Exam DP-203: Data Engineering on Microsoft Azure Master Cheat Sheet

Various modules and percentage involved in DP-203.

Skills measured

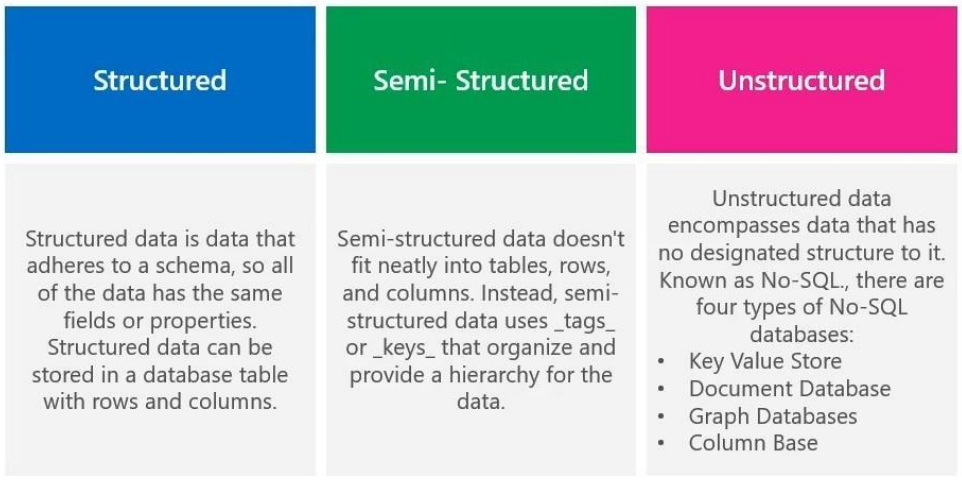
- Design and implement data storage (40-45%)
- Design and develop data processing (25-30%)
- Design and implement data security (10-15%)
- Monitor and optimize data storage and data processing (10-15%)

Data Storage:

Type of Data

Structured versus non-structured data

There are three broad types of data and Microsoft Azure provides many data platform technologies to meet the needs of the wide varieties of data



Azure Storage

4 configurations options available includes

1. Azure Blob
    - Massive storage for Text and binary
2. Azure Files
    - Mange files or share for cloud or on premise deployment
3. Azure Queues
    - Messaging store for reliable messaging between application components
4. Azure Tables
    - A NoSQL stores for schema less storage of structured data

Performance:

- Standard allows you to have any data service (Blob, File, Queue, and Table) and uses magnetic disk drives.
- Premium limits you to one specific type of blob called a page blob and uses solid-state drives (SSD) for storage.

Access tier:

- Hot
    - When the frequent operation is data retrieved.
- Cold
    - When the data is not often accessed.

Note:

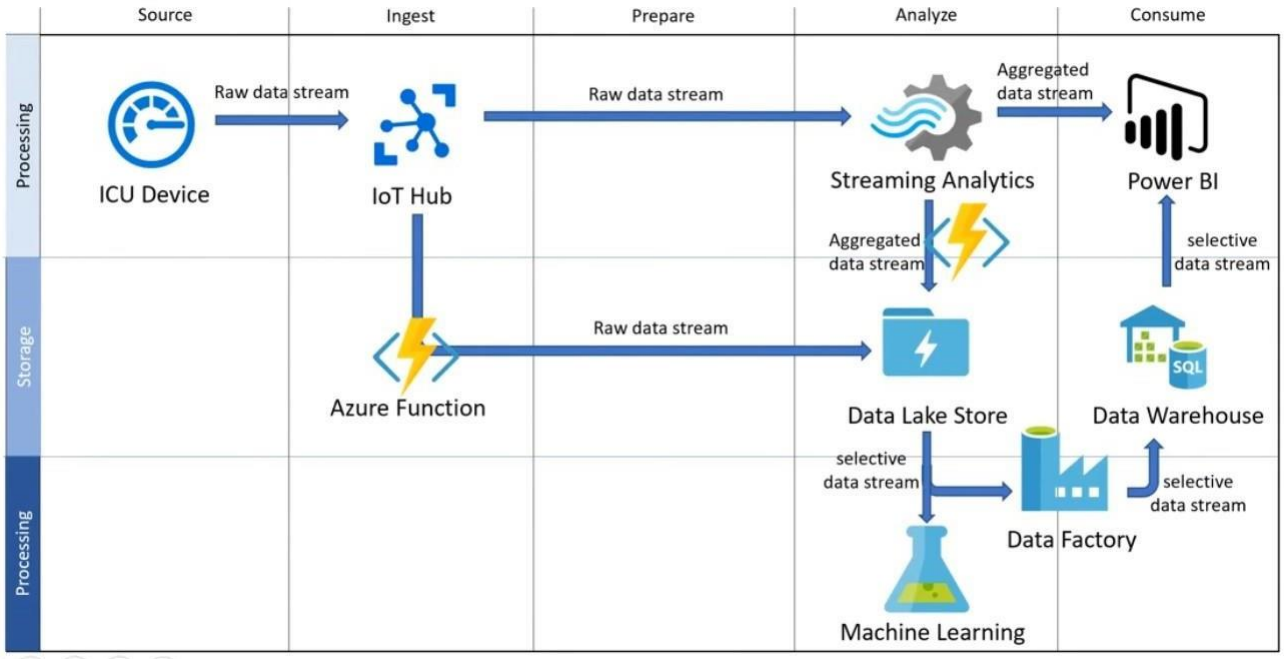
- Data Lake Storage (ADLS) Gen2 can be enabled in the Azure Storage. Hierarchical Namespace:
    - The ADLS Gen2 hierarchical namespace accelerates big data analytics workloads and enables file-level access control lists (ACLs)
- Account kind: StorageV2 (general purpose v2)
    - The current offering that supports all storage types and all of the latest features
- A storage account is a container that groups a set of Azure Storage services together.

Azure Blob Usage

- When we don't have to query on the data stored
- Less cost
- Works well with images and unstructured format

What service to use for Data?

Architecture and usage of different Azure services



Azure data bricks

- Apache Spark-based analytics platform
    - Simplifies the provisioning and collaboration of Apache Spark-based analytical solutions
- Enterprise Security
    - Utilizes the security capabilities of Azure
- Integration with other Cloud Services
    - Can integrate with variety of Azure data platform services and Power BI

Azure HD-Insight

- Deploy cluster of Hadoop or Storm or Spark

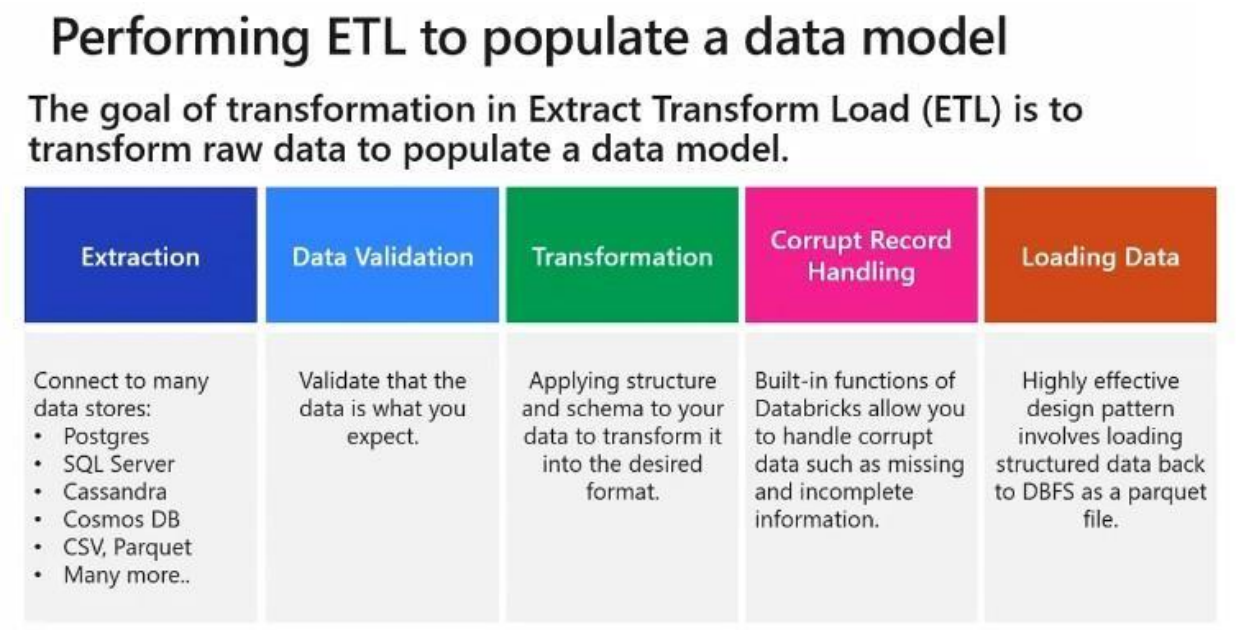
Azure Active Directory

- To guarantee security and manage person.
- Role and user permission to data bricks and data lake.

Reading Data in Azure Databricks

SQL	DataFrame
SELECT col_1 FROM myTable	df.select(col("col_1"))
DESCRIBE myTable	df.printSchema()
SELECT * FROM myTable WHERE col_1 > 0	df.filter(col("col_1") > 0)
..GROUP BY col_2	..groupBy(col("col_2"))
..ORDER BY col_2	..orderBy(col("col_2"))
..WHERE year(col_3) > 1990	..filter(year(col("col_3"))) > 1990)
SELECT * FROM myTable LIMIT 10	df.limit(10)
display(myTable)(text format)	df.show()
display(myTable)(html format)	display(df)

Performing ETL to populate a data model



Transformations usually performed on a dataset

- Basic Transformations
    - Normalizing values
    - Missing/Null data
    - De-duplication
    - Pivoting Data frames
- Advanced Transformations
    - User Defined functions
    - Joins and lookup tables
    - Multiple databases

COSMOS-DB

Can Build Globally Distributed Databases with Cosmos DB, it can handle

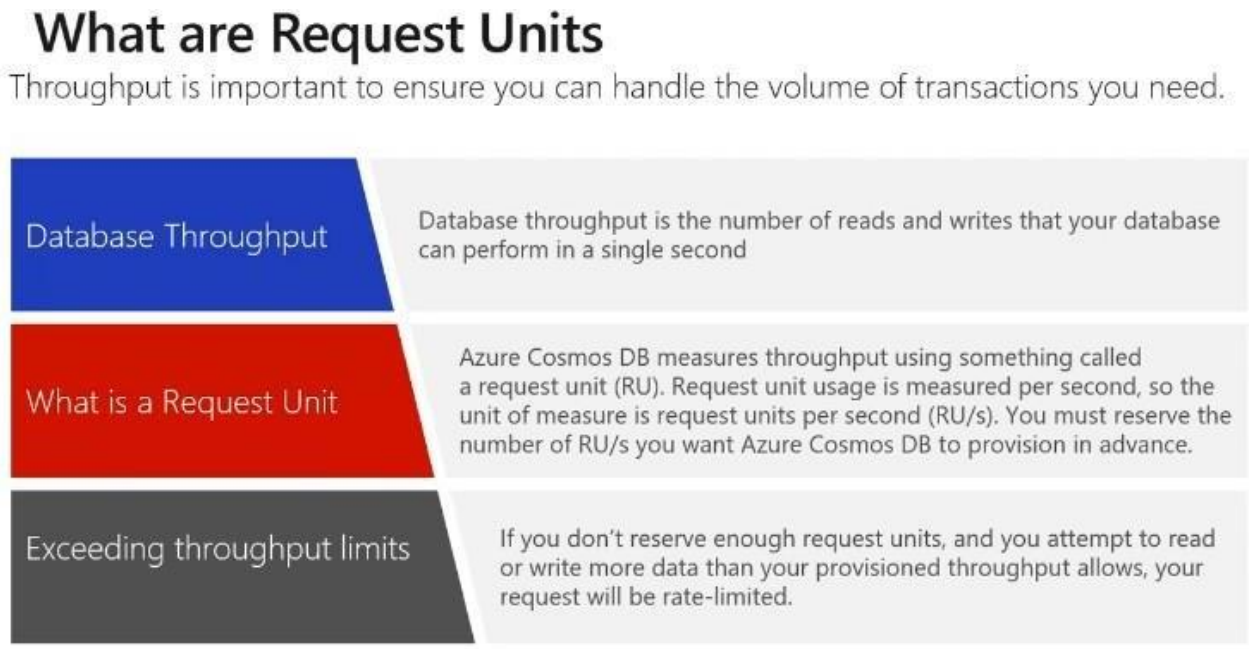
- Document databases
- Key value stores
- Column family stores
- Graph databases

Azure Cosmos DB indexes every field by default

Azure Cosmos DB (NoSQL)

- Scalability
- Performance
- Availability
- Programming Models

Request Units in Cosmos-DB



Request Unit (RU) for a DB

- A single RU is equivalent to 1 KB of Get request
- Creation, deletion and insertion require additional processing costing more RU.
- RU can be changed at any point of time
- Value of RU can be set via Capacity Planner
    - Upload the sample JSON doc
    - Define no of documents
    - Minimum RU = 400
    - Maximum RU = 215 thousand (If we require more throughput then a ticket needs to raised in the Azure portal for it)

Choosing Partition-Key

- Enable quick lookup of data
- Enable it to Auto scale when needed
- Selection of right partition key is important during development process
- Partition key is the value used to organise your data into Logical divisions.
    - e.g.: In a Retail scenario
        - ProductID and UserID value as a partition key is a good choice.

Note: A physical node can have 10 GB of information that means each Unique partition Key can have 10 GB of unique values.

Creating a Cosmos-DB

1. Click on resources and create it
2. Click on Data Explorer to create a Database name and the table
3. Use New Item tab to add the values to the table
4. UDF can also be created as Stored procedures in JavaScript.

We can also create the same using Azure CLI

```
az account list –output table     // Lists the set of Azure subscriptions that we
have

Az account set –subscription “<subscription name>”

az  group list –out table         // List of resource groups

export NAME=“<Azure Cosmos DB account name>”

export RESOURCE_GROUP=“<rgn>[sandbox resource group name]</rgn>”

Export LOCATION=“<location>”      // Data centre location

Export DB_NAME=“Products”

Az group create –name <name> –location <location>

Az cosmosdb create –name $NAME –kind GlobalDocumentDB –resource-group $RESOURCE_GROUP

Az cosmosdb database create –name $NAME –db-name $DB_NAME –resource-group
$RESOURCE_GROUP
```

```
Az cosmosdb collection create –collection-name “Clothing“ –partition-key-path
“/productId” –throughput 1000 - name $NAME –db-name $DB_NAME –resource-group
$RESOURCE_GROUP
```

After creating a COSMOSDB

- Navigate to Data Explorer
- Click on New container and Database
- A container can have multiple Databases

Cosmos DB fail over management

Cosmos DB Consistency Levels

Consistency Level	Guarantees
Strong	Linearizability. Reads are guaranteed to return the most recent version of an item

Consistency Level	Guarantees
Bounded Staleness	Consistent Prefix. Reads lag behind writes by at most k prefixes or t interval.
Session	Consistent Prefix. Monotonic reads, monotonic writes, read-your-writes, write-follows-reads.
Consistent Prefix	Updates returned are some prefix of all the updates, with no gaps.
Eventual	Out of order reads.

- Eventual consistency provide the weakest read consistency but offer lowest latency of both reads and writes. ‼️ ▶️

Question related to setting up latency ‼️ ▶️

What is the Latency I will have to use in order to provide the lower latency of reads and writes ‼️ ▶️ - Eventual Consistency

COSMOS-DB takes care of consistency of data when replicated ‼️ ▶️

AZURE SQL DATABASE CONFIGURATION

- DTUs (Database Transaction Unit)
  - Combined measure of Compute, storage, and IO resources
- VCores
  - Enables you to configure resources independently
  - Greater control over compute and storage resources
- SQL Elastic Pools ‼️ ▶️
  - Relate to eDTUs.
  - Enable you to buy set of compute and storage resources that are shared among all the databases in the pool.
  - Each database can use the resources they need.
- SQL Managed Instances

- Creates a database with near 100% compatibility with the latest SQL server.
- Useful for SQL Server customers who would like to migrate on-premises servers instance in a “lift and shift” manner.

shell.azure.com to start Azure shell

```
To connect to Database
jay@Azure:~$ az configure --defaults group=ms-dp-200 sql-server=jaysql01

jay@Azure:~$ az sql db list
O/P:


jay@Azure:~$ az sql db list | jq '[.[] | {name: .name}]'
O/P:
[
  {
    "name": "master"
  },
  {
    "name": "sqldbjay01"
  }
]

jay@Azure:~$ az sql db show --name sqldbjay01

az sql db show-connection-string --client sqlcmd --name sqldbjay01
O/P:
        "sqlcmd -S tcp:<servername>.database.windows.net,1433 -d sqldbjay01 -U
<username> -P <password> -N -l 30"

"sqlcmd -S tcp:sqldbjay01.database.windows.net,1433 -d sqldbjay01 -U jay -P “******”
-N -l 30"

SELECT name FEOM sys.tables; GO
```

SQL-DB does not take care of consistency of data when replicated, it needs to be done manually. ‼️ ▶

AZURE SQL-DW

3 types

- Enterprise DW
  - Centralized data store that provides analytics and decision support
- Data Marts
  - Designed for the needs of a single Team or business unit such as sales

- Operational Data Stores
    - Used as interim store to integrate real-time data from multiple sources for additional operations on the data.

2 Architectural way of building a DW

- Bottom-Up Architecture
    - Approach based on the notion of connected Data Marts
    - Depends on Star Schema
    - Benefit
        - Start departmental Data Mart
- Top-down Architecture
    - Creating one single integrated Normalized Warehouse
    - Internal relational constructs follow the rules of normalization

Azure SQL-DW Advantage

- Elastic scale & performance
    - Scales to petabytes of data
    - Massively Parallel Processing
    - Instant-on compute scales in seconds
    - Query Relational / Non-Relational
- Powered by the Cloud
    - Starts in minutes
    - Integrated with AzureML, PowerBI & ADF
    - Enterprise Ready

Azure-DW GEN-2

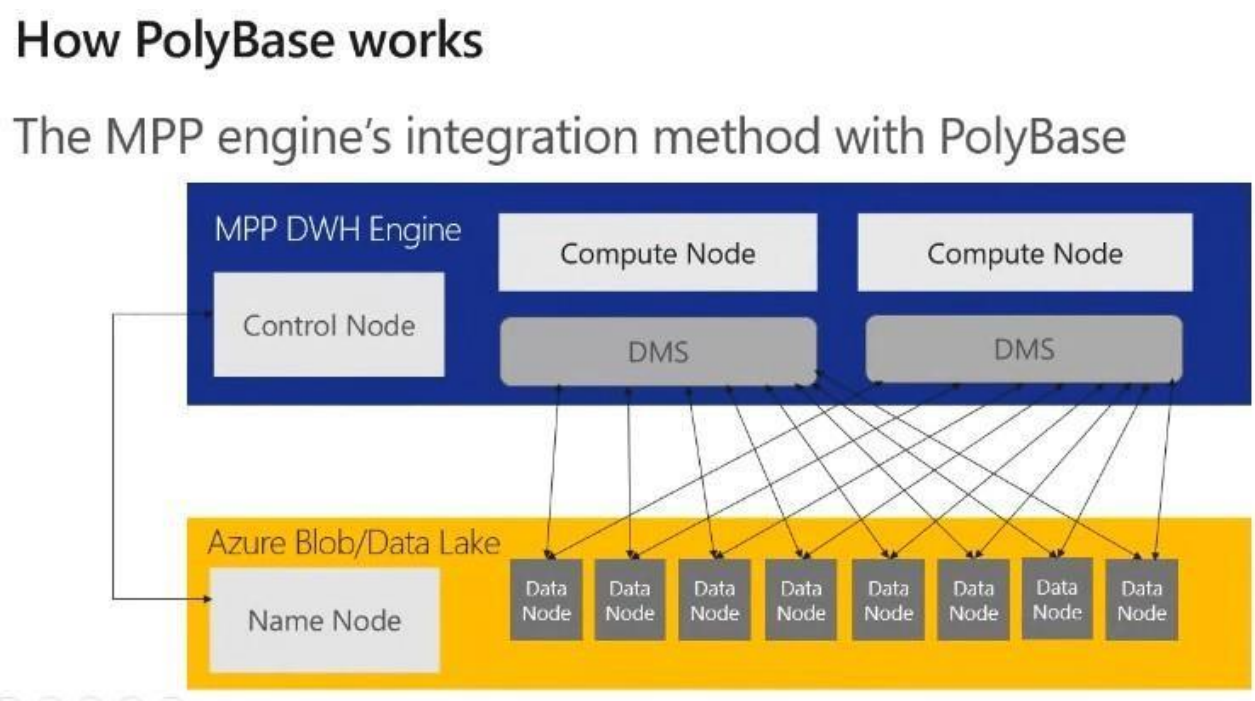
- Introduced Cache and tempDB to pull data from remote datasets
- Max DWU is 30Kc
- 120 connections and 128 queries
- MPP

Creation of Azure DW

- Create New resource
- DB
- SQL Data Warehouse

Using PolyBase to Load Data in Azure SQL Data Warehouse ‼⍰ ⯈

How PolyBase works ‼□ ▯



The MPP engine's integration method with PolyBase

- Azure SQLDW is a relational datawarehose store which use MPP architecture which takes advantage of the on demand Elastic scale of Azure compute and storage to load and process Petabytes of data
- Transfers data between SQLDW and external resource providing the fast performance
- Faster way to access Data Nodes

PolyBase ETL for DW are

- Extract the source data into Text file
- Load the data into Azure Blob Storage / Hadoop DataLake store

- Import the data into SQLDW staging table using PolyBase
- Transform the data (optional state)
- Insert the data into Partition tables

Create a Storage Account

- Go to Resource

Blobs

- REST-based object storage for Unstructured data.

Import the Blob file into SQL-DW

```
CREATE MASTER KEY;
CREATE DATABASE SCOPED CREDENTIAL AzureStorageCredential
WITH
        IDENTITY = ‘jayDW’,
        SECRET = ‘THE-VALUE-OF-THE-ACCESS-KEY’            -- put key1’s value here
;

CREATE EXTERNAL DATA SOURCE AzureStorage
WITH (
TYPE = HADOOP ,
LOCATION = ‘wasbs://data-files@demodwstorage.blob.core.windows.net‘,
CREDENTIAL = AzureStorageCredential
);

CREATE EXTERNAL FILE FORMAT TextFile
WITH (
 FORMAT_TYPE = DelimiteddText,
FORMAT_OPTIONS (FIELD_TERMINATOR = ‘,’)
);

– Load the data from Azure Blob storage to SQL Data Warehouse

CREATE TABLE [dbo].[StageDate]
WITH (
CLUSTERED COLUMNSTORE INDEX,
DISTRIBUTION = ROUND_ROBIN
)
AS
SELECT * FOM [dbo].[Temp];

– Create statistics on the new data

CREATE STATISTICS [DataKey] on [StageDate] ([DateKey]);
CREATE STATISTICS [Quarter] on [StageDate] ([DateKey]);
CREATE STATISTICS [Month] on [StageDate] ([Month]);
```

Import the Blob file into SQL-DW (Alternative)

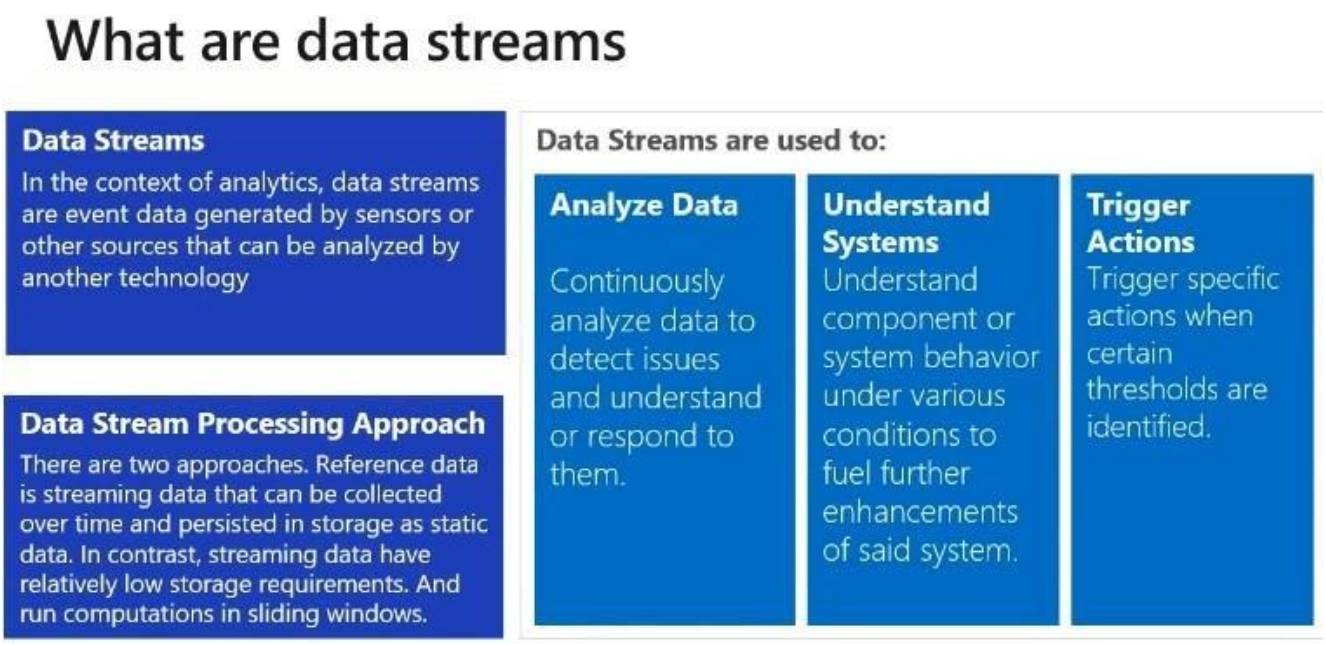
Import data from Blob Store to SQL DW

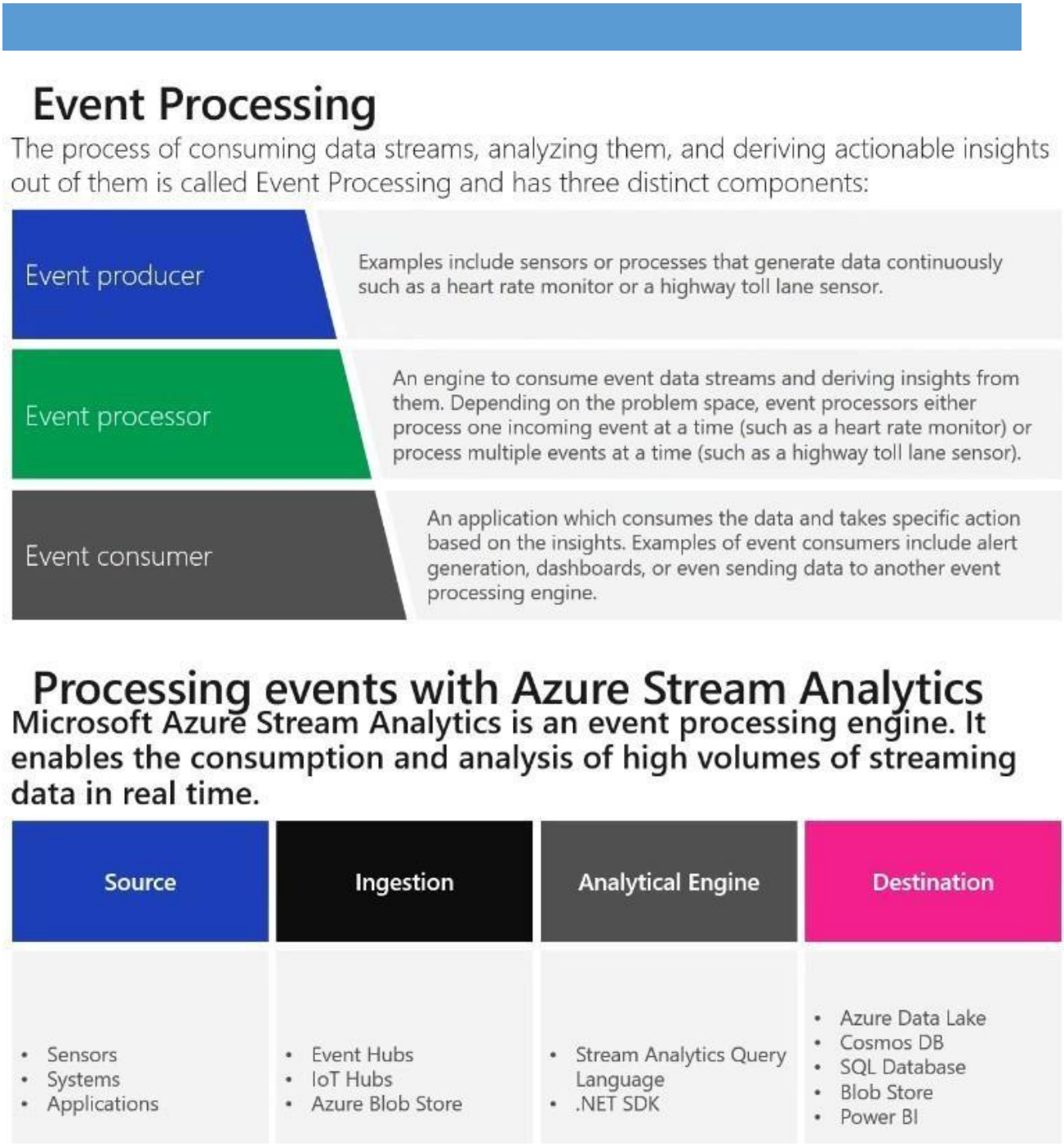
```
--STEP 1: Create an external data source for Hadoop
-- DROP EXTERNAL DATA SOURCE FXR_TEST_DSRC;
CREATE EXTERNAL DATA SOURCE FXR_TEST_DSRC
  WITH (  TYPE = HADOOP
        , LOCATION = 'hdfs://192.168.210.145:8020'
        , JOB_TRACKER_LOCATION = '192.168.210.145:8032‘
          ---- defaults:8021 - Cloudera 4.3; 8032 - HDP 2.x on Windows | Cloudera 5.1;
          ----          8050 - HDP 2.x on Linux; 50300 - HDP 1.3
       );
--STEP 2: Create an external file format for a Hadoop text-delimited file.
--DROP EXTERNAL FILE FORMAT FXR_Test_Format;
CREATE EXTERNAL FILE FORMAT FXR_Test_Format
  WITH (  FORMAT_TYPE = DELIMITEDTEXT
        , FORMAT_OPTIONS ( FIELD_TERMINATOR = N';‘
        , USE_TYPE_DEFAULT = TRUE
        , STRING_DELIMITER = '')
       );

--STEP 3: Create a new external table in SQL Server MPP SQL
-- DROP EXTERNAL TABLE Test;
CREATE EXTERNAL TABLE Test
    (name nvarchar(17), startzeitpunkt nvarchar(35),
     endzeitpunkt varchar(35), flms_system_realtime nvarchar(19),
     dummy nvarchar(19) NULL, Counter1DTonDur nvarchar(19),
     Counter1DMileage nvarchar(19), dummy2 nvarchar(2) NULL
    )
 WITH
    (LOCATION = '/user/fxr47511/pdwtest'
     , DATA_SOURCE = FXR_TEST_DSRC
     , FILE_FORMAT = FXR_Test_Format
     , REJECT_TYPE = value
     , REJECT_VALUE = 1000
    );
```

```
--STEP 4: Create a new external table in SQL Server MPP SQL
CREATE EXTERNAL TABLE dbo.Test_2
WITH (   LOCATION = '/user/fxr47511/pdwtest'
       , DATA_SOURCE = FXR_TEST_DSRC
       , FILE_FORMAT = FXR_Test_Format
       , REJECT_TYPE = value
       , REJECT_VALUE = 1000
     )
AS
SELECT T1.* FROM dbo.FactInternetSales T1
          JOIN dbo.DimCustomer T2 ON ( T1.CustomerKey = T2.CustomerKey )
```

Check Ingest Polybase in Data warehouse !!□ □

Data Streams





ORCHESTRATING DATA MOVEMENT WITH ADF AND SECURING AZURE DATA PLATFORMS

Azure Event Hubs:

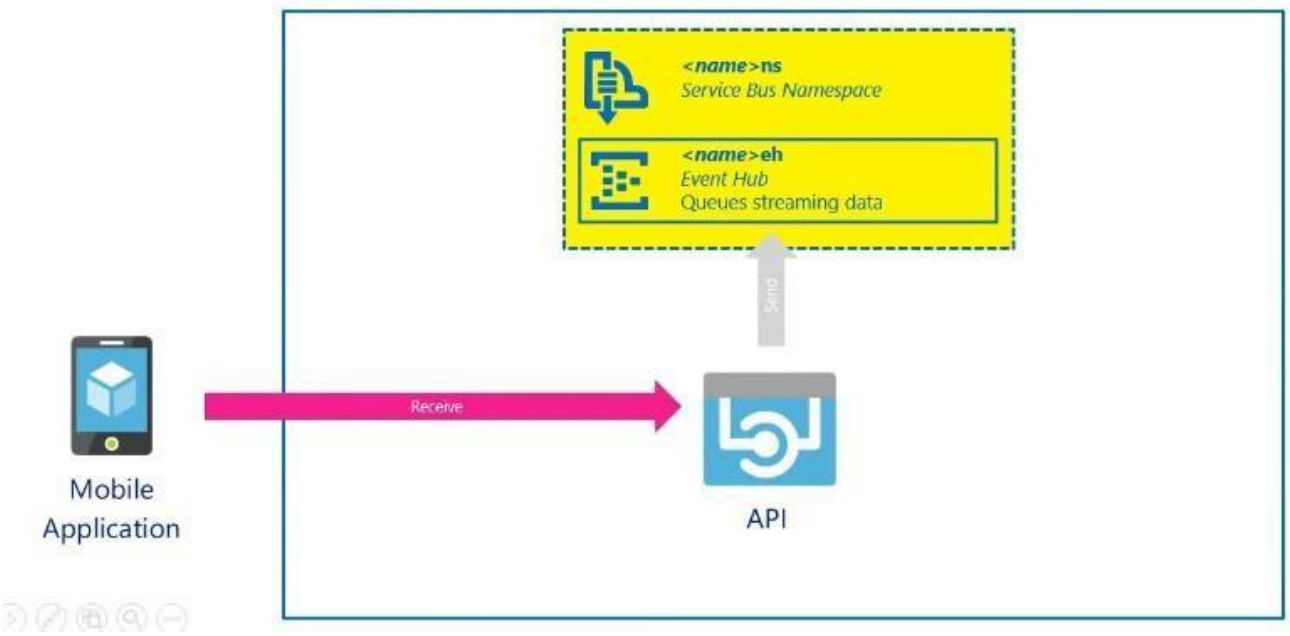
- Is a highly scalable publish-subscribe service that can invest millions of events per second and stream them into multiple applications

- A Event hub is a cloud-based event service capable of receiving and assessing millions of events per second.
- An Event is a small packet of information, a datagram that contain a notification.
- Events can be published individually or in batch.
- Single Publication or batch count can exceed 256KB.

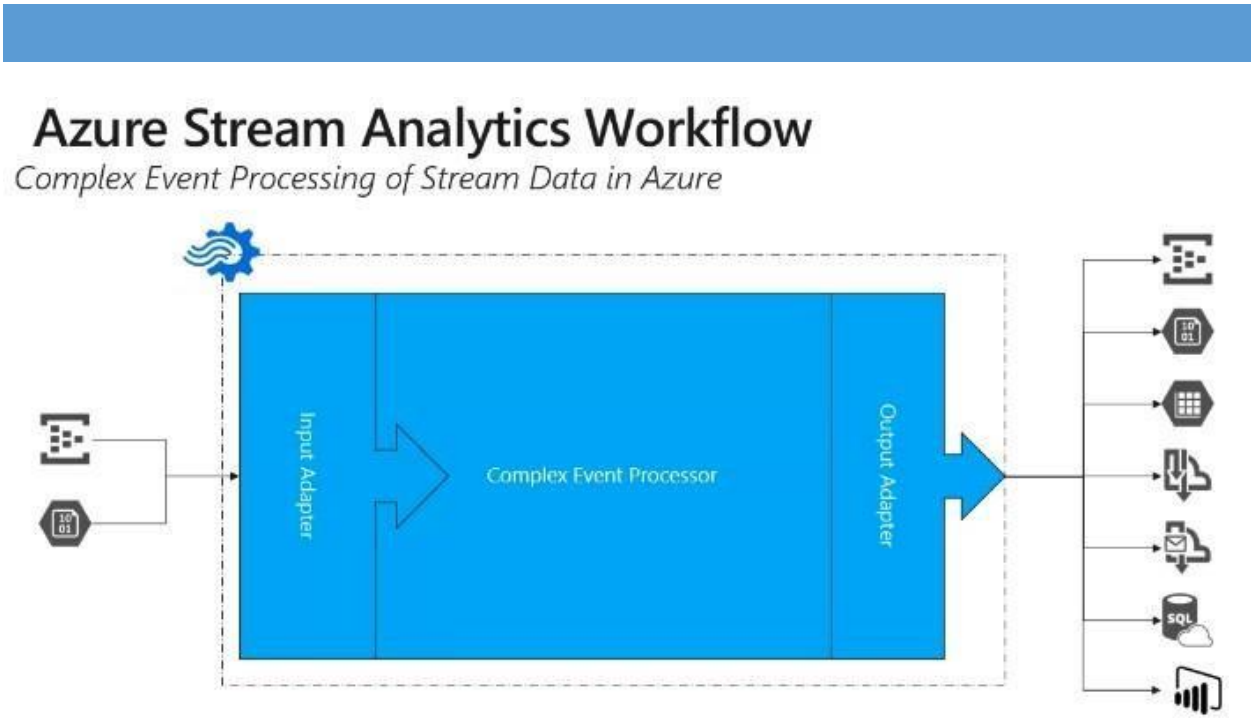
Create Event Hub

- Navigate to Entities
- Event Hub
- Shared Access policies
    - Policy will generate Primary key and Secondary key and the connection string

Configure Application to use Event Hubs



Azure Stream Analytics Workflow



Azure Data Factory - ADF

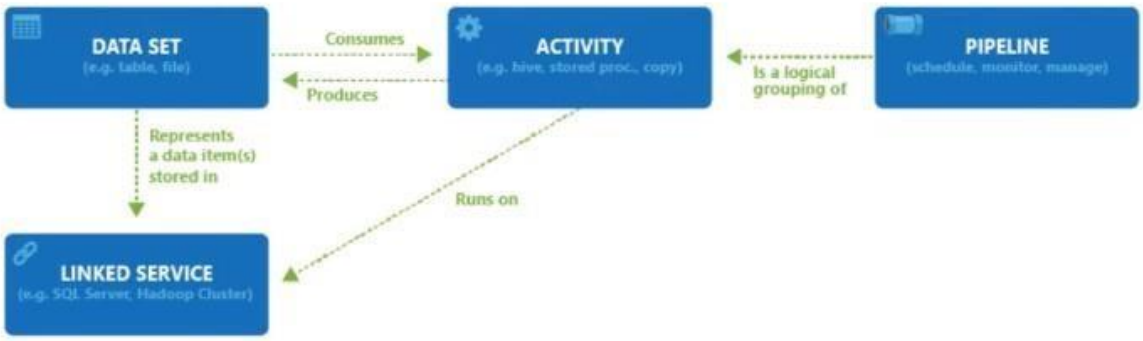
- Creates, orchestrates, and automates the movement, transformation and/or analysis of data through in the cloud.

The Data Factory Process

- Connect & collect
- Transform & Enrich
- Publish
- Monitor

Azure Data Factory Components

Azure Data Factory Components



Azure Data Factory Contributor Role

- Create, edit, and delete factories and child resources including datasets, linked services, pipelines, triggers, and integration runtimes.
- Deploy Resource Manager Templates. Resource Manager Deployment is the deployment method used by Data Factory in the Azure portal.
- Manage App Insights alerts for a data factory
- At the resource group level or above, lets users deploy Resource Manager Template.
- Create support tickets.

Linked Services

Linked services are much like connection strings, which define the connection information needed for Data Factory to connect to external resources.

Linked Services

Data Sources

Category	Data store	Supported as a source	Supported as a sink
Azure	Azure Blob storage	✓	✓
	Azure Data Lake Store	✓	✓
	Azure DocumentDB	✓	✓
	Azure SQL Database	✓	✓
	Azure SQL Data Warehouse	✓	✓
	Azure Search Index		✓
	Azure Table storage	✓	✓
Databases	Amazon Redshift	✓	
	DB2	✓	
	MySQL	✓	
	Oracle	✓	✓
	PostgreSQL	✓	
	SAP Business Warehouse	✓	
	SAP HANA	✓	
	SQL Server	✓	✓
	Sybase	✓	
	Teradata	✓	

Compute resource

Data transformation activity	Compute environment
Hive	HDInsight [Hadoop]
Pig	HDInsight [Hadoop]
MapReduce	HDInsight [Hadoop]
Hadoop Streaming	HDInsight [Hadoop]
Machine Learning activities: Batch Execution and Update Resource	Azure VM
Stored Procedure	Azure SQL, Azure SQL DW, or SQL Server
Data Lake Analytics U-SQL	Azure Data Lake Analytics
DotNet	HDInsight [Hadoop] or Azure Batch

Linked Service Example

Linked Services

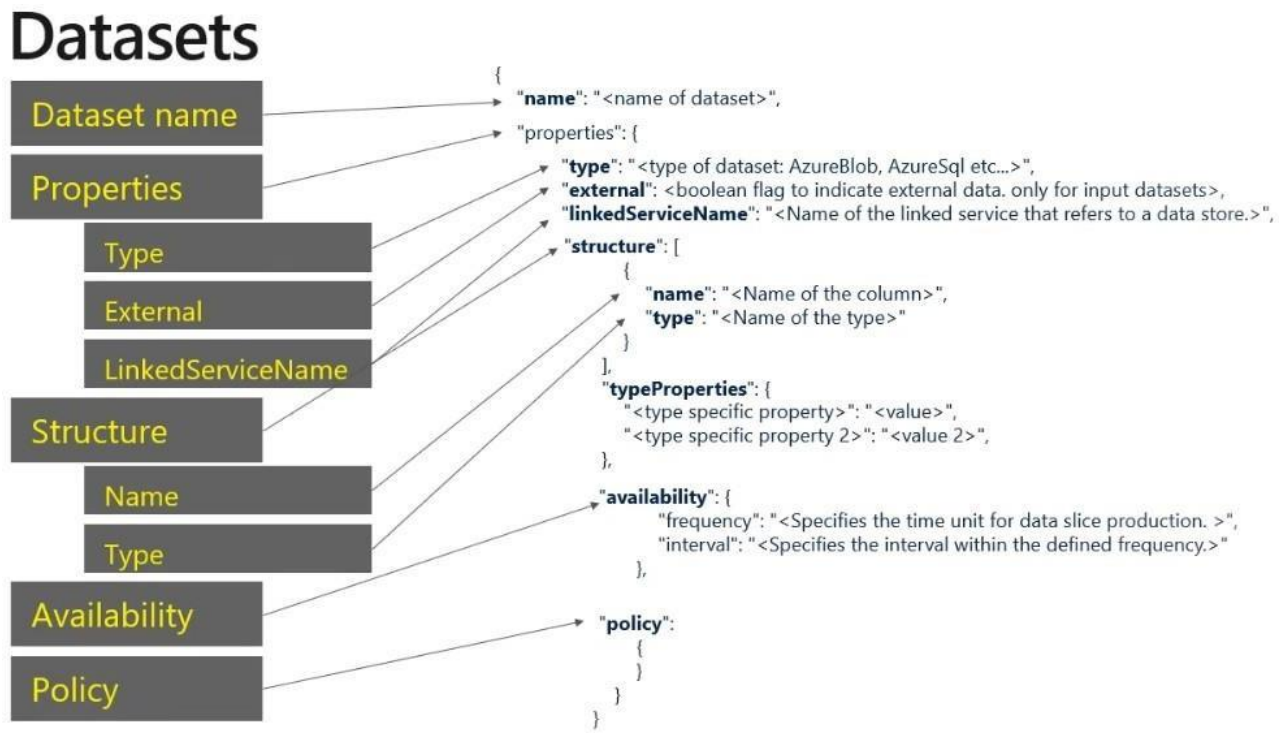
AZURE SQL DATABASE EXAMPLE

```
{
  "name": "AzureSqlLinkedService",
  "properties": {
    "type": "AzureSqlDatabase",
    "typeProperties": {
      "connectionString": "Server=tcp:ctosqldb.database.windows.net,1433;Database=EquityDB;User ID=ctesta-
oneill;Password=P@ssw0rd;Trusted_Connection=False;Encrypt=True;Connection Timeout=30"
    }
  }
}
```

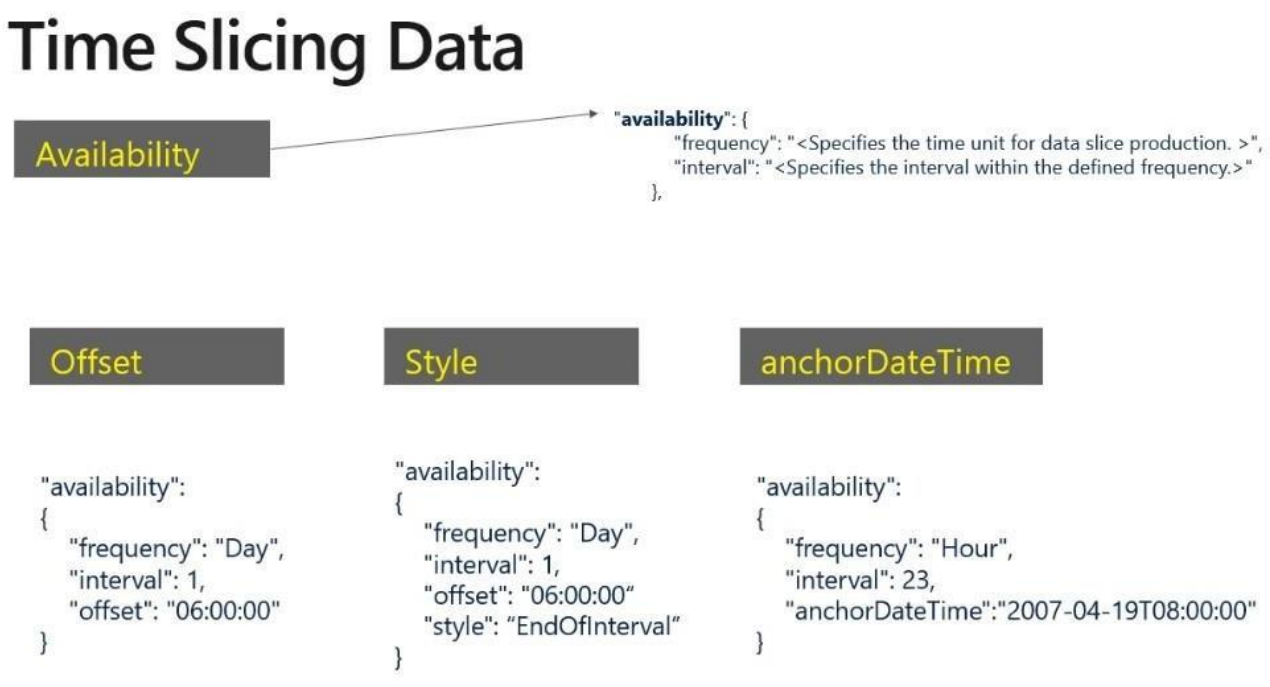
AZURE BLOB STORE EXAMPLE

```
{
  "name": "StorageLinkedService",
  "properties": {
    "type": "AzureStorage",
    "typeProperties": {
      "connectionString":
"DefaultEndpointsProtocol=https;AccountName=ctostorageaccount;AccountKey=087ubp097guh8*JON*&B*(97g9879"
    }
  }
}
```

Data Sets



Time Slicing Data



Data Factory Activities

Activities within ADF defines the actions that will be performed on the data and there are three categories including:

- Data movement activities
  - Simply move data from one data store to another.
  - A common example of this is in using Copy Activity.
- Data transformation activities
  - Use compute resource to change or enhance data through transformation, or it can call a compute resource to perform an analysis of the data
- Control Activities
  - Orchestrate pipeline activities that includes chaining activities in a sequence, branching, defining parameters at the pipeline level, and passing arguments while invoking the pipeline on-demand or from a trigger

Pipelines

- Pipeline is a grouping of logically related activities.
- Pipeline can be scheduled so the activities within it get executed.
- Pipeline can be managed and monitored.

Working with documents programmatically

- Create Storage Account
- Create ADF
- Create data workflow pipeline
- Add data bricks workbook to pipeline
- Perform analysis on the data

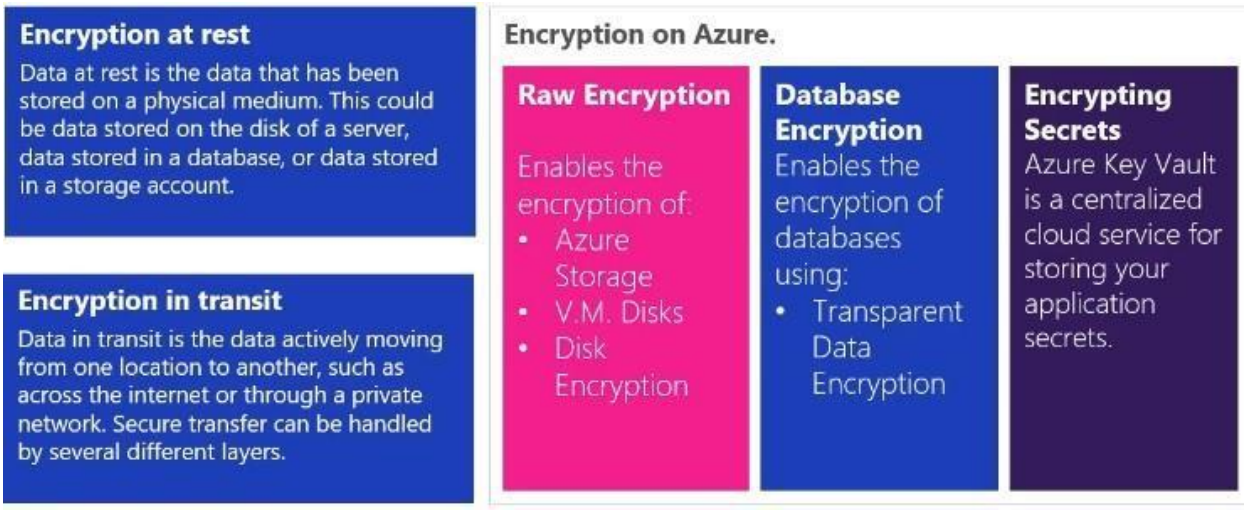
Network Security

Securing your network from attacks and unauthorized access is an important part of any architecture.

Internet Protection	Firewalls	DDoS Protection	Network Security Groups
Assess the resources that are internet-facing, and to only allow inbound and outbound communication where necessary. Make sure you identify all resources that are allowing inbound network traffic of any type.	To provide inbound protection at the perimeter, there are several choices: • Azure Firewall • Azure Application Gateway • Azure Storage Firewall	The Azure DDoS Protection service protects your Azure applications by scrubbing traffic at the Azure network edge before it can impact your service's availability.	Network Security Groups allow you to filter network traffic to and from Azure resources in an Azure virtual network. An NSG can contain multiple inbound and outbound security rules.

Identity and Access (Azure Active Directory (AD))

Encryption



Azure Key-Vaults (2 Ques) ‼□ □

- It is a centralised cloud service for storing your application secrets
- Provides secure access capability
- Key management can be done

Different Keys available

- RSA
- EC

Managing Encryption

Databases stores information that is sensitive, such as physical address, email address, and phone numbers. The following is used to protect this data:

Azure Data Lake Storage Gen2 Security Features

- Role Based Access Control
- Posix Complaint ACL
- AAD Oauth 2.0 Token
- Azure Services Integration

MONITORING, TROUBLESHOOTING DATA STORAGE AND OPTIMIZING DATA PLATFORMS

Azure Monitor

Azure Monitor provides a holistic monitoring approach by collecting, analysing, and acting on telemetry from both cloud and on-premises environments

Metric Data

- Provides quantifiable information about a system over time that enables you to observe the behaviour of a system.

Log Data

- Logs can be queried and even analysed using Log Analytics. In addition, this information is typically presented in the overview page of an Azure Resource in the Azure portal.

Alerts

- Alerts notify you of critical conditions and potentially take corrective automated actions based on triggers from metrics or logs.

Monitoring the network

Log Analytics within Azure monitor has the capability to monitor and measure network activity.

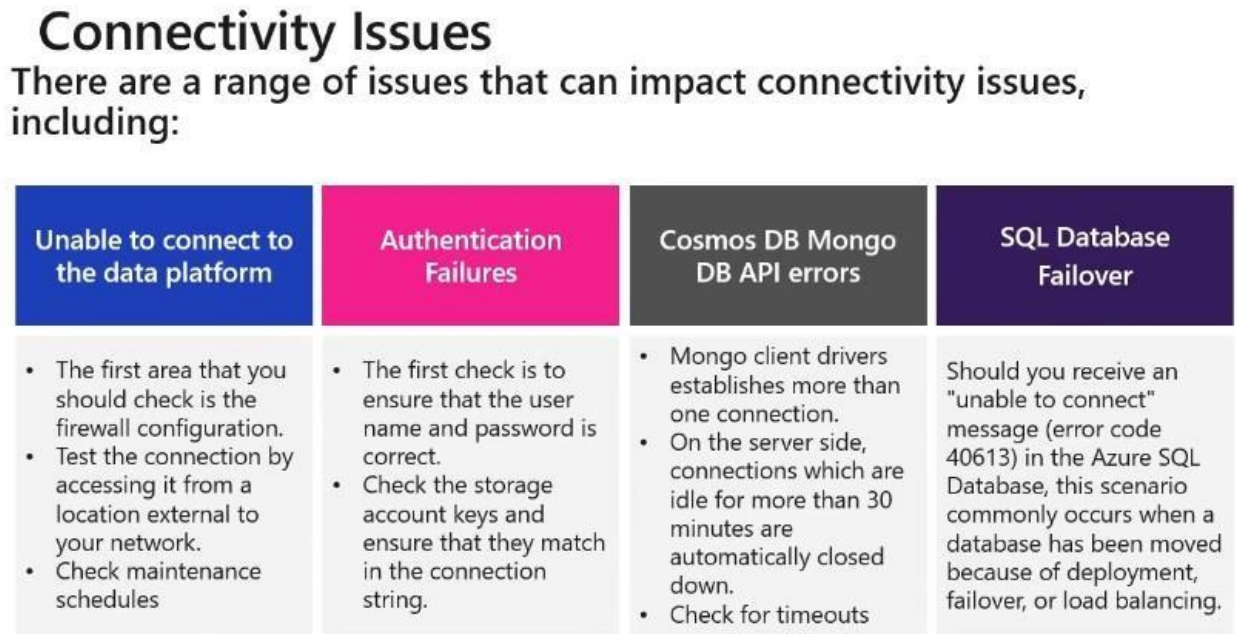
Network Performance Monitor

- Measures the performance and reachability of the networks that you have configured.

Application Gateway Analytics

- Contains rich, out-of-the box views you can get insights into key scenarios, including:
    - Monitor client and server errors.
    - Check requests per hour

Connectivity Issues



Performance Issues (To speed up query performance)

- Data Lake Storage
    - Ensure hierarchical Namespace is enabled
- SQL Database
    - Install the latest Document-DB SDK
    - Use direct mode as your connection mode when configuring your connection policy.
    - Increase the no of thread or tasks to decrease the wait time while fulfilling the requests.
    - Identify and add missing indexes.
- Cosmos DB
    - Avoid full scans on the collection, so query part of the collections
    - All UDF's and built-in function will scan across all the documents within the query
    - Use direct mode as your connection mode when configuring your connection policy.
    - Tune the page size for querying and read feeds for better performance using the x-ms-max-itime.count.header
    - For any partisient collections query in parallel to increase performance and leverage more throughput
    - Use direct https connectivity mode for best performance
- Colocation of Resources
    - Try increasing the RU between your collection
- SQL Data Warehouse
    - Ensure the statistics are up-to-date
    - Query optimizer

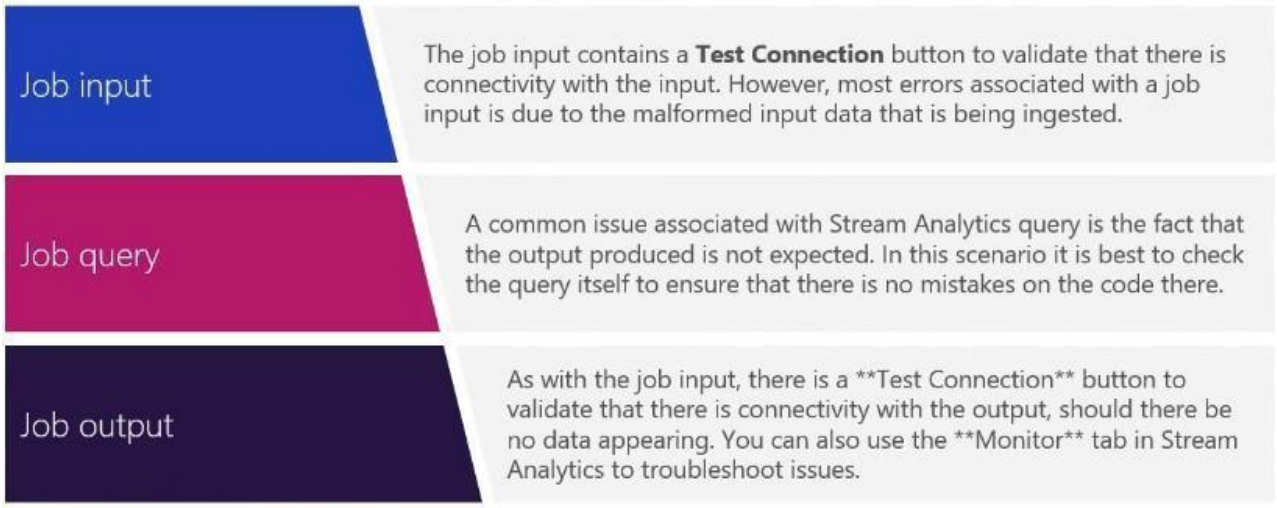
Storage Issues !!□ □

- Consistency

- Corruption

Troubleshoot Streaming data

When using Stream Analytics, a Job encapsulates the stream Analytic work and is made up of three components:



Troubleshoot batch data loads

When trying to resolve data load issues, it is first pragmatic to make the holistic checks on Azure, as well as the network checks and diagnose and solve the issue check. After that, then check:

Azure Blob and Data Lake Store	SQL Data Warehouse	Cosmos DB	SQL Database
Notwithstanding network errors; occasionally, you can get timeout or throttling errors that can be a symptom of the availability of the storage accounts.	• Make sure you are always leveraging Polybase. • Ensure CTAS statements are used to load data • Break data down into multiple text files. • Consider DWU usage	• Check that you have provisioned enough RU's • Review partitions and partitioning keys • Check for client connection string settings	• Check that you have provisioned enough DTU's • Review whether the database would benefit from elastic pools • A wide range of tools can be used to troubleshoot SQL Database

Data redundancy

Data redundancy is the process of storing data in multiple locations to ensure that it is highly available.

Azure Blob and Data Lake Store	SQL Data Warehouse	Cosmos DB	SQL Database
• Locally redundant storage (LRS) • Zone-redundant storage (ZRS) • Geo-redundant storage (GRS) • Read-access geo-redundant storage (RA-GRS)	SQL Data Warehouse performs a **geo-backup** once per day to a paired data center. The RPO for a geo-restore is 24 hours.	Azure Cosmos DB is a globally distributed database service. You can configure your databases to be globally distributed and available in any of the Azure regions.	• Check that you have provisioned enough DTU's • Review whether the database would benefit from elastic pools • A wide range of tools can be used to troubleshoot SQL Database

Disaster Recovery

There should be process that are involved in backing up or providing failover for databases in an Azure data platform technology. Depending on circumstances, there are numerous approaches that can be adopted.

Azure Blob and Data Lake Store	SQL Data Warehouse	Cosmos DB	SQL Database
Supports account failover for geo-redundant storage accounts. You can initiate the failover process for your storage account if the primary endpoint becomes unavailable.	SQL Data Warehouse performs a **geo-backup** once per day to a paired data center. Data warehouse snapshot feature that enables you to create a restore point to create a copy of the warehouse to a previous state.	Takes a backup of your database every **4 hours** and at any point of time Only the latest 2 backups are stored.	Creates database backups that are kept between 7 and 35 days Uses Azure read-access geo-redundant storage (RA-GRS) to ensure that they preserved even if data center is unavailable.

Scenarios

1. **Recommended service: Azure Cosmos-DB**

- Semi-structured: because of the need to extend or modify the schema for new product
- Azure Cosmos DB indexes every field by default

- ACID-compliant and faster while querying compared to other services

Advantages:

- Latency & throughput: High throughput and low latency
- Transactional support: Required
- Customers require a high number of read operations, with the ability to query on many fields within the database.
- The business requires a high number of write operations to track the constantly changing inventory.

2. **Recommended service: Azure Blob storage**

- Unstructured: Product catalog data
- Only need to be retrieved by ID.
- Customers require a high number of read operations with low latency.
- Creates and updates will be somewhat infrequent and can have higher latency than read operations.
- Latency & throughput: Retrievals by ID need to support low latency and high throughput. Creates and updates can have higher latency than read operations.
- Transactional support: Not required

3. **Recommended service: Azure SQL Database**

- Structured: Business data
- Operations: Read-only, complex analytical queries across multiple databases
- Latency & throughput: Some latency in the results is expected based on the complex nature of the queries.
- Transactional support: Required

Tips to remember, A day prior to the Exam.

Azure Service	Purpose
Azure SQL Data Sync	Synchronization of data between Azure Sql & On-premises SQL with bi-directional
Azure SQL DB Elastic pool's	Depend on eDTUs or Vcore's and max data size
Azure Data Lake Storage	Azure Storage with Hierarchical nature
Azure SQL Database Managed Instance	Data Migration between On-premise & Cloud with almost 100% compatibility e.g.: From on-premises or IaaS, self-built, or ISV provided environment to fully managed PaaS cloud environment, with as low migration effort as possible.
Azure Resource Manager Templates	Used when same operation needs to be performed frequently or daily basis with minimal effort eg: Clusters
Data Migration Assistant	Synchronize data from on-premises Microsoft SQL Server database to Azure SQL Database and to determine whether data will move without compatibility issues
Azure Data Warehouse	Used frequently for Analytical data store
Azure Data Factory	Orchestrate and manage the data lifecycle
Azure Data bricks (Spark)	In memory processing (or) support for usage of Scala, Java, Python, R languages (or) Cluster scale up or scale down
Data load between any of the two services SQl <=> Blob <=> Data-warehouse	99% of the cases we use CTAS(Create Table As Select) and not other operations such as Insert into, so on..

Azure Service	Purpose
Azure Database Migration Service (DMS)	A fully managed service designed to enable seamless migrations from multiple database sources to Azure data platforms with minimal downtime (online migrations).
Database Experimentation Assistant (DEA)	Helps you evaluate a targeted version of SQL Server for a specific workload. Customers upgrading from earlier versions of SQL Server (starting with 2005) to more recent versions of SQL Server can use the analysis metrics that the tool provides.
SQL Server Migration Assistant (SSMA)	A tool designed to automate database migration to SQL Server from Microsoft Access, DB2, MySQL, Oracle, and SAP ASE.

Azure Data Warehouse | Synapse Analytics

Azure Data Warehouse	Data distribution	Reason	Fit For
Small Dimension Table	Replicated	Data size usually less than 2 GB	star schema with less than 2 GB of storage after compression
Temporary/Staging Table	Round Robin	Data size usually less than 5 GB	No obvious joining key or good candidate column
Fact Table	Hash Distributed	Data Size is huge more than 100 GB	Large dimension tables

Azure Data Warehouse | Synapse Analytics Selection of Table Index

Type	Fit For
Heap	Staging or temporary table, Small tables with small lookups
Clustered index	Tables with up to 100 million rows, Large tables (more than 100 million rows) with only 1-2 columns heavily used
Clustered column store index (CCI) (default)	Large tables (more than 100 million rows)

Note:

- Preferred Index type is usually Clustered Colum-Store.
    - e.g.: Similar to Parquet file.

Data bricks - Cluster Configurations

	STANDARD	HIGH CONCURRENCY
Recommended for...	Single User	Multiple Users
Language Support	SQL, Python, R, and Scala	SQL, Python, and R (not Scala)
Notebook Isolation	No	Yes

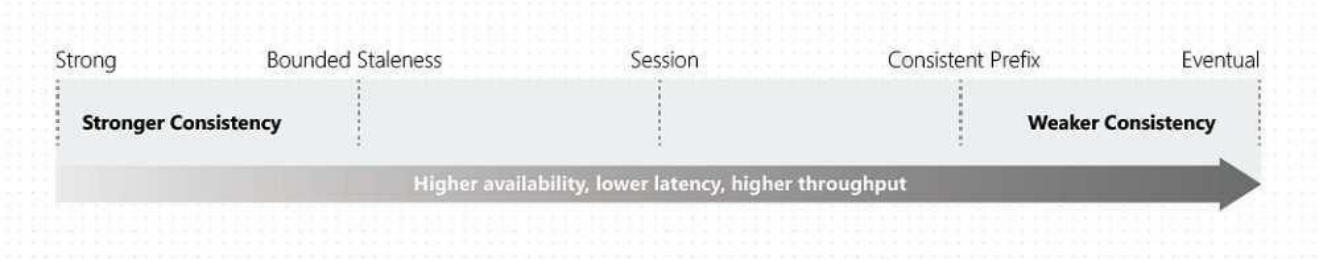
Azure Data Factory Triggers

TYPE	DESCRIPTION
Schedule	Runs on a wall-clock schedule (e.g. every X mins/h/d/w/m's).
Tumbling Window	A series of fixed-sized, non-overlapping, and contiguous time intervals.
Event-based	Runs pipelines in response to an event (e.g. Blob Created/Deleted).

Azure Data Factory Integration Runtime (IR) Usage

IR type	Public network	Private network
Azure	Data Flow Data movement Activity dispatch	
Self-hosted	Data movement Activity dispatch	Data movement Activity dispatch
Azure-SSIS	SSIS package execution	SSIS package execution

Cosmos DB Consistency level



Cosmos DB Entities by API

ENTITY	SQL	CASSANDRA	MONGODB	GREMLIN	TABLE
Container	Container	Table	Collection	Graph	Table
Item	Document	Row	Document	Node or Edge	Item
Database	Database	Key space	Database	Database	N/A

Azure SQL DB Dynamic Data Masking

Data Masking Attribute	Masking value	Example
Default	Zero value for numeric data types (or) Mask 4 or less string data type characters	Default value (0, xxxx, 01-01-1900)
Custom	Mask everything except characters at beginning and at the end	Custom string (prefix [padding] suffix)
Random Number	Returns a random number	Random number range e.g.: 0 to 100
Email	Mask first letter and domain	aXX@XXXX.com
Credit card	Exposes the last four digits of the designated fields	XXXX-XXXX-XXXX-1234

Performance Parameters

Azure Service	Parameters
Azure Stream Analytics	Depends on Streaming Unit
Azure SQL DB	Depends on DTU's
Azure Data warehouse	Depends on Cache used. Unit measured is Data warehouse Units (DWU)
Azure Cosmos DB	Depends on Data Integration Unit (or) Request Units (RU)

Pillars of Azure Architecture

Design for Performance and Scalability

- Scaling
    - Compute resources can be scaled in two different directions:
        - Scaling up is the action of adding more resources to a single instance.
        - Scaling out is the addition of instances.
- Performance When optimizing for performance, you'lll look at network and storage to ensure performance is acceptable. Both can impact the response time of your application and databases.
- Patterns and Practices
    - Partitioning
        - In many large-scale solutions, data is divided into seperate partitions that can be managed and accessed seperatly.
    - Scaling
        - Is the process of allocating scale units to match performance requiremnets. This can be done either automatically or manually
    - Caching
        - Is a mechanism to store frequently used data or assests (web pages, images) for faster retrieval.

Design for Availability and Recoverability

- Availabilty
    - Focus on maintaining uptime through small-scale incidents and temporary conditions like partial network outages.
- Recoverability
    - Focus on recovery from data loss and from large scale disasters.
    - **Recovery Ponit Objective**
        - The maximum duration of acceptable data loss.
    - **Recovery Time Objective**
        - The maximum duration of acceptable downtime.

Design Azure Data Storage Solutions

Azure Storage

BLOB

- It is also a backbone for creating a storage account that can be used as a Data Lake storage

CosmosDB

- Globally distributed and elastically scalable database.

i) Core (SQL) API

- Default API for Azure Cosmos DB
- Can query hierarchical JSON documents with a SQL-like language
- Uses Javascript's type system, expression evaluation, and function invocation.

ii) MongoDB API

- Allows existing MongoDB client SDKs, drivers, and tools to interact with the data transparently, as if they are running against an actual MongoDB database.
- Data is stored in document format, similar to Core (SQL)

iii) Cassandra API

- Using Cassandra Query language (CQL), the data will appear to be a partitioned row store.

iv) Table API

- The original table API only allows for indexing on the partition and row keys; there are no secondary indexes.
- Storing table data in Cosmos DB automatically indexes all the properties, requires no index management.
- Querying is accomplished by using OData and LINQ queries in code, and the original REST API for GET operations.

v) Gremlin API

- Provides a graph based view over the data. Remember that at the lowest level, all data in any Azure Cosmos DB is stored in an ARS format.
- Use a traversal language to query a graph database, and Azure Cosmos DB supports Apache Tinkepop's Gremlin language.

Analyze the storage decision criteria

- Does the schema change a lot?
  - A traditional document database is a good fit in these scenarios, making Core (SQL) a good choice.
- Is there important data about the relationships between items in the database?
  - Relationships that require metadata to be stored for them are best represented in a graph database.
- Does the data consist of simple key-value pairs?
  - Before Azure Cosmos DB existed, Redis or the Table API might have been a good fit for this kind of data; however, Core (SQL) API is now the better choice, as it offers a richer query experience, with improved indexing over the Table API.

Scenario's to choose different CosmosDB API's

Use Core (SQL) to store a product catalog

Problem analysis

- The system needs to support searching and sorting across many different properties. There is a structured relational database that can be used to import data.

Recommended API: Core (SQL)

- The existing app uses a traditional relational database, which means that none of the other APIs are currently being used.

Why not any of the other APIs?

API	Description
Azure Table	This API should only be used to allow existing apps that are based on the Table API access to Azure Cosmos DB. However, new projects should always choose Core (SQL).
Cassandra	This API isn't a good choice in this particular scenario, because the schema is unknown and will change over time.
Gremlin	This API isn't a good choice since the scenario doesn't need to process graph-based data.
MongoDB	MongoDB's lack of support for SQL-like queries give Core (SQL) an advantage for your existing relational database users.

Use the Gremlin (graph) API as a recommendation engine

- A graph database is the perfect fit to model this kind of data. Gremlin, the graph query language, will support the marketing department's requirements.

Use MongoDB to import historical order data

- Problem analysis
  - The operations team has semi-structured data that needs the flexibility to store many different supplier order formats.
- Recommended API: MongoDB
  - To allow the operations team to continue to use their existing app that uses MongoDB queries, your best option is to use the MongoDB API.
- Why not any of the other APIs?

```
- We are not looking into any relationships so Gremlin is not the right choice
- Other CosmosDB API's are not used since the existing queries are MongoDB native and
there MongoDB is the best fit
```

Use Cassandra for web analytics

- Problem analysis
- Recommended API: Cassandra
- Why not any of the other APIs?

API	Description
Azure Table	This API should only be used to allow existing apps that are based on the Table API access to Azure Cosmos DB.
Core(SQL)	All of the requirements for your web analytics application can be satisfied by Core (SQL), which makes your decision difficult when choosing between Core (SQL) and Cassandra. Since the web team is already using their Cassandra-based application, and because of their prior experience using the Cassandra Query Language (CQL) for some of their reporting, Cassandra is the right choice for this scenario, although Core (SQL) is still a close second choice.
Gremlin	This API isn't a good choice because the data isn't graph based.
MongoDB	The flexibility of a document-based data store is not enough of a reason to use

Use the Azure Table API to store IoT data

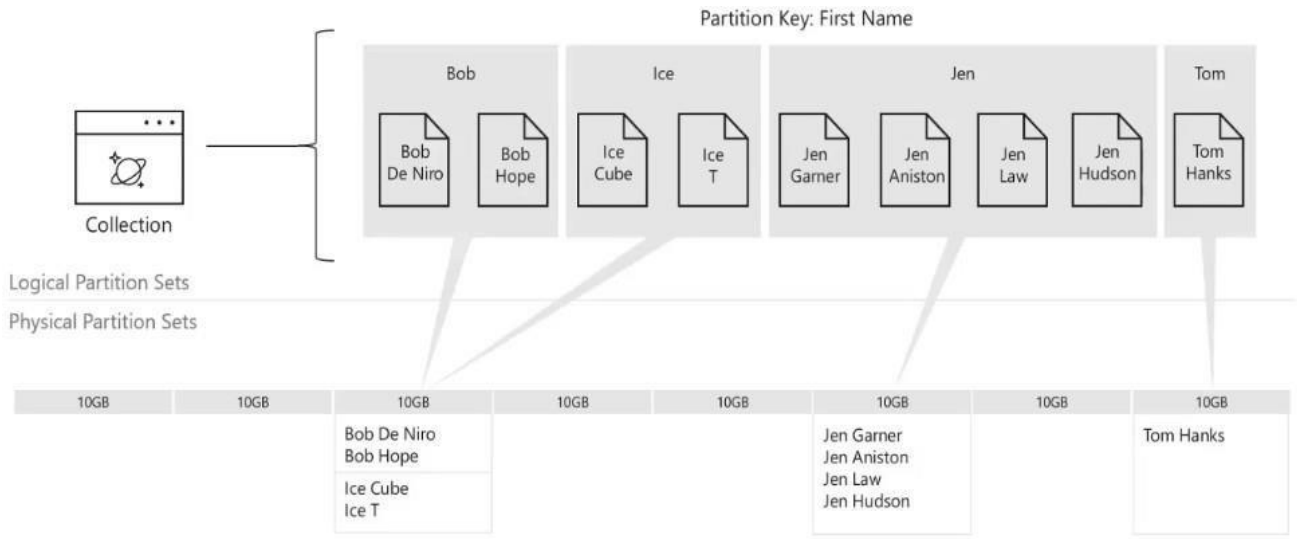
- Problem analysis
- Recommended API: Azure Table

API	Description
Cassandra	This API isn't a good choice because of the existing Azure Table Storage database, and because of the requirements to import and reuse application code.
Core(SQL)	This API would be the best choice if you were designing a new system; however, since this scenario consists of a legacy application with a large existing Azure Table Storage dataset, the Azure Table API is the correct choice.
Gremlin	This API isn't a good choice because this scenario doesn't need to process graph-based data, and because of the requirements to import and reuse application code.
MongoDB	This API isn't a good choice because of the existing Azure Table Storage database, and because of the requirements to import and reuse application code.

Request Unit Considerations for CosmosDB

- Item size
- Item indexing
- Item property count
- Indexed properties
- Data consistency
- Query patterns
- Script usage

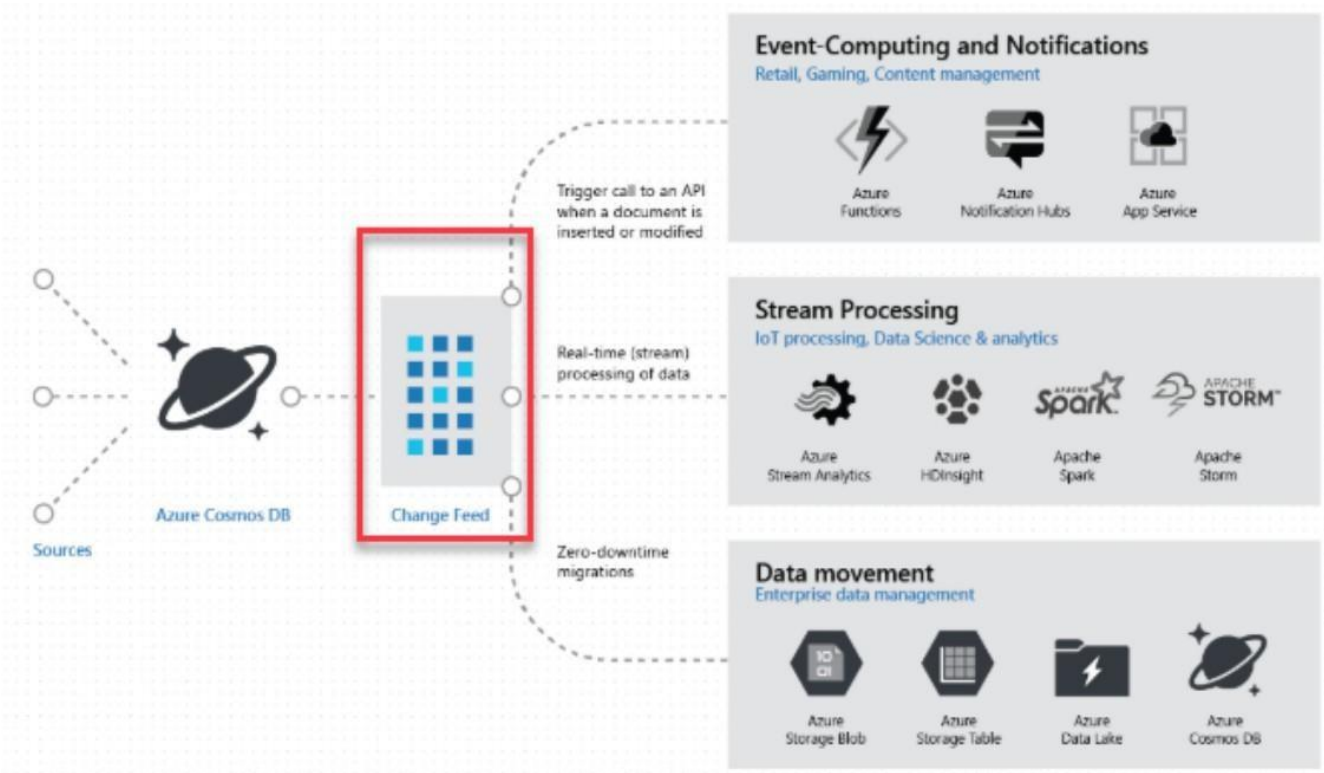
CosmosDB Partition Design



- Items are placed into logical partitions by partition key
- Partition keys should generally be based on unique values
- Ideally the partition key should be part of a query to prevent "fan out"
- Logical partitions are mapped to physical partitions
- A physical partition always contains at least one logical partition
- Physical partitions are capped at 10GB
- As physical partitions fill-up they will seamlessly split
- Logical partitions cannot be split

Cosmos DB Change Feed feature

- Enables you to build efficient and scalable solutions for each of the patterns shown below



When to use Azure Cosmos DB

Deploy Azure Cosmos DB when need a NoSQL database of the supported API model, at planet scale, and with low latency performance.

Currently, Azure Cosmos DB supports five-nines uptime (99.999 percent).

It can support response times below 10 ms when it's provisioned correctly.

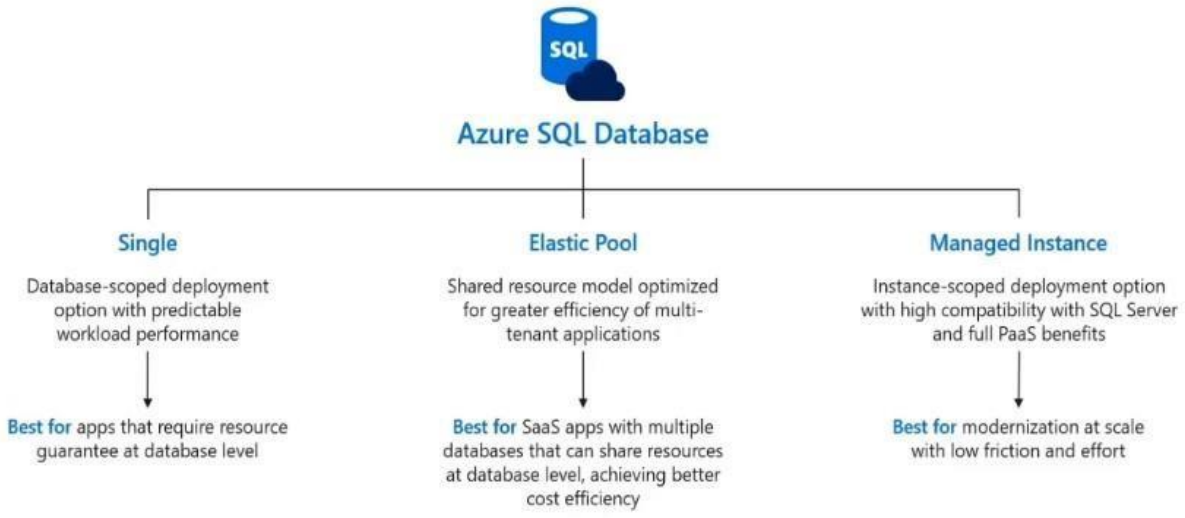
Differences between Azure Storage tables and Azure Cosmos DB tables

- Query results from Azure Cosmos DB are not sorted in order of partition key and row key as they are from Storage tables.
- Row keys in Azure Cosmos DB are limited to 255 bytes.
- Batch operations are limited to 2 MBs.
- Cross-Origin Resource Sharing (CORS) is not currently supported by Azure Cosmos DB.
- Table names are case-sensitive in Azure Cosmos DB. They are not case-sensitive in Storage tables.

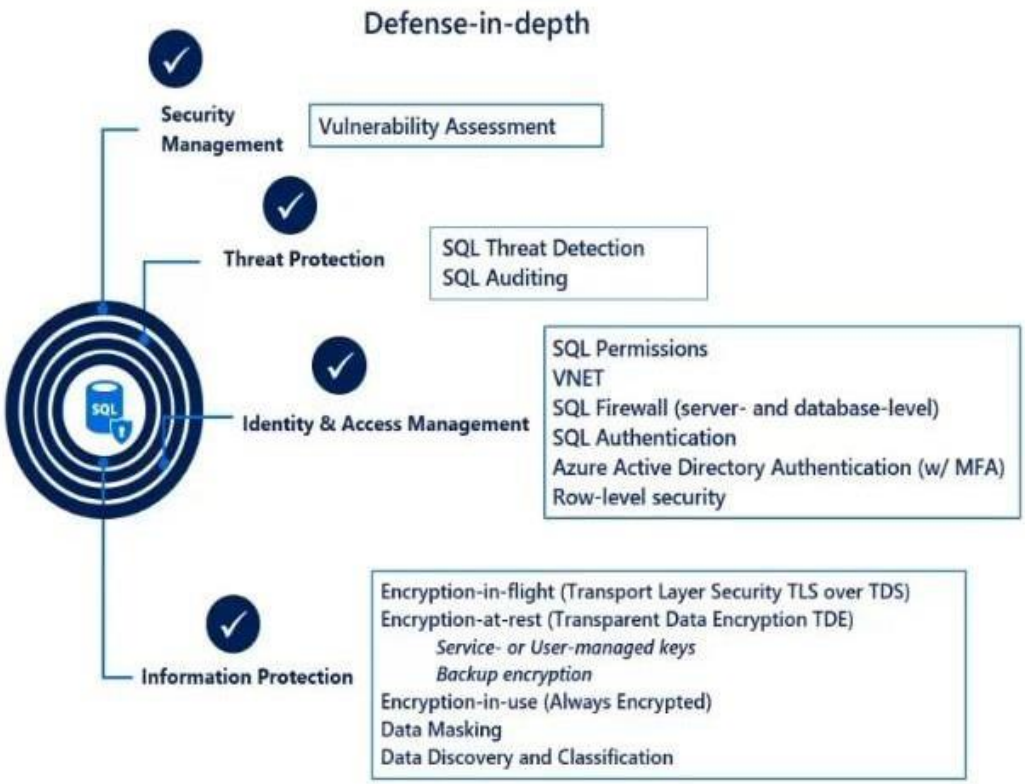
How to choose a storage location

Priority	Azure Storage Tables	Azure Cosmos DB Tables
Latency	Responses are fast, but there is no guaranteed response time.	< 10 ms for reads, < 15 ms for writes
Throughput	Maximum 20,000 operations/sec	No upper limit on throughput. Over 10 million operations/sec/table.
Global distribution	Single region for writes. A secondary read-only region is possible with read-access geo-redundant replication.	Replication of data for read and write to more than 30 regions.
Indexes	A single primary key on the partition key and the row key. No other indexes.	Indexes are created automatically on all properties.
Data consistency	Strong in the primary region. If you are using read-access geo-redundant replication, it may take time for changes to reach the secondary region.	You can choose from five different consistency levels depending on your needs for availability, latency, throughput, and consistency.
Pricing	Optimized for storage.	Optimized for throughput.
SLAs	99.99% availability.	99.99% availability for single region and relaxed consistency databases. 99.999% availability for multi-region databases.

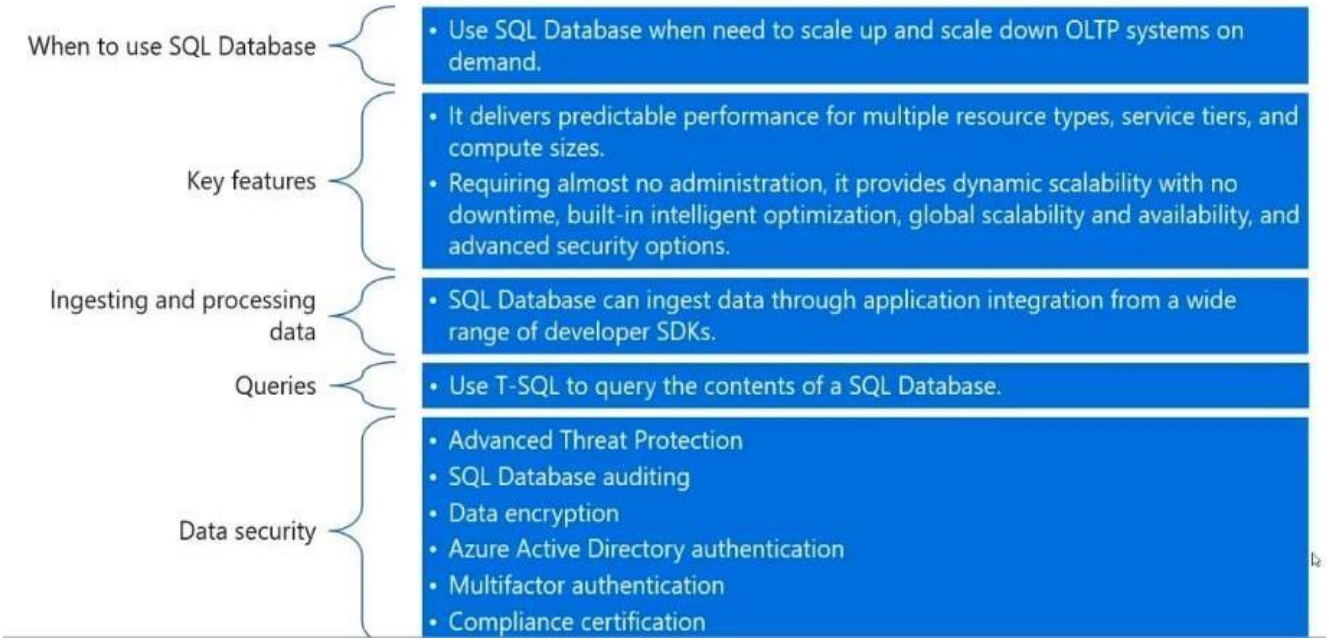
Azure SQL Database hosting options



SQL Security



Scenario to choose Azure SQL



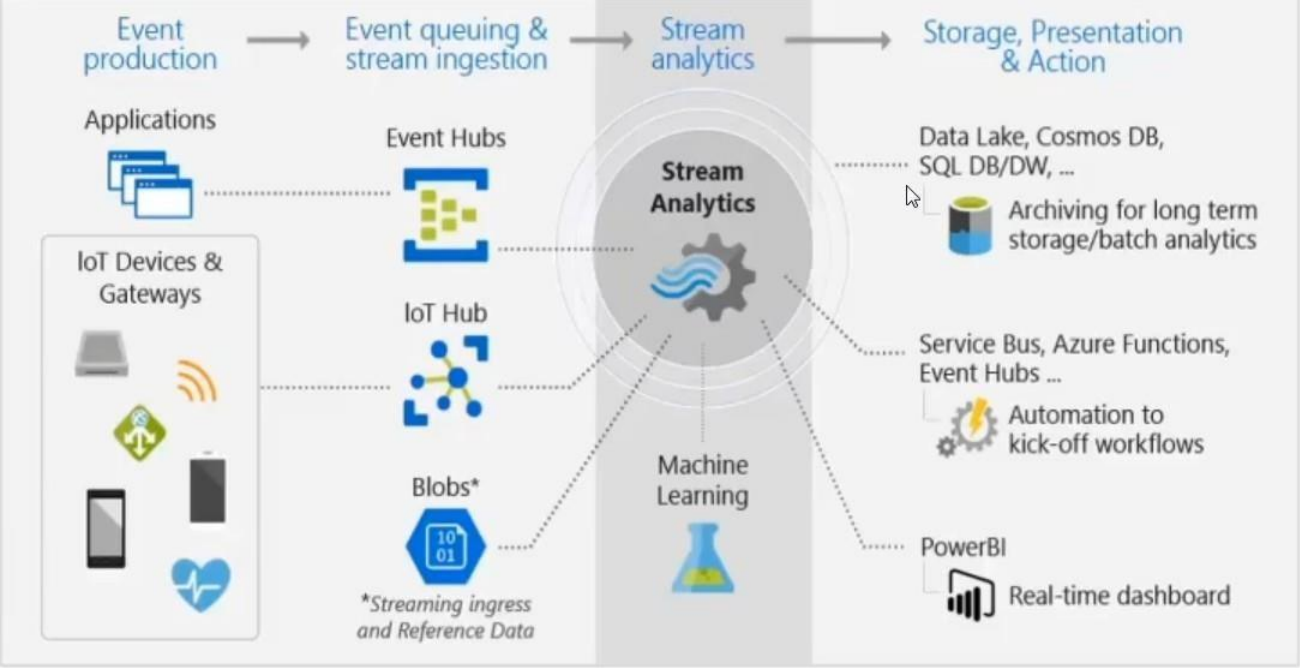
Scenario to choose Azure Synapse

When to use SQL Data Warehouse	Data loads can increase the processing time for on-premises data warehousing solutions.
Key features	SQL Data Warehouse uses massively parallel processing (MPP) to quickly run queries across petabytes of data
	SQL Data Warehouse can also pause and resume the compute layer.
Ingesting and processing data	SQL Data Warehouse uses the extract, load, and transform (ELT) approach for bulk data.
Queries	Load data fast by using PolyBase with additional Transact-SQL constructs such as CREATE TABLE and AS SELECT.
Data Security	SQL Data Warehouse supports both SQL Server authentication and Azure Active Directory.
	SQL Data Warehouse supports security at the level of both columns and rows.

Scenario to choose Azure Data Lake Storage Gen2

- Data Lake Storage is designed to store massive amounts of data for big-data analytics.
- Data Lake Storage Gen2 reduces computation times, making the research faster and less expensive.
- The compute aspect that sits above this storage can vary.
- The aspect can include platforms like HDInsight, Hadoop, Cloudera, Azure Databricks, and Hortonworks.

Azure Stream Analytics



Scenario to choose Stream Analytics

In real time, data is ingested from applications or IoT devices and gateways into an event hub or IoT hub.

The event hub or IoT hub then streams the data into Stream Analytics for real-time analysis.

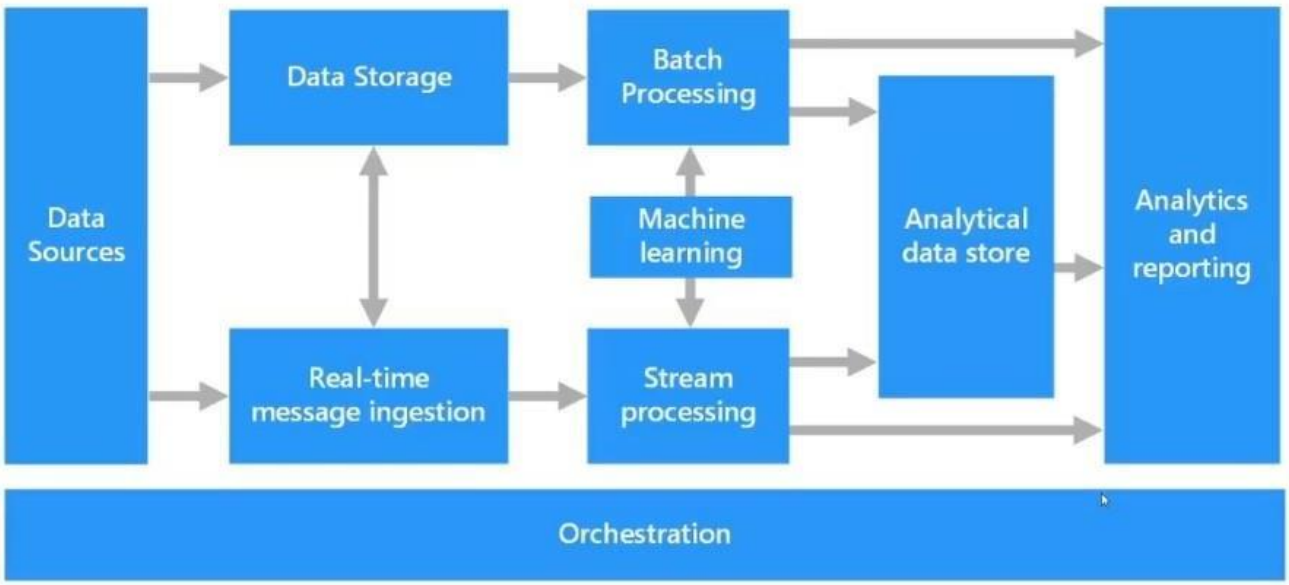
Batch systems process groups of data that are stored in an Azure Blob store.

They do this in a single job that runs at a predefined interval.

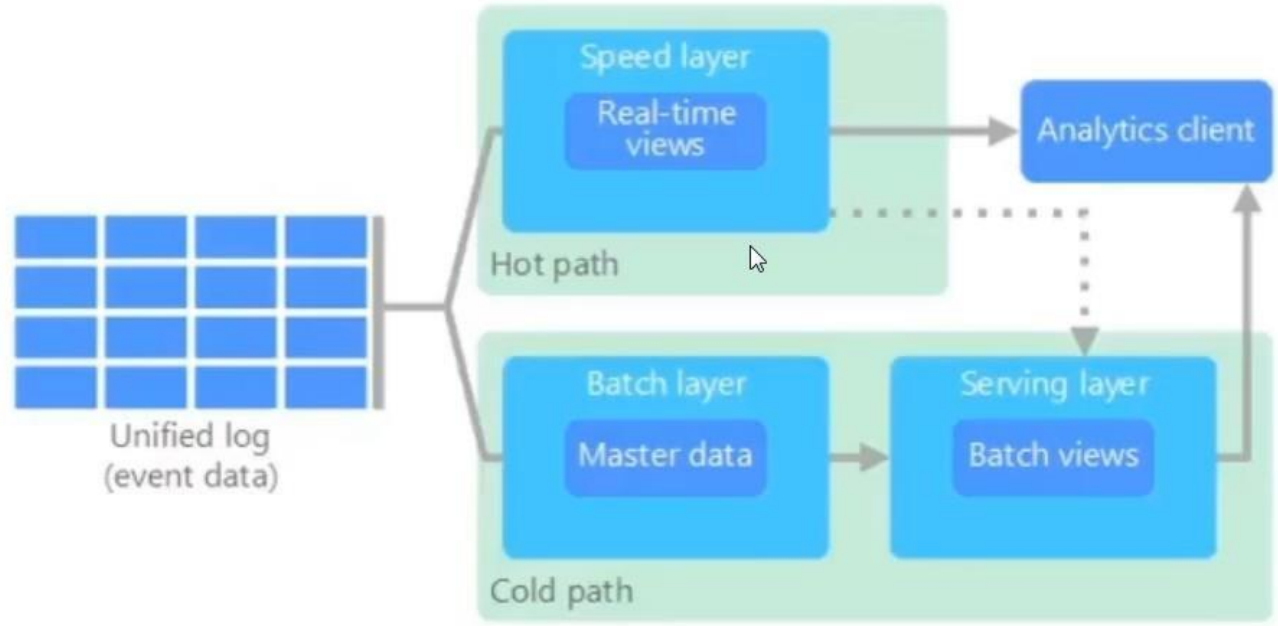
Don't use batch systems for business intelligence systems that can't tolerate the predefined interval.

3. Design Data Processing Solutions

Components of Big Data architecture



Lambda Architecture

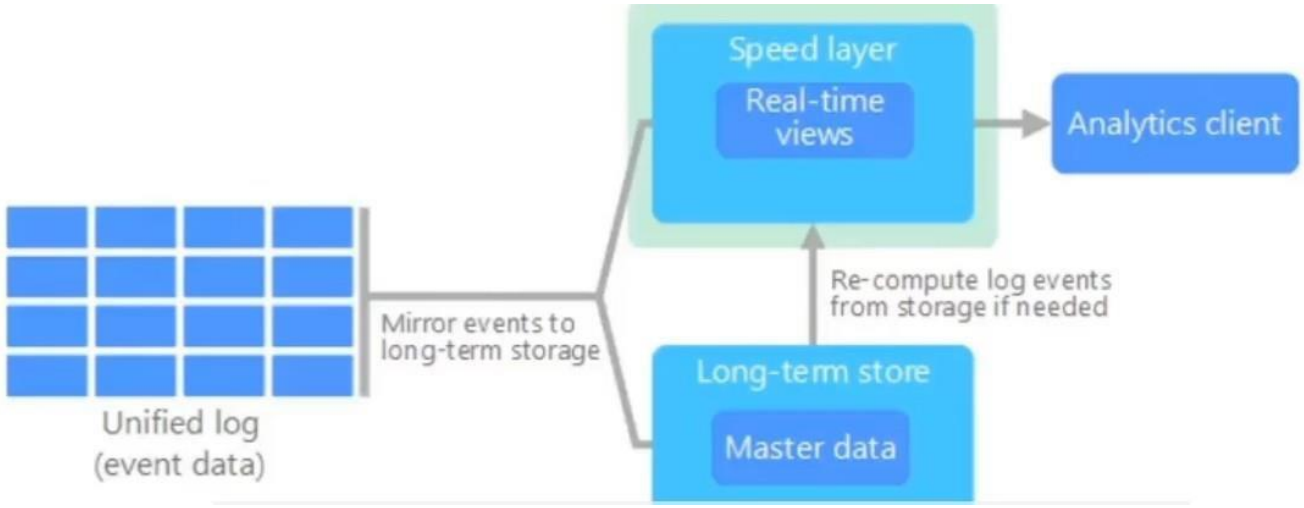


- When working with very large data sets, it can take a long time to run the sort of queries that clients need.
- Often require algorithms such as Spark/ Map reduce that operate in parallel across the entire data set.
- The results are then stored separately from the raw data and used for querying.
- Drawback to this approach is that it introduces latency

The lambda architecture, addresses this problem by creating two paths for data flow:

- Batch layer (cold path)
- Speed layer (hot path)

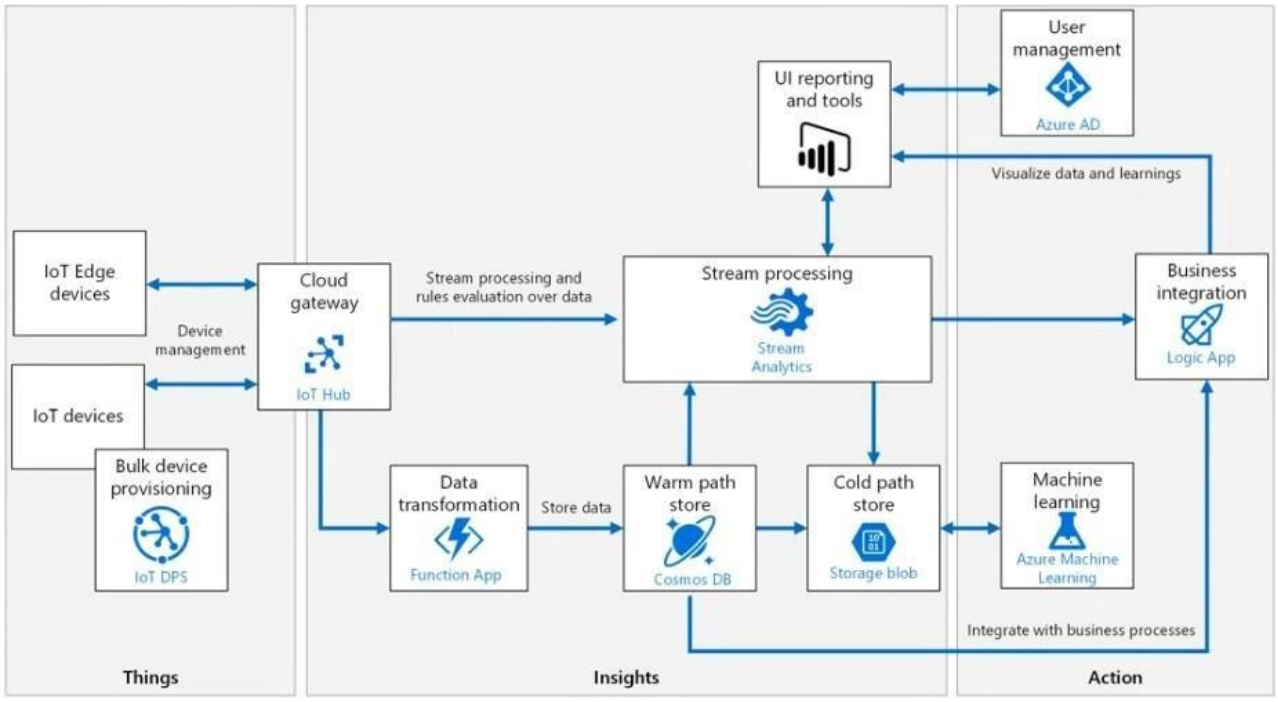
Kappa Architecture



- A drawback to the lambda architecture is its complexity.
- Processing logic appears in two different places - the cold and hot paths - using different frameworks
- This leads to duplicate computation logic and the complexity of managing the architecture for both paths.
- The kappa architecture was proposed by Jay Kreps as an alternative to the lambda architecture.
- All data flows through a single path, using a stream processing system.

IOT

Azure IOT Reference Architecture

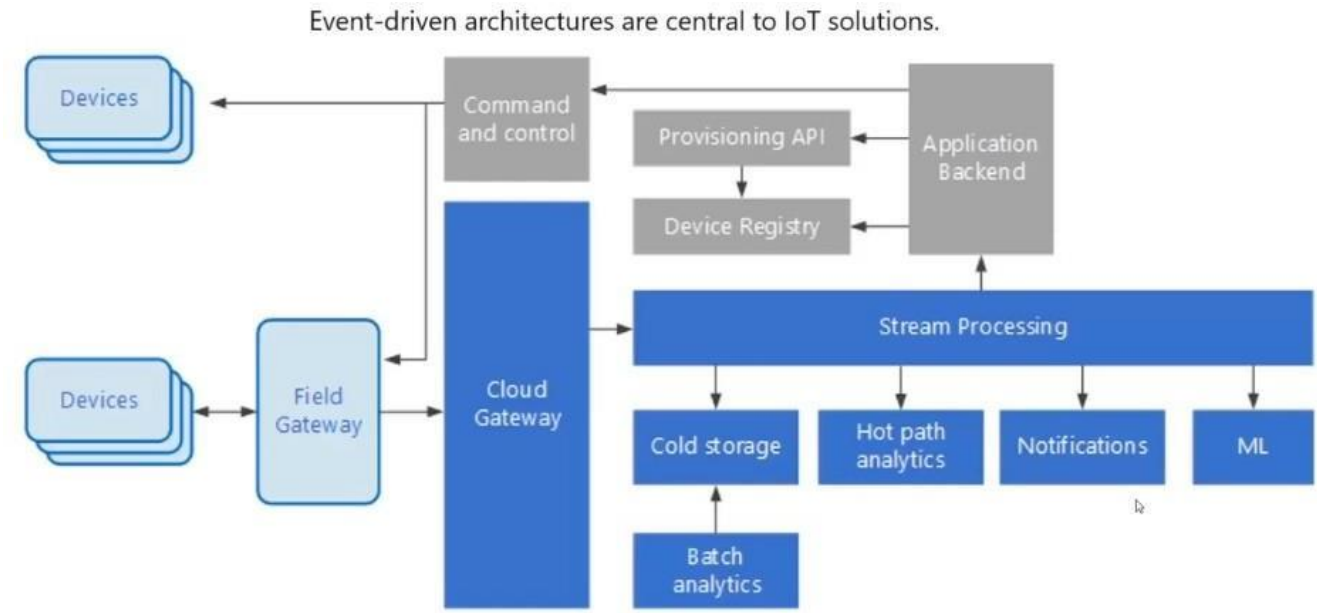


IOT Edge devices: Devices cannot be constantly connected to the cloud in this case IOT edge devices contian some processing analysis logic within it. So that there is no constant dependency for the cloud.

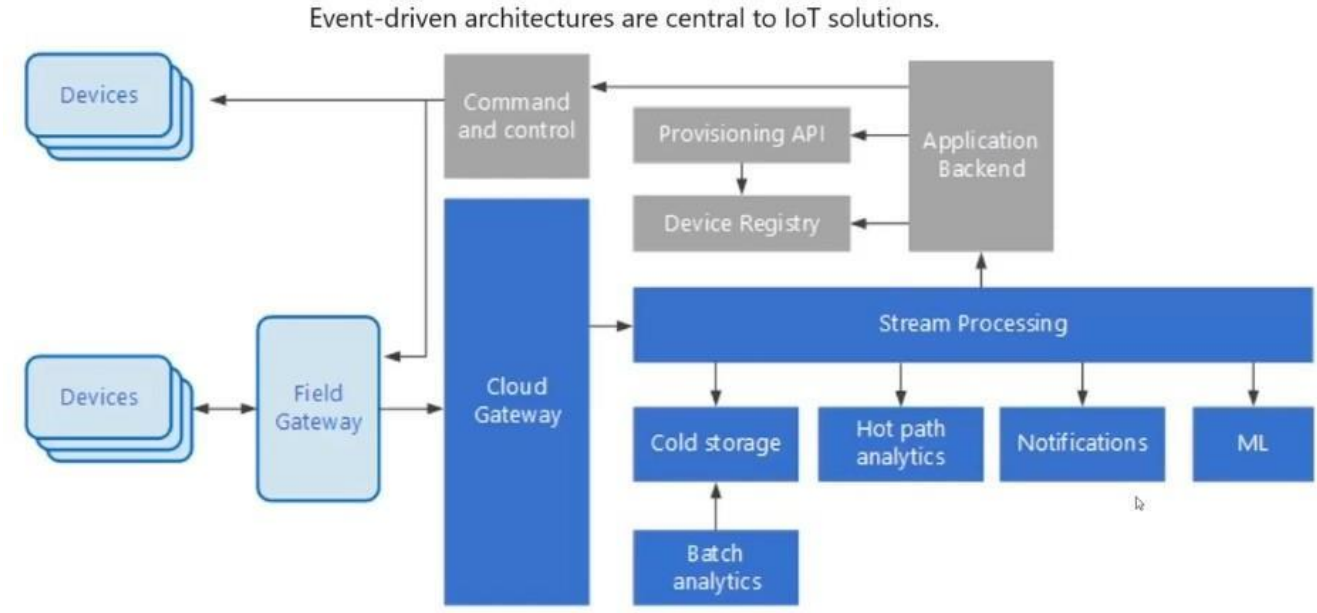
- eg: Shipment containers

IOT devices: Are constantly connected to the cloud which provides capability tp perform data processing and analysis

Cloud Gateway (IOT Hub): Provides a cloud for a device to connect securely to the cloud and send data. It acts a message broker between the devices and the other azure services.



Batch Processing



Scenario's to use Batch Processing

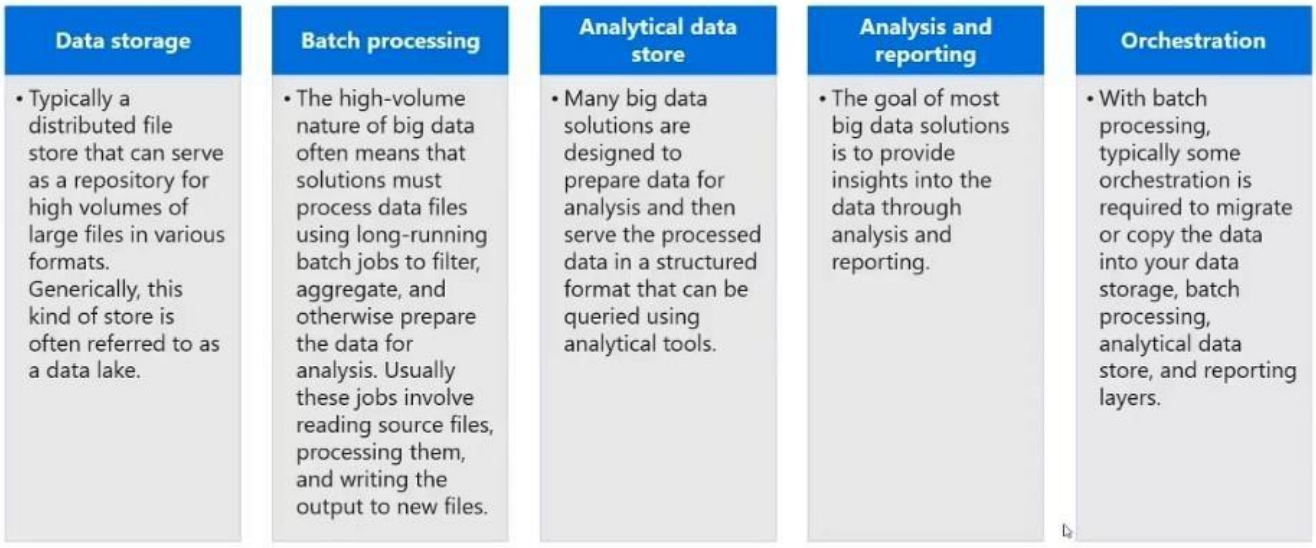
- From simple data transformations to a more complete ETL (extract-transform-load) pipeline

- In a big data context, batch processing may operate over very large data sets, where the computation takes significant time.
- One example of batch processing is transforming a large set of flat, semi-structured CSV or JSON files into a schematized and structured format that is ready for further querying.

Design considerations for Batch processing

- Data format and encoding
    - When files use an unexpected format or encoding
        - Example is text fields that contain tabs, spaces, or commas that are interpreted as delimiters
    - Data loading and parsing logic must be flexible enough to detect and handle these issues.
- Orchestrating time slices
    - Often source data is placed in a folder hierarchy that reflects processing windows, organized by year, month, day, hour, and so on.
    - Can the downstream processing logic handle out-of-order records?

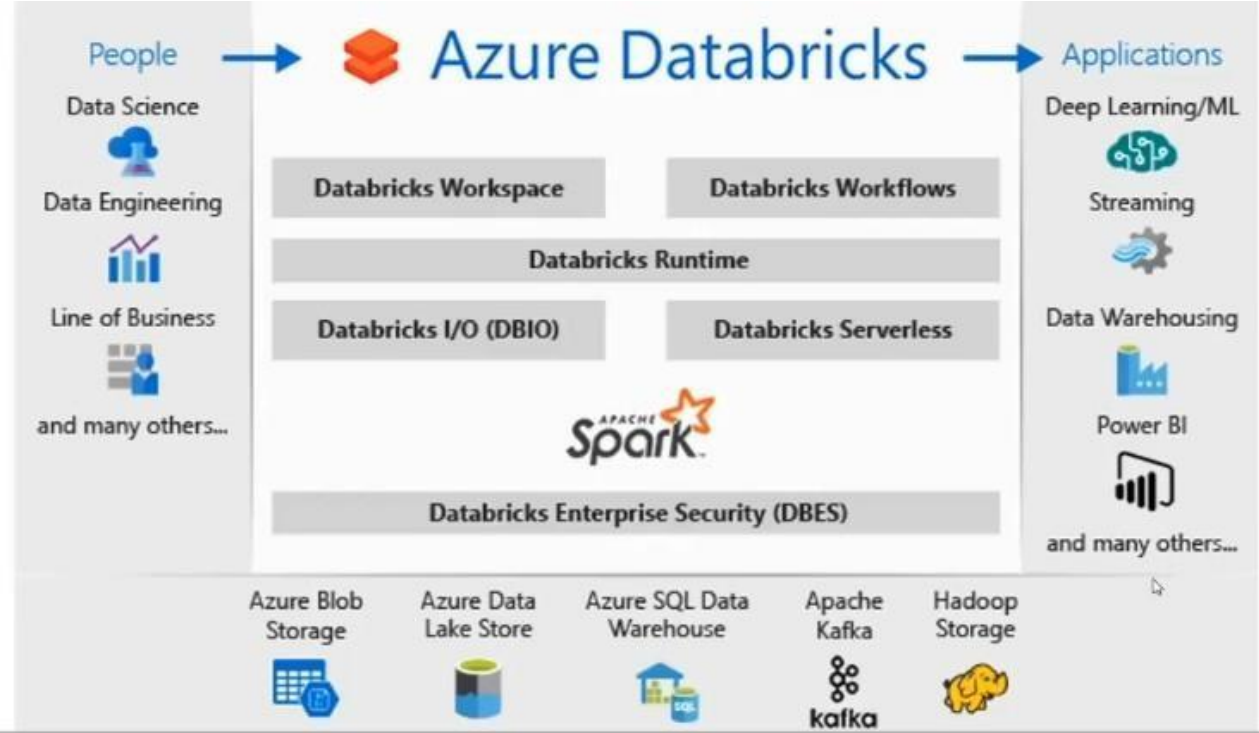
Batch processing Logical components



Batch processing Technology choices

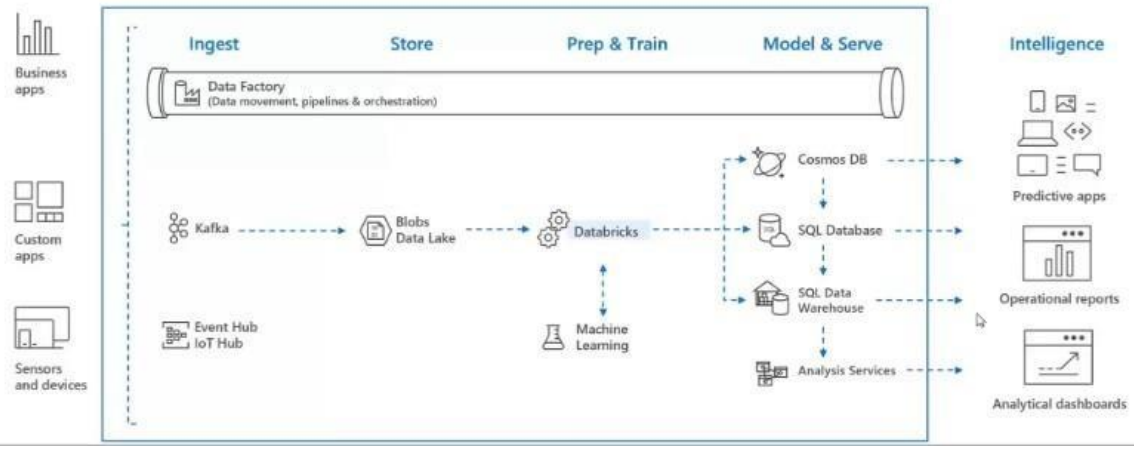
Data Storage	Batch Processing	Analytical data store	Analytics and Reporting	Orchestration
• Azure Storage Blob Containers • Azure Data Lake Store	• U_SQL • Hive • Pig • Spark	• Azure Synapse Analytics • Spark SQL • HBase • Hive	• Azure Analysis Services • Power BI • Microsoft Excel	• Azure Data Factory • Oozie and Scoop

Batch processing with Azure Databricks



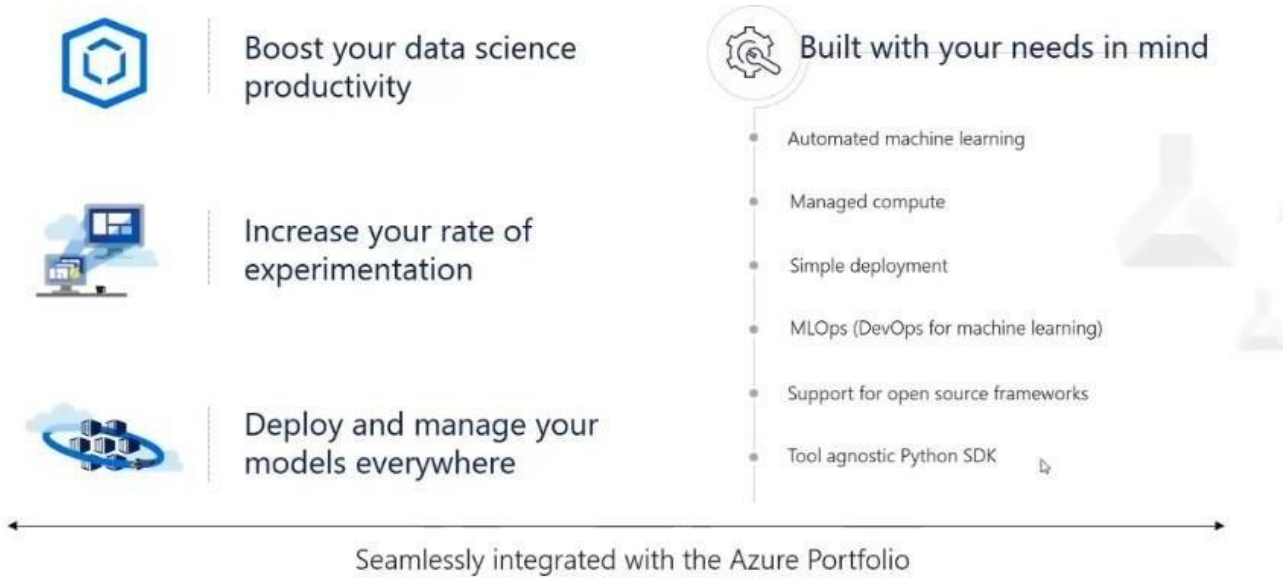
- Fast cluster start times, auto termination, autoscaling
- Built-in integration with Azure Blob Storage, ADLS, Azure Synapse, and other services.
- User authentication with Azure Active Directory.
- Web-based notebooks for collaboration and data exploration.
- Supports GPU-enabled clusters.

Usage of Azure Databricks

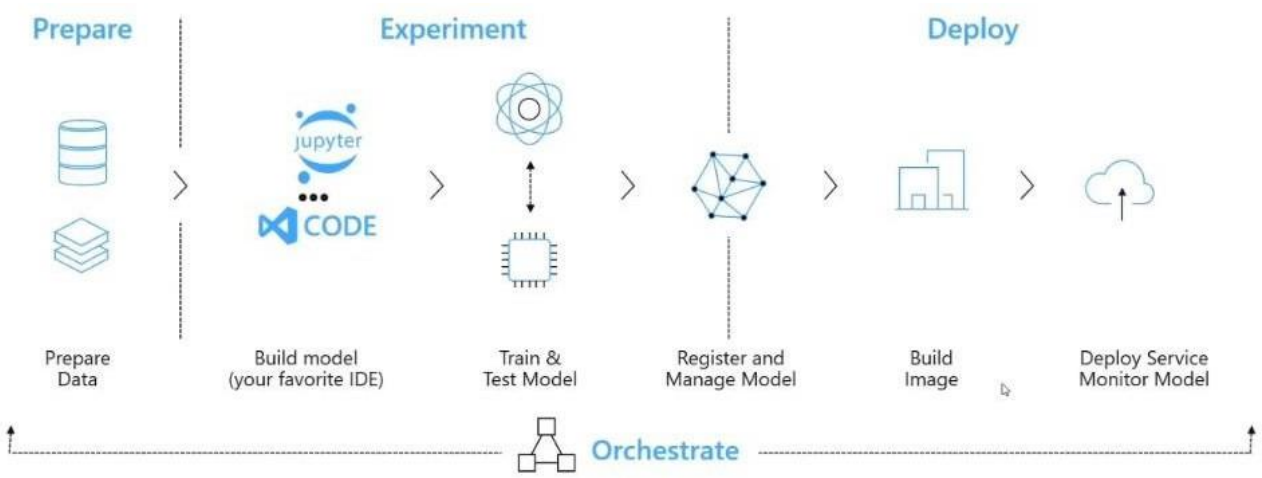


- To read data from multiple data sources such as Azure Blob Storage, ADLS, Azure Cosmos DB, or SQL DW and turn it into breakthrough insights using spark.

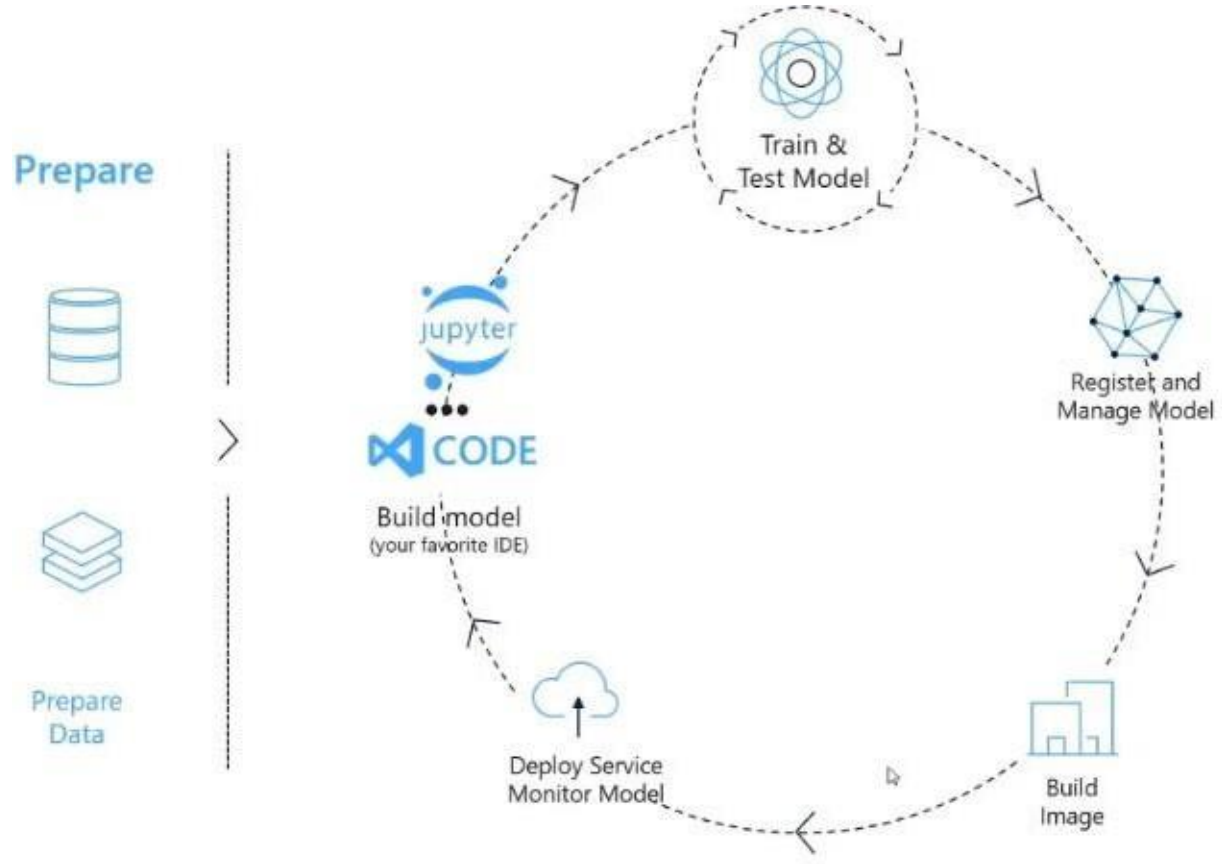
Azure Machine Learning



Machine Learning Typical E2E process



Devops Loop for Data Science

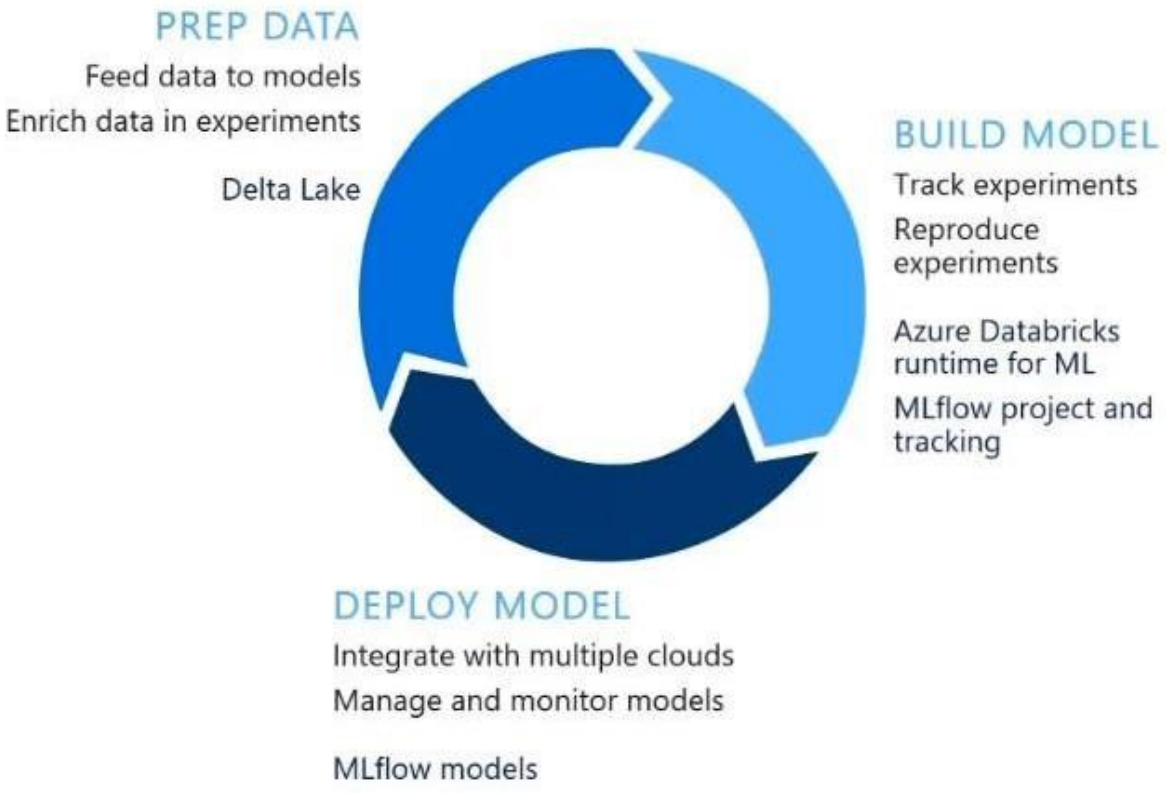


Azure Databricks + Azure ML

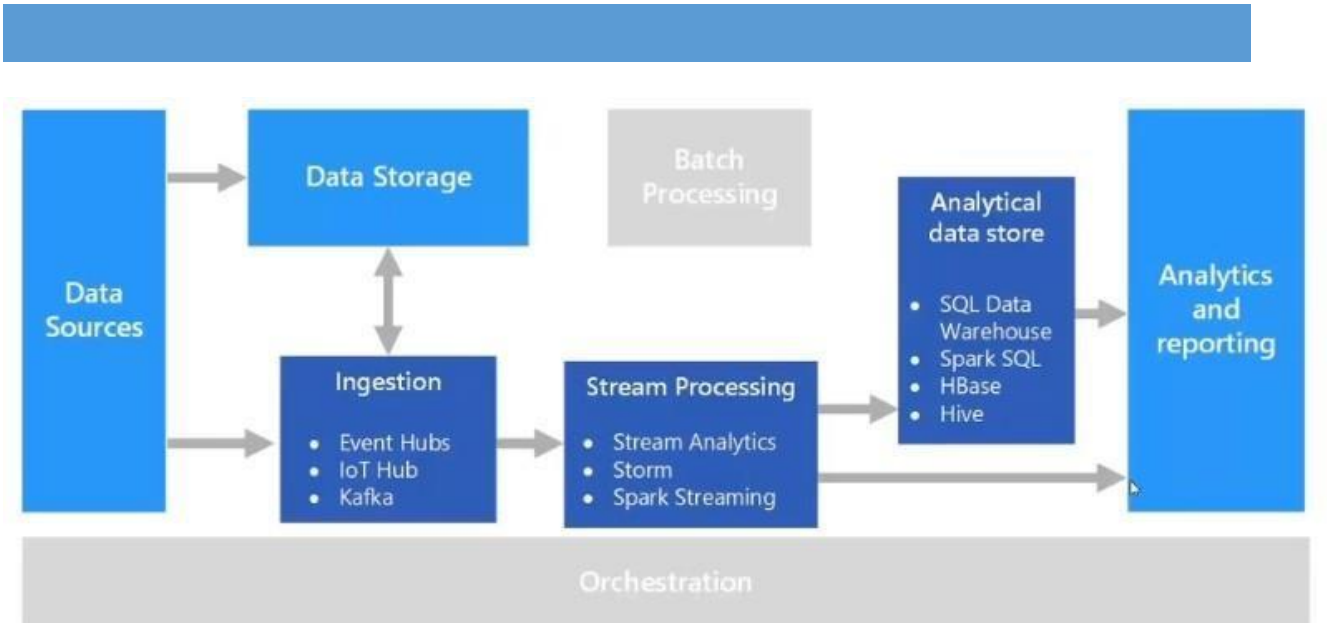
- Log experiments and models in a central place

- Maintain audit trails centrally
- Deploy models seamlessly in Azure ML
- Manage your models in Azure ML

Standardizing the ML lifecycle on Azure Databricks



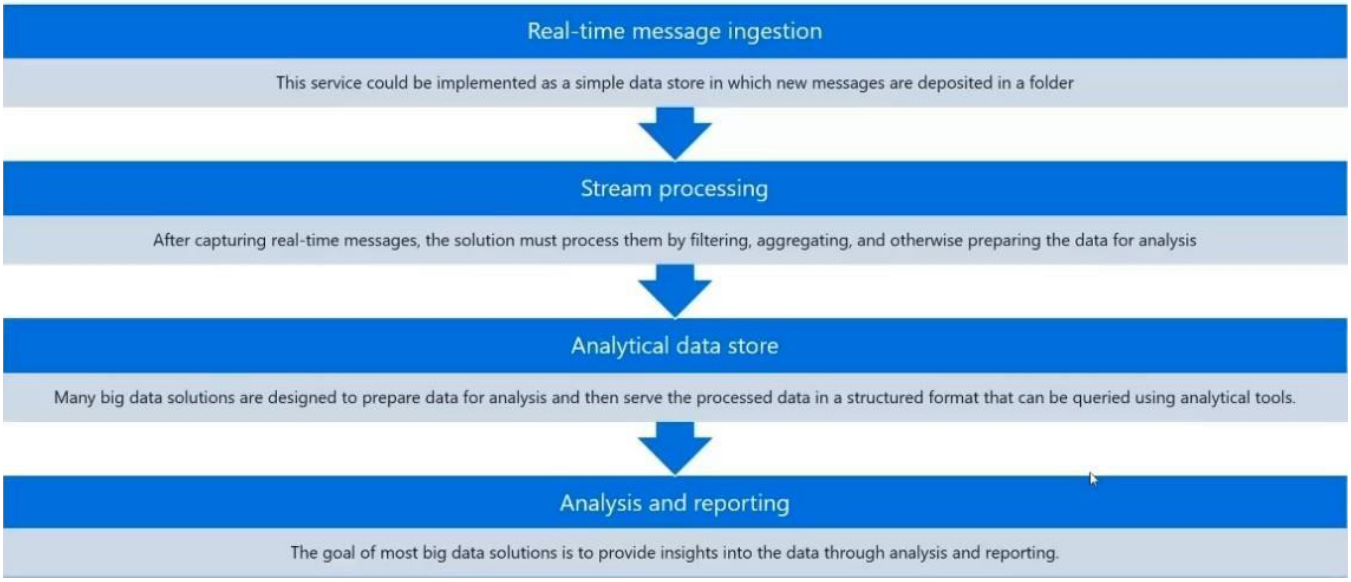
Realtime Processing



Challenges

- One of the big challenges of real-time processing solutions is to ingest, process, and store messages in real time, especially at high volumes.
- Processing must be done in such a way that it does not block the ingestion pipeline.
- The data store must support high-volume writes.
- Another challenge is to act on data quickly such as generating alerts in real time or presenting the data in a real-time (or near real-time) dashboard.

Real Time Processing Architecture - Logical COmponents



Real Time Processing Technology choices

Real-time message ingestion	Data storage	Stream processing	Analytical data store	Analytics and reporting
• Azure Event Hubs • Azure IoT Hub • Apache Kafka	• Azure Storage Blob Containers or Azure Data Lake Store	• Azure Stream Analytics • Storm • Spark Streaming	• Azure Synapse Analytics, HBase, Spark, or Hive	• Azure Analysis Services, Power BI, and Microsoft Excel

Azure Data Factory (ADF)

Data Integration Service: Serverless, Scalable, Hybrid

Data Movement and Transformation @Scale
Cloud & Hybrid w/ 90+ connectors provided
Up to 4 GB/s, ETL/ELT in the cloud

Hybrid Pipeline Model
Seamlessly span: on premise, Azure, other clouds & SaaS
Run on-demand, scheduled, data-availability or on event

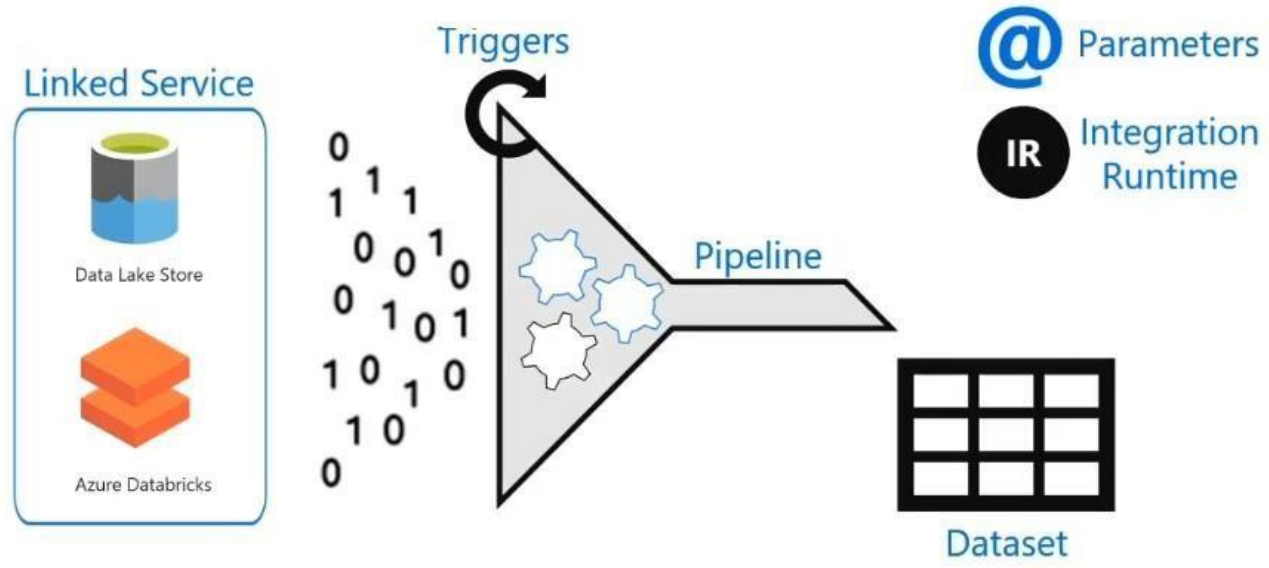
Author & Monitor
Programmability w/ multi-language SDK
Visual Tools

SSIS Package Execution
Lift existing SQL Server ETL to Azure
Use existing tools (SSMS, SSDT)

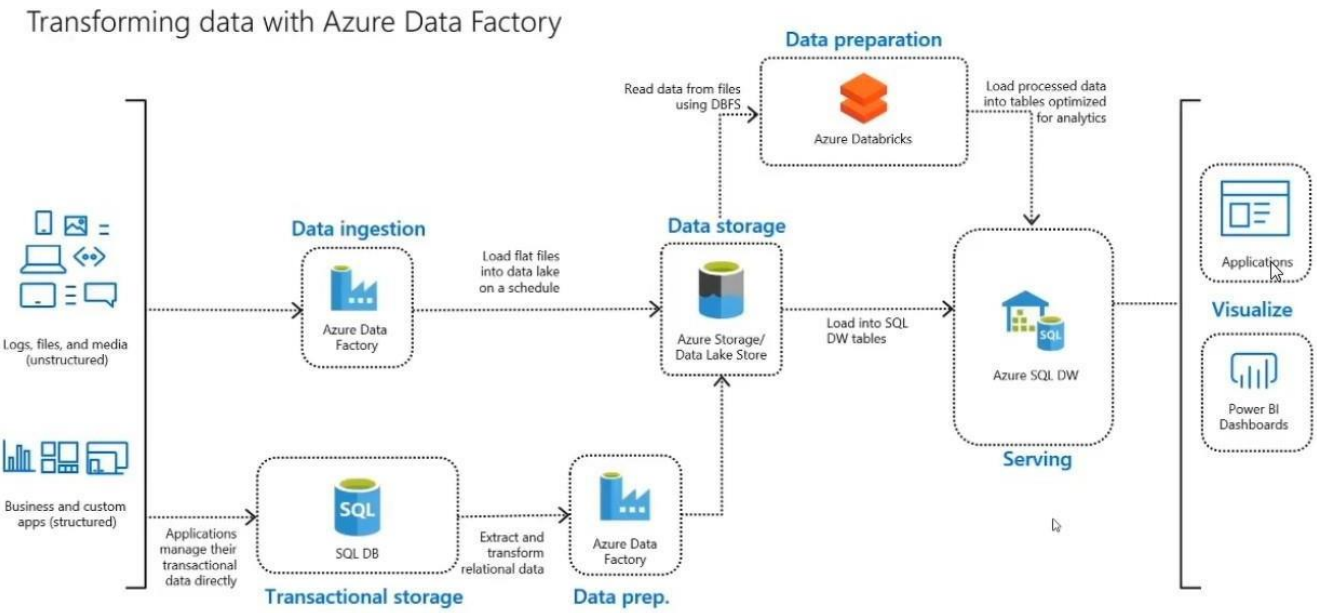
A cloud-based data integration service that allows you to orchestrate and automate data movement and data transformation.

- Connect & collect
- Transform and enrich

ADF Components



Data Transformation in Azure

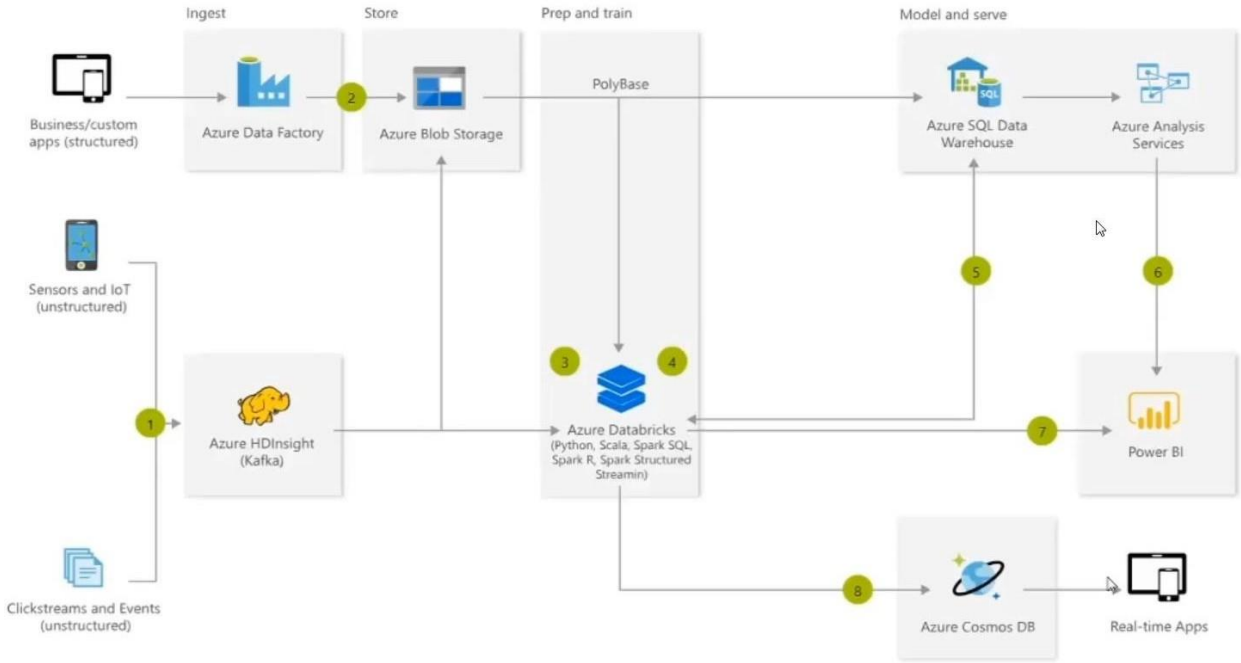


Scenario to use ADF

Ingest data using ADF to bootstrap your analytics workload

KEY SCENARIO	WHY ADF
Data migration for data lake & EDW 1. Big data workload migration from AWS S3, on-prem Hadoop File System, etc 2. EDW migration from Oracle Exadata, Netezza, Teradata, AWS Redshift, etc	• **Tuned for perf & scale:** PBs for data lake migration, tens of TB for EDW migration • **Cost effective:** serverless, PAYG • Support for **initial snapshot & incremental catch-up**
Data ingestion for cloud ETL 1. Load as-is from variety of data stores 2. Stage for data prep and rich transformation 3. Publish to DW for reporting or OLTP store for app consumption	• **Rich built-in connectors:** file stores, RDBMS, NoSQL. • **Hybrid connectivity:** on-prem, other public clouds, VNet/VPC • **Enterprise grade security:** AAD auth, AKV integration • **Developer productivity:** code-free authoring, CICD • **Single-pane-of-class monitoring** & Azure Monitor integration

Real Time Analytics



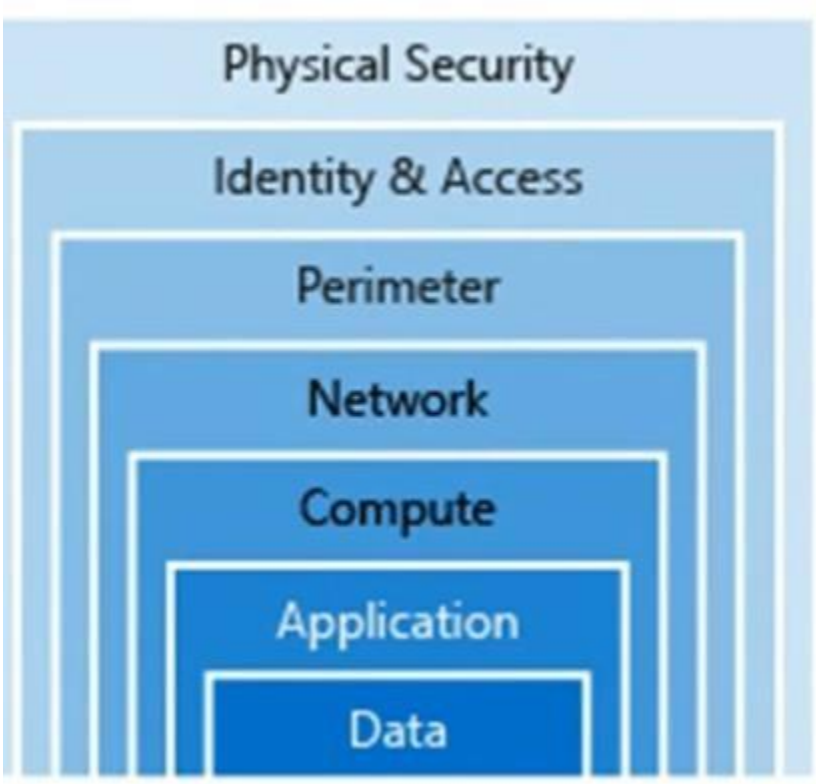
Complexities in Stream Processing

- Complex Data
    - Diverse data formats (json, avro, binary, ...)
    - Data can be dirty, latem out of order

- Complex Workloads
    - Combining Streaming with interactive queries
    - Machine learning

Design for Data Security and Compliance

Design for security



Identity Management

Identifying users that access your resources in an important part of security design.

- Identity as a security Layer
- Single sign-on
    - With SSO, users only need to remember one ID and one password. Access across database systems or applications is granted to a single identity tied to a user. -SSO with Azure Active Directory

- Azure AD is a cloud based identity service. It has built-in support for synchronizing with your existing on-premises AD or can be used stand-alone, This means that all your applications, whether on premise, in the cloud (including Office 365) or even mobile can share the credentials.

Infrastructure Protection

Role Based Access Control

Roles are defined as collections of access permissions. Security principals are mapped to roles directly or through group membership.

Role and Management groups:

- Roles are sets of permissions that users can be granted to. Management groups add the ability to group subscriptions together and apply policy at an even higher level.

Privileged Identity Management:

- Azure AD Privileged Identity Management (PIM) is an additional paid-for offering that provides oversight of role assignments, self-service, and just-in-time role activation.

Providing identities to services

An Azure service can be assigned an identity to ease the management of service access to other Azure resources.

Service Principals:

- It is an identity that is used by a service or application. Like other identities it can be assigned roles.

Managed identities:

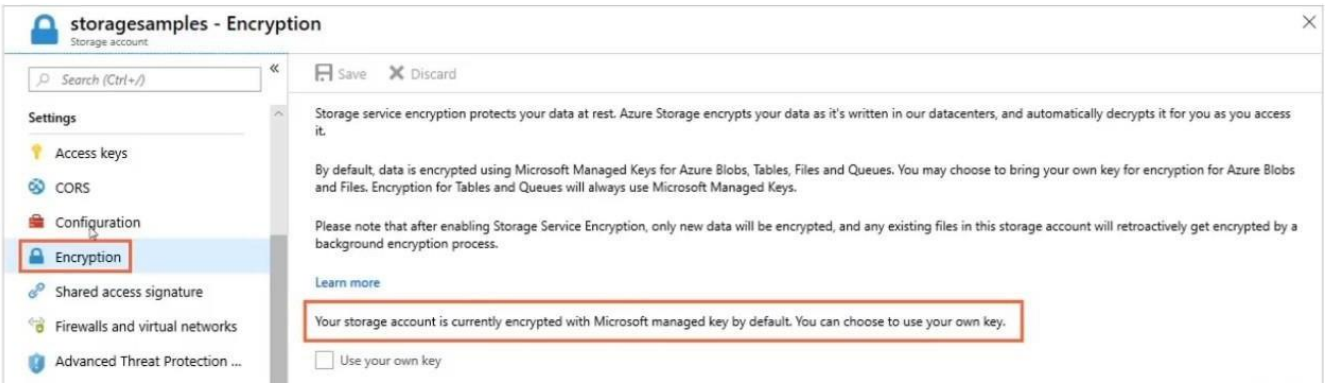
- When you create a managed identity for a service, you create an account on the Azure AD tenant. Azure infrastructure will automatically take care of authentication.

Securing Azure Storage

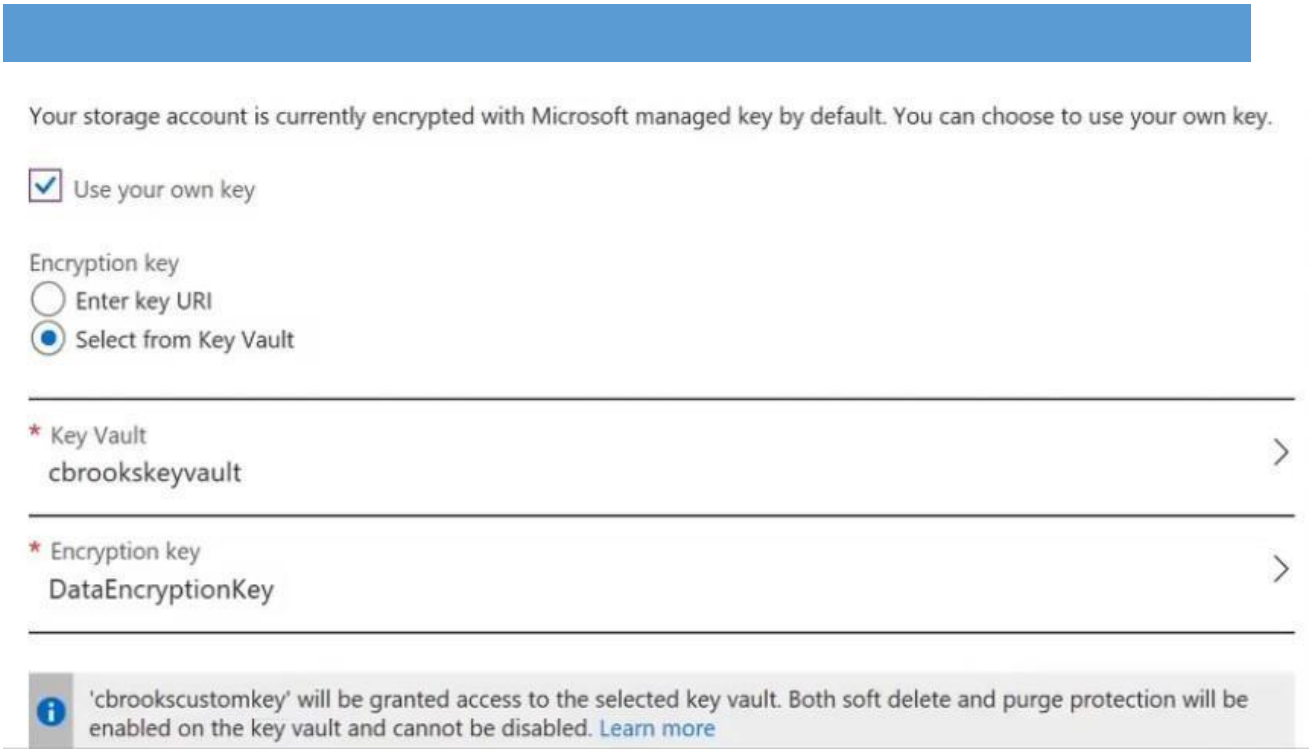
- Azure services such as Blob storage, Files share, Table storage, and Data Lake Store all build on Azure Storage.
- High-level security benefits for the data in the cloud:
    - Protect the data at rest
        - That is encrypt the data before persisting it into the storage and decrypt while retrieving. eg: Blob, Queue
    - Protect the data in transit
    - Support browser cross-domain access
    - Control who can access data
    - Audit storage class

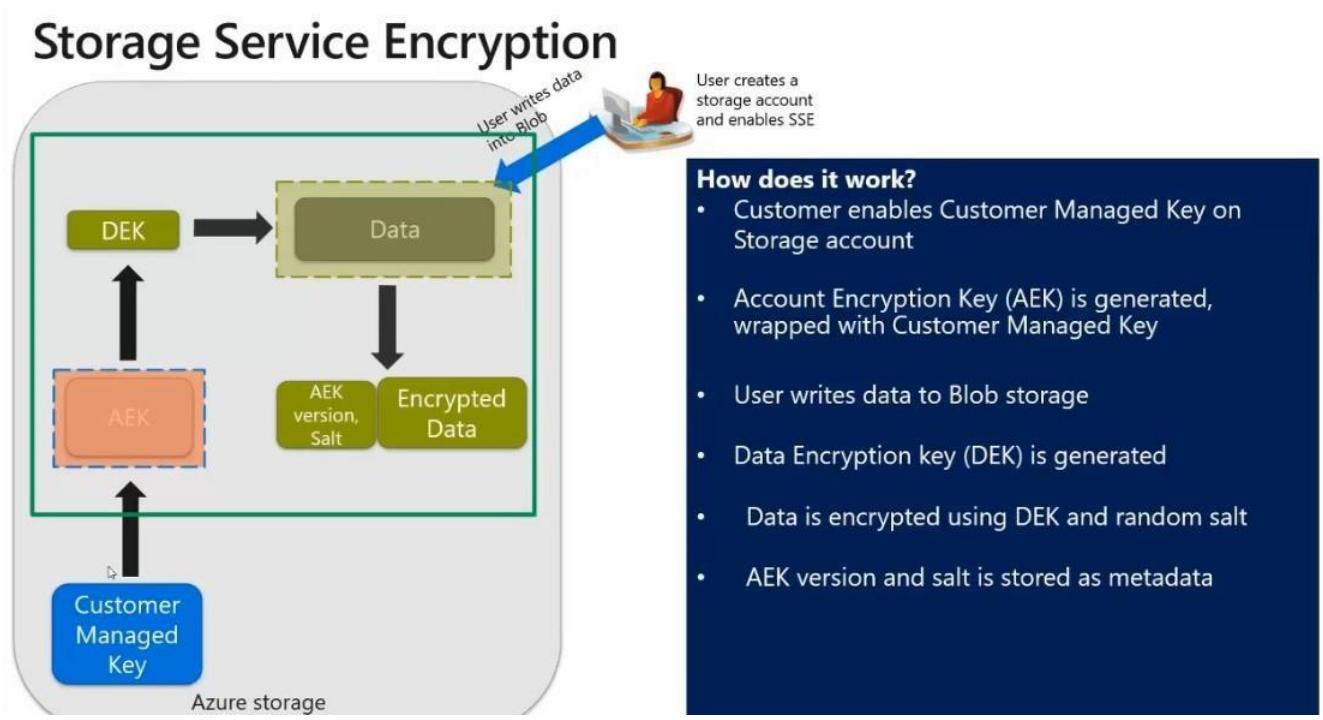
Encryption at rest - Azure Storage Service Encryption (SSE)

- All storage data encrypted at rest - protected from physical breach
    - By default, one master key per account, managed by Microsoft
    - Optionally, protect the master key with your own key in Azure Key Vault
    - Each write encrypted with a unique derived key
- All data writtern to storage is encrypted with SSE i.e, 256 bit advanced standard AES cipher. SSE automatically encrypts data on writting to Azure storage. This feature cannot be disabled



- For VM's Azure lets you encrypt virtual hard-disks by using Azure disk encryption. This encryption uses bit locker for windows images and uses DEM encrypt for Linux.
- Azure key Vault stores the keys automatically to help you control and manage disk encryption, keys and secret automatically.





Encryption at rest models

Client-side Encryption	Server-side Encryption	
With customer managed keys	With service managed keys	With customer managed keys
Customer manages keys in location of choice Azure Dedicated HSM	Microsoft manages keys	Customer manages keys via Azure Key Vault
MS services cannot see decrypted data	MS services see decrypted data	
REDUCED cloud functionality	Full cloud functionality	

Azure Key-Vault

- Safeguard cryptographic keys and other secrets used by cloud apps and services.

Encryption in transit

- Keep your data secure by enabling transport-level security between Azure and the client.
- Always use HTTPS to secure communication over the public internet.

- When you call the REST APIs to access objects in storage accounts, you can enforce the use of HTTPS by requiring secure transfer for the storage account.

Cross Origin Resource Sharing (CORS) support

- Azure Storage supports cross-domain access through cross-origin resource sharing (CORS)
- It is an optional flag that can be applied on storage accounts. The flag adds appropriate headers when you use http requests to retrieve resources from storage account.
- It uses HTTP headers so that a web application at one domain can access resources from a server at a different domain.
- By using CORS, web apps ensure that they load only authorizes content from authorized sources.

Identity-Based Access Control for Azure Blob Storage

Grant access to user and service identities from Azure Active Directory

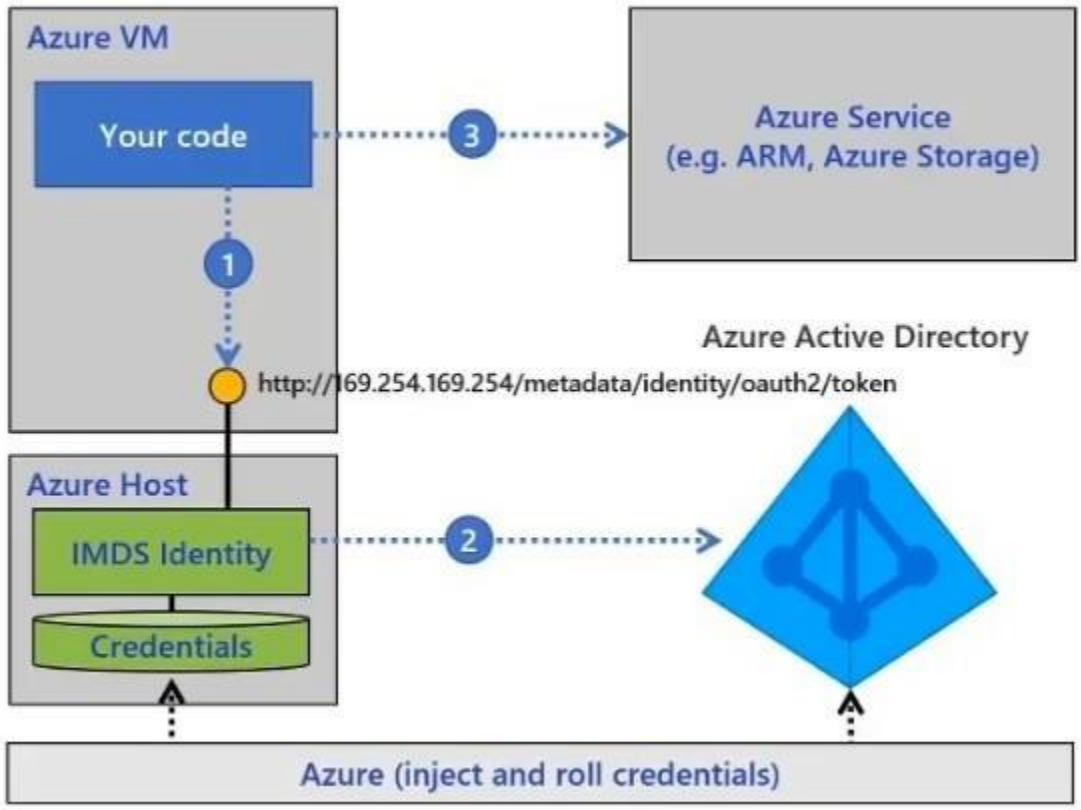
- Federate with enterprise identity systems
- Leverage powerful AAD capabilities including 2-factor and biometric authentication, conditional access, identity protection and more.

Control access with role-based access control (RBAC)

- Grant access to storage scopes ranging from entire enterprise down to one blob container
- Define custom roles that match your security model
- Leverage Privileged Identity Management to reduce standing administrative access.

AAD Authentication and RBAC currently support AAD, OAuth and RBAC on Storage Resource Provider via ARM.

Managed identities for Azure resources



- Auto-managed identity in Azure AD for Azure resource.
- Use the MSI endpoint to get access tokens from Azure AD (no secretes required).
- Direct authentication with services, or retrieve creds from Azure key vault
- No additional charge for MSI.

Managed Shared Access Policies and Signatures

Storage Explorer provides the ability to manage access policies for containers.

- A shared access signature (SAS) provides you with a way to grant limited access to other clients, without exposing your account key.
- Provides delegated access to resources in your storage account.

Types of Shared Access Signature (SAS)

- Service level
    - Service level SAS are defined on a resource under a particular service.
    - Used to allow access to specific resources in a storage account.

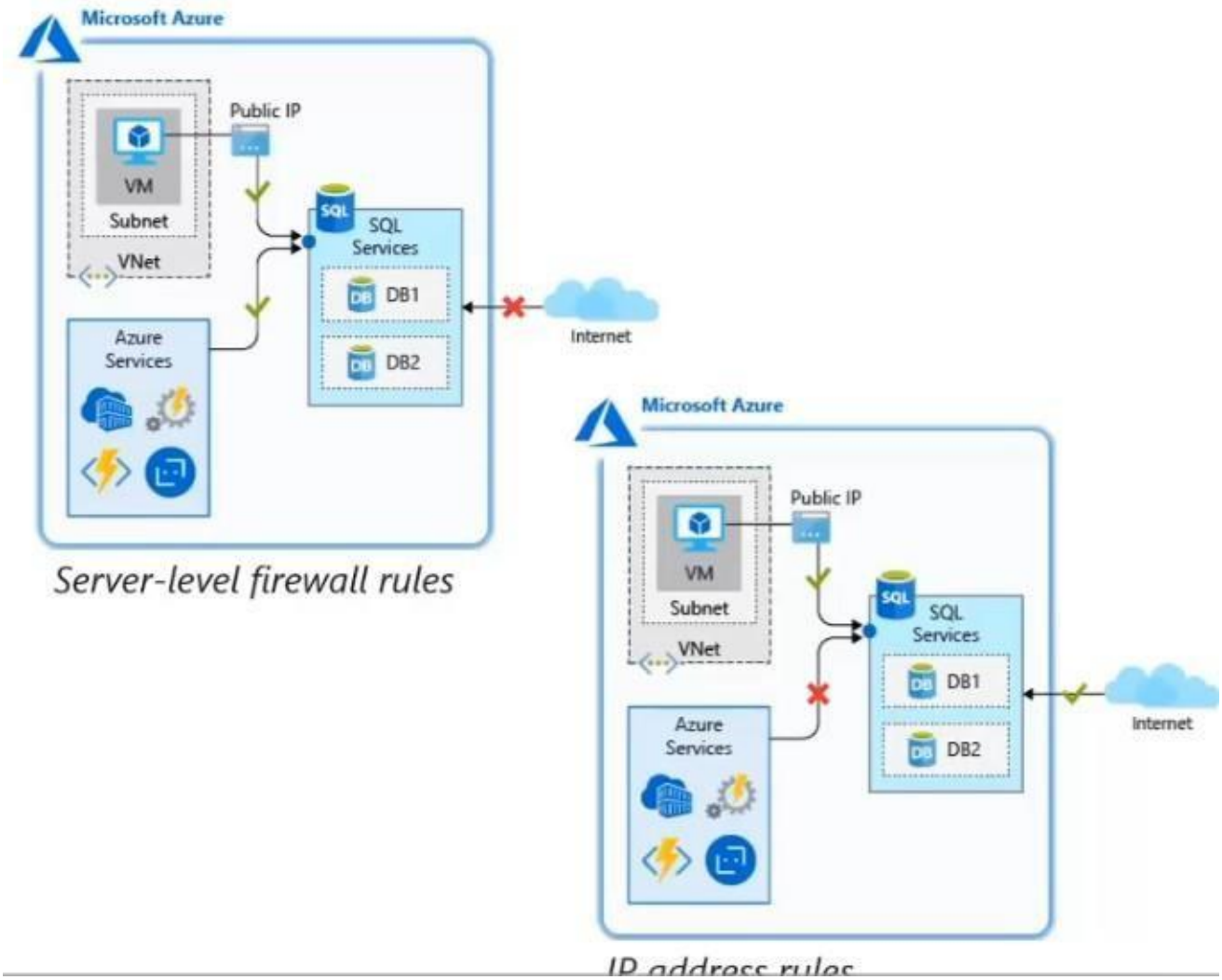
    - For example, to allow an app tp retrieve a list of files in a file system or to download a file.

- Account level
    - Targets the storage account and can apply to multiple services and resources
    - For example, you can use an account-level SAS to allow the ability to create file systems.

Immutability Policies

- Support for time-based retention
    - container level configuration
    - RBAC support and policy auditing
    - BLobs cannot be modified or deleted for N days
- Support for legal holds with tags
    - Container level configuration
    - Blobs cannot be modified or deleted when legal hold is set.
- Support for all Blob tiers
    - Applies to hot, cool and cold data
    - Policies retained when data is tiered
- SEC 17a-4(f) complaint

Firewall rules



- Azure SQL DB has a built-in firewall that is used to allow and deny network access to both the db server itself, as well as individual db.
- Server-level firewall rules
    - Allow access to Azure services
    - IP address rules
    - Virtual network rules
- Database-level firewall rules
    - IP address rules

Network Security

Network security is protecting the communication of resources within and outside of your network. The goal is to limit exposure at the network layer across your services and systems

Internet protection:

- By assessing the resources that are internet-facing, and only allow inbound and outbound communication when necessary. Ensure that they are restricted to only ports/protocols required.

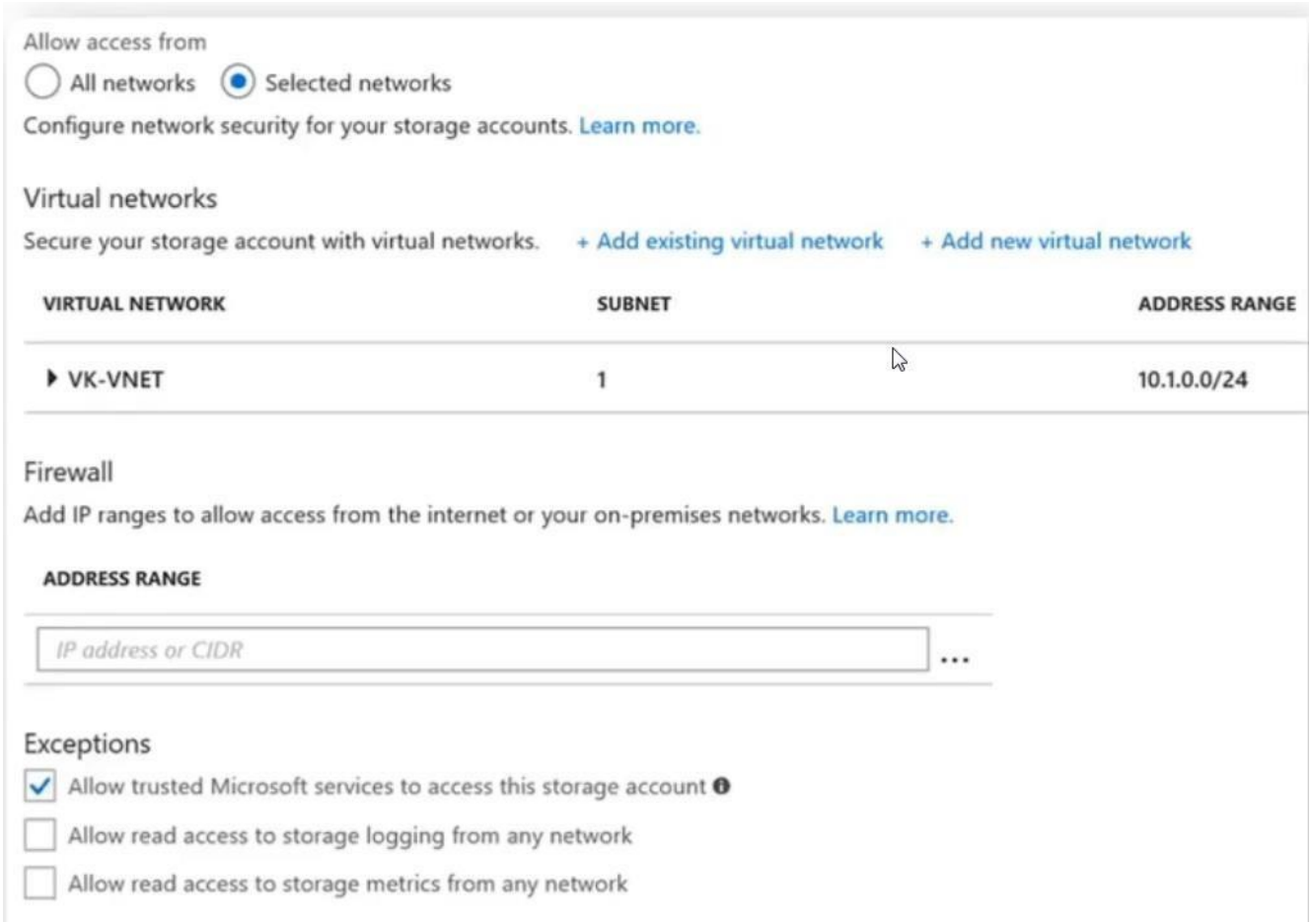
VIrtual network security:

- To isolate Azure services to only allow communication from virtual networks, use VNet service endpoints. With service endpoints, Azure service resources can be secured to your virtual network.

Network Integration:

- VPN connections are a common way of establishing secure communication channels between networks, and this is no different when working with virtual networking on Azure. Connection between Azure VNets and an on-premises VPN device is a great way to provide secure communication.

Firewalls and VNET access



- Storage Firewall
    - Block internet access to data
    - Grant access to clients in specific vnet
    - Grant access to clients from on-premise networks via public peering network gateway

Private endpoints

- Azure private endpoint is a fundamental building block for private link in Azure. It enables service like Azure VM to communicate privately with private link resources.
- It is a network interwork interface that connects you privately and securely to service powered by Azure Private link.
- A private endpoint assigns a private IP address from your Azure Virtual Network (VNET) to the storage account.

- private endpoint enables communication from the same VNet, regionally peered VNets, globally peered VNets, and on-premises using VPN or Express Route, and services powered by private link.
- It secures all traffic between your VNet and the storage account over a private link.

Advanced threat protection for Azure Storage

- An additional layer of security intelligence that detects unusual and potentially harmful attempts to access or exploit storage accounts
- These security alerts are integrated with Azure Security Center.

Secure Azure Cosmos DB Data

- Using Firewall settings
- Add inbound and Outbound networks

Azure SQL database - Secure your data in transit, at rest and on display

- TLS network encryption
    - Azure SQL DB enforces Transport Layer Security (TLS) encryption at all times for all connections, which ensures all data is encrypted "in transit" between the database and the client.
- Transparent Data Encryption (TDE)
    - Protects your data at rest using TDE.
    - TDE performs real-time encryption and decryption of the DB, associated backups, and transaction log files at rest without requiring changes to the application.
- Dynamic data masking
    - By using the, we can limit the data that is displayed to the user.
    - Policy-based security feature that hides the sensitive data in the result set of a query over designated DB fields, while the data in the DB is not changed e.g: phone numbers, credit card numbers.

Enable Database Auditing

- For SQL Server you can create audits that contain specifications for server-level events and specifications for database-level events.
- Audited events can be written to the event logs or to audit files

- There are several levels of auditing for SQL Server, depending on government or standards requirements for your installation.
- Azure SQL DB and Azure Synapse Analytics auditing tracks database events and writes them to an audit log in your Azure storage account.
- Enable Threat detection to know any malicious activities on SQL DB or potential security threats.

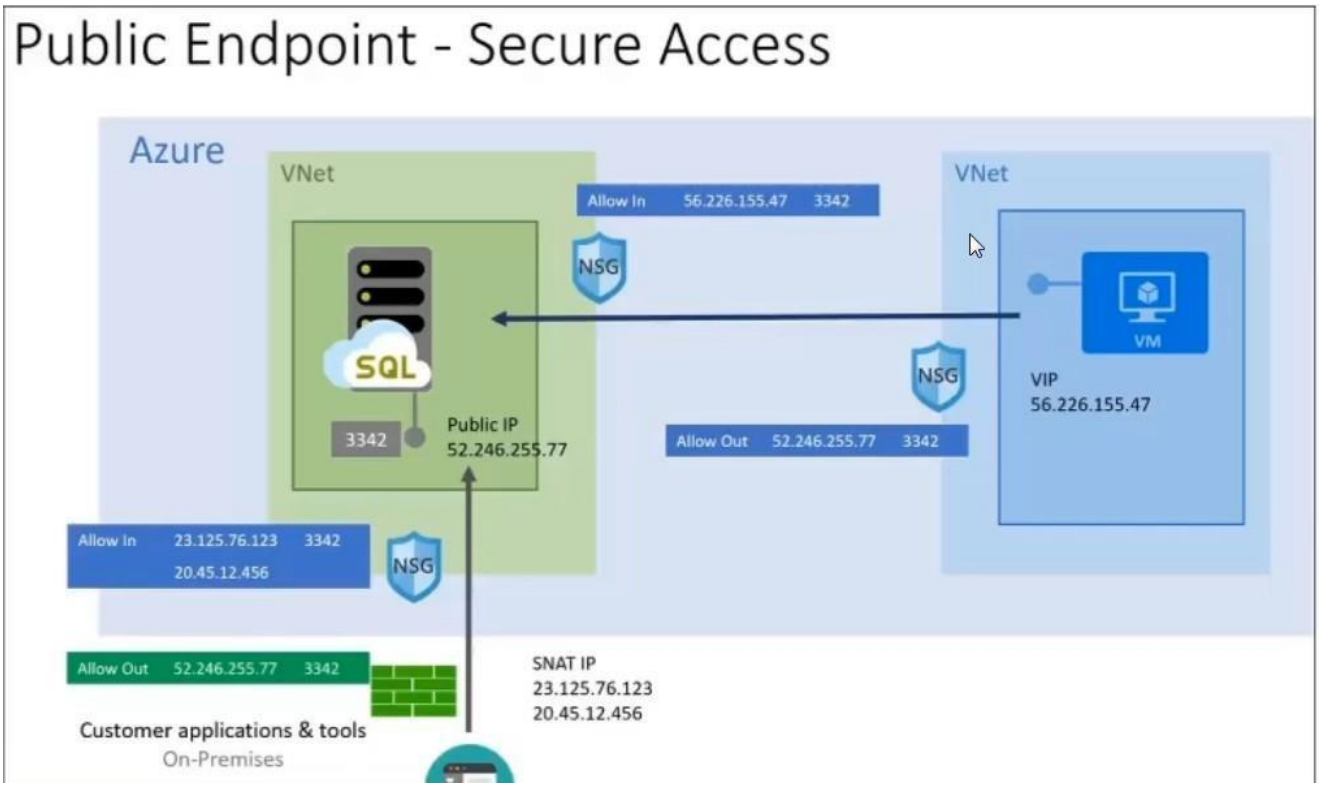
Use an Azure SQL Database managed instance securely with public endpoints

- A SQL DB managed instance provides a private endpoint to allow connectivity from inside its VNET.

Scenarios where you need to provide public endpoint connection

- The managed instance must integrate with multi-tenant-only PaaS offerings. -You need higher throughput of data exchange than is possible when you're using VPN.
- Company policies prohibit PaaS inside corporate networks.

Managed Instance - Lock down inbound and outbound connectivity



- A managed instance has a dedicated public endpoint address.
- In the client-side outbound firewall and in the NSG rules, set this public endpoint IP address to limit outbound connectivity.
- Use a NSG to limit access to the managed instance public endpoint on port 3342.

Azure SQL Database and Azure Synapse Analytics data discovery & classification

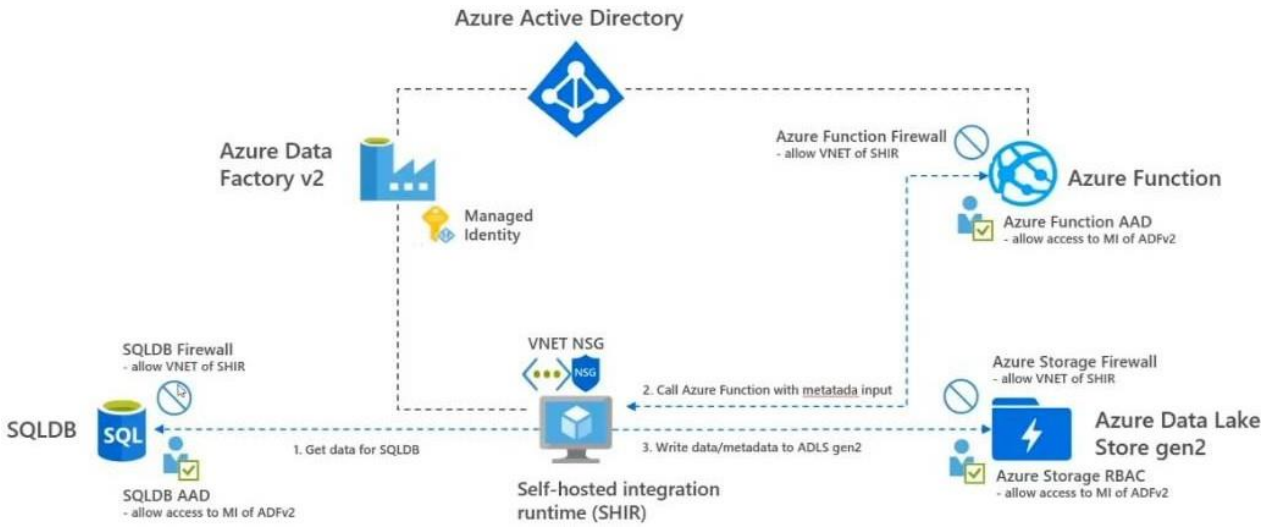
- Discovery & recommendations
    - The classification engine scans your DB and identifies columns containing potentially sensitive data. It then provides you an easy way to review and apply the appropriate classification recommendations via the Azure portal.
- Labelling
    - Sensitivity classification labels can be persistently tagged on columns using new classification metadata attributes introduced into the SQL Engine. This metadata can then be utilized for advanced sensitivity-based auditing and protection scenarios.
- Query result set sensitivity
    - The sensitivity of query result set is calculated in real time for auditing purposes.
- Visibility
    - The DB classification state can be viewed in a detailed dashboard in the portal.

Discover, classify & label sensitive columns

The classification includes two metadata attributes:

- Labels
    - The main classification attributes used to define the sensitivity level of the data stored in the column.
- Information Types
    - Provide additional granularity into the type of data stored in the column.

Architecture of Azure Data Factory



Security aspects that are part of the above architecture

- AAD access control
  - SQLDB, ADLS Gen2 and Azure function only allow the Managed Identity (MI) of ADFv2 to access the data. This means that no keys need to be stored in ADFv2 or Key vaults.
  - To secure ADLS Gen2 account:
    - Add RBAC rule that only MI of ADFv2 can access ADLS Gen2
    - Add firewall rule that only VNET of Self Hosted Integrated Runtime (SHIR) can access ADLS Gen2 container.
- Firewall rules
  - SQLDB, ADLS Gen2 and Azure function all have firewall rules in which only the VNET of the SHIR is allowed as inbound network.
  - To secure SQLDB:
    - Add Database rule that only MI of ADFv2 can access SQLDB
    - Add firewall rule that only VNET of SHIR can access SQLDB

Miscellaneous

Serverless Computing

Containers

- A container is a method running applications in a virtualized environment. The virtualization is done at the OS level, making it possible to run multiple identical application instances within the same OS.

Azure Kubernetes Service (AKS)

- Azure Kubernetes Service allows you to set up virtual machines to act as your nodes. Azure hosts the Kubernetes management plane and only bills for the running worker nodes that host your containers.

Azure Container Instance (ACI)

- It is a serverless approach that lets you create and execute containers on demand. You're charged only for the execution time per second.

Performance Bottlenecks

Azure Monitor

- A single management point for infrastructure-level logs and monitoring for most of your Azure services.

Log Analytics

- You can query and aggregate data across logs. This cross-source correlation can help you identity issues or performance problems that may not be evident when looking at logs or metrics individually.

Application performance management

- Telemetry can include individual page request times, exceptions within your application, and even custom metrics to track business logic. This telemetry can provide a wealth of insight into apps.

Tips to remember, A day prior to the Exam.

Azure Service	Type of File
Azure CosmosDB	Graph Databases
Azure Hbase and HDInsight	Column Family in-memory key-value store

Azure Service	Usage
Azure Synapse	Data analytics
Azure Search	Search engine databases
Azure Timeseries Insights	Time series databases
Azure Blob	Object store
Azure FileStorage	Shared files

Azure Cosmos DB Usage:

- For Real-Time Customer Experiences
- Telemetry stores for IOT
- Migrate NoSQL apps

Azure Cosmos DB Authentication

- It uses two types of keys to authenticate users and provide access to its data and resources.

Key Type	Resources
Master Keys	Used for administrative resources: database accounts, databases, users, and permissions

Key Type	Resources
Resource tokens	Used for application resources: containers, documents, attachments, stored procedures, triggers, and UDFs

Azure Cosmos DB SLA for Read/Write operation

Operation type	Single region	Multi-region (single region writes)	Multi-region (multi-region writes)
Writes	99.99	99.99	99.999
Reads	99.99	99.999	99.999

Azure Storage Service

Storage account type	Supported services	Supported performance tiers	Supported access tiers	Replication options	Deployment model[1]
General-purpose V2	Blob, File, Queue, Table, and Disk	Standard, Premium[5]	Hot, Cool, Archive[3]	LRS, ZRS[4], GRS, RA-GRS	Resource Manager
General-purpose V1	Blob, File, Queue, Table, and Disk	Standard, Premium[5]	N/A	LRS, GRS, RA-GRS	Resource Manager, Classic

Azure Storage Availability

Scenario	LRS	ZRS	GRS	RA-GRS
Node unavailability within a data center	Yes	Yes	Yes	Yes
An entire data center (zonal or non-zonal) becomes unavailable	No	Yes	Yes	Yes
A region-wide outage	No	No	Yes	Yes

Azure Data Factory

- Does not store any data except for linked service credentials for cloud data stores, which are encrypted by using certificates.

Azure Databricks (2 types of clusters)

- Interactive clusters are used to analyze data collaboratively with interactive notebooks.
- Job clusters are used to run fast and robust automated workloads using the UI or API.

Azure SQL Database - Security Overview

LAYER	TYPE	DESCRIPTION
Network	IP Firewall Rules	Grant access to databases based on the originating IP address of each request.
Network	Virtual Network Firewall Rules	Only accept communications that are sent from selected subnets inside a virtual network.
Access Management	SQL Authentication	Authentication of a user when using a username and password.
Access Management	Azure AD Authentication	Leverage centrally managed identities in Azure Active Directory (Azure AD).
Authorization	Row-level Security	Control access to rows in a table based on the characteristics of the user/query.
Threat Protection	Auditing	Tracks database activities by recording events to an audit log in an Azure storage account.

LAYER	TYPE	DESCRIPTION
Threat Protection	Advanced Threat Protection	Analyzing SQL Server logs to detect unusual and potentially harmful behavior.
Information Protection	Transport Layer Security (TLS)	Encryption-in-transit between client and server.
Information Protection	Transparent Data Encryption (TDE)	Encryption-at-rest using AES (Azure SQL DB encrypted by default).
Information Protection	Always Encrypted	Encryption-in-use (Column-level granularity; Decrypted only for processing by client).
Information Protection	Dynamic Data Masking	Limits sensitive data exposure by masking it to non-privileged users.
Security Management	Vulnerability Assessment	Discover, track, and help remediate potential database vulnerabilities.
Security Management	Data Discovery & Classification	Discovering, classifying, labeling, and protecting the sensitive data in your databases.
Security Management	Compliance	Been certified against a number of compliance standards.

Azure SQL - Network Access Controls

CONTROL	DESCRIPTION
Allow Azure Services	When set to ON, other resources within the Azure boundary can access the SQL resource.
IP firewall rules	Use this feature to explicitly allow connections from a specific IP address.
Virtual Network firewall rules	Use this feature to allow traffic from a specific Virtual Network within the Azure boundary.

Encryption on Azure

Type	Technique or service used	Enables encryption of
Raw Encryption	-	Azure Storage, VM Disks, Disk Encryption
Database Encryption	Transparent Data Encryption	Databases and SQL DW
Encypting Secrets	Azure Key Vault	Storing application secrets

HDInsight cluster types to run Apache Hive queries

Cluster Type	Usage
Interactive Query	To optimize for ad hoc, interactive queries
Apache Hadoop	To optimize for hive queries used as batch process
Spark & HBase	Run hive queries

SQLDW or Synapse

To achieve fastest loading speed for moving data into a DW table

- load data into a staging table. Define the staging table as a heap and use round-robin for the distribution option.

Criteria to select a Distribution column

- Has many Unique values
- Does not have Nulls, or has only a few Nulls
- Is not a date column
- Use a Column from Group BY, not from where clause

Distribution Type

- Round Robin for small Fact tables
- Hash distributed for large Fact tables

Data corruption checks

- We create a user-defined restore point before data is uploaded. Delete the restore point after data corruption checks complete.

Simple yet very powerful Hacks:

(You can use these hacks when uncertain about any question or scenario to make quick decisions)

- IOT HUb, Event Hub, Blob are the three ways to bring data into Stream Analytics
- Anything related to RBAC Identities majority cases answer would be Service Srincipal
- In MySQL sharding is the best way to partition the data.
    - Criteria to select a column for sharding
        - Unique (data should be well distributed)
- Cosmos DB Partition keys should generally be based on unique values.
- For a Database using a nonclustered columnstore index will improve performance on analytics and not clustered columnstore index.

- You can use Azure Event Hubs, IoT hub, and Azure Blob storage for **streaming data input**.
- Azure Stream Analytics Supports Azure SQL DB or Azure Blob storage for **reference data input**.
- Primary key and secondary key grant access to remotely administer the **Storage account**.
- Event Hubs Capture creates files in Avro format.
- If notebooks are involved with scheduling or autoscale of clusters it is databricks.
- RBAC support for databricks via Premium clusters.
- If there is a question based on IOT Hub or Event Hub probability of the answer being Stream Analytics for processing is maximum.
- If you see the term "Relationship" or nodes and vertices in CosmosDB question by default the option in Gremlin API.
- If hierarchical or Big Data related storage involved then ADLS Gen2.
- If flat file related storage then blob.