for analytical workloads.
Data Analyst (DA)	Explore and analyze data to create visualizations and charts to enable organizations to make informed decisions. They're responsible for designing and building scalable models, cleaning and transforming data, and enabling advanced analytics capabilities through reports and visualizations. A data analyst processes raw data into relevant insights based on identified business requirements to deliver relevant insights.

Transactional (records transactions)

High volume
- Accessed very quickly
- OLTP (Online Transaction Processing)
- Normalized data
- Can be complex to query
- ACID (Atomicity, Consistency, Isolation, Durability)
  - Atomicity guarantees that each transaction is treated as a single unit, which either succeeds completely, or fails completely.
  - Consistency ensures that a transaction can only take the data in the database from one valid state to another.
  - Isolation ensures that concurrent execution of transactions leaves the database in the same state that would have been obtained if the transactions were executed sequentially.
  - Durability guarantees that once a transaction has been committed, it will remain committed even if there's a system failure such as a power outage or crash.
- Inherently complex

Analytical (supports analysis)

- OLAP (Online Analytical Processing)
- Analytical Processing System Steps
  a. Data Ingestion
  b. Transformation/ Processing (Batch/ Streaming)Query/ Visualization



Batch Processing

Vast amounts of data need to be transferred into a data analysis system and the data is not real-time.

An example of ineffective batch-processing would be to transfer small amounts of real-time data, such as a financial stock-ticker.

Stream Processing

Streaming data processing is beneficial in most scenarios where new, dynamic data is generated on a continual basis.

Stream processing is ideal for time-critical operations that require an instant real-time response.

Differences between Batch and Streaming

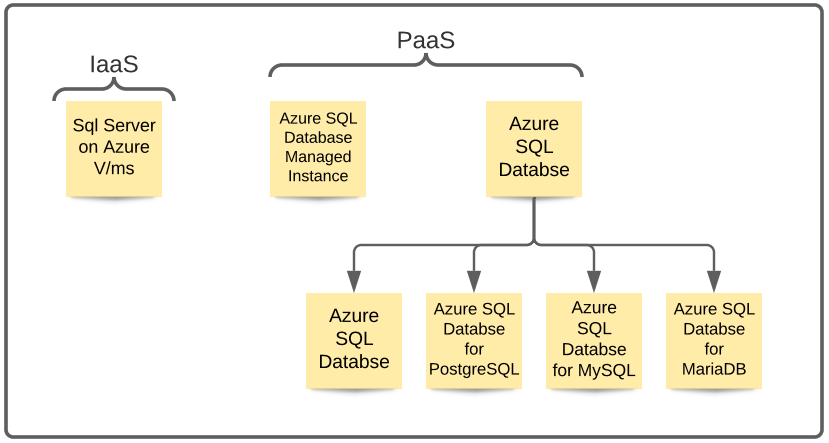
Data Scope: Batch processing can process all the data in the dataset. Stream processing typically only has access to the most recent data received, or within a rolling time window (the last 30 seconds, for example).

Data Size: Batch processing is suitable for handling large datasets efficiently. Stream processing is intended for individual records or micro batches consisting of few records.

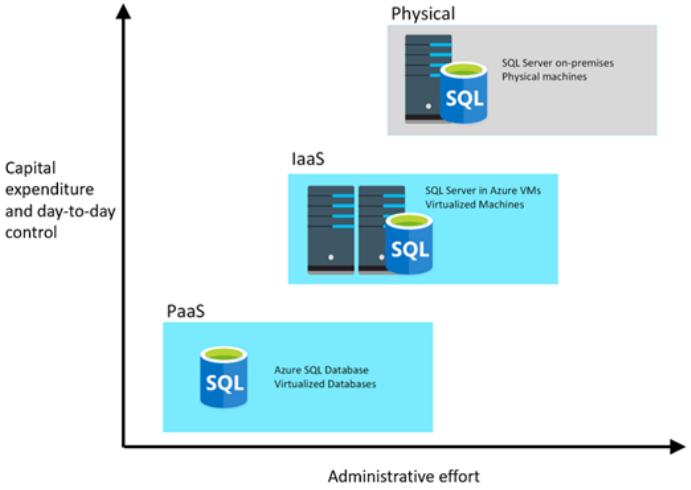
Performance: The latency for batch processing is typically a few hours. Stream processing typically occurs immediately, with latency in the order of seconds or milliseconds. Latency is the time taken for the data to be received and processed.

Analysis: You typically use batch processing for performing complex analytics. Stream processing is used for simple response functions, aggregates, or calculations such as rolling averages.

Relational Overview

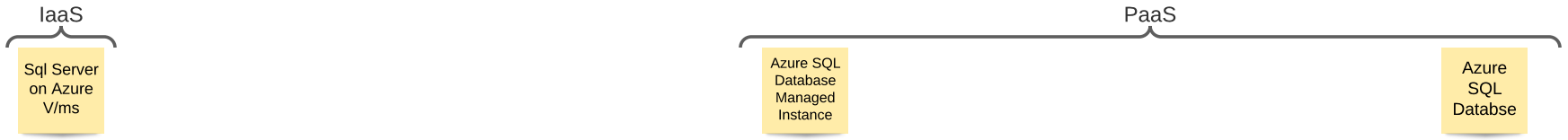


- Tables with columns and rows
- ALL rows in the same table have the same columns
- Referential Integrity
- The main characteristics of a relational database are:
  - All data is tabular. Entities are modeled as tables, each instance of an entity is a row in the table, and each property is defined as a column.
  - All rows in the same table have the same set of columns.
  - A table can contain any number of rows.
  - A primary key uniquely identifies each row in a table. No two rows can share the same primary key.
  - A foreign key references rows in another, related table. For each value in the foreign key column, there should be a row with the same value in the corresponding primary key column in the other table.
- Use SQL for DDL and DML
- OLTP
  - Banking solutions
  - Online retail applications
  - Flight reservation systems
  - Many online purchasing applications
- Indexing utilized (clustered – where/ how the data is stored on disk and non-clustered stores a reference to where the data lives)
- Views – virtual tables



	On-premises	Cloud
Personal control of data security	X	
Scalable		X
Hardware maintained		X
Software maintained		X
Low capital expenditure		X
Low operational expenditure	X	

Relational Details



SQL Server on Azure v/m's

- Lift and shift – fast migration to the cloud
- Best for migration to cloud or extending environment to cloud to create hybrid
- Good for
  - Dev/ test environments
  - Becoming lift/shift readyResizing the v/ms

Azure SQL Database Managed Instance

- Fully controllable instance of SQL Server in the cloud
- Still automates backups/ patching/ monitoring
- Relies on other Azure services
- Lift and sift ready without having to manage V/ms
- Linked servers, message broker, database mail ALL supported
- Near 100% compatibility w/ SQL Server Enterprise
- SQL Server logins and Azure AD supported

Azure SQL Database

- Managed db server in cloud for your DB to sit on top of
- Single DB
  - Quickly setup a single DB
  - Scale the DB
  - Just create tables, load and go
  - Resource pre-allocated to your managed server
  - Also has a serverless option but your data would like on same machine as someone else's
- Elastic Pool
  - Multiple of your DB's can share the same resources (pool) multi-tenancy
  - Create the pool then your DBs use them
  - Great for varying resource requirements – helps reduce cost
- Lost cost / minimal administration
- Great for new cloud projects
- Easy scaling w/o upgrading hardware, etc.
- Automatically patched and upgraded SQL Server software
- HA of 99.99%
- Point in time restore
- Can be replicated to multiple regions
- Advanced threat protection
- Audits can be sent to Azure audit log
- Encryption
  - In Transit (TLS)
  - At rest (TDE)
  - Always (AE)
- CANNOT USE LINKED SERVERS!!!

Azure SQL Database

DML (SELECT, INSERT, UPDATE, DELETE)
DDL (CREATE, ALTER, DROP, RENAME)

MySQL	MariaDB	PostgreSQL
• Very popular • Available as free Community edition or paid-for, and more functional, Standard and Enterprise editions • Azure Database for MySQL is based on the free Community edition, but adds high availability and scalability	• Compatible with Oracle Database • Built-in support for temporal data	• Can store both relational and non-relational data • Can store geometric data • Extensible

Database Migration Service

- All restore of backup directly to DB in cloud (MySQL, MariaDB, or PostgreSQL)
- Can help ease the pain of moving over

Azure SQL Database

Querying:
- Transact SQL (T-SQL)

Tools for Azure SQL:
- Azure Portal
- cmd line / Cloud Shell using SQLCMD
- Azure Data Studio
- SSMS
- SSDT

Azure SQL Database for MariaDB

- Community Edition
- High availability features built-in.
- Predictable performance.
- Easy scaling that responds quickly to demand.
- Secure data, both at rest and in motion.
- Automatic backups and point-in-time restore for the last 35 days.
- Enterprise-level security and compliance with legislation.
- The system uses pay-as-you-go pricing so you only pay for what you use.

Querying:
- PL\SQL (oracle)

Tools for MySQL:
- MySQL workbench
- Azure cloud shell

Azure SQL Database for MySQL

- PaaS MySQL Community w/ HA and scalability
- Automate backups / point in time restore

Querying:
- PL\SQL (oracle)

Tools for MySQL:
- MySQL workbench
- Azure cloud shell

Azure SQL Database for PostgreSQL

- Can only use a list of supported extensions
- Single Server: (three pricing tiers based on load)
  - Basic
  - General Purpose
  - Memory Optimized
- Hyperscale (Citus)
  - Scales queries across multiple server nodes
  - DB split across nodes based on a partition/ sharding key
- HA included
- pgAdmin Tool included and used to manage these
- azure_sys captures queries for tuning/ monitoring

Querying:
- pgSQL (PostgreSQL)

Tools for PostgreSQL:
- pSQL from Azure Cloud Shell or locally from cmd prompt (need to install pSQL client)
- pgAdmin
- Azure Data Studio using PostgreSQL extension

Provisioning Relational Database

Definitions

Provisioning – act of running a series of tasks to create and configure a service, Azure will setup the various other resources, you just specify the size of these resources

Scaling – increasing or decreasing resources

Options/ Parameters to provision MySQL/ PostgreSQL

- Subscription
- Resource group
- Server name (unique – public facing)
- Data source
- Location
- Version
- Compute and Storage
  - Basic – dev/ testing, small infrequent use
  - General Purpose – business workloads (web/ mobile apps)
  - Memory Optimized – in memory performance (transactional / analytical high performance apps
- Admin username
- Admin Password

Tools Used

- Azure Portal
- Azure CLI – cmd prompt or on the cloud shell (great for automation)
- Azure PowerShell – commandlets
- Azure Resource Manager Templates – JSON files to template deployment of resources – then run CLI or PowerShell to run the template

Connectivity/ Security

Connectivity/ Firewalls

- Default is to ***deny*** all access to all
- Firewalls and virtual Networks page > selected networks >
  - Virtual networks (on Azure cloud)
  - Firewall (on-prem)
  - Exceptions (access to any other services in your subscription)

Access Control

- Defines what a user or app can do with a resource once they have been authenticated
- Azure RBAC (Role Based Access Control)
  - Principal – object (user/ group/ service principal)**WHO**
  - Definition – collection of permissions (Built-in and custom)**WHAT**
  - Scope – set of resources that the access applies to**WHERE**
- Setup in IAM (Access Control) page in Azure Portal

******* Allow only the connection and communication necessary to allow service to operate*******

Connecting from within Azure:

- Connection policy of redirect by default
- After established connection to the DB through the gateway, all subsequent requests will go straight to DB
- If connectivity is dropped the app will need to reconnect through the gateway

Connecting from Outside Azure:

- Uses proxy mode by default
- Connection first established through gateway and all subsequent requests flow through that gateway

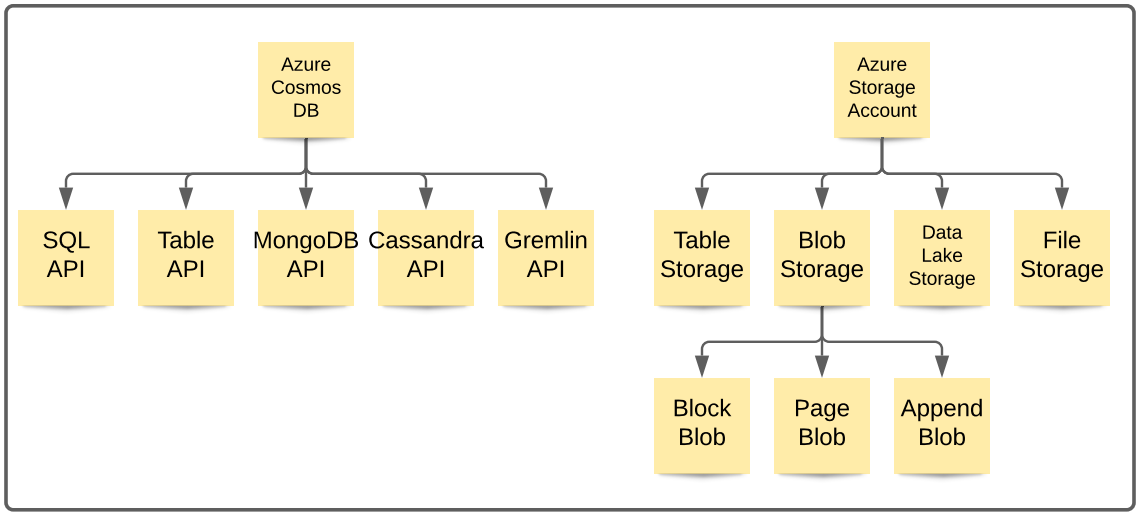
DosGuard:

- Tracks failed logins from IP address
- Blocks for a bit if repeated failures
- Validates all connections currently to server
- Encrypts all communications
- Packet inspection

PostgreSQL/ MySQL Read Replicas

- Up to 5 replicas
- Replicated asynchronously
- Helps read-intensive workloads

Non - Relational



- Capture data very quickly
- Very flexible
- Collections/ Containers
- Two entities in the same collection can have different fields
- Self-describing with labels on each field
- Can nest the subfields
- Application must but built to parse this data
- Generally use a key and value format where the application uses the key, filtering on other fields is challenging unless advanced indexing is used
- Use Cases:
  - IOT/ Telemetry data
  - Retail/ Marketing Data
  - Gaming
  - Web App/ Mobile Apps
- Semi Structured Data Formats
  - JSON (row based)
  - AVRO (row based – highly compressible)
  - ORC (column based)
  - Parquet (column based)

NoSQL Database Types

Key/ Value Stores

- Azure Table Storage/ Cosmos DB w/ Table API
- Quickest for inserting/ querying
- Items stored in key order
- No access to values for filtering, selection based only on keys
- Inserts and deletes only (no concept of update)
- Good for stream ingestion

Document DB

- Azure Cosmos DB w/ SQL API
- Documents have a unique ID used by the dbms
- JSON format or other encodings (XML, YAML, BSON)
- Can query based on field values (some dbms's even support indexing based on these values)
- Some allow updates (can cause fragmentation), other only allow insert/delete
- Focus is on the query capabilities

Column Family DB

- Apache Cassandra / Azure Cosmos w/ Cassandra API
- Organizes data into rows and columns
- Can then divide the columns into families that are stored together on diskFetch data in row by key

Graph Database

- Azure Cosmos DB w/ Gremlin API
- Entities , focus on relationships between them
- Nodes and Edges – both have properties and the edges have a directionEffectively traverse the graph of complex relationship with Gremlin API as opposed to storing this data in a RDBMS and needing to construct complex recursive queries to output the same response.

Non - Relational Details

Azure Table Storage

- Key-value model
- Rows and columns
- Semi-structured data
- Must have a key, columns can vary by row
- No concept of relationships, stored procedures, secondary indexes, foreign keys
- De-normalized
- Much faster retrieval
- Partitioned data
- When searching, use the partition key to improve searching performance
- Key [partition key, row key]
- Items inside of a partition are stored in row key value order
- Point queries, range by partition key are best!
- Numeric, string, binary column types
- Schemaless – can adapt easily – flexible structure
- Pros:
  - It's simpler to scale
  - Hold semi-structured data
  - No complex relationships to maintain
  - Row insertion is fast
  - Data retrieval is fast
- Cons:
  - Transactional updates across multiple entities aren't guaranteed
  - No referential integrity
  - Difficult to sort and filter on non-key data
- ·Excellent for:
  - Storing TBs of structured data capable of serving web apps (product catalogs, consumer information)
  - Storing data sets that don't require complex joins, FK, and SPs (IOT data collection)
  - Capturing event logging and performance monitoring data
- Intended for storage of very large volumes of data
- Automatically handles the partitioning
- HA w/ in a region (seamlessly transition) – Geo-redundancy is possible for additional cost (apps will need to re-point)
- Security and RBAC

Azure Blob Storage

- Massive unstructured data
- 3 types
  - Block blob
    - 4.7 TB Max
    - Large binary objects with infrequent access
  - Page Blob
    - Optimized to support random R/W
    - 8TB max
    - Azure uses page blobs to build virtual disk storage for their v/m's
  - Append Blob
    - Block blob optimized for appending
    - Add blocks to end of append blob
    - Cannot delete or modify existing blocks
    - 4 MB per block – 196 Gb for blob
- Create a blob inside of containers/ folders similar to file structure
- 3 access tiers
  - Hot (default)
    - Accessed frequently
    - SSD storage
    - Higher storage cost, lower retrieve cost
  - Cool
    - Lower cost/ lower performance
    - Lower storage cost, Higher retrieve cost
    - Infrequent data
- Archive
    - Lowest cost/ performance
    - Historical data access very rarely
    - Can take hours for data to become available
    - Need to "re-hydrate" the blob to cool or hot tier to retrieve the data
- Lifecycle management
- Image data
- Replicated in-region 3x (HA) can setup geo-redundancy at an additional cost
- Versioning. You can maintain and restore earlier versions of a blob.
- Soft delete. This feature enables you to recover a blob that has been removed or overwritten, by accident or otherwise.
- Snapshots. A snapshot is a read-only version of a blob at a particular point in time.
- Change Feed. The change feed for a blob provides an ordered, read-only, record of the updates made to a blob.

Azure File Storage

- Create files/ shares in cloud that can be accessed anywhere
- SMB 3.0
- 100TB / storage account
- File max 1 TiB (2^40 bytes)
- 2000 concurrent connections/ shared file
- Azure portal or AzCopy utility
- Azure file sync service
- Standard (HDD) or Premium (SSD)
- Use cases:
  - Migrating existing file driven apps to cloud
  - Share server data across on-prem and cloud
- Replicated in region – can geo-replicate
- Encryption at rest (default) can enable encryption in transit

Azure Cosmos DB

- NoSQL DB
- Partitioned set of documents
- Document – collection of fields identified by key
- Fields can vary, can contain child documents
- JSON
- 2 MB/ document
- APIs:
  - SQL API
  - Table API (enables switch from table storage to Cosmos DB w/o rewriting application code)
  - MongoDB (run existing mongo backed apps without rewrite)
  - Cassandra API (column/ family DB)
  - Gremlin API – Graph DBs
- Containers contain the documents(partitioned by partition column)
- Additional indexing – can search for non-key fields in documents
- Highly scalable
- Virtually no admin overhead
- Replicated in region (HA) – 99.999% uptime – seamless failover within region
- Geo-replicated at additional cost
- 10ms latency
- Encrypted @ rest and in motion by default
- Row-level authentication
- Use Cases:
  - IOT and telematics
  - Retail and marketing
  - GamingWeb and mobile apps

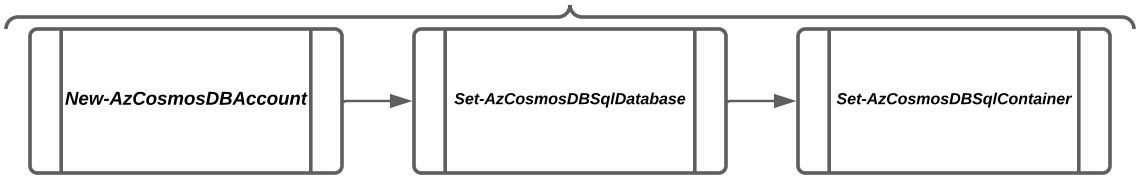
Non - Relational Provisioning - Cosmos DB

Azure Cosmos DB

- Azure Portal (interactively)
- Azure CLI
- Azure PowerShell
- Azure Resource Manager Templates
- DB Account > DB > [Container (Partitioned)] > Documents > Fields
- Throughput measured in RU/ second – amount of resources required to read a 1KB document w/ 10 fields
- Specify throughput by DB or by container (by DB – shared by all containers)
- If under-provisioned – db will throttle requests
- Can increase/ decrease resources at any time

Azure CLI

Azure PowerShell



Configure

Configure Connectivity

(Relational storage denies ALL by default – Non-Relational allows all by default)

Configure Authentication

- Requires access key to authenticate – anyone with this key can access – course grained auth
- Azure AD with RBAC
- Advanced Security available – more money

Replication

- Multi-region container and DB
- At account provisioning you can choose to replicate but won't have control as to where
- After provisioning you can select where you want it replicated to
- Can configure automated failover between regions (HA)
- Default behavior – primary region allows writes and replicated does not
- Can enable mulit-region writes – can cause conflicts through, last update to the data will stick
- Replication is Asynchronous

Consistency

(applications can decrease the consistency level, but they cannot increase it)

- **Eventual**: least consistent – changes won't be lost, they'll appear eventually, but they might not appear immediately
- **Consistent Prefix**: changes will appear in order, although there may be some delay before they become visible
- **Session**: if an app makes a number or changes, they will be visible to that application, in order. Other apps may see old data, although any changes will appear in order on their side.
- **Bounded Staleness**: there's a lag between writing and then reading the updated data – you specify the staleness in time or number of previous versions that data will be inconsistent for
- **Strong**: all write are only visible after the changes are confirmed as written successfully to the replicas. UNAVAILABLE if distributing data across multiple global regions.

Data Operations

- Data Explorer in Portal
- Cosmos DB Migration Tool
  - Github
  - Multisource to existing or new container
  - Can multithread for large uploads (increase RU's for this or use Auto-Scale)
- Azure Data factory
- Custom app (Cosmos DB BulkExecution library)
- Custom app (Cosmos DB SQL API)

Query Azure Cosmos DB

- SQL API - run sql-like queries against it
  - Returns JSON documents
  - JOIN fields in document to fields in subdocument
  - CANT JOIN TWO DIFFERENT DOCUMENTS
  - SELECT / FROM [container]/ WHERE/ ORDER BY
  - DISTINCT/ TOP
  - BETWEEN inclusive range
  - IS_DEFINED whether a field exists in a document
  - COUNT()/ SUM(), AVG(), MIN(), MAX()

NO GROUP BY CLAUSE

Non - Relational Provisioning - Storage Account

Azure Storage Account

- Portal
- Subscription
- Resource Group
- Storage acct name (unique as it is public)
- Location
- Performance Tier (see below)
- Account kind (see below)
- Replication (see below)
- Access Tier (see below)

Azure CLI

```
az storage account create
```

Azure PowerShell

```
New-AzStorageAccount
```

SKU – combo of performance tier and rep options
Kind – BlobStorage, BlockBlobStorage, FileStorage, Storage, Storagev2
Access tier – Hot/ Cool

Configure Authentication

- Requires access key to authenticate – anyone with this key can access – course grained auth
- Azure AD with RBAC
- Advanced Security available – more money

Performance Tier

- Standard (HDD)
- Premium (SSD)

Access Tier

- Hot
- Cool

Account Kind

- General Purpose V2 (blobs, files, queues, tables) – recommended for most
- General Purpose (legacy)
- BlockBlobStorage (Premium ONLY – block blobs/ append blobs)
- FileStorage (Premium Only – for file storage)
- BlobStorage (legacy)

Storage account type	Supported services	Redundancy options
General-purpose V2	Blob, File, Queue, Table, Disk, and Data Lake Gen2[2]	LRS, GRS, RA-GRS, ZRS, GZRS, RA-GZRS[3]
General-purpose V1	Blob, File, Queue, Table, and Disk	LRS, GRS, RA-GRS
BlockBlobStorage	Blob (block blobs and append blobs only)	LRS, ZRS[3]
FileStorage	File only	LRS, ZRS[3]
BlobStorage	Blob (block blobs and append blobs only)	LRS, GRS, RA-GRS

Replication

- LRS – Local redundant storage
  - Synchronously
  - Replicated w/ in physical location of primary region 3 times
- ZRS – Zone redundant storage
  - Synchronously
  - Replicated across 3 azure Availability Zones in primary region
- GRS – Geo-Redundant Storage
  - Locally redundant as above
  - Also replicated to secondary location
  - Does not expose geo-redundant copy for consumption – only there to restore from
- RA-GRS – Read Access Geo-Redundant Storage
  - Extension of GRS
  - Provides read-only access to the geo-redundant data in secondary location
  -

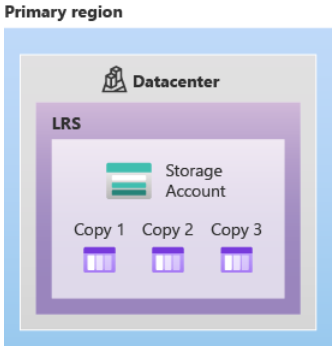
NOTE: Premium storage accounts only support LRS

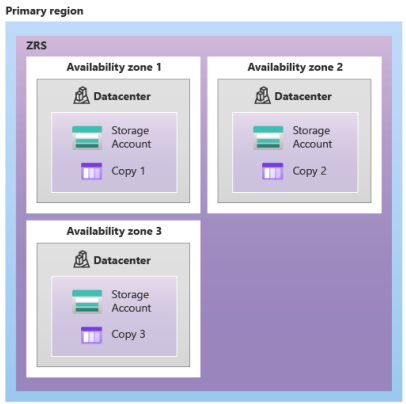
Azure Data Lake Storage

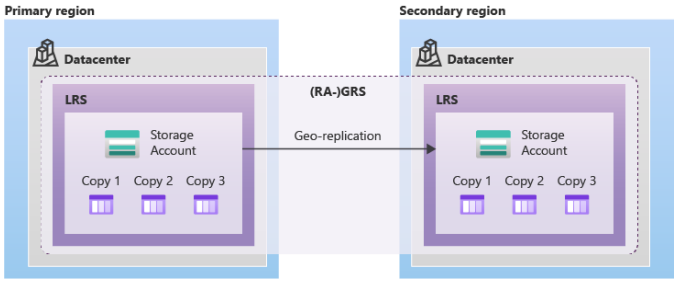
- Must enable this when building storage account, cannot enable after the fact
- Enable hierarchical namespace true

Configure Storage Accounts

- Enable/ disable secure comms w/ the service
- Switch default access tier cool/ hot
- Change the way the account is replicated
- Enable/ disable integration with Azure AD
- Encryption:
- Auto encrypted w/ Microsoft generated/ managed keys
- Can provide and manage your own keys
- SAS (Shared Access Signatures)
- Limited rights to resources for a specified time period, from a specified device IP(s)
- Only use for data intended to be public







Non - Relational Provisioning - Blob Storage

Blob Storage

- Blobs are stored in containers in a storage account
- Containers must have unique name inside of storage account
- Can set access level or use RBAC
- Portal – Containers page of storage account

Manage Azure Blob Storage

- Page Blobs – for v/m disk storage
- Append Blobs – logs or archive data
- Block blobs – image data
- Create container in storage account to hold blobs
- Can create folders in containers to further organize blobs

	Azure CLI	Azure PowerShell
Create Container for Blob	***az storage container create***	***Get-AzStorageAccount | New-AzStorageContainer***
Upload Blobs	***az storage blob upload [-batch]***	***Get-AzStorageAccount | New-AzStorageBlobContent***
List Blobs	***az storage blob list***	***Get-AzStorageAccount | Get-AzStorageBlob***
Download Blobs	***az storage blob download [-batch]***	***Get-AzStorageAccount | New-AzStorageBlobContent***
Delete Blobs	***az storage blob delete [-batch]***	***Get-AzStorageAccount | Remove-AzStorageBlob***
Delete Containers	***az storage container delete [-batch]***	***Get-AzStorageAccount | Remove-AzStorageContainer***

Non - Relational Provisioning - Blob Storage

File Storage

- Files Shares are created in a storage account
- Can set access level or use RBAC
- Portal – File Shares page of storage account

Manage Azure File Storage

- Azure Portal
- Azure Storage Explorer (download)
- Create Share > Add directories to the share
- Upload and Download
  - Manually (portal / storage explorer) or connecting via mounted drive
  - Automatically – azcopy utility CLI (fault tolerant)1 file or multiple (-- recursive flag and point to directory instead of file)

Create Share

Azure CLI

az storage share create

Azure PowerShell

Get-AzStorageAccount | New-AzStorageShare

Send to File Storage

azcopy copy "myfile.txt" https://<storage-account-name>.file.core.windows.net/<file-share-name>/myfile.txt<SAS-token>

azcopy copy "myfolder" "https://<storage-account-name>.file.core.windows.net/<file-share-name>/myfolder<SAS-token>" –recursive

Pull From File Storage

azcopy copy https://<storage-account-name>.file.core.windows.net/<file-share-name>/myfile.txt<SAS-token> "myfile.txt"

azcopy copy "https://<storage-account-name>.file.core.windows.net/myshare/myfolder<SAS-token>" "localfolder" --recursive

Analytics

	ETL	ELT
Improved data privacy and compliance	X	
Data lake support		X
Does not require specialist skills	X	
Ideal for large volumes of data		X

Data analytics activity	Purpose
Descriptive analytics	Helps answer questions about what has happened, based on historical data.
Diagnostic analytics	Helps answer questions about why things happened.
Predictive analytics	Helps answer questions about what will happen in the future.
Prescriptive analytics	Helps answer questions about what actions should be taken to achieve a goal or target.
Cognitive analytics	Helps to draw inferences from existing data and patterns

- **Bar and column charts:** Bar and column charts enable you to see how a set of variables changes across different categories. For example, the first chart below shows how sales for a pair of fictitious retailers vary between store sites
- **Line charts:** Line charts emphasize the overall shape of an entire series of values, usually over time.
- **Matrix:** A matrix visual is a tabular structure that summarizes data. Often, report designers include matrixes in reports and dashboards to allow users to select one or more element (rows, columns, cells) in the matrix to cross-highlight other visuals on a report page.
- **Key influencers:** A key influencer chart displays the major contributors to a selected result or value. Key influencers are a great choice to help you understand the factors that influence a key metric. For example, what influences customers to place a second order or why sales were so high last June.
- **Treemap:** Treemaps are charts of colored rectangles, with size representing the relative value of each item. They can be hierarchical, with rectangles nested within the main rectangles.
- **Scatter:** A scatter chart shows the relationship between two numerical values. A bubble chart is a scatter chart that replaces data points with bubbles, with the bubble size representing an additional third data dimension. A dot plot chart is similar to a bubble chart and scatter chart, but can plot categorical data along the X-Axis.
- **Filled map.** If you have geographical data, you can use a filled map to display how a value differs in proportion across a geography or region. You can see relative differences with shading that ranges from light (less-frequent/lower) to dark (more-frequent/more).

Modern Data Warehouse Analytics in Azure

Azure Data Factory

- ETL
- Streaming and batch data
- Orchestration engine
- Linked service – provides info needed for DF to connect to a source or destination
- Dataset – the data you want to ingest or store
- Connect to a dataset to output/ input viaa linked service
- Pipelines
- Don't have to be linear
- Mapping columns
- Run manually / run off a trigger

SSIS

- On prem
- ADF allows run existing SSIS packages as part of a pipeline in cloud
- Azure Feature Pack for SSIS to talk to Azure services

Polybase

- SQL Server / Azure Synapse Analytics
- Run T-SQL on external data abstracted as tables
- Transfer data from external data sources to tables
- Can join to polybase tables

Azure Data Lake Storage

- Repository for large quantities of raw data
- Fast to load/ update needs to be processed for analysis
- Staging point
- Directories/ sub-directories
- POSIX file and directory permissions RBAC
- Compatible with HADOOP HDFS

Azure Databricks

- Apache spark environment in Azure
- Big data processing, streaming, ML
- GUI to define and test steps
- Use languages R, python, scala
- SPARK code using notebooks
- Structured stream processing
- Can incorporate these notebooks in pipeline / pass params / etc.

Azure HDInsight

- Big data processing service
- Can use frameworks such as Hadoop M/R, Apache Spark, Apache Hive (SQL-like), Apache Kafka (clusteres streaming service/ real-time ingestion), Apache Storm (scalable, fault tolerant platform for streaming data), R, etc.

Azure Analysis Services

- Build tabular models to support OLAP
- Various data sources
- Can cache data in-memory or create dynamically

- Use Azure Analysis Services for:
- Smaller volumes of data (a few terabytes).
- Multiple sources that can be correlated.
- High read concurrency (thousands of users).
- Detailed analysis, and drilling into data, using functions in Power BI.
- Rapid dashboard development from tabular data.

Azure Synapse Analytics

- Analytics engine
- Ingest from external sources – transform/aggregate for analytics processing
- Store data to repeat queries on
- MPP (control noes / compute nodes)
- SQL Pools or Spark Pools
- SQL Pools –
  - Azure SQL Database
  - Azure Storage
  - Uses polybase to query external data
  - Can store data
  - Can add/ remove compute nodes
- Spark Pools –
  - Spark notebooks
  - Can save in Azure storage or data lake
  - Auto-scaling

- Use Azure Synapse Analytics for:
  - Very high volumes of data (multi-terabyte to petabyte sized datasets).
  - Very complex queries and aggregations.
  - Data mining, and data exploration.
  - Complex ETL operations. ETL stands for Extract, Transform, and Load, and refers to the way in which you can retrieve raw data from multiple sources, convert this data into a standard format, and store it.
  - Low to mid concurrency (128 users or fewer).

Power BI

- Data Viz Platform
- Power BI Desktop
- Power BI Service
- Mobile Apps
- Workflow – Build report in Desktop publish to service
- Create visualizations of the underlying data

Power BI Apps

Collection of ready-made, preset visuals and reports that are shared with an entire organization

Dashboards

Collection of visuals from a single page that you share with others

Reports

A collection of visualizations that appear together on one or more pages

Reports let you create many visualizations, on multiple pages, and let you arrange in whatever way you want

Tiles

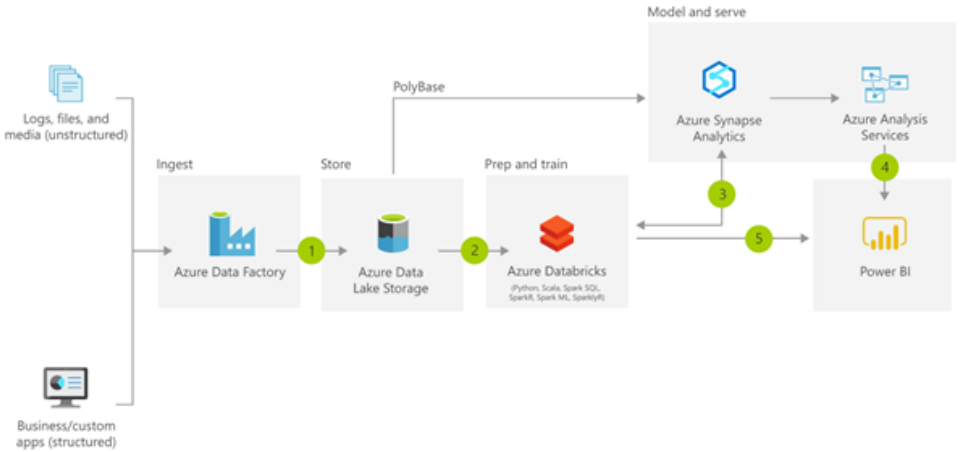
A single visualization on a report or dashboard

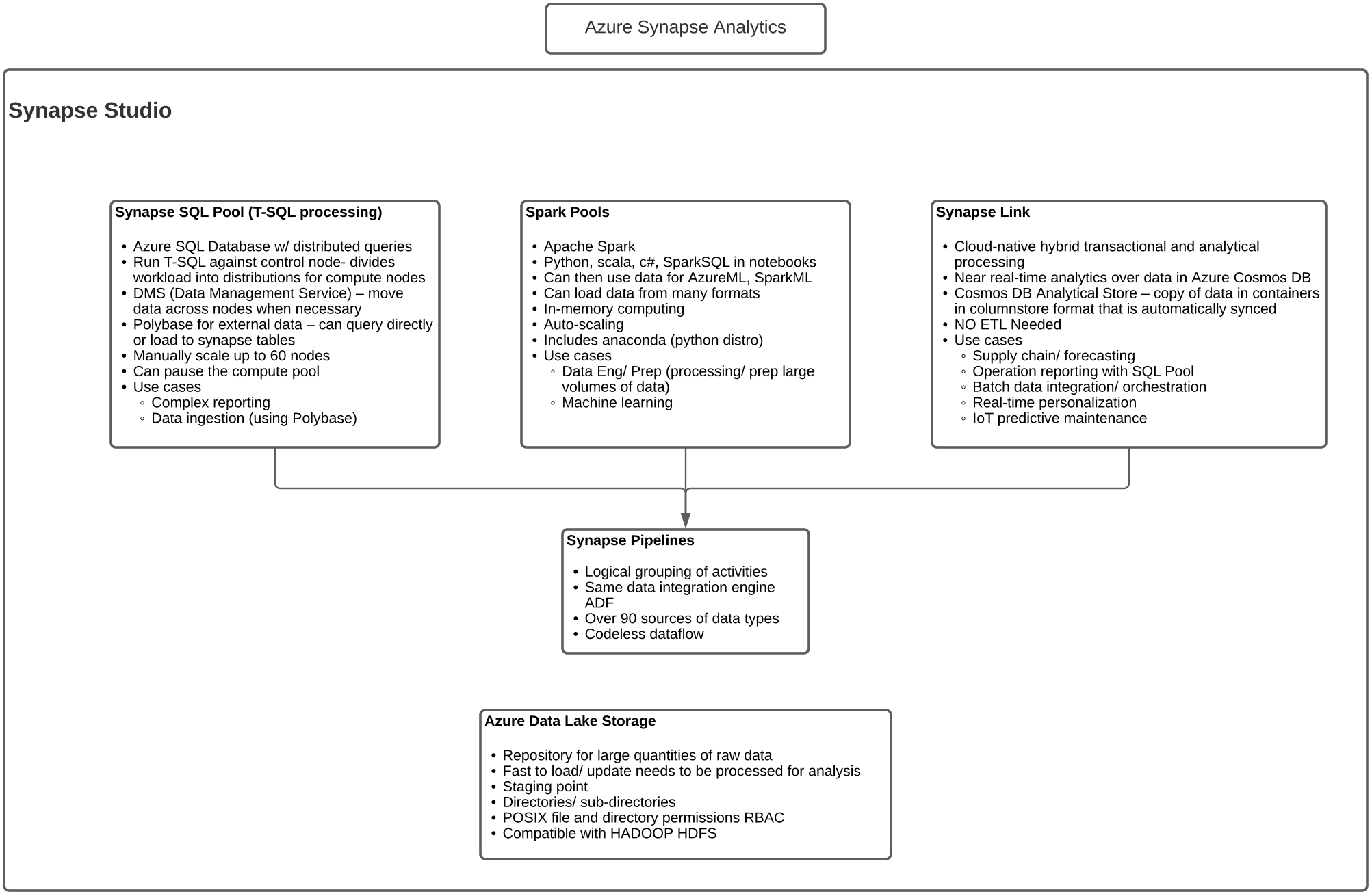
DataSets

Collection of data used to create visuals

Can be a combo of multiple data sources (filter and combine)

Multitude of data connections included




Azure Synapse Analytics
Synapse Studio
Synapse SQL Pool (T-SQL processing)
• Azure SQL Database w/ distributed queries
• Run T-SQL against control node- divides workload into distributions for compute nodes
• DMS (Data Management Service) – move data across nodes when necessary
• Polybase for external data – can query directly or load to synapse tables
• Manually scale up to 60 nodes
• Can pause the compute pool
• Use cases
◦ Complex reporting
◦ Data ingestion (using Polybase)
Spark Pools
• Apache Spark
• Python, scala, c#, SparkSQL in notebooks
• Can then use data for AzureML, SparkML
• Can load data from many formats
• In-memory computing
• Auto-scaling
• Includes anaconda (python distro)
• Use cases
◦ Data Eng/ Prep (processing/ prep large volumes of data)
◦ Machine learning
Synapse Link
• Cloud-native hybrid transactional and analytical processing
• Near real-time analytics over data in Azure Cosmos DB
• Cosmos DB Analytical Store – copy of data in containers in columnstore format that is automatically synced
• NO ETL Needed
• Use cases
◦ Supply chain/ forecasting
◦ Operation reporting with SQL Pool
◦ Batch data integration/ orchestration
◦ Real-time personalization
◦ IoT predictive maintenance
Synapse Pipelines
• Logical grouping of activities
• Same data integration engine ADF
• Over 90 sources of data types
• Codeless dataflow
Azure Data Lake Storage
• Repository for large quantities of raw data
• Fast to load/ update needs to be processed for analysis
• Staging point
• Directories/ sub-directories
• POSIX file and directory permissions RBAC
• Compatible with HADOOP HDFS