**Operational system:**

An **Operating System** can be defined as an **interface between user and hardware**. It is responsible for **the execution of all the processes, Resource Allocation,**[**CPU**](https://www.javatpoint.com/cpu-full-form)**management, File Management and many other tasks.**

**Characteristics of operational system:**

**Process management:**

* Which operator should work first?
* How much time each processor should work?
* Allocating process to processor
* Deallocating process from processor

**Memory management:**

* How much memory should be allocated for process?
* How much memory should be deallocated for process?
* How much memory is being used currently?
* How much memory is free to use?

**I/O device managemnets(mouse keyboard,printer):**

* Which process gets the device for how much time?
* Managing the connection of these devices?
* Plug and play detects and configures new devices?
* Error handling manage device failure.

**File management:**

* The os organize and manipulate the files and directories.
* It provides file access and create deletion and permissions.

**Security:**

* The os ensure data security and user authentication**.**
* It protects against unauthorized access and malicious software.

**Concurrency Control**:

* The OS handles simultaneous execution of processes.
* It prevents conflicts and ensures data consistency.

**Need of separate operational and information systems:**

1. **Operational Systems (OLTP - Online Transaction Processing)**:

* **Purpose**: Operational systems **handle day-to-day business transactions**, such as inventory management, order processing, and payroll.
* **Data Focus**: They deal **with real-time data values**, ensuring smooth running of business processes.
* **Optimization**: **Operational databases are optimized for transactional** workloads, including indexing, hashing, and efficient queries.
* **Response Time**: They provide **sub-second response times.**
* **Volume**: Suited for **small volumes of data**.

1. **Informational Systems (MIS - Management Information Systems)**:

* **Purpose**: Informational systems **support decision-making** and management activities.
* **Data Focus**: They collect, **analyze, and report data** to aid planning, control, and decision-making.
* **Optimization**: **Optimized for complex queries** and multidimensional views.
* **Response Time**: Typically have **response times ranging from a few seconds to minutes.**
* **Volume**: Designed for **large volumes of historical data.**

what we are doing in Business Intelligence?

we want to use and find raw data and transform this data in such a way That we can transform it into a meaningful insight.

Business Intelligence is made up of different.

strategies,

procedures,

technologies

infrastructures,

example:**data analytisc**

data gathering

data storing

reporting

data visualization

data mining

predictive analytics

**DATA WAREHOUSE:**.

It is central repository of information where the data is collected from different source and stored in the structure format. It is a database that stores information oriented to satisfy decision-making requests. It includes historical data derived from transaction data from single and multiple sources.

Data Warehouse Systems are called as Online-Analytical Processing (OLAP) Systems.

**Characteristics of Data Warehouse:**

**subject-oriented:**

Data warehouses typically provide a concise and straightforward view around a particular subject, such as customer, product, or sales, instead of the global organization's ongoing operations.

**integrated:**

A data warehouse integrates various heterogeneous data sources like RDBMS, flat files, and online transaction records.

**time-variant:**

The data collected in a data warehouse is identified with a particular time period. The data in a data warehouse provides information from the historical point of view.

**Non-volatile:**

Non-volatile means the previous data is not erased when new data is added to it. A data warehouse is kept separate from the operational database and therefore frequent changes in operational database is not reflected in the data warehouse.

Goals of Data Warehousing

* To help reporting as well as analysis.
* Maintain the organization's historical information.
* Be the foundation for decision making.

Need of data warehouse:

* Store historical data.
* Make strategic decisions.
* For data consistency & quality .
* High response time.
* Business users.

## Benefits of Data Warehouse

* Understand business trends and make better forecasting decisions.
* Data Warehouses are designed to perform well enormous amounts of data.
* Data warehousing provide the capabilities to analyze a large amount of historical data.

## Data warehousing is an efficient method to manage demand.

* Queries that would be complex in many normalized databases could be easier to build and maintain in data warehouses.

## Operational Database:

## The Operational Database is the source of information for the data warehouse. It includes detailed information used to run the day to day operations of the business. The data frequently changes as updates are made and reflect the current value of the last transactions.

## Operational Database Management Systems also called as OLTP (Online Transactions Processing Databases), are used to manage dynamic data in real-time.

## Data Warehouse(OLAP) and the operational Database (OLTP) database are both relational databases.

**On-Line Transaction Processing (OLTP)** [[OLAP vs OLTP in English | Online Transaction Processing VS Online Analytical Processing difference](https://www.youtube.com/watch?v=yuF7ocrvvUg)](https://www.youtube.com/watch?v=yuF7ocrvvUg)

* OLTP System **handle with operational data**.
* OLTP manage many concurrent customers and queries touching only an individual record.
* OLTP system focuses primarily **on the current data within an enterprise** or department,
* OLTP system usually uses an entity-relationship (ER) data model and application-oriented database design.
* OLTP system are typically related to tasks like **updating customer records, processing orders,** or managing inventory.( **Read/write**)
* **Volume of data is not very large**.
* **Concurrency Control and Recovery**: Due to frequent updates, OLTP systems require mechanisms to handle**concurrency** (multiple users accessing data simultaneously) and**recovery** (in case of system failures).
* **Users:** OLTP systems are customer-facing and designed for use by frontline workers such as store clerks, hotel reservation specialists, and online shoppers.
* **Examples:**
* **ATM centers**
* **Online banking transactions**.

Online Analytical Processing (OLAP) [OLAP Operations in Data Warehouse | Slice Dice Roll-up Drill-down Pivot | Example of location, time (youtube.com)https://www.youtube.com/watch?v=4NKbZMsddBI](https://www.youtube.com/watch?v=4NKbZMsddBI) (**slice, dice, drill down, roll up, pivot**)

* OLAP handle with **Historical Data**.
* OLAP system must have the **capability to operate on millions of files** to answer a single query.
* OLAP systems also deal with data that originates from various organizations, integrating information from many data stores.
* OLAP system typically uses either a star or snowflake model and subject-oriented database design.
* OLAP systems are mostly **read-only** methods because of these data warehouses stores historical data.
* **Volume of data is very large**. OLAP data are stored on multiple storage media.
* **Users**: OLAP systems are **business-facing** and used by data scientists, analysts, and business decision-makers .

**Examples**:

* + **Data warehouse systems**
  + **Spotify analyzing user songs for personalized recommendations**
  + **Netflix’s movie recommendation system**

# **Data Warehouse Architecture:**

* A data warehouse architecture is a method of **defining the overall architecture of data communication processing and presentation** that exist for end-clients computing (enterprise)
* payroll accounts payable product purchasing and inventory control are designed for online transaction processing **(OLTP)**. **Gather detailed data from day to day operations**.
* Data warehouse must be **restructured tables de-normalized, data cleansed of errors and redundancies** and new fields and keys added to reflect the needs to the user for sorting, combining, and summarizing data.
* **Source(Operational database Management system,falt files)**
* **Staging layer and persistence layer**
* **Cleaning**
* **Core/data warehouse**
* **Data mart**
* **Visualization(tabule,powerbi)**

**SOURCE:**

An **operational system** is a method used in data warehousing to refer to a **system** that is used to process the day-to-day transactions of an organization.

**Flat Files:**

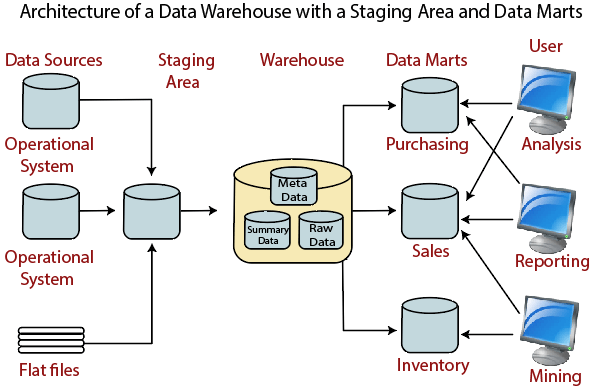
A **Flat file** system is a system of files in which transactional data is stored, and every file in the system must have a different name.

**Data Warehouse Staging Area** is a **temporary location where a record from source systems is copied**. where we  clean and process your operational information **before put it into the warehouse.**

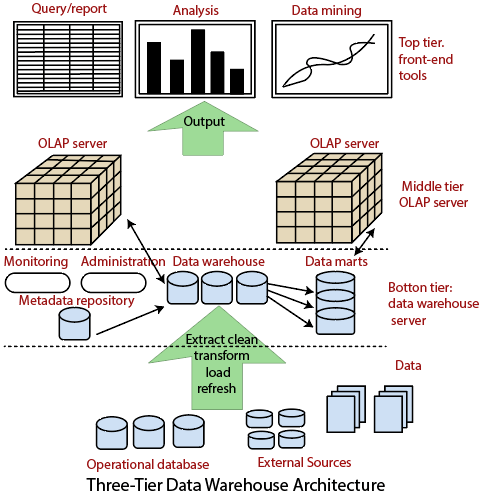
**DATA MART:**

A data mart is a **segment of a data warehouses** that can **provided information for reporting and analysis on a section, unit, department** or operation in the company, eg., sales, payroll, production, etc.

* **Subset of dwh**
* **Usability+acceptance**
* **performance**



**Three-Tier Data Warehouse Architecture**, which is commonly **used in designing data warehouses**. This architecture provides a structured approach to building a data warehouse, ensuring optimal performance and scalability. Here are the three tiers involved:



1. **Bottom Tier (Data Warehouse Server)**:
   * The bottom tier serves as the foundation of the data warehouse. It typically consists of an **RDBMS (Relational Database Management System)** that stores the actual data.
   * Data from various sources, including operational databases and external systems, is **extracted** (**extracted using application program interfaces called a gateway**)and loaded into this tier.
   * Specialized data marts may also reside here, focusing on specific business areas or subject domains.
   * A **metadata repository** is part of this tier, storing information about **the data warehouse structure, schema, dimensions, hierarchies, and more.**
2. **Middle Tier (OLAP Server)**:
   * The middle tier is responsible for **fast querying** of the data warehouse. It enhances performance by providing efficient access to aggregated data.
   * Within this tier, you’ll find the **OLAP (Online Analytical Processing) server**.
   * Two common OLAP models are used:
     + **Relational OLAP (ROLAP)**: An extended relational DBMS that maps multidimensional functions to standard relational operations.
     + **Multidimensional OLAP (MOLAP)**: A specialized server directly implementing multidimensional data and operations.

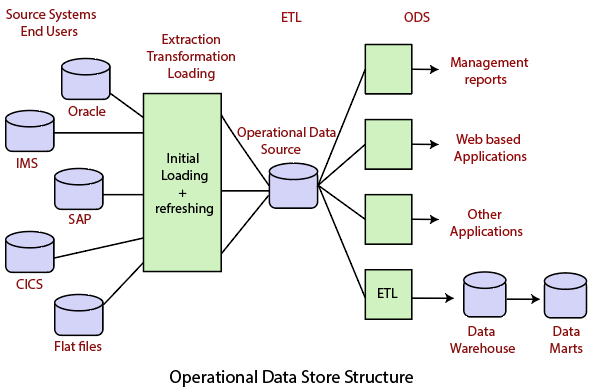
<https://www.youtube.com/watch?v=QnDcu-0YsG0%20> (explanation of rolap,molap)

1. **Top Tier (Front-End Tools)**:
   * The top tier is **where end users interact with the data warehouse**. It includes various **front-end tools for reporting, analysis, and visualization**.

**Operational data source:**

**An operating data source is where a company gathers and stores data on its daily operations, such as transactions, interactions and processes for analysis and decision making.**

* The **ODS** is an **integrated**. That is, it is a group of subject-oriented record from a variety of systems to provides an enterprise-wide view of the information.
* **ODS** is up-to-date and **follow the current status of the data.**
* **ODS** **does not contain historical information**.
* **ODS** **is volatile**. ODS frequently changes **as new data refreshes** the ODS.



**Facts and dimensions:**

**What is facts?**

A fact table is a **primary table in a dimensional model**.

**Grain: Most atomic level facts are defined**

A Fact Table contains

1. **Primary key**
2. **Foreign key to dimension table**
3. **Measurements/facts**

**What is dimension table?**

* A dimension table contains **dimensions of a fact**.
* They are joined to **fact table via a foreign key**.
* Dimension tables are **de-normalized tables**.
* Dimensions **offers descriptive characteristics** of the **facts** with the **help of their attributes**.
* **Dimension Attributes are the various columns** in a dimension table.

Facts vs dimensions

ETL : Extract, Transform and Load

**ETL is a process that extracts the data from different source systems, then transforms the data (like applying calculations, concatenations, etc.) and finally loads the data into the Data Warehouse system.**

**Two types of ETL:**

* **Initial ETL**
* **Incremental ETL**
  + **Four types of incremental ETL**

1. **Append: small amount of data added every time in one or more tables.**
2. **In-place update or slowly changing dimensions: we are not adding any data, but we are going to exciting data and update the things.**
3. **Complete replacement: while adding data we are going to add new data and the same time we are going to replacement of exiting data(refresh).**
4. **Rolling append: removing old data (certain duration of data)and appending new data.**

**Type of transformations models:**

* + - **Data value unification. (modifying row data to makes row data common in both tables)**
    - **Data type and size unification (resize data type of column and making one column which insert both table data)**
    - **De-duplication (removing duplicates)**
    - **Drooping columns or vertical slicing (removing unnecessary columns which are not used for analytical purpose)**
    - **Value-based row filtering or horizontal slicing (removing row from table and keeping only necessary information like units, departments wise )**
    - **Correcting known errors (removing wrongly inserted values in table)**

**Azure Storage platform** :

it includes the following data services:

* **Azure Blobs**: A massively scalable object store for text and binary data. Also includes support for big data analytics through Data Lake Storage Gen2.
* **Azure Files**: Managed file shares for cloud or on-premises deployments.
* **Azure Queues**: A messaging store for reliable messaging between application components.
* **Azure Disks**: Block-level storage volumes for Azure VMs.
* **Azure Tables:** NoSQL table option for structured, non-relational data.

**Access tiers:**

* Hot, cool, cold, and archive tiers can be set at the blob level, during or after upload.
* Data in the cool and cold access tiers can tolerate slightly lower availability, but still requires high durability, retrieval latency.
* **Hot access tier**: Optimized for storing data that is accessed frequently (for example, images for your website).
* **Cool access tier**: Optimized for data that is infrequently accessed and stored for at least 30 days (for example, invoices for your customers).
* **Cold access tier**: Optimized for storing data that is infrequently accessed and stored for at least 90 days.
* **Archive access tier**: Appropriate for data that is rarely accessed and stored for at least 180 days, with flexible latency requirements (for example, long-term backups). Archive storage stores data offline and offers the lowest storage costs, but also the highest costs to rehydrate and access data.

**Azure Storage Explorer:**

Azure Storage Explorer is a standalone app that provides a graphical interface to manage files and blobs in your Azure Storage Account.

# zero trust model

**When you create a resource group in Azure, you specify a region for that resource group**. **This region is where metadata about the resource group itself is stored, such as tags, policies, and access control settings.**

1. **Resource Group Region**:

* The **region assigned to the resource group is immutable**.
* It **remains fixed** after creation.
* **All the resources within that resource group inherit the same region as the resource group itself**.

1. **Resource Region**:

* **Each individual resource (such as a virtual machine, storage account, or App Service) can have its own specific region.**
* **By default, when you create a resource within a resource group, it is placed in the same region as the resource group.**
* **While you can create resources in different regions within the same resource group, it’s generally recommended to keep them in the same region for better performance and management.**

1. **Access Control Lists (ACLs)**:
   * ACLs allow you to apply **a finer-grained level of access to directories and files within Data Lake Storage Gen2**.
   * You can **define custom ACLs to restrict access to specific files or folders.**
   * However, **ACLs do not directly prevent deletion; they control read, write, and execute permissions**.
2. **Resource Locks**:
   * Resource locks are a way **to prevent accidental deletion of Azure resources.**
   * You can apply **a lock at the resource group level or the individual resource level (such as a storage account).**
   * When a lock is applied, **it prevents any modification or deletion of the resource until the lock is removed.**
   * There are two types of locks:
     + **Delete Lock**: **we can perform read/write operation but we cannot delete**(Prevents deletion of the resource)
     + **Read-Only Lock**: Allows **read operations but prevents modifications.** (Prevents deletion of the resource)
   * Applying a delete lock to **the parent storage account would indeed prevent accidental deletion of the entire account, including its child resources (like folders in Data Lake)**.

**Combine ACLs and Resource Locks**:

* + - **Set up appropriate ACLs on the specific files or folders you want to protect.**
    - **Apply a delete lock to the storage account to safeguard against accidental deletion.**

**Types of Encryptions in azure cloud:**

This **encryption mechanisms** help protect data at **different stages like transit(moving data),at rest, begin processed to have integrity for Azure cloud service.**

**Transport Layer Security(TLS):** TLS encrypts **data transmitted over networks,** ensuring **secure communication between client and server**.

* It is commonly used for web traffic encryption such as **HTTPS**

**Azure Disk Encryption:** This feature **encrypts the operating system and data disks** of Azure virtual machine(VMs).

It help protect data **against unauthorized access, both while in use and when stored on disk**.

**Azure Key Vault: Azure key vault provides a centralized cloud service for securely storing and managing cryptographic keys, secrets and certification.**

**Azure Storage Service Encryption:** SSE **automatically encrypts** data stored in **Azure Blob storage, Azure Files and Azure Queue storage**.

It simplifies data protection **by handling encryption and decryption transparently**

**Azure SQL Database Transparent Data Encryption(TDE):TDE** encrypts **SQL Database data files** that data is **encrypted before being written to storage and decrypted when read from storage**.

Azure sql database manages **the encryption keys are stored securely** and are used to encrypt and decrypt the data. **Users do not have direct access to these keys**.

**Explain encryption by SSE?**

* Data encryption at rest focusing on **MMK(Microsoft- Managed key) and CMK(Customer-Management key)** in Azure.
* **Microsoft- Managed key(MMK):** Azure manages and **rotates the encryption keys used for SSE**. This option provides a convenient way to **enable encryption without managing keys manually.**
* **Customer-Management key(CMK):** users can **bring their encryption keys and manage them using Azure Key vault**. This option gives users **more control over their encryption keys** and allows them to meet **specific compliance requirement**.

**Access keys and SAS keys:**

**Access keys:** An access key is a **long-term credential associated** with your Azure subscription or resource. It consists **of 2keys are used to provide redundancy and security purpose** .This keys are used to **authenticate and authorize access to resource** within the azure account.

* Azure storage account provide **built-mechanism for rotation and maintaining continuous access of data.**
* Access keys are used to access and manage **Azure storage accounts,such as uploading files,deleting blobs or creating container.**

**Shared Access Signature(SAS):**A **SAS** provides more granular control and **temporary access to specific resources**. It allows you to delegate access to someone **without sharing your account** keys.

* You can specify constrains such as **permissions, start and expiry time** and IP address.
* Sas token allowing a **client application to upload/download files to/from** that container for limited **time period.**

**What are end points in azure cloud?**

**Public Endpoint:** A public endpoint is **accessible from the internet**, **allowing external users** or **services to interact with your Azure resource**.

These endpoints are used for **web application ,Api or services** that need to be accessed by **clients out side azure.**

**Private Endpoint:** A private endpoint is accessible only from **within virtual network or specific network environment** providing secure and private **access to azure resources** **without exposing them to the public internet.**

**Service Endpoint:** A service endpoint **extends your virtual network to the azure service** over a direct connection, allowing traffic to the azure service to stay within the Microsoft Azure backbone network.

**What is difference between DTU(Database transaction unit) and VCore?**

**Vcore:**

* A virtual core (vCore) represents a **logical CPU** and offers you the option to choose between **generations of hardware and the physical characteristics of the hardware** (for example, the **number of cores, the memory, and the storage size).**
* This model allows you to choose **compute, memory, and storage resources** based on your workload needs.
* **General Purpose**, **Business Critical**, and **Hyperscale** service tiers.
* In the vCore-based purchasing model, your costs depend on the choice and usage of

1. Service tier

2.Hardware configuration

3.Compute resources (the number of vCores and the amount of memory)

4.Reserved database storage

5.Actual backup storage

**DTU:**

* A database transaction unit (DTU) represents a **blended measure of CPU, memory, reads, and writes.**
* The DTU-based purchasing model offers a **set of preconfigured bundles of compute resources and included storage** to drive different levels of application.
* **Basic**, **Standard**, and **Premium** service tiers for Azure SQL Database
* **https://keyvaulttestkv1.vault.azure.net/**

**Difference between Shared Access Signature URL and end point URL?**

**Sas is used to give access for individual component(datalake,blod,file,table) in resource or along with whole resource access in Microsoft explorer.**

**End point url is used for get access for storage into azure datafactory to fetch the folders form storage(resource).**

**Role base access control:**

**Owner:**

**Grants full access to manage all resources, including the ability to assign roles in Azure RBAC**.

**Contributer:**

* **Grants full access to manage all resources.**
* **But does not allow you to assign roles in Azure RBAC, manage assignments**

**Reader:**

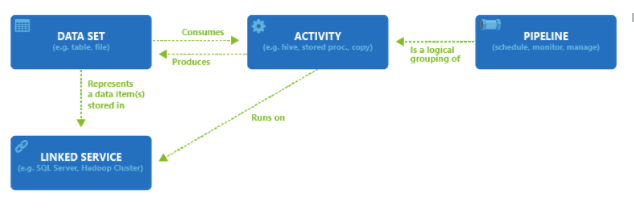
**View all resources, but does not allow you to make any changes**.

**RBAC administrator:**

* **Manage access to Azure resources by assigning roles using Azure RBAC.**
* **This role does not allow you to manage access using other ways, such as Azure Policy.**

**Azure Data Factory:**

* It is thecloud-based ETL and data integration service that allows you to create data-driven workflows for orchestrating data movement and transforming data at scale.
* Data orchestration involves **bringing data together from various sources**. These sources could be databases, files, APIs, or other data stores.



**Pipeline:**

* A pipeline is a logical grouping of activities that performs a unit of work. Together, the activities in a pipeline perform a task.
* the pipeline allows you to manage the activities as a set instead of managing each one individually.

**Activity:**

* Activities represent a processing step in a pipeline.
* **activity** defines a specific action to be performed within a data-driven workflow.
* Data Factory supports three types of activities: data movement activities, data transformation activities, and control activities.

| **Control activity** | **Description** |
| --- | --- |
| [Append Variable](https://learn.microsoft.com/en-us/azure/data-factory/control-flow-append-variable-activity) | Add a value to an existing array variable. |
| [Execute Pipeline](https://learn.microsoft.com/en-us/azure/data-factory/control-flow-execute-pipeline-activity) | Execute Pipeline activity allows a Data Factory or Synapse pipeline to invoke another pipeline. |
| [Filter](https://learn.microsoft.com/en-us/azure/data-factory/control-flow-filter-activity) | Apply a filter expression to an input array |
| [For Each](https://learn.microsoft.com/en-us/azure/data-factory/control-flow-for-each-activity): | ForEach Activity defines a repeating control flow in your pipeline. This activity is used to iterate over a collection and executes specified activities in a loop. The loop implementation of this activity is similar to the Foreach looping structure in programming languages. |
| [Get Metadata](https://learn.microsoft.com/en-us/azure/data-factory/control-flow-get-metadata-activity) | GetMetadata activity can be used to retrieve metadata of any data in a Data Factory or Synapse pipeline. |
| [If Condition Activity](https://learn.microsoft.com/en-us/azure/data-factory/control-flow-if-condition-activity) | The If Condition can be used to branch based on condition that evaluates to true or false. The If Condition activity provides the same functionality that an if statement provides in programming languages. It evaluates a set of activities when the condition evaluates to true and another set of activities when the condition evaluates to false. |
| [Lookup Activity](https://learn.microsoft.com/en-us/azure/data-factory/control-flow-lookup-activity) | Lookup Activity can be used to read or look up a record/ table name/ value from any external source. This output can further be referenced by succeeding activities. |
| [Set Variable](https://learn.microsoft.com/en-us/azure/data-factory/control-flow-set-variable-activity) | Set the value of an existing variable. |
| [Until Activity](https://learn.microsoft.com/en-us/azure/data-factory/control-flow-until-activity) | Implements Do-Until loop that is similar to Do-Until looping structure in programming languages. It executes a set of activities in a loop until the condition associated with the activity evaluates to true. You can specify a timeout value for the until activity. |
| [Validation Activity](https://learn.microsoft.com/en-us/azure/data-factory/control-flow-validation-activity) | Ensure a pipeline only continues execution if a reference dataset exists, meets a specified criterion, or a timeout has been reached. |
| [Wait Activity](https://learn.microsoft.com/en-us/azure/data-factory/control-flow-wait-activity) | When you use a Wait activity in a pipeline, the pipeline waits for the specified time before continuing with execution of subsequent activities. |
| [Web Activity](https://learn.microsoft.com/en-us/azure/data-factory/control-flow-web-activity) | Web Activity can be used to call a custom REST endpoint from a pipeline. You can pass datasets and linked services to be consumed and accessed by the activity. |
| [Webhook Activity](https://learn.microsoft.com/en-us/azure/data-factory/control-flow-webhook-activity) | Using the webhook activity, call an endpoint, and pass a callback URL. The pipeline run waits for the callback to be invoked before proceeding to the next activity. |

**Datasets:**

* Datasets represent data structures within the data stores, which simply point to or reference the data you want to use in your activities as inputs or outputs.

**Linked Services:**

* Linked Services provide the connection information that's needed for Data Factory to connect to external resources.
* It is bridge between your data stores and the data factory or workspace.

**Integration Runtime:**

* The Integration Runtime (IR) is the compute infrastructure used by Azure Data Factory and Azure Synapse pipelines.
* An integration runtime provides the bridge between the activity and linked Services. It's referenced by the linked service or activity and provides the compute environment where the activity either runs on or gets dispatched from.
* **data integration capabilities across different network environments:  
  Data Flow:**

Execute a [Data Flow](https://learn.microsoft.com/en-us/azure/data-factory/concepts-data-flow-overview) in a managed Azure compute environment.  
**Data movement:**

Copy data across data stores in a public or private networks (for both on-premises or virtual private networks)  
**Activity dispatch:**

Dispatch and monitor transformation activities running on a variety of compute services such as Azure Databricks, Azure HDInsight, ML Studio (classic), Azure SQL Database, SQL Server, and more.

**SSIS package execution:**

Natively execute SQL Server Integration Services (SSIS) packages in a managed Azure compute environment.

* **The three types of IR are:**
* Azure
* Self-hosted
* Azure-SSIS

**Azure Integration Runtime:**

* This is the default runtime and is fully managed by azure data factory.it is used to move data between cloud data stores.
* Run Data Flows in Azure
* Run copy activities between cloud data stores.

**Self-hosted IR network:**

* If you want to perform data integration securely in a private network environment that doesn't have a direct line-of-sight from the public cloud environment, you can install a self-hosted IR in your **on-premises environment behind a firewall, or inside a virtual private network.**
* This runtime is **installed on your on-premises environment and allows data factory to access data within your private network**.
* It is used when data source or destinations are located **on-premises(own serves or data centers) or in virtual private network(virtual machine).**
* Running copy activity between a cloud data stores and a data store in private network.

**Azure SSIS Integration Runtime:**

* It provide the infrastructure to execute SSIS packages in Azure.
* This runtime allows you to lift and shift your existing SQL Server integration services (ssis)packages to azure dat factory.

**Trigger:**

* It is mechanism that initiates the execution of a pipeline or pipeline run based on a defined schedule or event.

**TYPES OF TRIGGERS:**

**Schedule Trigger:**

Trigger runs a pipeline on a defined schedule such as day, hour, daily,, weekly, monthly

**Thumbling Window:**

Triggers a pipeline at regular intervals defined by a window size.

**Event-Based Trigger:**

Trigger initiates the execution of a pipeline in response to an event, such as the arrival of a file in azure blob storage or message from azure Event hub.

**Data movement trigger:**

Trigger monitor data changes in source dataset and trigger a pipeline run when new or update data is detected.

**Control flow:**

* Control flow refers to the orchestration and sequencing of activities within a data pipeline.
* It allows you to define the logical flow of data movement and data processing operations.
* Adf include activities such as data ingestion, transformation, loading, as well as branching, looping and conditional execution based on success or failure of preceding activity.

**Parameters:**

* Parameters are key-value pairs of read-only configuration.  Parameters are defined in the pipeline.
* The arguments for the defined parameters are passed during execution from the run context that was created by a trigger or a pipeline that was executed manually.

**Dataflow:**

Enable your data engineers to develop data transformation logic without needing to write code. Data flows are run as activities within Azure Data Factory pipelines that use scaled-out Apache Spark clusters.

1. **Synapse Workspace**:
   * A **Synapse workspace** serves as a **securable collaboration boundary** for cloud-based enterprise analytics in Azure.
   * It is deployed in a specific region and includes an associated **ADLS Gen2 account** (Azure Data Lake Storage Gen2) and file system (for storing temporary data).
   * Workspaces are organized under a **resource group**.
   * Within a Synapse workspace, you can perform analytics using both **SQL** and **Apache Spark**.
   * Resources for SQL and Spark analytics are organized into **SQL pools** and **Spark pools**, respectively.

Create table parameters\_ADLS\_to\_ADLS (

objectname varchar(200),

source\_container varchar(200),

source\_foldername varchar(200),

source\_filename varchar(200),

target\_container varchar(200),

target\_foldername varchar(200)

);

insert into [dbo].[parameters\_ADLS\_to\_ADLS]

(objectname,source\_container,source\_foldername,source\_filename,target\_container,target\_foldername)

values

('semicolonobject','refine','source\_1','TopMovies\_Part1.csv','dynamic','semicolon'),

('semicolonobject','refine','source\_1','TopMovies\_Part2.csv','dynamic','semicolon'),

('commaobject','refine','source\_1','TopMovies\_Part3.csv','dynamic','comma'),

('semicolonobjcet','mainfile','productfile','producct\_1.csv','dynamic','semicolon'),

('commaobjcet','mainfile','productfile','producct\_2.csv','dynamic','comma')

**----------------------------------------------------------------------------------------------------**

CREATE TABLE [dbo].[topmovies\_semicolon](

   [Index] [BIGINT] NULL,

   [MovieTitle] varchar(500) NULL

);

CREATE TABLE [dbo].[topmovies\_comma](

   [Index] [BIGINT] NULL,

   [MovieTitle] varchar(500) NULL

);

CREATE TABLE [dbo].[parameters\_ADF](

   [FolderName] varchar(100) NULL,

   [FileName] varchar(100) NULL,

   [SqlName] varchar(100) NULL

);

truncate table [dbo].[parameters\_ADF]

select \*from [dbo].[parameters\_ADF]

insert into [dbo].[parameters\_ADF] (FolderName,FileName,SqlName) values ('semicolon',';','topmovies\_semicolon');

insert into [dbo].[parameters\_ADF] (FolderName,FileName,SqlName) values ('comma',';','topmovies\_comma');

---------------------------------------------------------------------------------

CREATE TABLE [dbo].[parameters\_ADF\_different\_sqltables](

   [FolderName] varchar(100) NULL,

   [FileName] varchar(100) NULL,

   [SqlName] varchar(100) NULL

);

create table dbo.product(

ProductID Bigint NOT NULL,

Name varchar(200) NOT NULL,

ProductNumber varchar(200) NOT NULL,

Color varchar(200),

StandardCost float NOT NULL,

ListPrice float NOT NULL,

Size int ,

Weight float,

ProductCategoryID INT NOT NULL,

ProductModelID INT NOT NULL)

CREATE TABLE [dbo].[movies](

    [Index] int,

   [MovieTitle] varchar(200)

);

Incremental load from on premise to azure

create table data\_source\_table(

[personID] int ,

[Name] varchar(200),

[LastModifytime] data

);

insert into [dbo].[data\_source\_table]

(PersonID,Name,LastModifytime)

values

(1, 'joshi','9/1/2017 12:56:00 AM'),

(2, 'satya','9/2/2017 5:23:00 AM'),

(3, 'vardan','9/3/2017 2:36:00 AM'),

(4, 'adina','9/4/2017 3:21:00 AM'),

(5, 'ajsv','9/5/2017 8:06:00 AM');

create table watermarktable(

TableName varchar(255),

WatermarkValue datetime,

);

INSERT INTO watermarktable

VALUES ('data\_source\_table','1/1/2010 12:00:00 AM')

CREATE PROCEDURE usp\_write\_watermark @LastModifiedtime datetime, @TableName varchar(50)

AS

BEGIN

UPDATE watermarktable

SET [WatermarkValue] = @LastModifiedtime

WHERE [TableName] = @TableName

END

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| country | country\_code | continent | population | indicator | daily\_count | date | rate\_14\_day | source |

Create table DeathsInUKAndIndia (

[country]  varchar(255),

[country\_code] varchar(255),

[continent] Varchar(255),

[population] BigInt,

[indicator] Varchar(255),

[daily\_count] BigInt,

[date] date,

[rate\_14\_day] decimal(10,6),

[source] Varchar(255)

);

|  |  |  |  |
| --- | --- | --- | --- |
| daily\_count | date | rate\_14\_day | source |

Create table AllOverDeaths (

[country]  varchar(255),

[country\_code] varchar(255),

[continent] Varchar(255),

[population] BigInt,

[indicator] Varchar(255),

[daily\_count] BigInt,

[date] date,

[rate\_14\_day] decimal(10,6),

[source] Varchar(255)

);

**Create table CountryWiseResponseMeasure (**

**Country Varchar(255),**

**Response\_measure Varchar(255),**

**Change int,**

**date\_start Varchar(255),**

**date\_end Varchar(255)**

**);**

**Create table HospitalAdmissions (**

**Country Varchar(255),**

**Indicator Varchar(255),`**

**Date date,**

**Year\_Week Varchar(255),**

**Value decimal,**

**Source Varchar(255),**

**URL Varchar(255)**

**);**

**Create table Testing (**

**Country Varchar(255),**

**CountryCode Varchar(255),**

**Year\_Week Varchar(255),**

**New\_Cases BigInt,**

**Tests\_Done BigInt,**

**Population BigInt,**

**Testing\_Rate decimal,**

**Postivitu\_Rate decimal,**

**Testing\_Data\_Source Varchar(255)**

**);**

**Create table parameters (**

**ObjectName Varchar(255),**

**FolderName Varchar(255),**

**FileName Varchar(255),**

**SqlTableName Varchar(255)**

**);**

**Insert into dob.parameters**

**(ObjectName,FolderName,FileName,SqlTableName)**

**Values**

**(object\_case\_deaths\_uk\_ind\_only,Ingest,case\_deaths\_uk\_ind\_only.csv, DeathsInUKAndIndia),**

**(object\_cases\_deaths,Ingest,cases\_deaths.csv, AllOverDeaths),**

**(object\_country\_response,Ingest, country\_response.csv, CountryWiseResponseMeasure),**

**(object\_hospital\_admission,Ingest, hospital\_admissions.csv, HospitalAdmissions),**

**(object\_testing,Ingest,testing.csv, Testing)**

**CREATE PROCEDURE config\_srcfileNames**

**As**

**Begin**

**Select \* from dob.parameters;**

**End;**

PowerShell: <https://learn.microsoft.com/en-us/training/paths/powershell/>

Azure storage:<https://docs.microsoft.com/en-us/azure/storage/common/storage-introduction>

 Az104 :Azure administration

AZ500: Azure security

Az900:<https://learn.microsoft.com/en-us/credentials/certifications/exams/az-900/>

DP900: <https://learn.microsoft.com/en-us/credentials/certifications/exams/dp-900/>

DP203: <https://learn.microsoft.com/en-us/credentials/certifications/exams/dp-203/>

Github:<https://github.com/deeksharm/DP203> (for powershell-command to work with powershell)

[9:45 AM] Sharma, Deeksha (Cognizant)

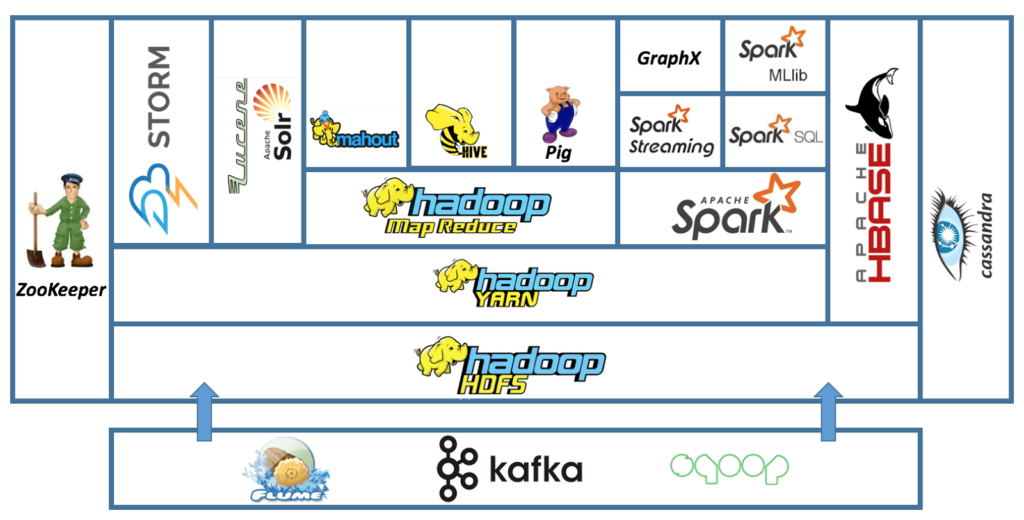
Azuredatafactory:<https://learn.microsoft.com/en-us/azure/data-factory/tutorial-incremental-copy-overview>

# Free credits :<https://www.microsoftazuresponsorships.com/Balance>

Databricks:<https://docs.databricks.com/en/getting-started/overview.html>

Policy in databricks:<https://learn.microsoft.com/en-us/azure/databricks/administration-guide/clusters/policies>

[Populate slowly changing dimensions in Azure Synapse Analytics pipelines - Training | Microsoft Learn](https://learn.microsoft.com/en-us/training/modules/populate-slowly-changing-dimensions-azure-synapse-analytics-pipelines/)



### A computer screen shot of a computer program Description automatically generated

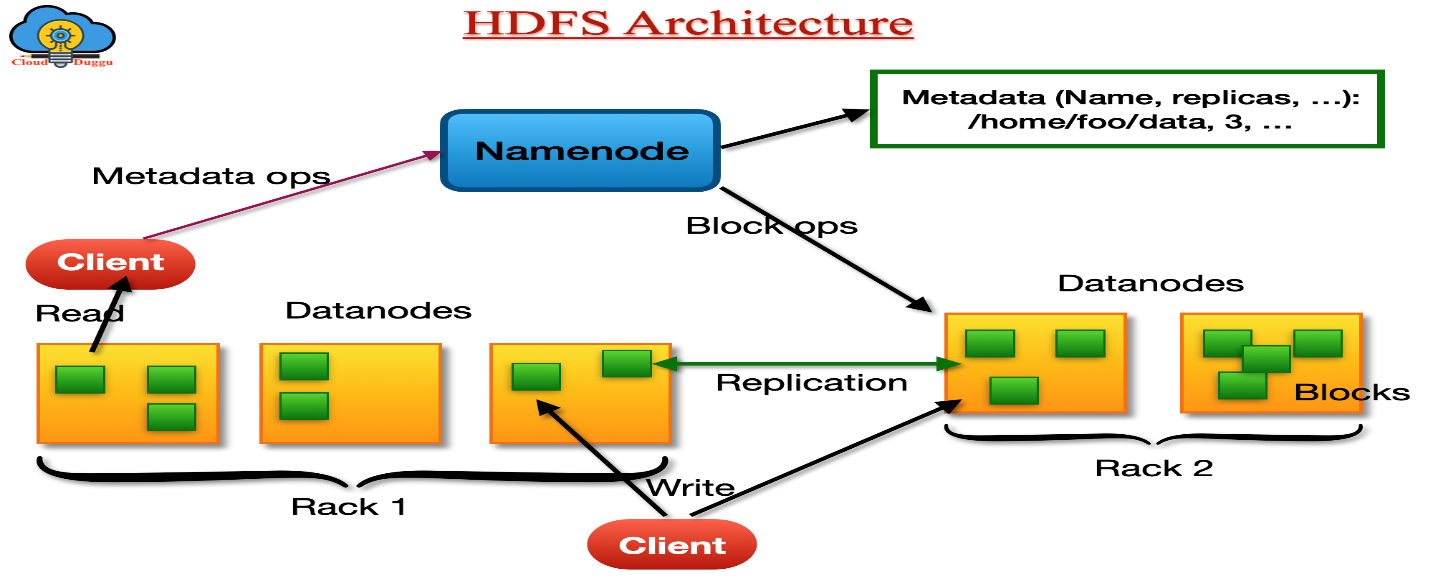
### YARN(Yet Another Resource Negotiator):

YARN is a Framework on which MapReduce works. YARN performs 2 operations that are Job scheduling and Resource Management. The Purpose of Job schedular is to divide a big task into small jobs so that each job can be assigned to various slaves in a Hadoop cluster and Processing can be Maximized. Job Scheduler also keeps track of which job is important, which job has more priority, dependencies between the jobs and all the other information like job timing, etc. And the use of Resource Manager is to manage all the resources that are made available for running a Hadoop cluster.

### A diagram of a software system Description automatically generated

### <https://intellipaat.com/blog/tutorial/hadoop-tutorial>

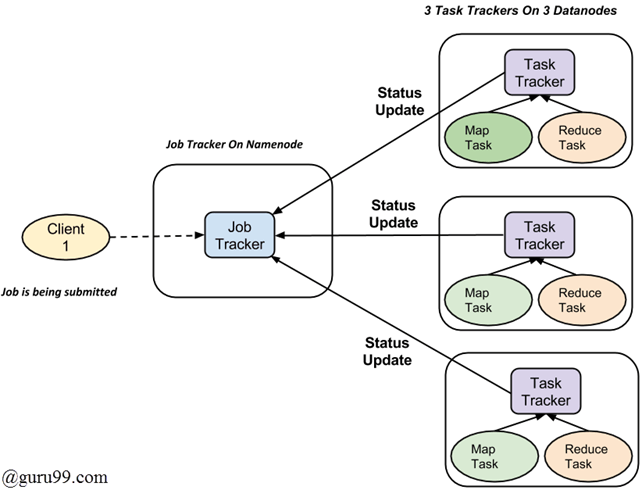
<https://www.geeksforgeeks.org/hadoop-yarn-architecture>



### <https://www.interviewbit.com/blog/hdfs-architecture>

<https://data-flair.training/blogs/hadoop-hdfs-architecture>

<https://www.cloudduggu.com/hadoop/hdfs>



<https://www.guru99.com/introduction-to-mapreduce.html>

* The **map phase** in Hadoop MapReduce is responsible for **transformation**.
* It takes a set of data and converts it into another set of data, where individual elements are broken down into tuples (key/value pairs).
* The **reduce phase** in Hadoop MapReduce is responsible for **aggregation**.
* It takes the output from the map phase (the intermediate key-value pairs) and combines those data tuples into a smaller set of tuples.