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

Structured	Semi- Structured	Unstructured
Structured data is data that adheres to a schema, so all of the data has the same fields or properties. Structured data can be stored in a database table with rows and columns.	Semi-structured data doesn't fit neatly into tables, rows, and columns. Instead, semi-structured data uses _tags_ or _keys_ that organize and provide a hierarchy for the data.	Unstructured data encompasses data that has no designated structure to it. Known as No-SQL., there are four types of No-SQL databases: • Key Value Store • Document Database • Graph Databases • Column Base

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.

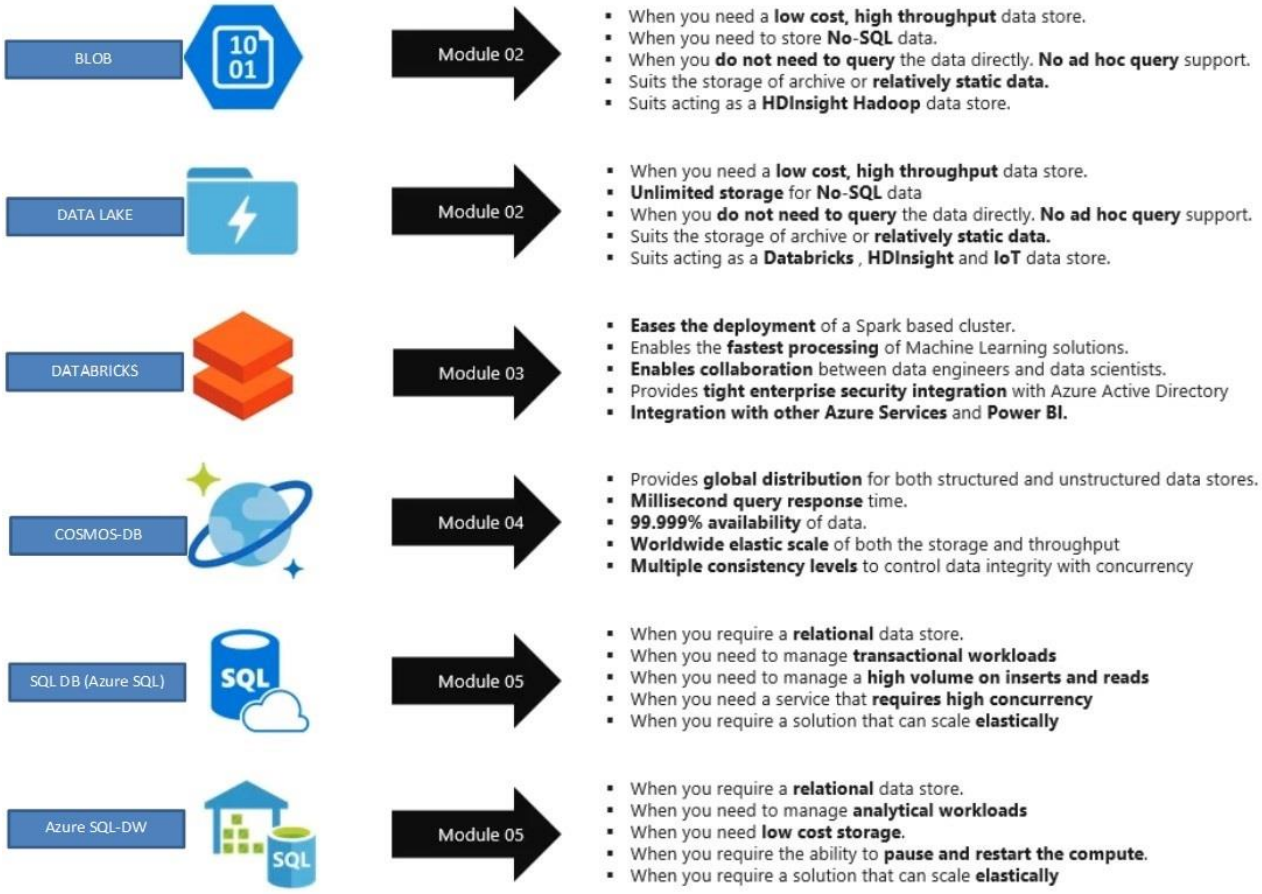
Note:

- 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.

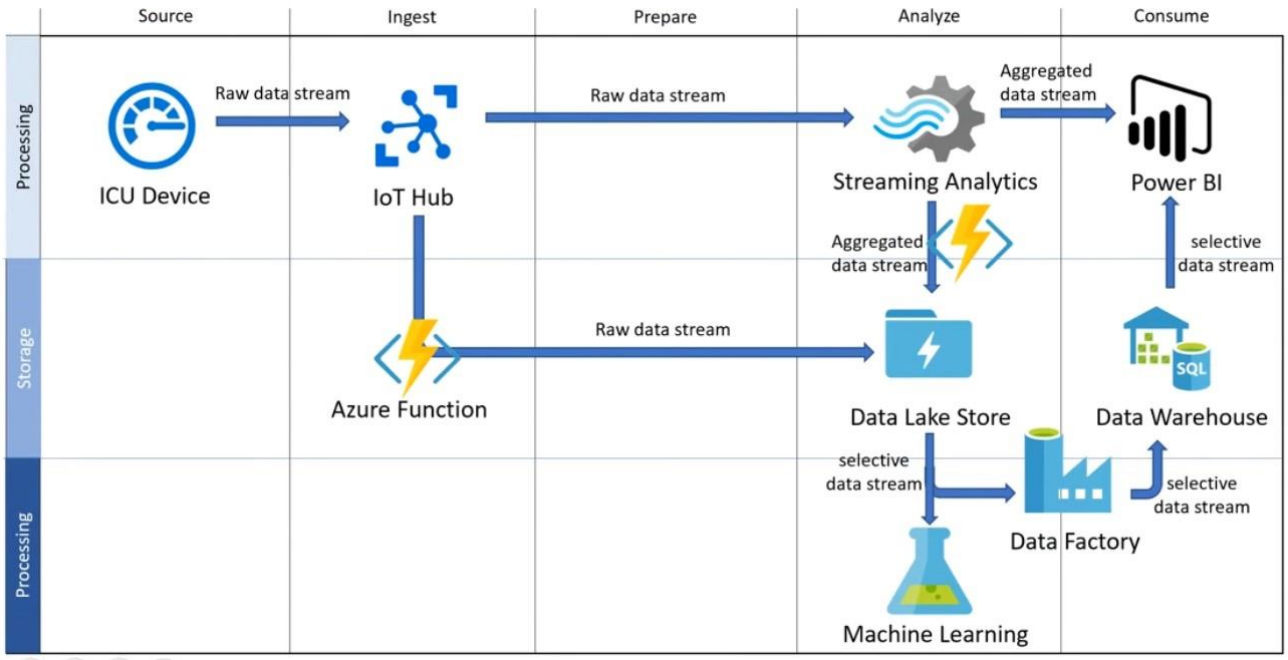
Azure Blob Usage

- 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

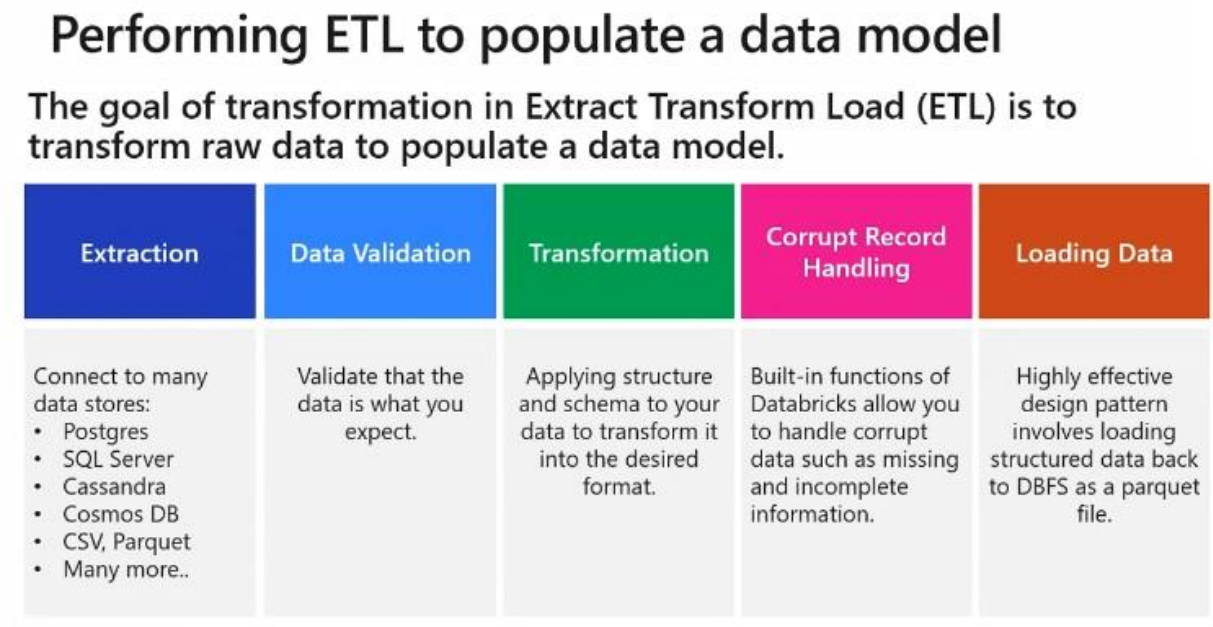
Azure Active Directory

- 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

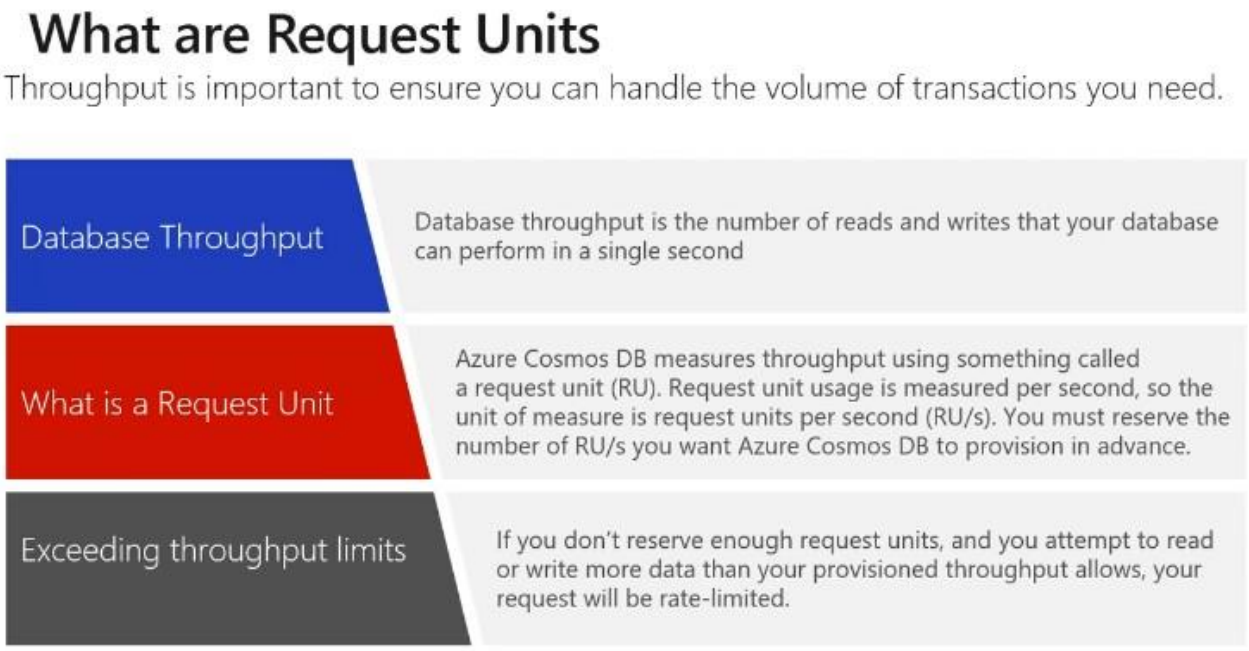
- Document databases
- Key value stores
- Column family stores
- Graph databases

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

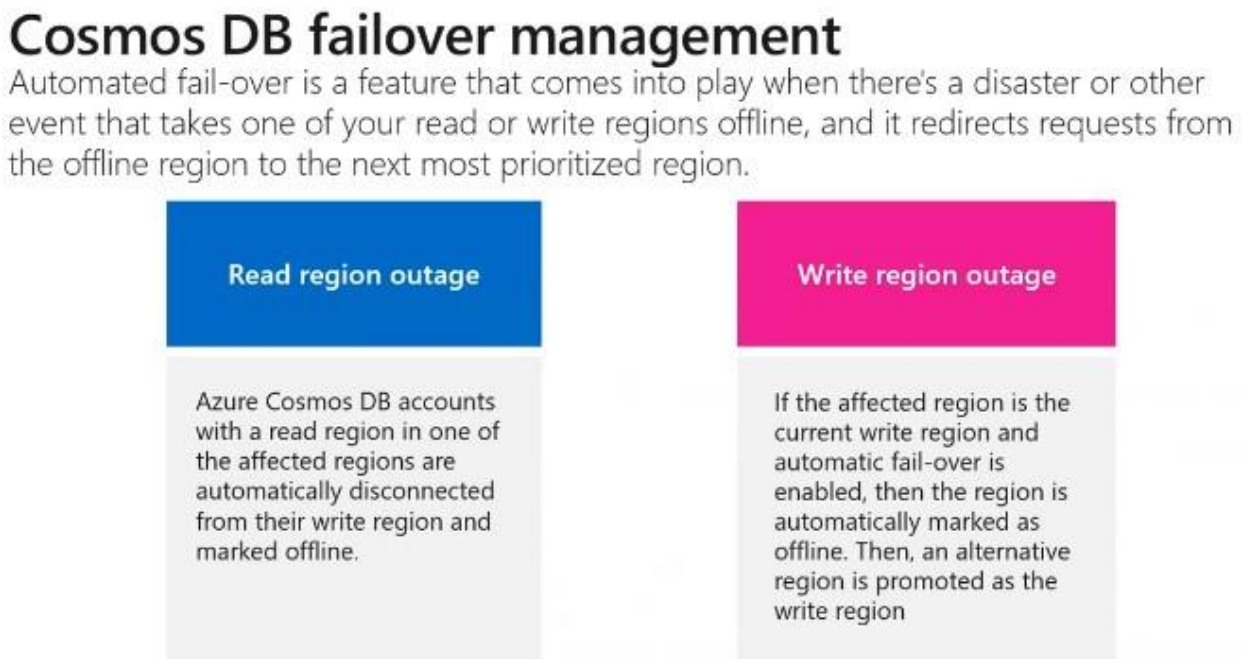
```
az account list –output table    // Lists the set of Azure subscriptions that we
have

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

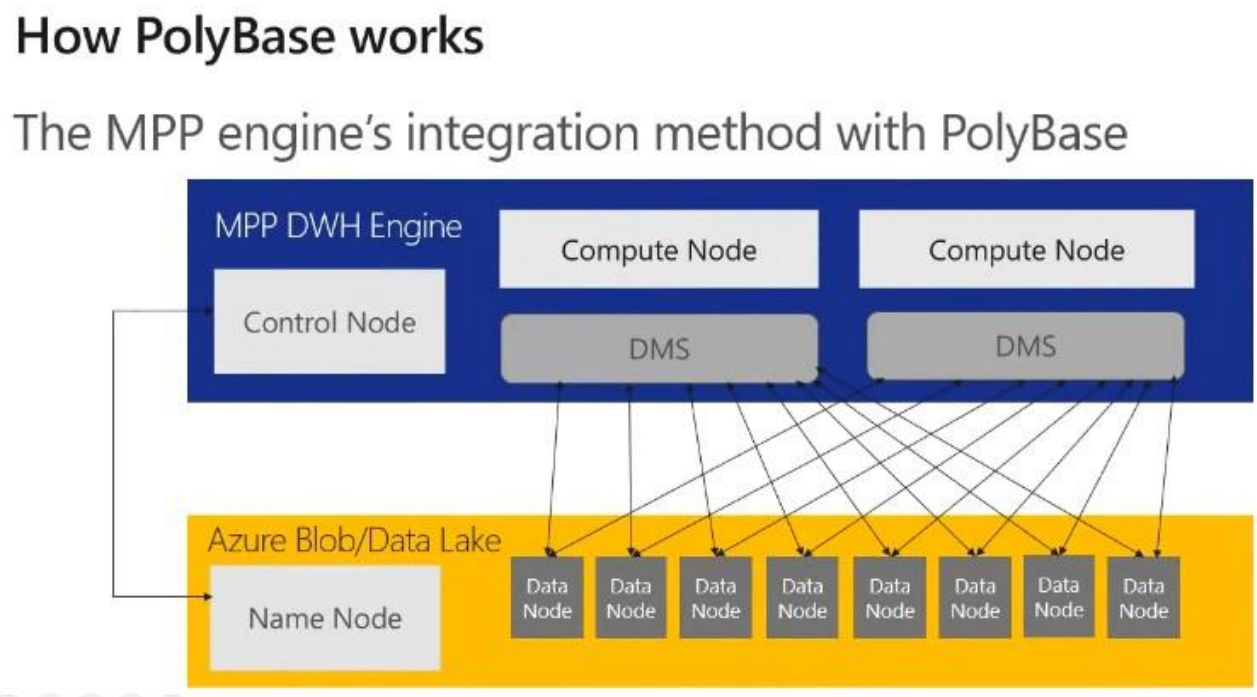
- Introduced Cache and tempDB to pull data from remote datasets
- Max DWU is 30Kc
- 120 connections and 128 queries
- MPP

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)

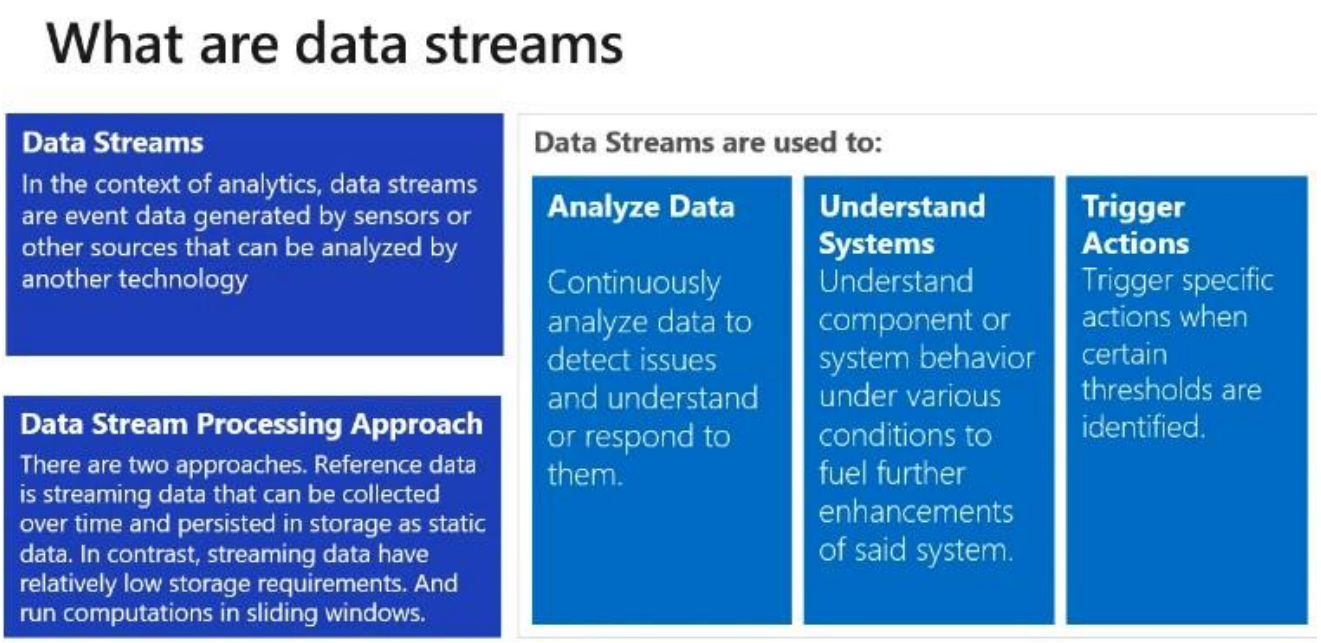
Import data from Blob Store to SQL DW

```
--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



Event Processing

The process of consuming data streams, analyzing them, and deriving actionable insights out of them is called Event Processing and has three distinct components:

Event producer	Examples include sensors or processes that generate data continuously such as a heart rate monitor or a highway toll lane sensor.
Event processor	An engine to consume event data streams and deriving insights from them. Depending on the problem space, event processors either process one incoming event at a time (such as a heart rate monitor) or process multiple events at a time (such as a highway toll lane sensor).
Event consumer	An application which consumes the data and takes specific action based on the insights. Examples of event consumers include alert generation, dashboards, or even sending data to another event processing engine.

Processing events with Azure Stream Analytics

Microsoft Azure Stream Analytics is an event processing engine. It enables the consumption and analysis of high volumes of streaming data in real time.

Source	Ingestion	Analytical Engine	Destination
• Sensors • Systems • Applications	• Event Hubs • IoT Hubs • Azure Blob Store	• Stream Analytics Query Language • .NET SDK	• Azure Data Lake • Cosmos DB • SQL Database • Blob Store • Power BI

ORCHESTRATING DATA MOVEMENT WITH ADF AND SECURING AZURE DATA PLATFORMS

Azure Event Hubs:

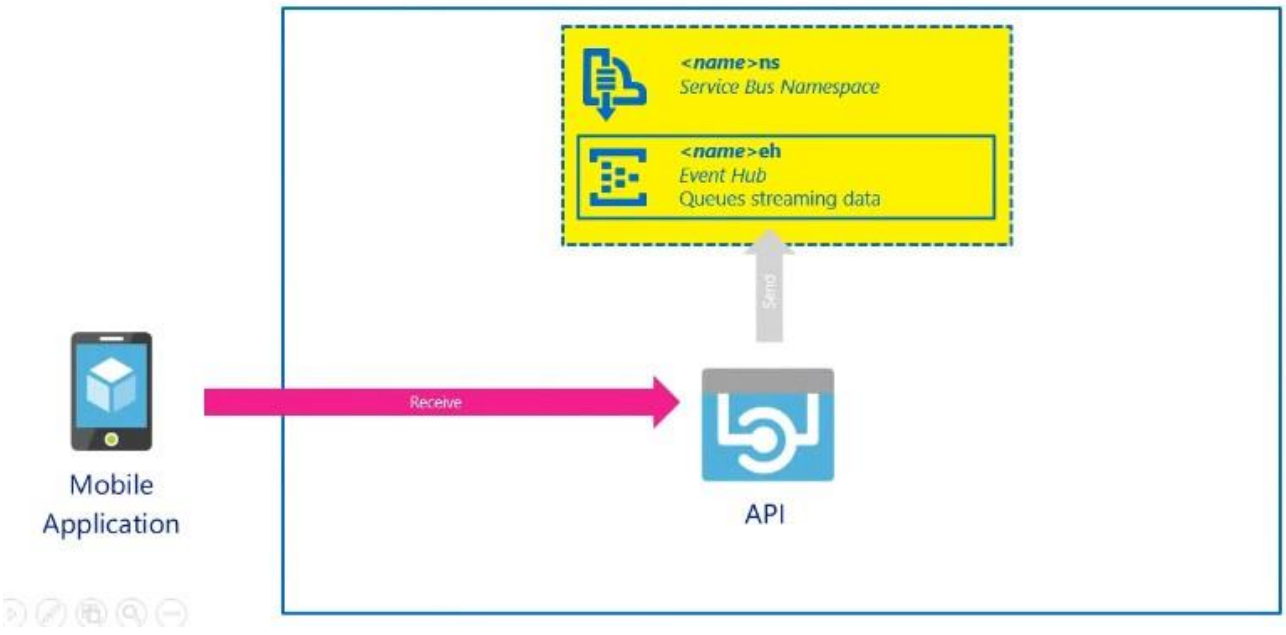
- Is a highly scalable publish-subscribe service that can invest millions of events per second and stream them into multiple applications

- 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.

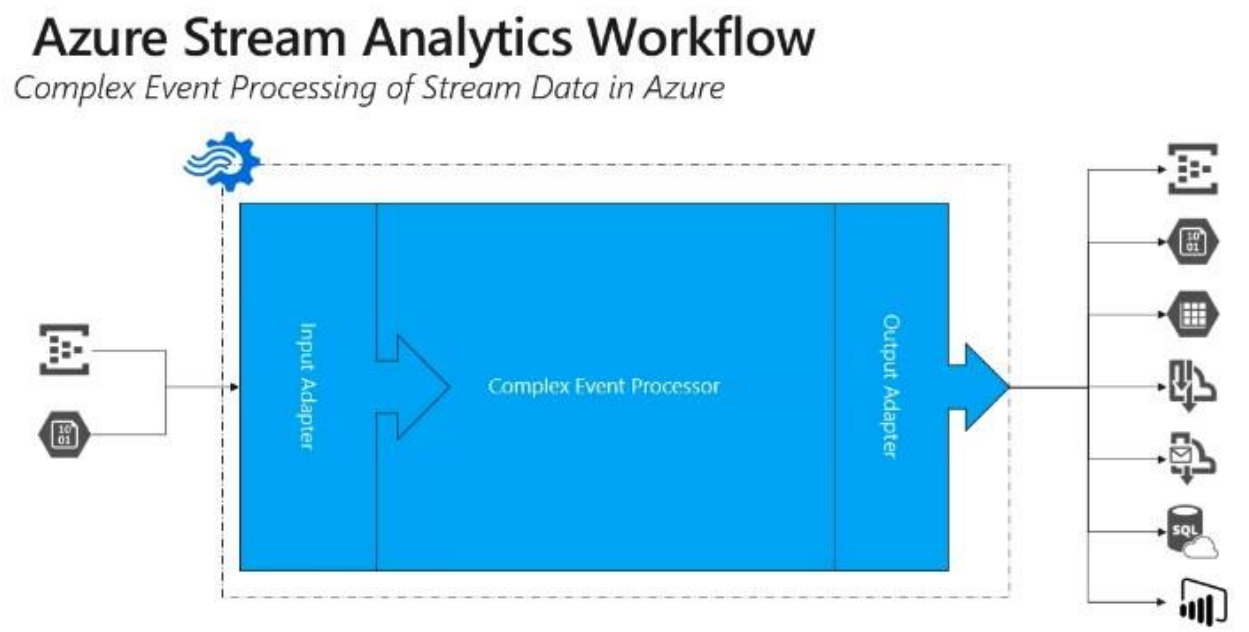
Create Event Hub

- 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

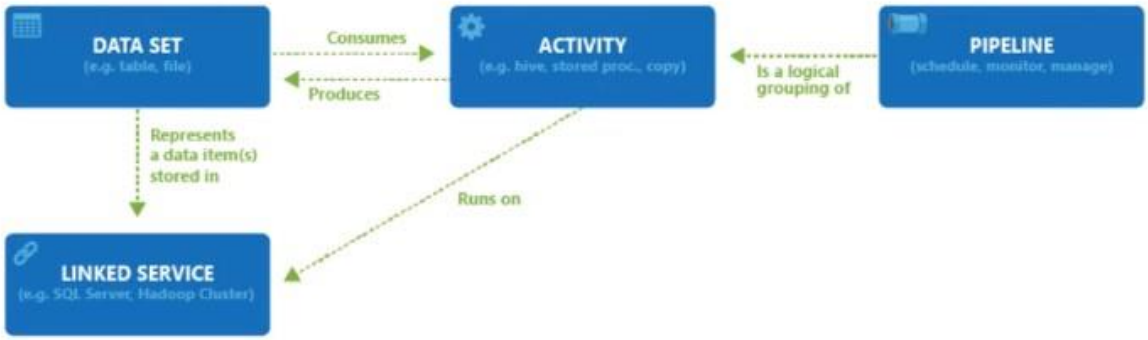
- Creates, orchestrates, and automates the movement, transformation and/or analysis of data through in the cloud.

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

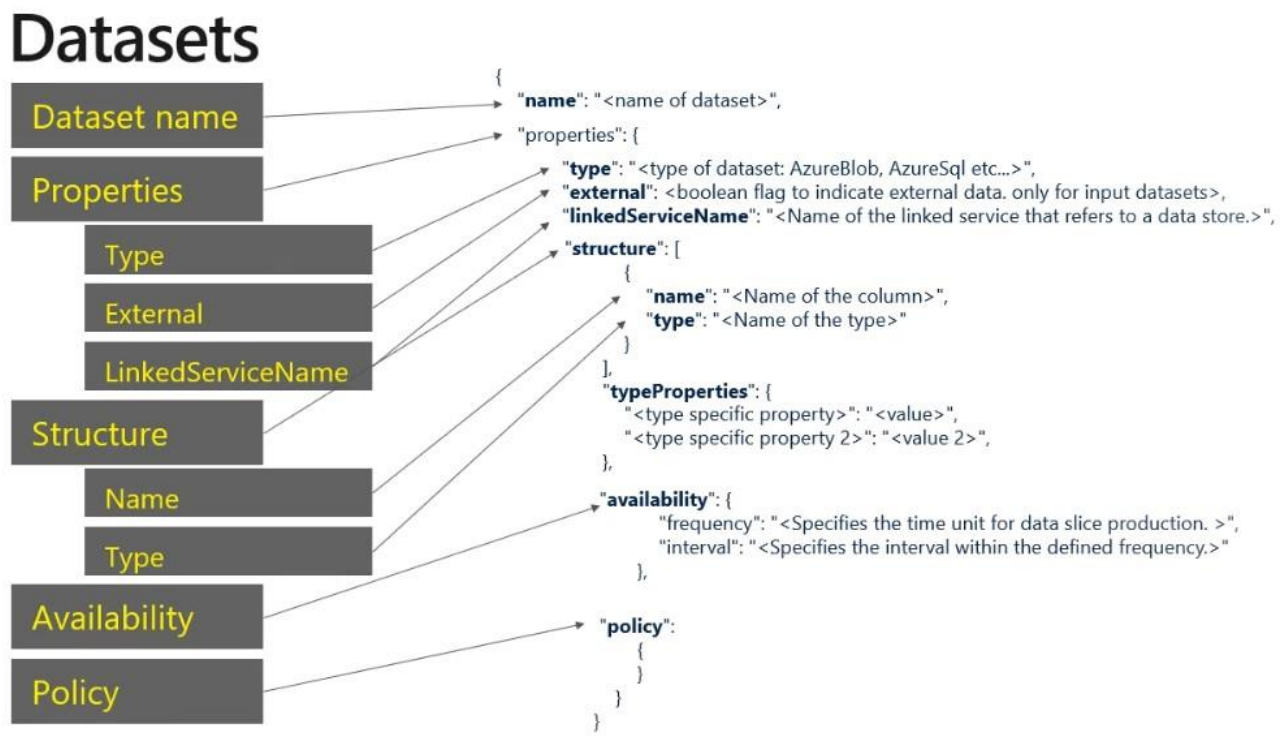
AZURE SQL DATABASE EXAMPLE

```
{
  "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"
    }
  }
}
```

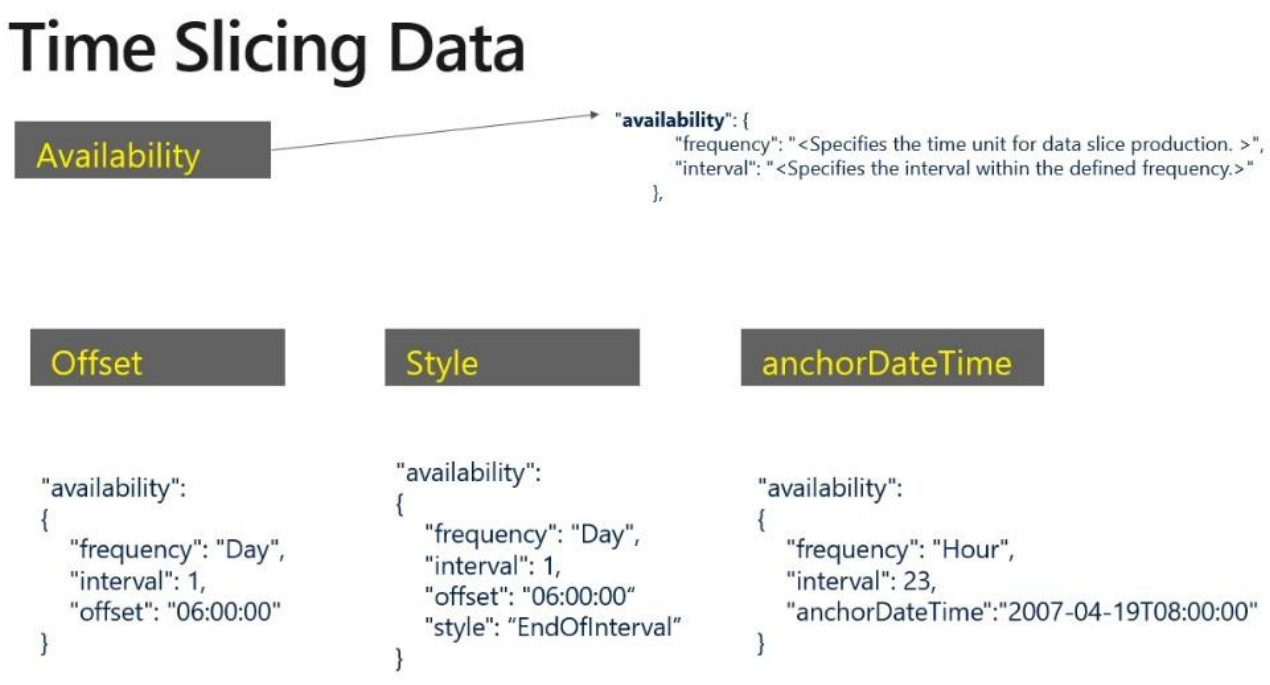
AZURE BLOB STORE EXAMPLE

```
{
  "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.