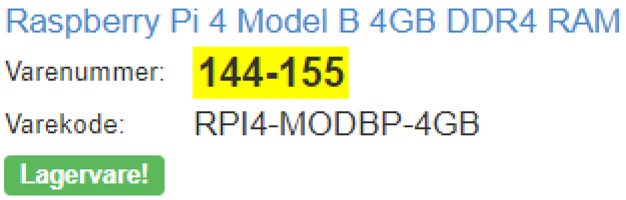
Azure Storage

May 2022

Camilla Gaardsted, SuperUsers A/S

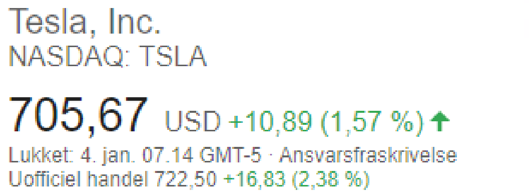
Data



Raspberry Pi 4 Model B 4GB DDR4 RAM

Varenummer: 144-155

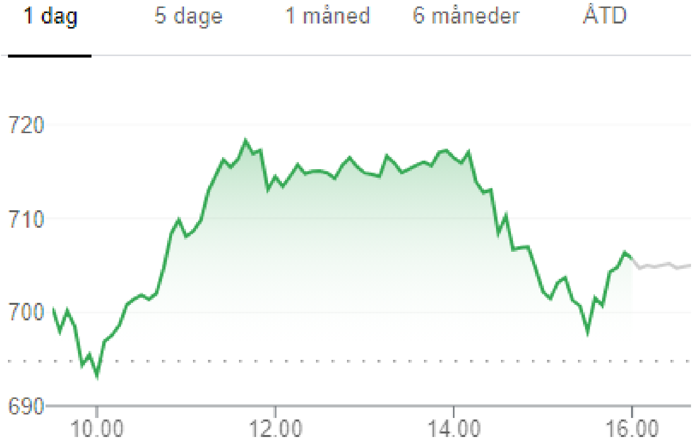
Varekode: RPI4-MODBP-4GB

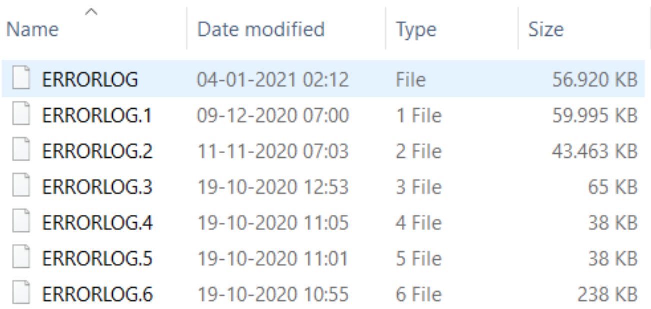


Tesla, Inc.
NASDAQ: TSLA

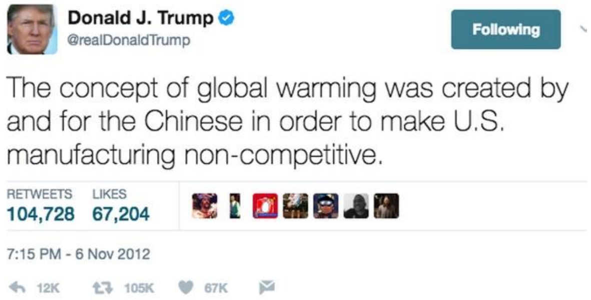
705,67 USD +10,89 (1,57 %) ↑

Lukket: 4. jan. 07.14 GMT-5 · Ansvarsfraskrivelse
Uofficiel handel 722,50 +16,83 (2,38 %)





Name	Date modified	Type	Size
ERRORLOG	04-01-2021 02:12	File	56.920 KB
ERRORLOG.1	09-12-2020 07:00	1 File	59.995 KB
ERRORLOG.2	11-11-2020 07:03	2 File	43.463 KB
ERRORLOG.3	19-10-2020 12:53	3 File	65 KB
ERRORLOG.4	19-10-2020 11:05	4 File	38 KB
ERRORLOG.5	19-10-2020 11:01	5 File	38 KB
ERRORLOG.6	19-10-2020 10:55	6 File	238 KB



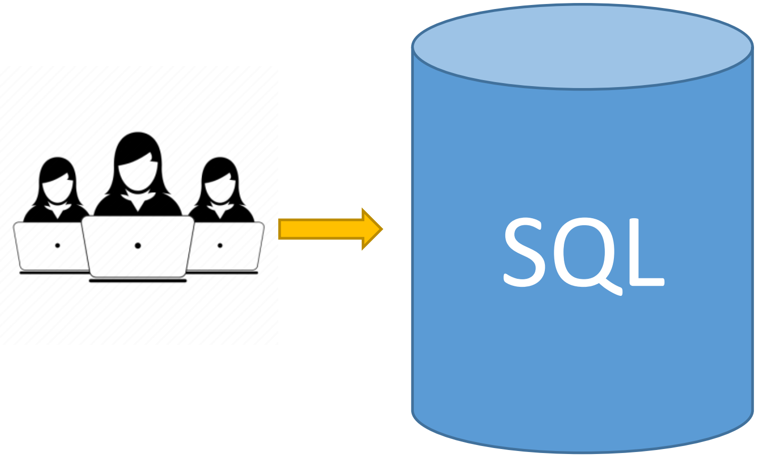
Donald J. Trump
@realDonaldTrump

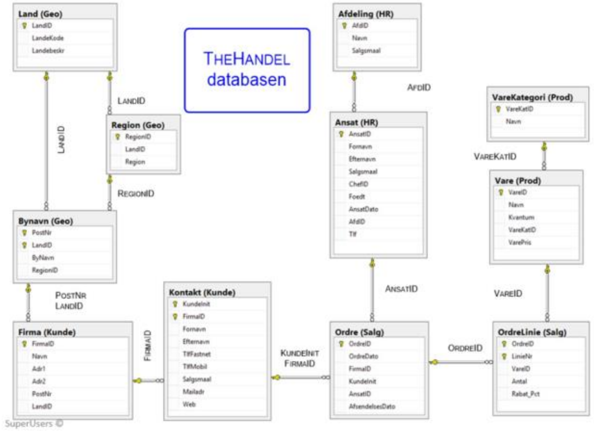
The concept of global warming was created by and for the Chinese in order to make U.S. manufacturing non-competitive.

RETWEETS 104,728 LIKES 67,204

7:15 PM - 6 Nov 2012

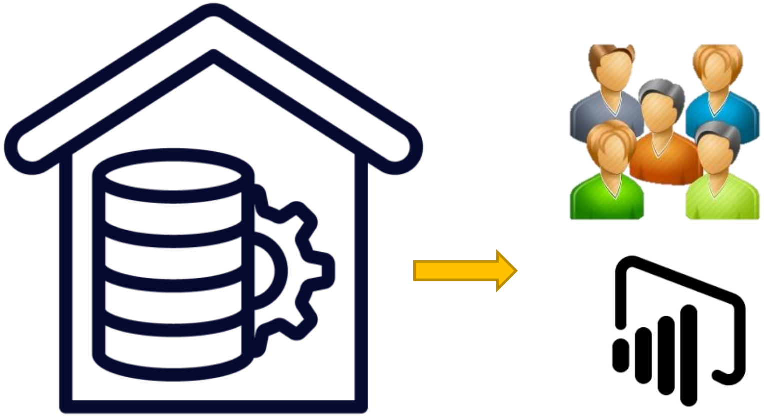
Relationelle databaser - OLTP

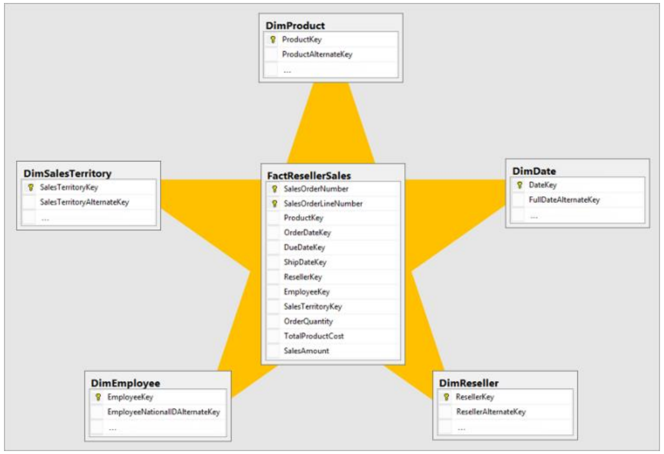




- Data er normaliseret
- Håndterer enkeltrækker
- Mange tabeller
- Samtidige brugere danner data
- Mange transaktioner
- Optimeret til ændringer:
  - INSERT/UPDATE/DELETE
- ANSI SQL
- Forretningsprocessen (business)

Enterprise Datawarehouse - OLAP





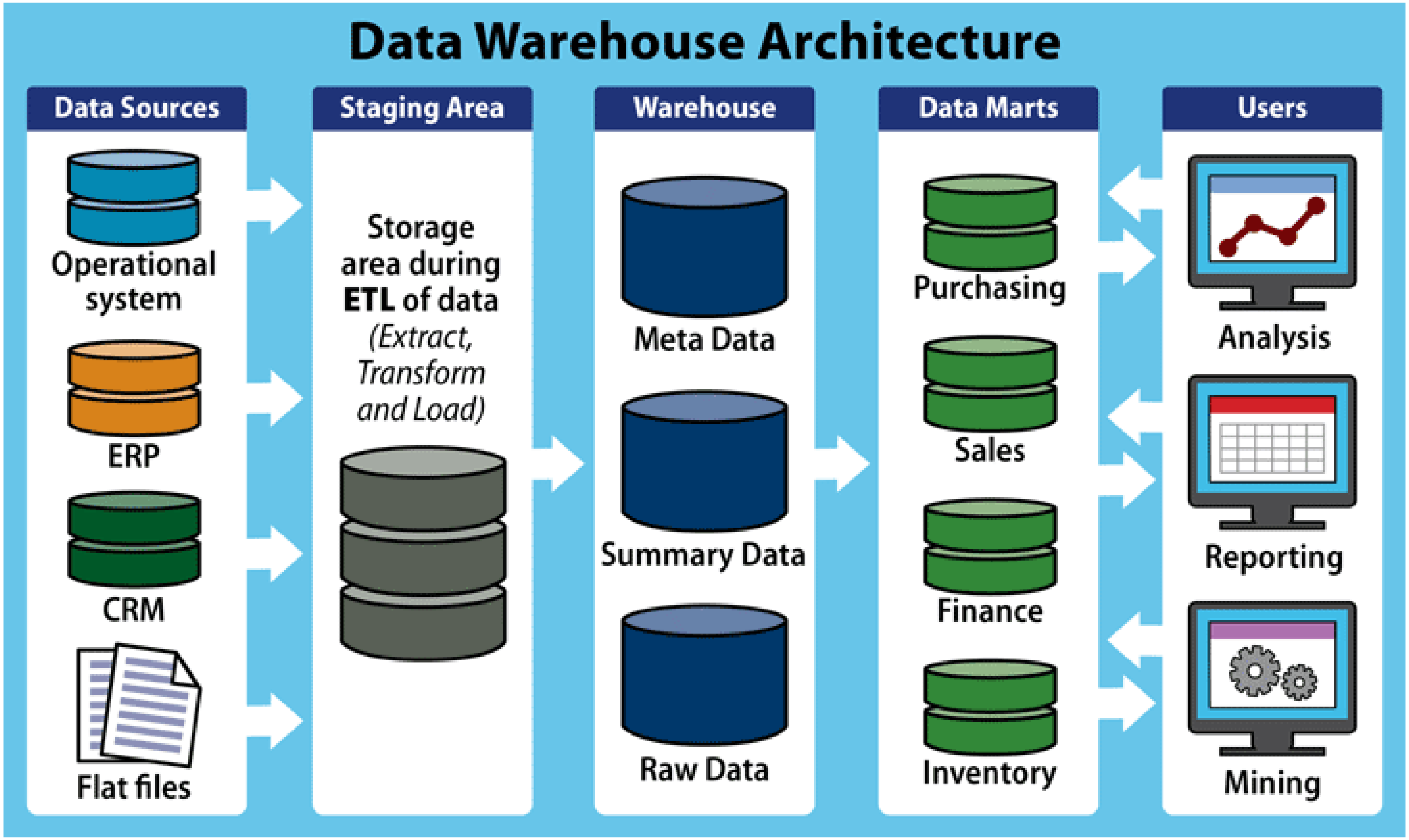
- Data er denormaliseret
- Håndterer rækker aggregeret
- Stjerneskema (dimensional model)
- Data ankommer periodisk
- Historisk data
- Optimeret til læsning:
    - SELECT
- ANSI SQL
- BI – Indsigt i forretningen

Azure Synapse Analytics

amazon REDSHIFT

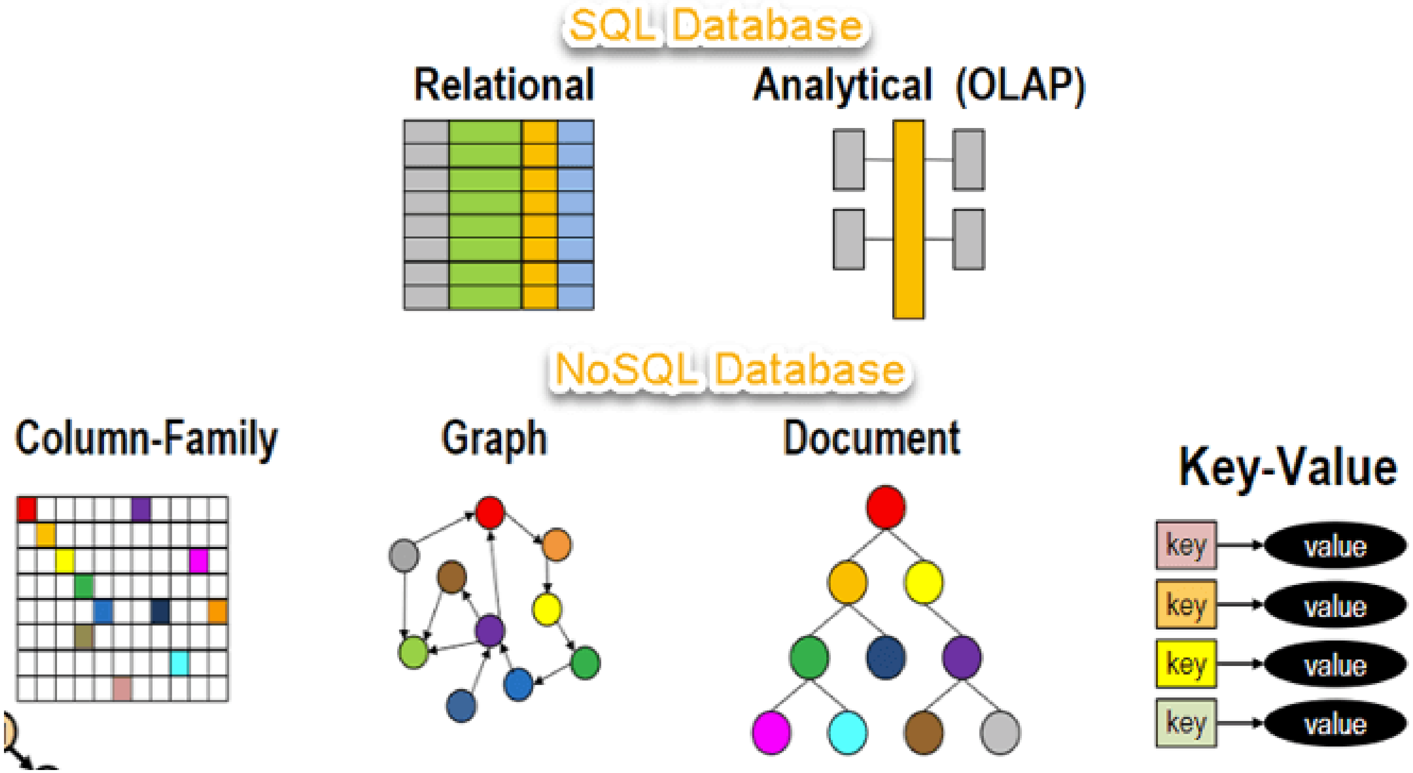
snowflake

Datawarehouse - Architecture



Kilde:https://www.datamation.com/big-data/top-15-data-warehouse-tools/

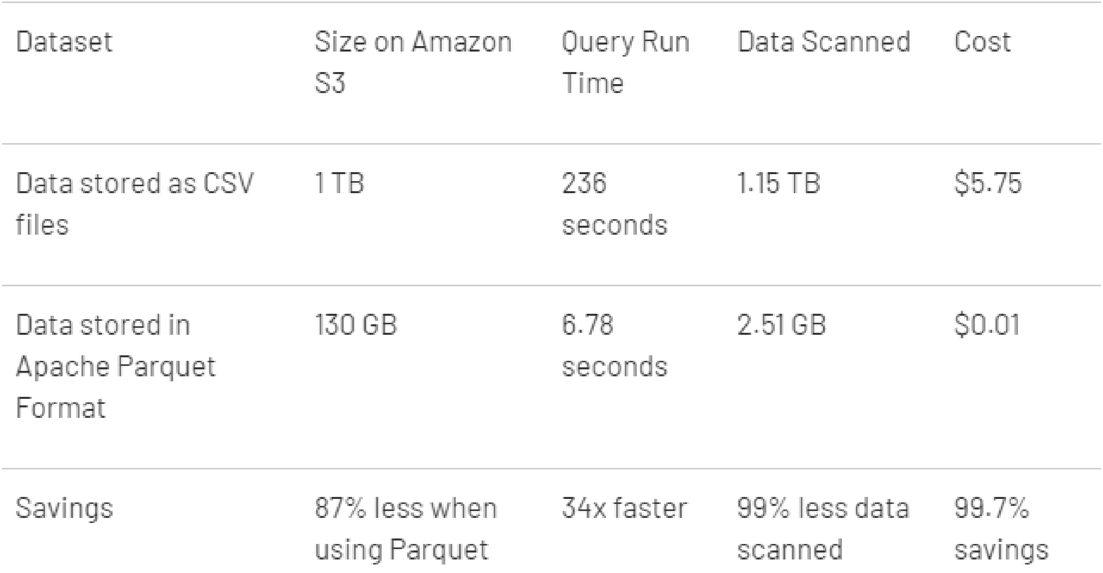
NoSQL Nye måder at gemme data på



Billede fra https://www.guru99.com/

Data filformater

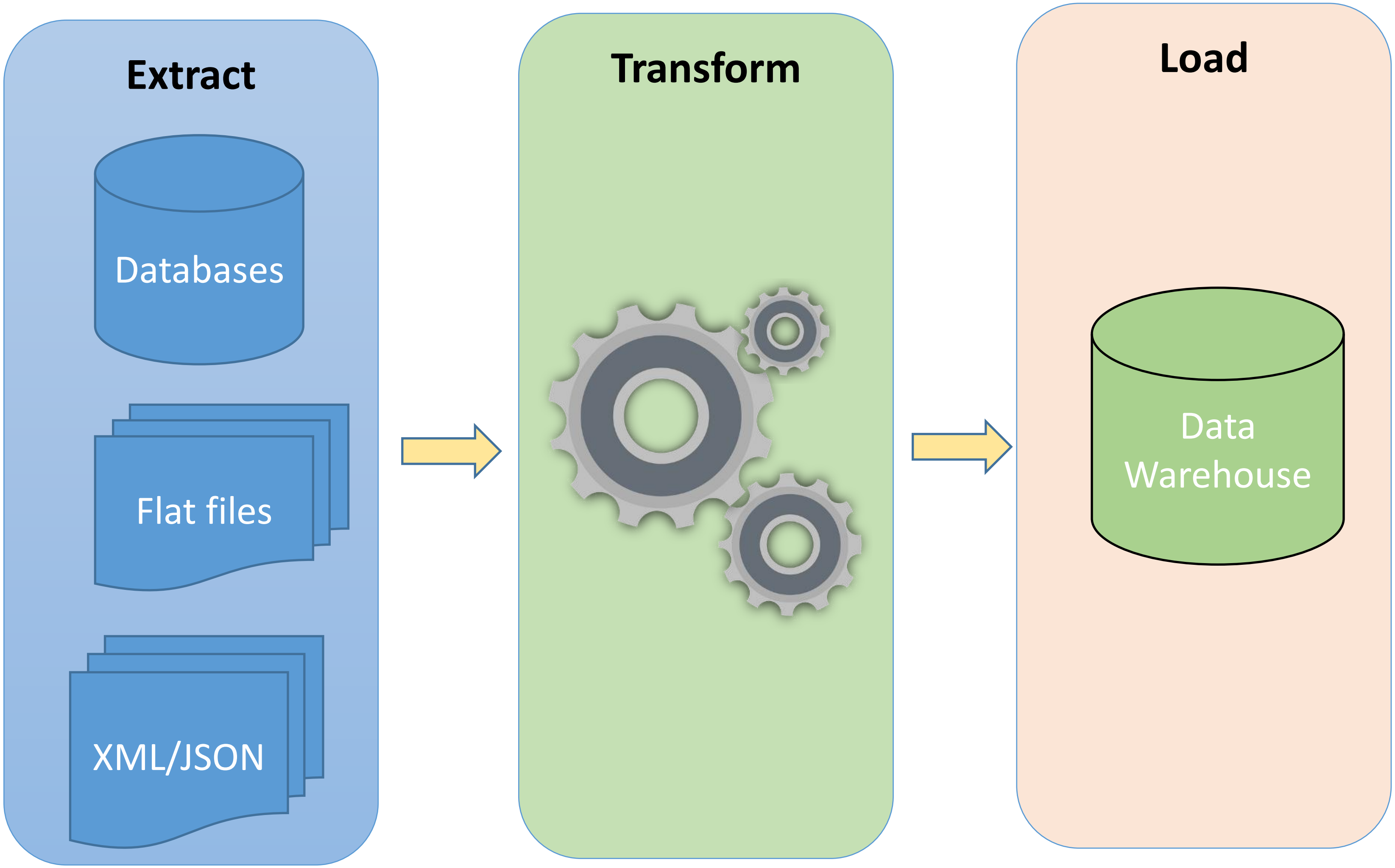
- CSV
- APACHE PARQUET
- AVRO
- JSON



Dataset	Size on Amazon S3	Query Run Time	Data Scanned	Cost
Data stored as CSV files	1 TB	236 seconds	1.15 TB	$5.75
Data stored in Apache Parquet Format	130 GB	6.78 seconds	2.51 GB	$0.01
Savings	87% less when using Parquet	34x faster	99% less data scanned	99.7% savings

Kilde:https://databricks.com/glossary/what-is-parquet

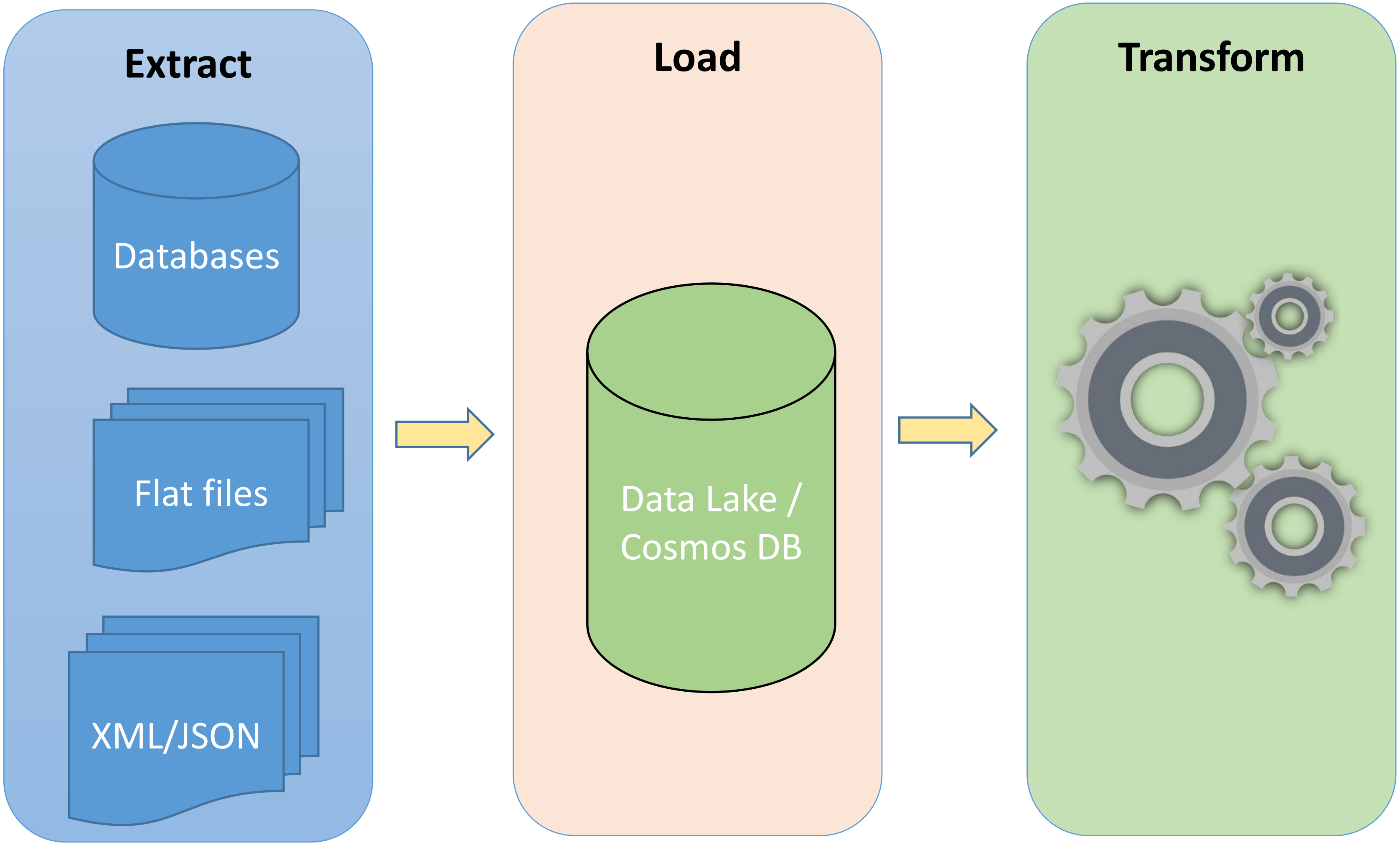
Classic ETL


Extract
Databases
Flat files
XML/JSON
Transform
Load
Data Warehouse

Big Data

Big data er et begreb indenfor datalogi, der bredt dækker over indsamling, opbevaring, analyse, processering og fortolkning af enorme mængder af data [Kilde: https://da.wikipedia.org/wiki/Big_data]

Modern ELT



Extract

Databases

Flat files

XML/JSON

Load

Data Lake / Cosmos DB

Transform

Hadoop, Spark and HDInsights

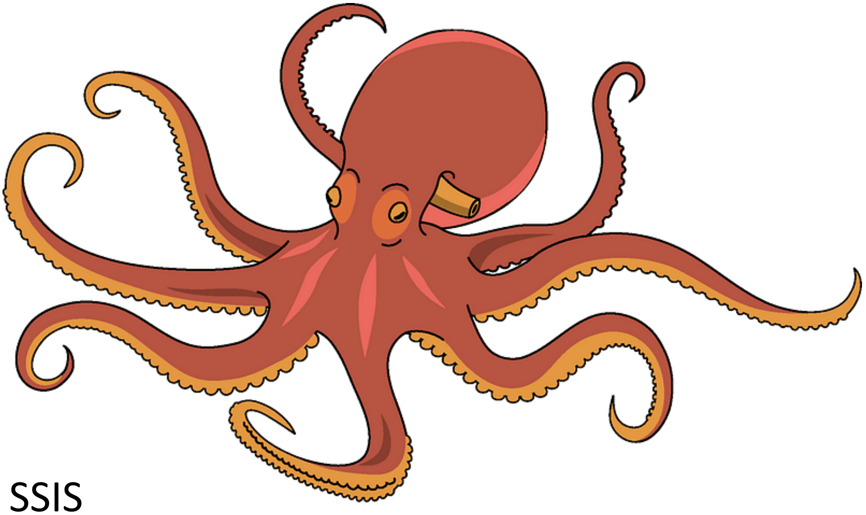
Parallel distributed processing of data

- Hadoop
- Spark Cluster
- Kafka
- Azure HDInsights

SQL Server Integration Services

SQL Server Integration Services (SSIS, 2005+)

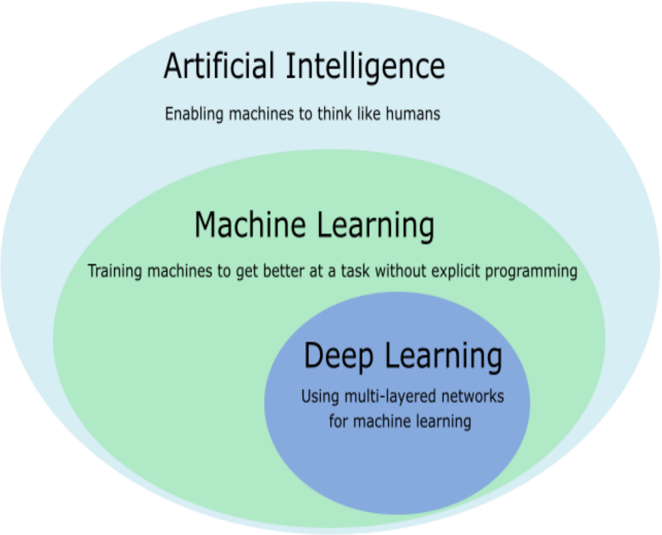
- ETL Tool from MS distributed with MS SQL Server
- Jobs/packages for data movement, transformations, processing, backup/restore
- Run once/scheduled/ad hoc



Azure Data Factory (ADF)

- Cloud udgaven/erstatningen af SSIS
- Pipelines i ADF svarer til pakker i SSIS
- Kan køre SSIS pakker

AI og ML



Artificial Intelligence

Enabling machines to think like humans

Machine Learning

Training machines to get better at a task without explicit programming

Deep Learning

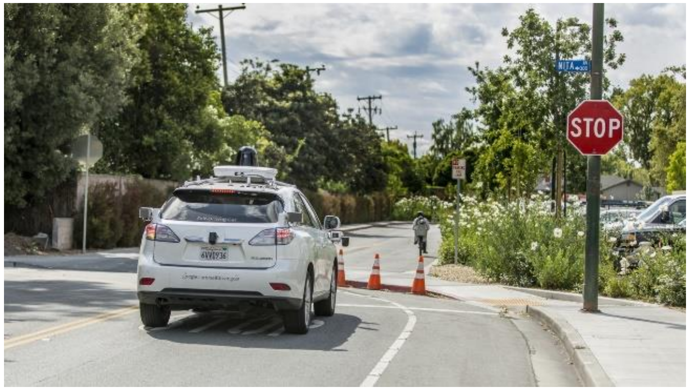
Using multi-layered networks for machine learning

Artificial Intelligence (AI)

- Perform tasks normally requiring human intelligence

Machine Learning (ML)

- Computer science + statistics
- Feed the alogorithm with data
- Recognize patterns in data
- Predictions for new data

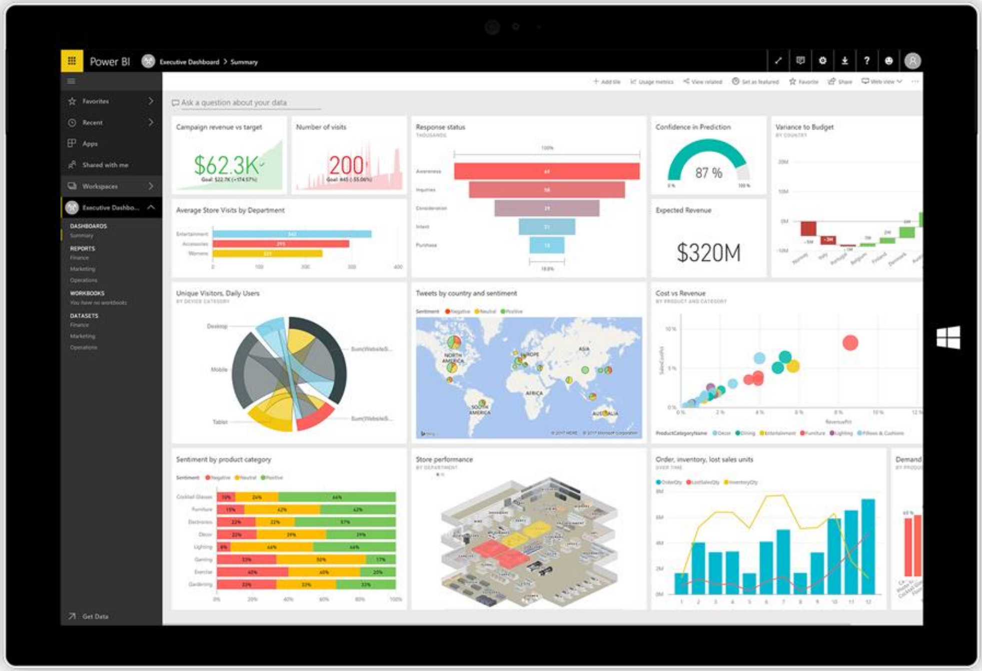


Deep Learning

- Artificial neural network (hidden layers – deeper)
- Algorithms analyzes data using a logic structure like humans

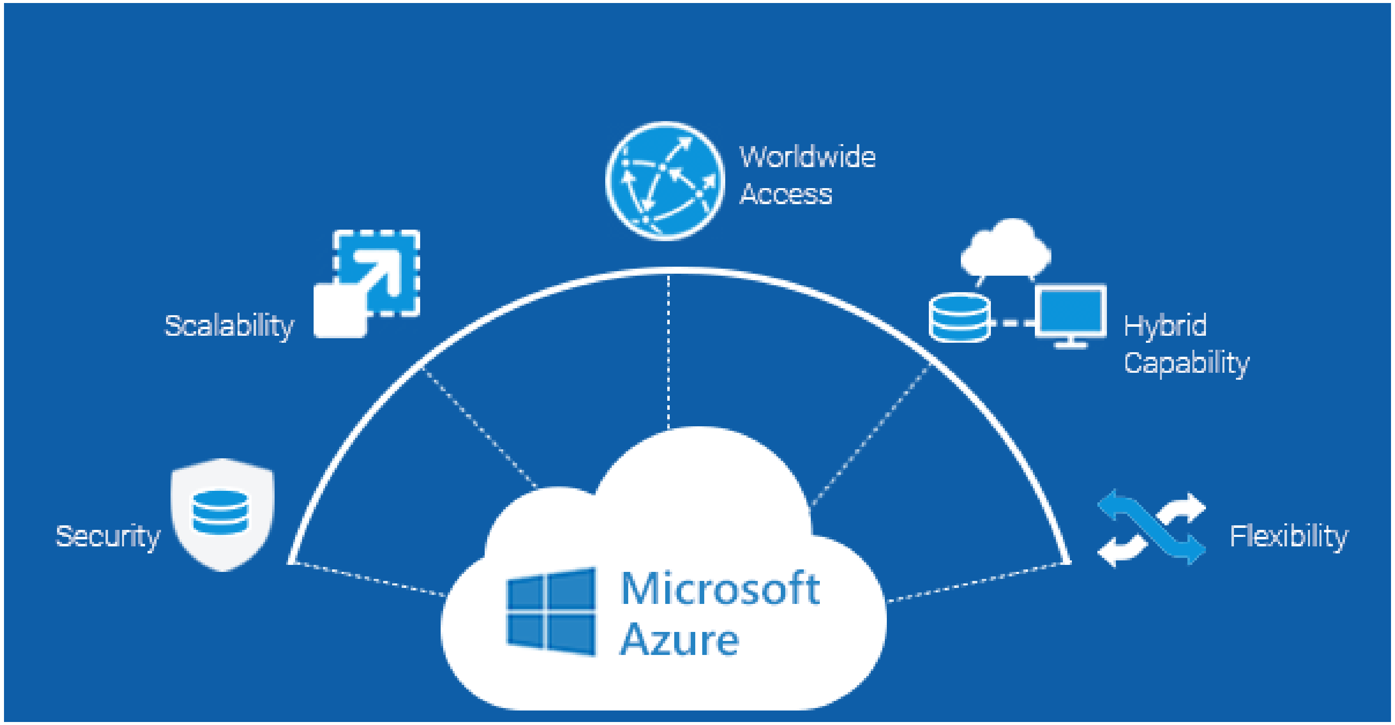
Power BI

Portefølje af produkter og services til at transformere, visualisere og præsentere data i interaktive rapporter.



Azure

Store, transform, process, analyze, and visualize data



Azure Data storage

Azure ressource navn	Type
Storage account	Blob
Storage account	Fileshare
Storage account	Table storage
Storage account	Queue storage
Storage account	Data Lake
(CosmosDB)	Gremlin, SQL, etc
(Azure SQL Database)	Relationel database
Azure Synapse Analytics	Data warehouse + mere
Azure Event Hub	
(Azure IoT Hub)	

Azure data processing

Teknologi	Input	Output
Azure Stream Analytics	Event/IoT Hub, Datalake	Blob, Power BI stream
Azure Databricks	Alt (Python til rådighed)	
Azure HDInsigths		
Azure Synapse Analytics	Alt (Python til rådighed)	
Azure Data Factory		

Data Lake

En sø af data hvor vi bare hælder alle mulige former for rå data ind som filer

TXT, CSV
JSON,XML
PDF,
AVRO,
PARQUET

Weather
Tweets
Images
Logs
Audio
Videos
etc

Data Lake

En datalake har diverse mangler, som vi er forvænt med eksisterer i database verdenen. En datalake har

- Ingen transaktioner
- Intet skema
- Svingende data kvalitet
- Mangel på konsistens/isolation ved skrivning/læsning

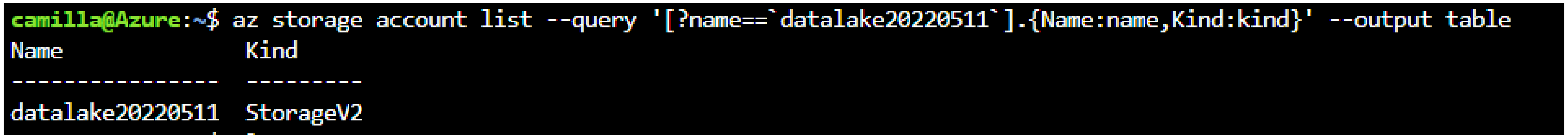
Til gengæld kan den indeholde alt muligt ustruktureret data i filer

Azure CLI

AzCLI is a command line interface for Azure

Free Cross platform shell tool

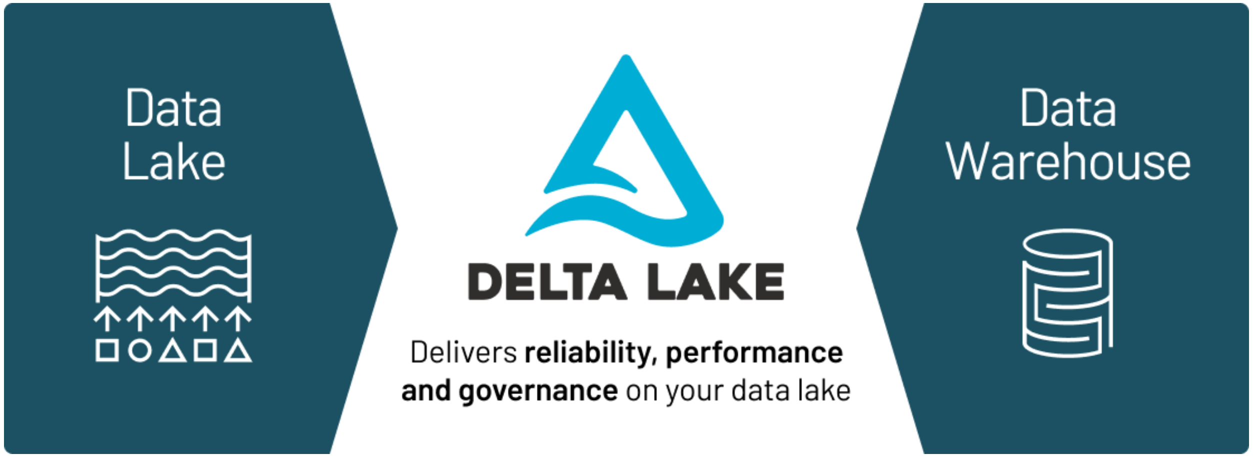
- Available in bash/PowerShell in portal.azure.com
- Can be downloaded and run locally
- Ouput default is json – use table:



```
camilla@Azure:~$ az storage account list --query '[?name==`datalake20220511`].{Name:name,Kind:kind}' --output table
Name              Kind
----------------  ---------
datalake20220511  StorageV2
```

Lakehouse

Databricks tilbyder et lakehouse



Data Lake

DELTA LAKE

Delivers **reliability, performance and governance** on your data lake

Data Warehouse

Azure Synapse Analytics har lakehouse arkitektur

Azure Synapse Analytics

Cosmos DB

Cosmos DB is a fully managed NoSQL database

- Single-digit millisecond response time
- Automatic and instant scalability

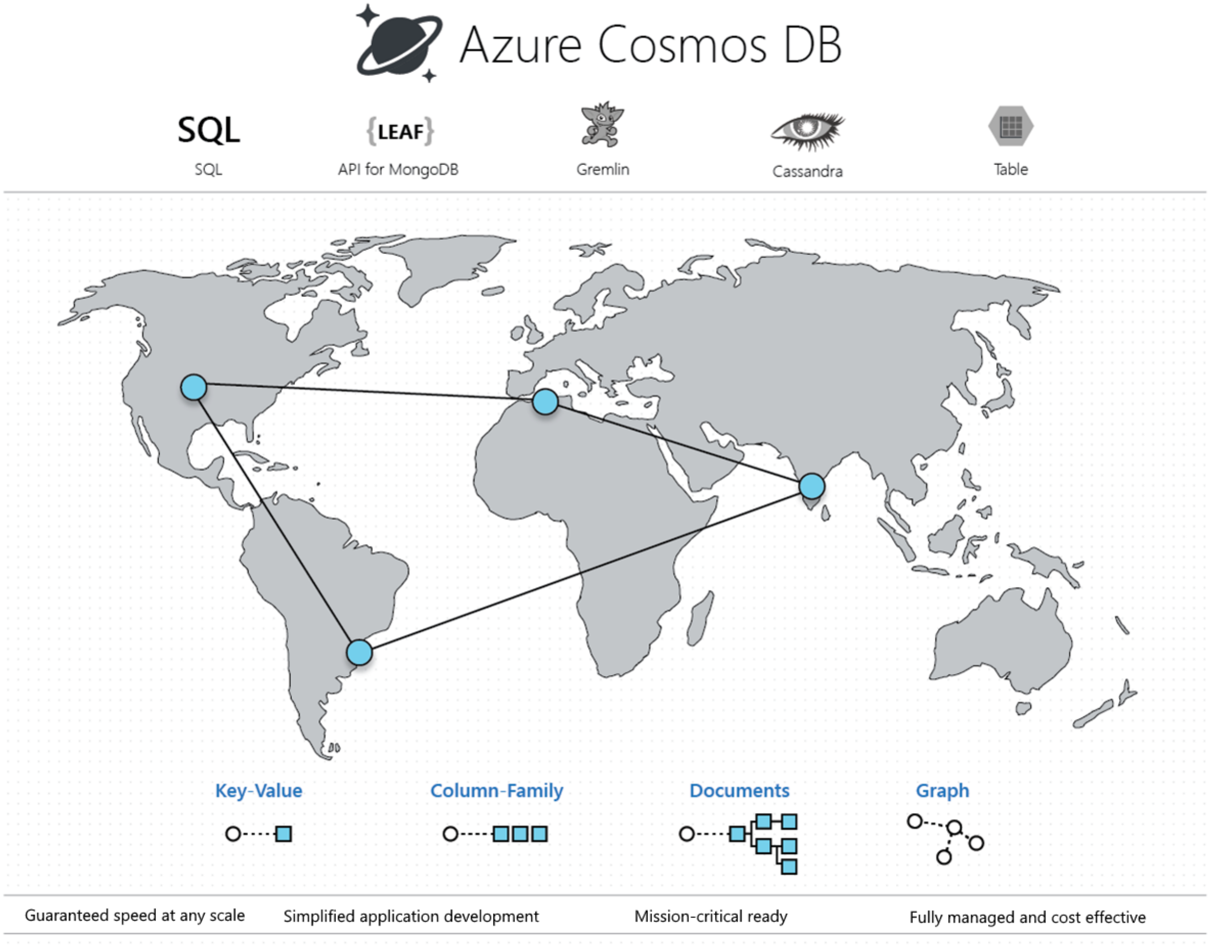
A database belongs to *one* Azure Cosmos DB account with a unique DNS name

- https://<accountname>.documents.azure.com

API is determined at account level

One account can have many databases (same API)

Azure Cosmos DB



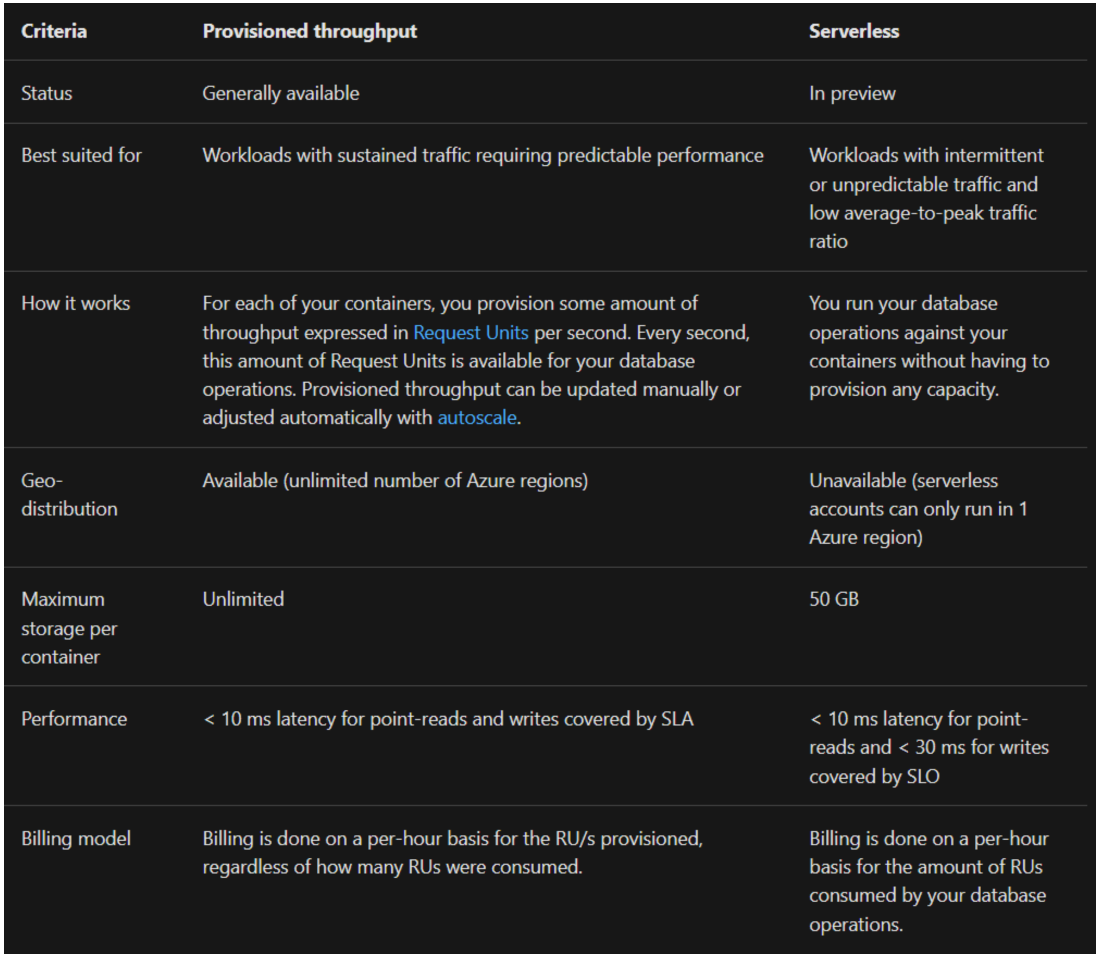
Azure Cosmos DB API

One Azure Cosmos DB Account

5 different APIs (wire protocol and storage format)

- Core (SQL) (GlobalDocumentDB) (default)
- MongoDB API
- Cassandra
- Azure Table
- Gremlin (graph)

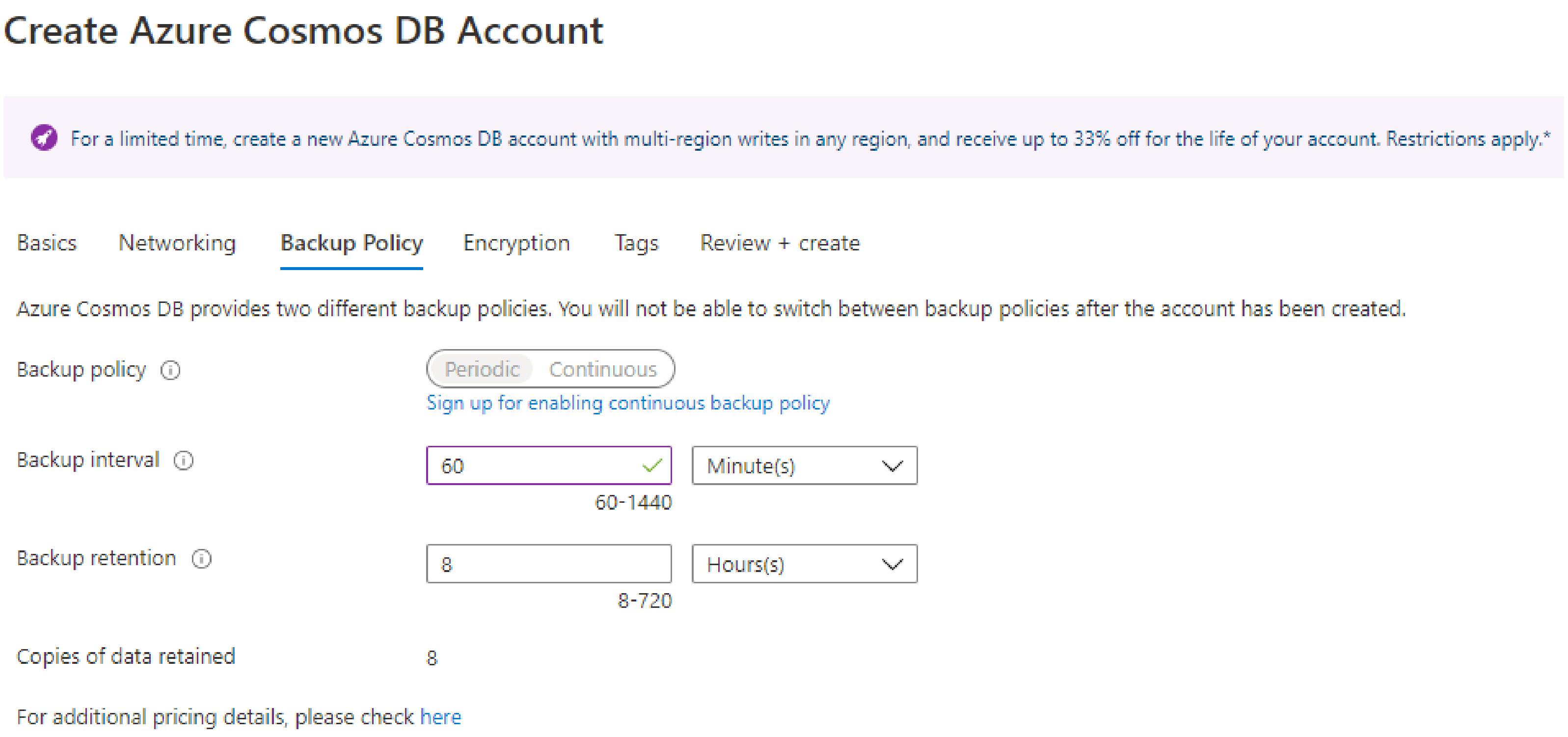
CosmosDB – Capacity mode



Criteria	Provisioned throughput	Serverless
Status	Generally available	In preview
Best suited for	Workloads with sustained traffic requiring predictable performance	Workloads with intermittent or unpredictable traffic and low average-to-peak traffic ratio
How it works	For each of your containers, you provision some amount of throughput expressed in Request Units per second. Every second, this amount of Request Units is available for your database operations. Provisioned throughput can be updated manually or adjusted automatically with autoscale.	You run your database operations against your containers without having to provision any capacity.
Geo-distribution	Available (unlimited number of Azure regions)	Unavailable (serverless accounts can only run in 1 Azure region)
Maximum storage per container	Unlimited	50 GB
Performance	< 10 ms latency for point-reads and writes covered by SLA	< 10 ms latency for point-reads and < 30 ms for writes covered by SLO
Billing model	Billing is done on a per-hour basis for the RU/s provisioned, regardless of how many RUs were consumed.	Billing is done on a per-hour basis for the amount of RUs consumed by your database operations.

Azure Cosmos DB – Backup policy

Defined at Account level



Create Azure Cosmos DB Account

For a limited time, create a new Azure Cosmos DB account with multi-region writes in any region, and receive up to 33% off for the life of your account. Restrictions apply.*

Azure Cosmos DB provides two different backup policies. You will not be able to switch between backup policies after the account has been created.

Backup policy ⓘ [✓ Periodic] Continuous

Sign up for enabling continuous backup policy

Backup interval ⓘ 60 ✓ Minute(s)

60-1440

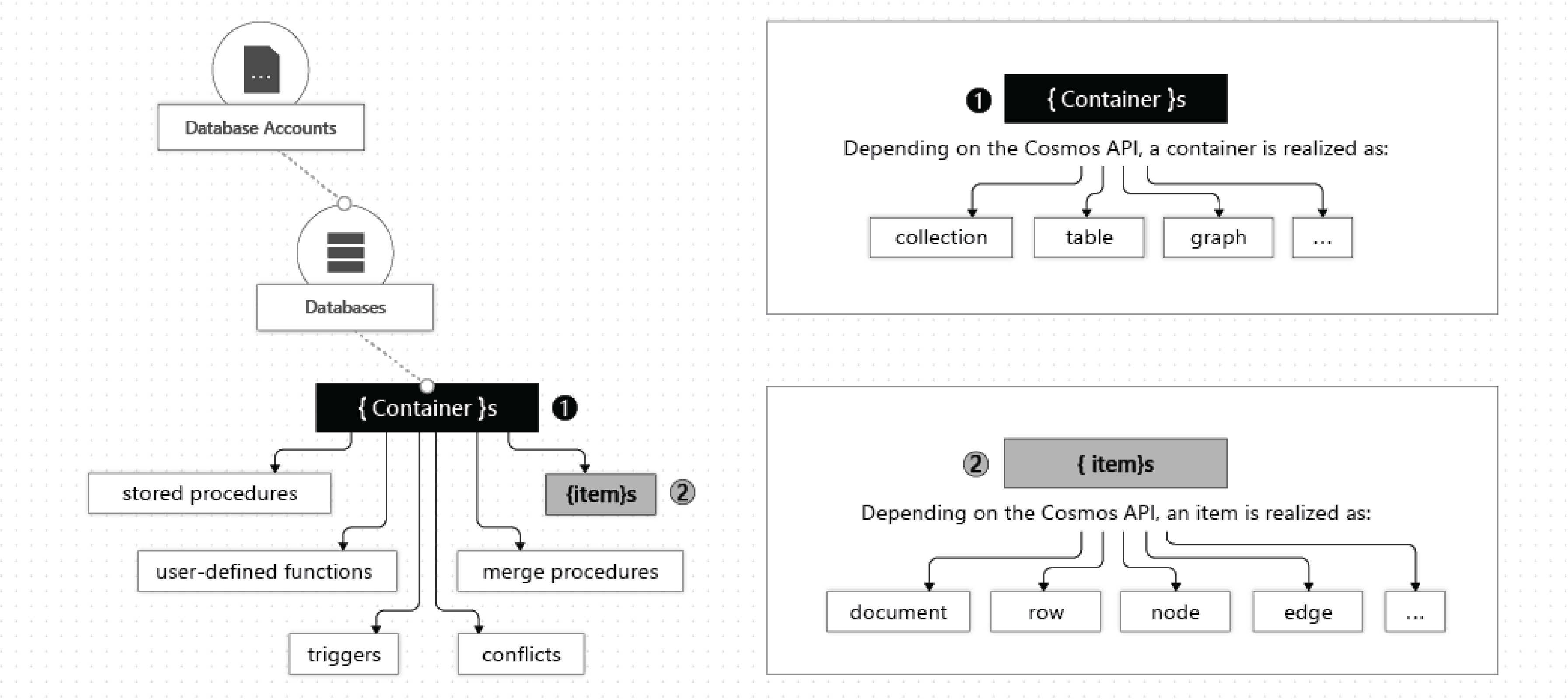
Backup retention ⓘ 8 Hours(s)

8-720

Copies of data retained 8

For additional pricing details, please check here

CosmosDB – container/items



CosmosDB - Container

- A Cosmos container is a schema-free collection of JSON items
- An Azure Cosmos container is the unit of scalability both for provisioned throughput and storage.
- A container is horizontally partitioned and then replicated across multiple regions.
- The items that you add to the container and the throughput that you provision on it are automatically distributed across a set of logical partitions based on the partition key.

Cosmos DB Container Partitionkey

Logical partitions are formed based on the value of a *partition key* that is associated with each item in a container.

Each item in a container has an *item ID* (unique within a logical partition)

The partition key and the *item ID* creates the item's *index*, which uniquely identifies the item.

Once you select your partition key, it is not possible to change it in-place.

Cosmos DB – Partition key

For all containers, your partition key should:

- Be a property that has a value which does not change. If a property is your partition key, you can't update that property's value.
- Have a high cardinality (wide range of possible values)
- Spread request unit (RU) consumption and data storage evenly across all logical partitions.

CosmosDB – Request Units (RU)

1 RU = cost to read a 1 KB document

5 RU = cost to write a 1 KB document

Minimum er 400 RU

Provision RU throughput at

- Database level(all containers share the amount)
- Container level (dedicated amount)

View Query Stats in Data Explorer

Possible to mix shared/dedicated, but container cannot change mode later

CosmosDB – Throughput

- Database
- Container

Autopilot(preview)

- 0.1*Tmax < T < Tmax
- 4 levels – max determines max size

CosmosDB - RU

- While you estimate the number of RUs per second to provision, consider the following factors:
- **Item size**: As the size of an item increases, the number of RUs consumed to read or write the item also increases.
- **Item indexing**: By default, each item is automatically indexed. Fewer RUs are consumed if you choose not to index some of your items in a container.
- **Item property count**: Assuming the default indexing is on all properties, the number of RUs consumed to write an item increases as the item property count increases.
- **Indexed properties**: An index policy on each container determines which properties are indexed by default. To reduce the RU consumption for write operations, limit the number of indexed properties.
- **Data consistency**: The strong and bounded staleness consistency levels consume approximately two times more RUs while performing read operations when compared to that of other relaxed consistency levels.

CosmosDB - Index

Auto indexing for a container. Can be turned off

- Consistent
- None

Including and excluding property paths

Can be set in the portal under Scale & Settings

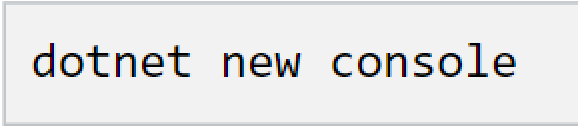
Composite index is allowed

Cosmos DB – SQL API development

Platform options:

- .NET
- Python
- Java
- Node.js
- Xamarin

dotnet CLI cmd tool example in VSCode



```
dotnet new console
```

Default consistency level

Angives på en Azure Cosmos DB account

Der er 5 muligheder Strong -> Eventual

Eksempler med musiknoder som forklarer levels:

https://docs.microsoft.com/en-us/azure/cosmos-db/consistency-levels

Cosmos DB – SQL API

Completely different from ANSI SQL

A document is a JSON item

The result of a query is a valid JSON value

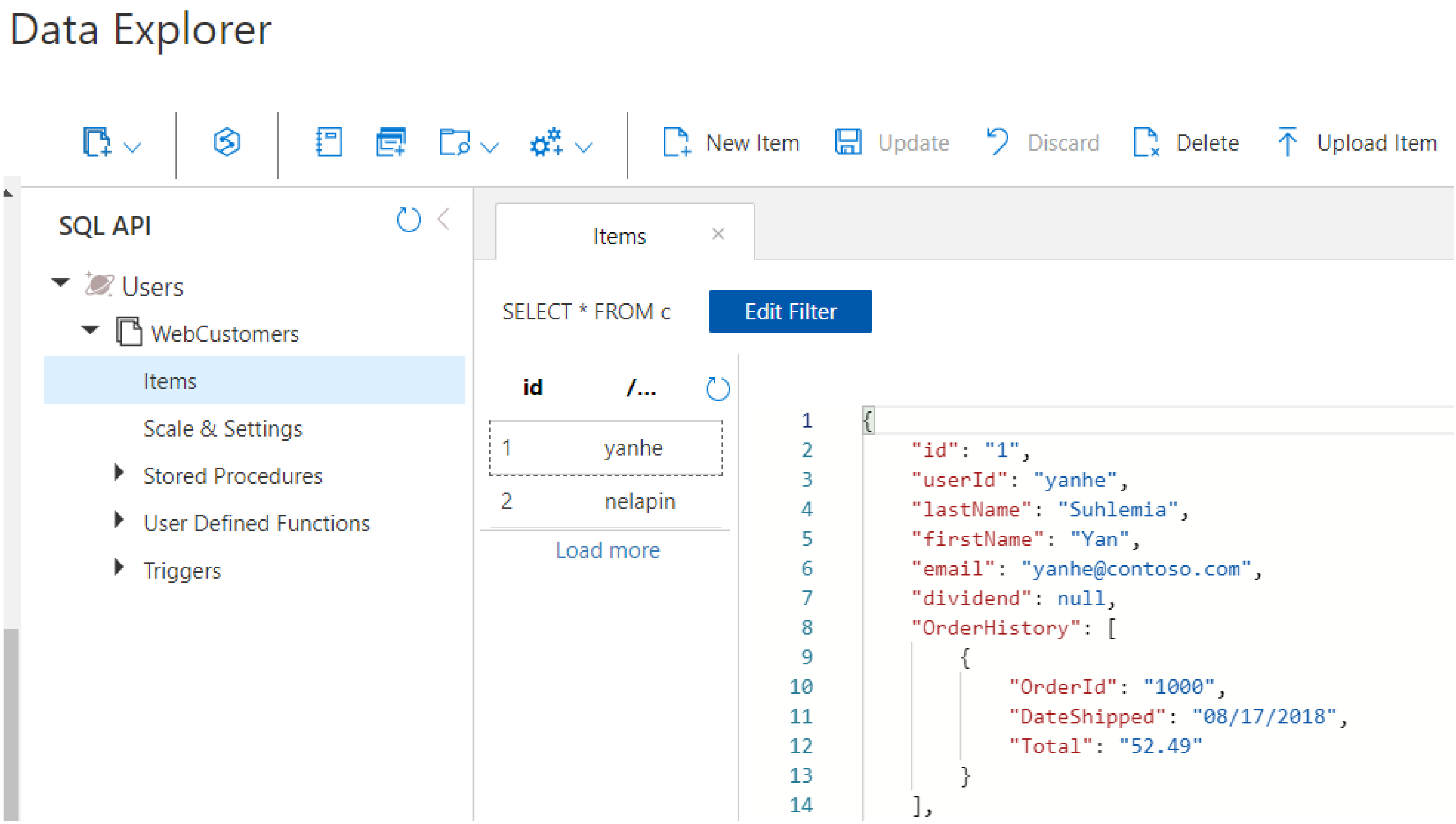
SQL API works on JSON values, it deals with tree-shaped entities instead of rows and columns

NB Case sensitive and beware of number/string

Refer to the tree nodes at any arbitrary depth, like Node1.Node2.Node3….Node<n>

Point reads (key/value lookup) vs SQL queries

Azure Cosmos DB – Data Explorer



SELECT * FROM c

id	/...
1	yanhe
2	nelapin

```
{
    "id": "1",
    "userId": "yanhe",
    "lastName": "Suhlemia",
    "firstName": "Yan",
    "email": "yanhe@contoso.com",
    "dividend": null,
    "OrderHistory": [
        {
            "OrderId": "1000",
            "DateShipped": "08/17/2018",
            "Total": "52.49"
        }
    ],
```

Azure Cosmos DB – Data Explorer

Data Explorer is a tool for the Cosmos DB SQL API in the Azure Portal

Browse/View/Create/Delete

- Databases
- Containers

Browse/View/Create/Update/Delete

- Items
- Stored Procedures
- User Defined Functions
- Triggers

Azure Cosmos DB – DB SQL API

.NET Querying (JSON) documents via

- LINQ
- SQL

Java, Python etc API

Cosmos DB – Resource tokens

Azure Cosmos DB uses two types of keys to authenticate users and provide access to its data and resources:

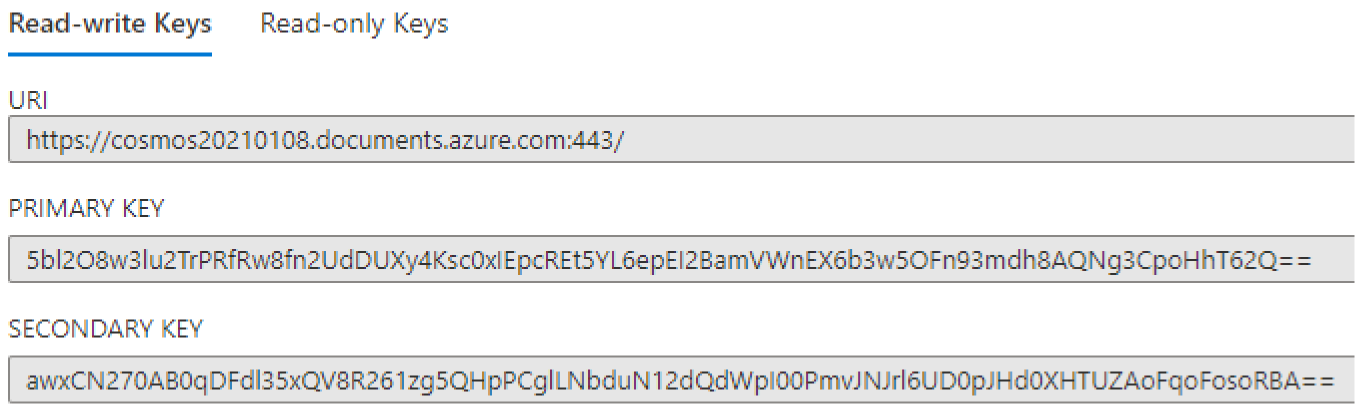
- Primary Keys
- Resource tokens

Used for application resources: containers, documents, attachments, stored procedures, triggers, and UDFs

Azure Cosmos DB – Security Access

Two account keys for administrative resources: database accounts, databases, users, and permissions

Two account keys for Read-only access on account



Read-write Keys Read-only Keys

URI

https://cosmos20210108.documents.azure.com:443/

PRIMARY KEY

5bl2O8w3lu2TrPRfRw8fn2UdDUXy4Ksc0xlEpcREt5YL6epEl2BamVWnEX6b3w5OFn93mdh8AQNg3CpoHhT62Q==

SECONDARY KEY

awxCN270AB0qDFdl35xQV8R261zg5QHpPCglLNbduN12dQdWpl00PmvJNJrl6UD0pJHd0XHTUZAoFqoFosoRBA==

Cosmos DB – Database Users

A database can contain zero or more users

Permissions on a resource

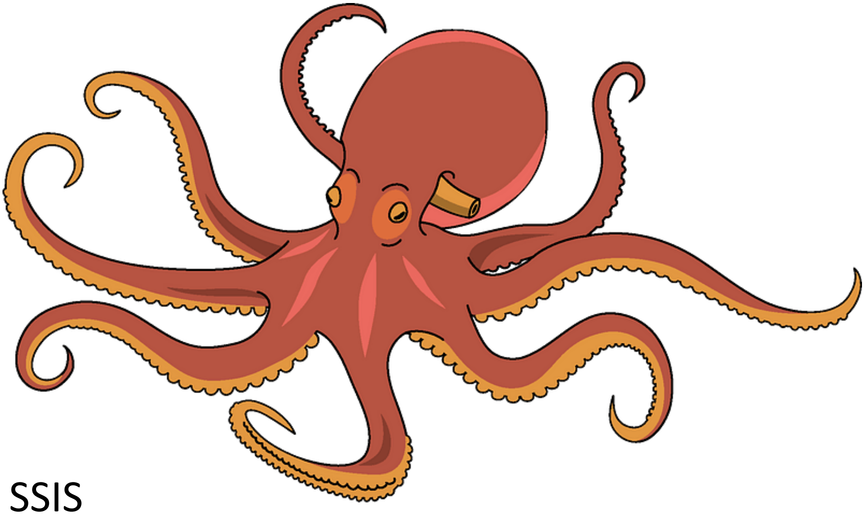
- All (full permission)
- Read (no write, update or delete)

Running a stored procedure requires All permission on the container

SQL Server Integration Services

SQL Server Integration Services (SSIS, 2005+)

- ETL Tool from MS distributed with MS SQL Server
- Jobs/packages for data movement, transformations, processing, backup/restore
- Run once/scheduled/ad hoc



Azure Data Factory (ADF)

- Cloud udgaven/erstatningen af SSIS
- Pipelines i ADF svarer til pakker i SSIS
- Kan køre SSIS pakker

Azure Data Factory - history

Data movement, transformations, processing data, ETL tools:

- SQL Server Integration Services (SSIS, 2005+)
- Azure Data Factory v1 (2015+)
- Azure Data Factory v2 (2018+)

Pipelines i ADF svarer til pakker i SSIS

Azure Data Factory (ADF)

Data integration

Data Processing

ETL and ELT (skema for datamodel)

SSIS integration runtime

Administrate ADF via GUI or json files

The serverless integration service does the rest..

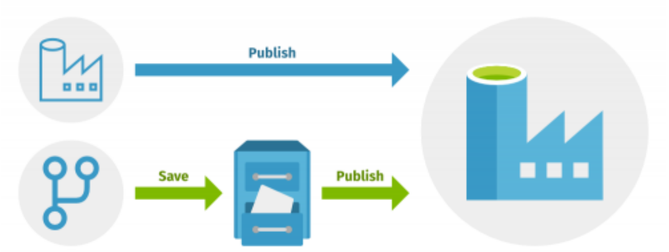
Azure Data Factory - Steps

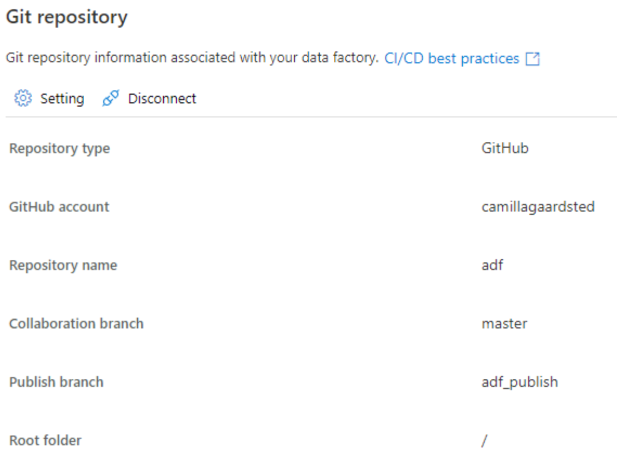
1. Connect to all the required sources of data and processing, such as software-as-a-service (SaaS) services, databases, file shares, and FTP web services.
2. Move the data to a centralized location for subsequent processing.
3. Process, analyze and/or transform data
4. Export/load data into destination

Azure Data Factory – Git repository

Git integration afgør om man har en Save eller Publish knap

Git via: Azure DevOps or GitHub





Git repository

Git repository information associated with your data factory. CI/CD best practices

Setting Disconnect

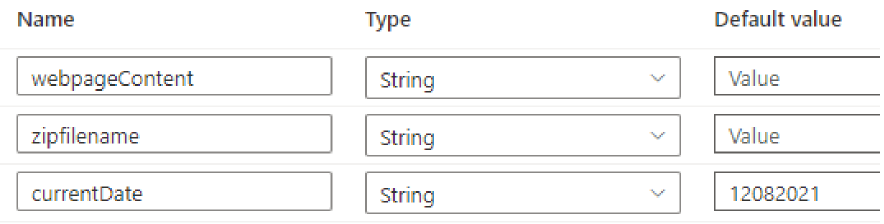
Repository type	GitHub
GitHub account	camillagaardsted
Repository name	adf
Collaboration branch	master
Publish branch	adf_publish
Root folder	/

Azure Data Factory

- Linked services (connection to the data source)
- Datasets (structure of the data)
- Pipeline
- Activities
- Data flows
- Triggers
- Integration runtimes (IR)

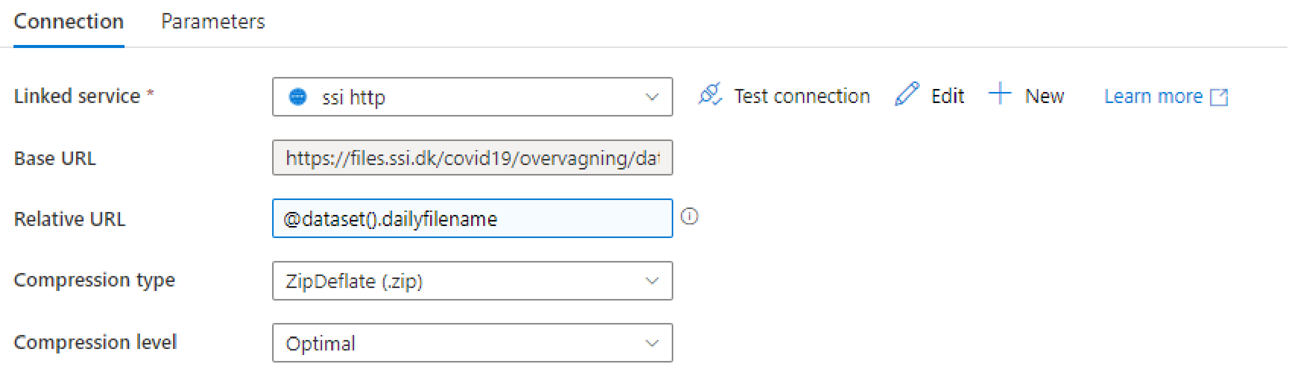
ADF – Dynamic pipeline

Pipeline variables (global in pipeline)



Name	Type	Default value
webpageContent	String	Value
zipfilename	String	Value
currentDate	String	12082021

Dataset parameters



Linked service *	ssi http
Base URL	https://files.ssi.dk/covid19/overvagning/dat
Relative URL	@dataset().dailyfilename
Compression type	ZipDeflate (.zip)
Compression level	Optimal

ADF – Linked services

A linked service is connection information e.g. like a connectionstring

View/edit via ADF->Monitor->Linked services

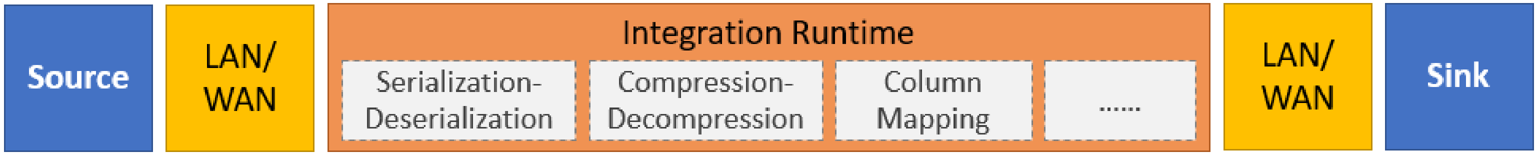
ADF - Dataset

- Data store
- Format

ADF – Copy Activity

Source options (many)
Sink options (limited)

You can use the Copy activity to copy files as-is between two file-based data stores, in which case the data is copied efficiently without any serialization or deserialization.



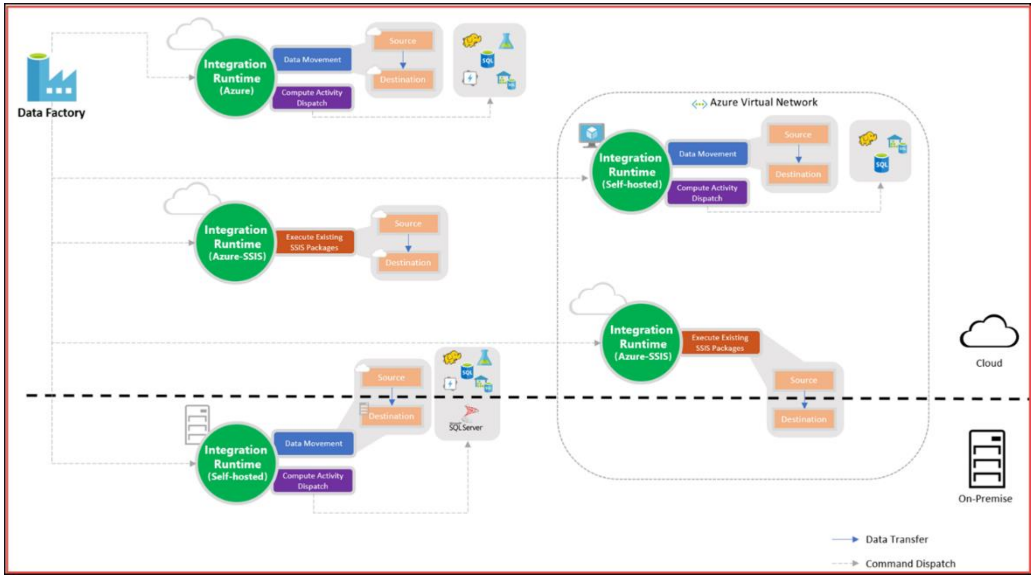
ADF - Data flow activity

- Mapping data flow
  - Uses Azure Databricks cluster
  - Visual flow
  - Each step in the flow is a transformation
  - Preview data via Debug
  - Handles also inserts, updates, deletes and upserts

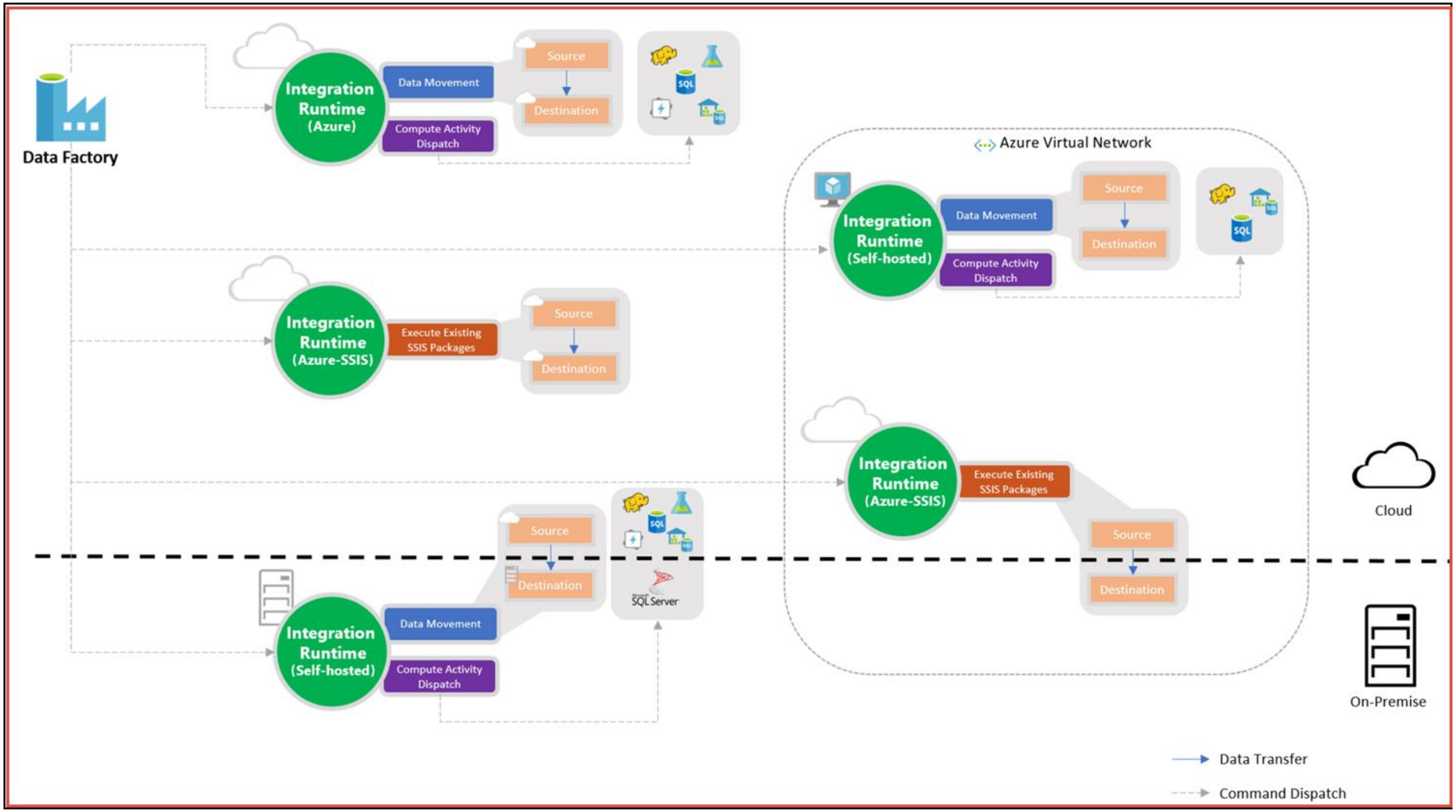
- Power Query (fremtid uvis???)
  - Uses a managed Spark environment
  - Power Query Online mashup editor (M)
  - Not all M commands are supported!

ADF – Integration Runtime (IR)

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


Data Factory
Integration Runtime (Azure)
Data Movement
Compute Activity Dispatch
Source
Destination
Azure Virtual Network
Integration Runtime (Self-hosted)
Data Movement
Compute Activity Dispatch
Source
Destination
SQL
Integration Runtime (Azure-SSIS)
Execute Existing SSIS Packages
Source
Destination
Integration Runtime (Azure-SSIS)
Execute Existing SSIS Packages
Source
Destination
Cloud
Source
Destination
SQL
SQL Server
Integration Runtime (Self-hosted)
Data Movement
Compute Activity Dispatch
On-Premise
Data Transfer
Command Dispatch

ADF – Integration Runtimes


Data Factory
Integration Runtime (Azure)
Data Movement
Compute Activity Dispatch
Source
Destination
Azure Virtual Network
Integration Runtime (Self-hosted)
Data Movement
Compute Activity Dispatch
Source
Destination
SQL
Integration Runtime (Azure-SSIS)
Execute Existing SSIS Packages
Source
Destination
Integration Runtime (Azure-SSIS)
Execute Existing SSIS Packages
Source
Destination
Cloud
Source
Destination
SQL
SQL Server
Integration Runtime (Self-hosted)
Data Movement
Compute Activity Dispatch
On-Premise
Data Transfer
Command Dispatch