UNIT-4
CLOUD STORAGE AND DATABASE SERVICES

Prepared By:

D.R.Gandhi

Lecturer in I.T.

Dr. S & S. S. Ghandhy College Of Engineering & Technology ,Surat

Learning Outcomes

- Cloud Storage Solutions
  - Object storage, block storage, and file storage in the cloud
  - Data consistency and durability
- Cloud Databases
  - Types of cloud databases (SQL, NoSQL)
  - Data scaling and replication

Cloud Storage Solutions

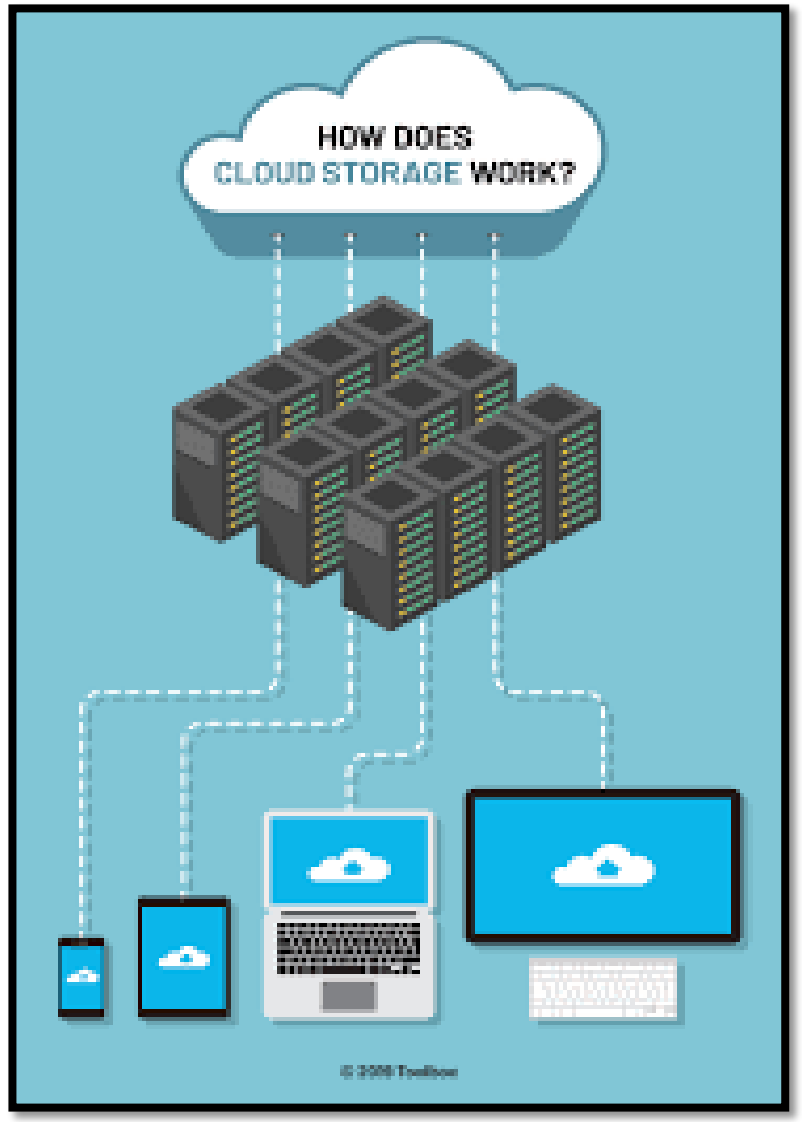
Cloud storage:-

- *Cloud storage is a data deposit model in which digital information such as documents, photos, videos and other forms of media are stored on virtual or cloud servers hosted by third parties.*
- It allows you to transfer data on an offsite storage system and access them whenever needed.
- *Cloud storage is a service model in which data is transmitted and stored on remote storage systems, where it is maintained, managed, backed up and made available to users over a network -- typically, the internet.*
- Users generally pay for their cloud data storage on a per-consumption, monthly rate.

Cloud Storage Solutions

Cloud storage:-

- **Cloud storage is a virtual locker where we can remotely stash any data.**
- When we upload a file to a cloud-based server like Google Drive, One Drive, or iCloud that file gets copied over the Internet into a data server that is cloud-based actual physical space where companies store files on multiple hard drives.



Cloud Storage Solutions

Features of Cloud storage:-

- It has a greater availability of resources.
- Easy maintenance is one of the key benefits of using Cloud computing.
- Cloud computing has a Large Network Access.
- It has an automatic system.
- Security is one of the major components and using cloud computing you can secure all over the networks.

Examples:-

- Google Drive, Dropbox, Microsoft One Drive, Amazon S3, and iCloud.
- These services provide users with a convenient, scalable, and accessible way to store and manage their data in the cloud, without the need to maintain physical storage infrastructure.

Cloud Storage Solutions

Advantages of Cloud storage:-

- **Scalability** – Capacity and storage can be expanded and performance can be enhanced.
- **Flexibility** – Data can be manipulated and scaled according to the rules.
- **Simpler Data Migrations** – As it can add and remove new and old data when required and eliminates disruptive data migrations.
- **Recovery** -In the event of a hard drive failure or other hardware malfunction, you can access your files on the cloud.

Cloud Storage Solutions

Disadvantages of Cloud storage:-

- Data centers require electricity and proper internet facility to operate their work, failing which system will not work properly.
- Support for cloud storage isn't the best, especially if you are using a free version of a cloud provider.
- When you use a cloud provider, your data is no longer on your physical storage.
- Cloud-based storage is dependent on having an internet connection. If you are on a slow network you may have issues accessing your storage.

Cloud Storage Solutions

Managed Cloud storage:-

- Managed cloud storage is a fully-managed service provided by cloud vendors, where the cloud provider takes care of the entire storage infrastructure, including hardware, software, and maintenance tasks.
- The key characteristics of managed cloud storage are:
- *Fully managed:* The cloud provider handles all aspects of storage management, including provisioning, scaling, backups, redundancy, and updates.
- *High availability and durability:* Managed cloud storage services are designed to provide high availability and durability, with data replicated across multiple data centers and geographic regions for fault tolerance and data protection.

Cloud Storage Solutions

Managed Cloud storage:-

- *Automatic scaling:* Storage capacity can be automatically scaled up or down based on your needs, without the need for manual intervention.
- *Pay-as-you-go pricing:* You pay only for the storage space you use, typically on a per-gigabyte or per-terabyte basis.
- **Examples:**
- *Amazon Elastic File System (EFS),*
- *Amazon Simple Storage Service (S3),*
- *Google Cloud Storage,*
- *Microsoft Azure Blob Storage*

Cloud Storage Solutions

Unmanaged Cloud storage:-

- Unmanaged cloud storage, also known as self-managed or bring-your-own-storage (BYOS), refers to storage services where the cloud provider offers the underlying infrastructure, but the customer is responsible for managing and maintaining the storage solution.
- In this model, the customer has more control and flexibility but also bears the responsibility for storage management tasks.
- Examples:
- *Amazon Elastic Block Store (EBS)*
- *Google Cloud Persistent Disk*
- *Microsoft Azure Managed Disks*

Cloud Storage Solutions

Unmanaged Cloud storage:-

- The key characteristics of unmanaged cloud storage are:
- *Self-managed:* The customer is responsible for provisioning, scaling, configuring, and maintaining the storage solution, including software updates, backups, and redundancy.
- *Lower costs:* Unmanaged cloud storage can be more cost-effective than managed services, as you only pay for the underlying infrastructure resources.
- *Customization and control:* Customers have greater control and flexibility over the storage configuration, allowing them to customize the solution to their specific needs.

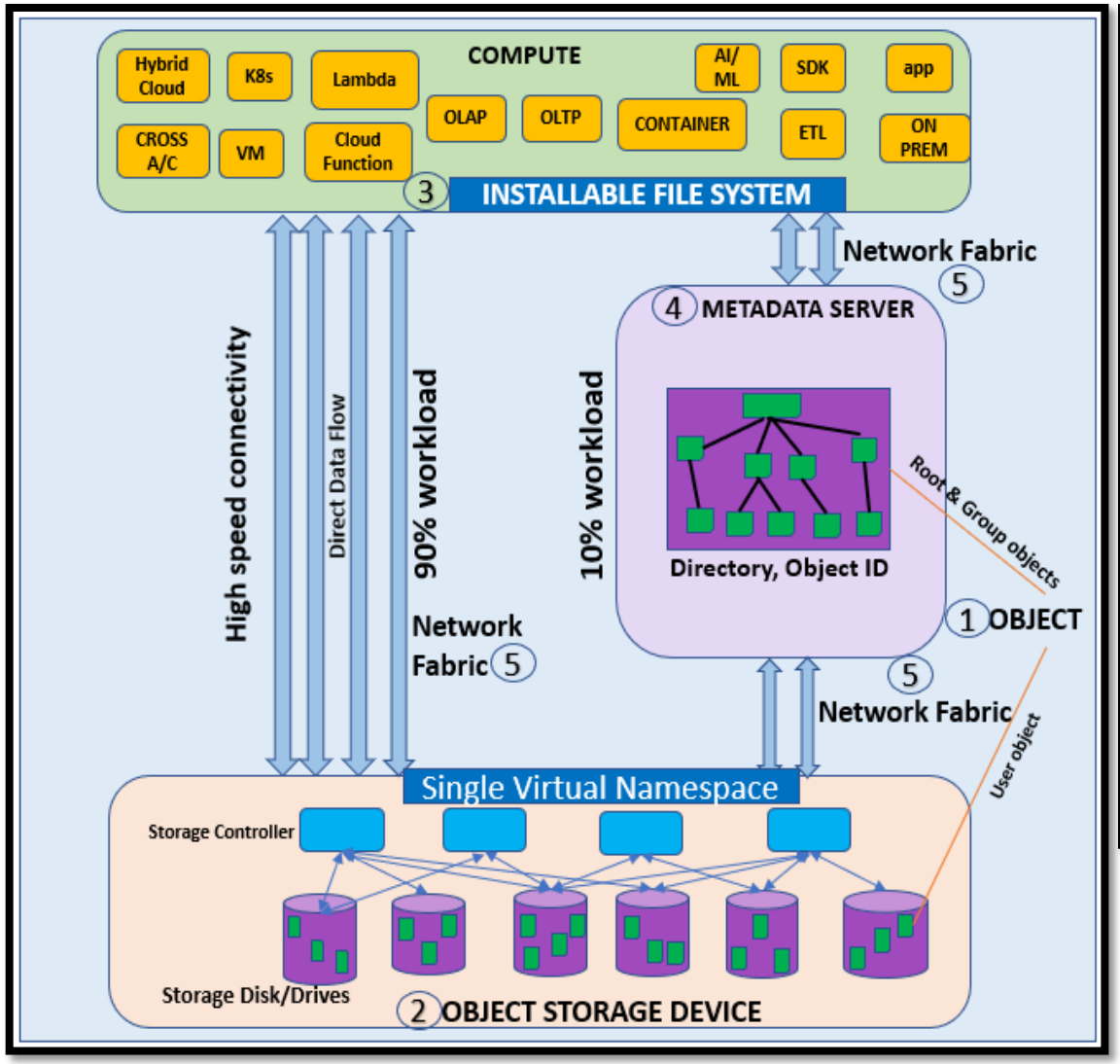
Cloud Storage Solutions

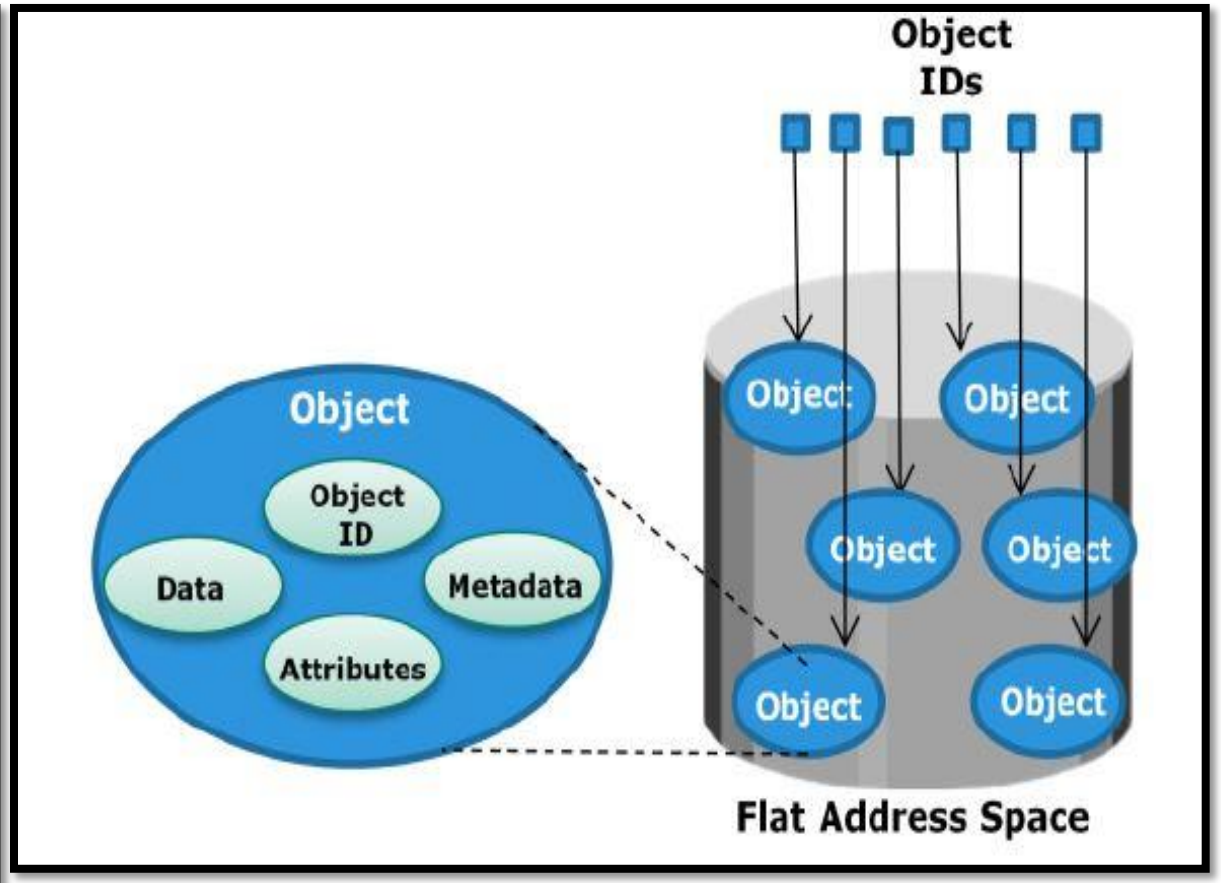
- There are three types of cloud storage solutions.
- ***Object Storage***
- ***Block Storage***
- ***File Storage***

Object Storage

- Object storage, also known as object-based storage, is a computer data storage architecture designed to handle large amounts of unstructured data.
- ***Object storage is a data storage architecture for storing unstructured data, which sections data into units—objects—and stores them in a structurally flat data environment.***
- Each object includes the *data*, *metadata*, and a *unique identifier* that applications can use for easy access and retrieval.
- Object storage does not store these objects in folders, as in a traditional file-based hierarchy — instead all objects are stored together in a single "*data lake" (also called a "data pool").* For this reason, object storage can store vast amounts of data very quickly.

Object Storage





Object Storage

- Each object typically includes the following components:
- **Object Data:** This is the actual data or file content being stored, such as documents, images, videos, or any other type of data.
- **Metadata:** Metadata is additional information about the object, such as the file name, size, date created, owner, access permissions, and custom metadata tags.
- **Unique Identifier:** Each object is assigned a unique identifier, often a long string of characters, which serves as the object's address or key within the object storage system.
- Popular object storage services in the cloud include Amazon Simple Storage Service (S3), Google Cloud Storage, Microsoft Azure Blob Storage, and IBM Cloud Object Storage.

Object Storage

- **Working of Object Storage:-**
- With object storage, the data blocks of a file are kept together as an object, together with its relevant metadata and a custom identifier, and placed in a flat data environment known as a storage pool.
- When you want to access data, object storage systems will use the unique identifier and the metadata to find the object you need, such as an image or audio file.
- You can locate and access objects using RESTful APIs, HTTP, and HTTPS to query object metadata.
- Since objects are stored in a **global storage pool**, it's fast and easy to locate the exact data you need.
- The **flat environment** enables you to scale quickly, even for petabyte or Exabyte loads.
- Storage pools can be spread across multiple object storage devices and geographical locations, allowing for unlimited scale.

Object Storage

- **Advantages of Object Storage:-**
- **Scalable** to any amount of data
- **Searchable** via metadata and unique identifiers
- **Not complex,** with no file hierarchies and no need for data to be reformatted or structured
- **Resilient,** as cloud storage providers have many server pools for failover
- **Low-cost** since customers pay only for the storage they need

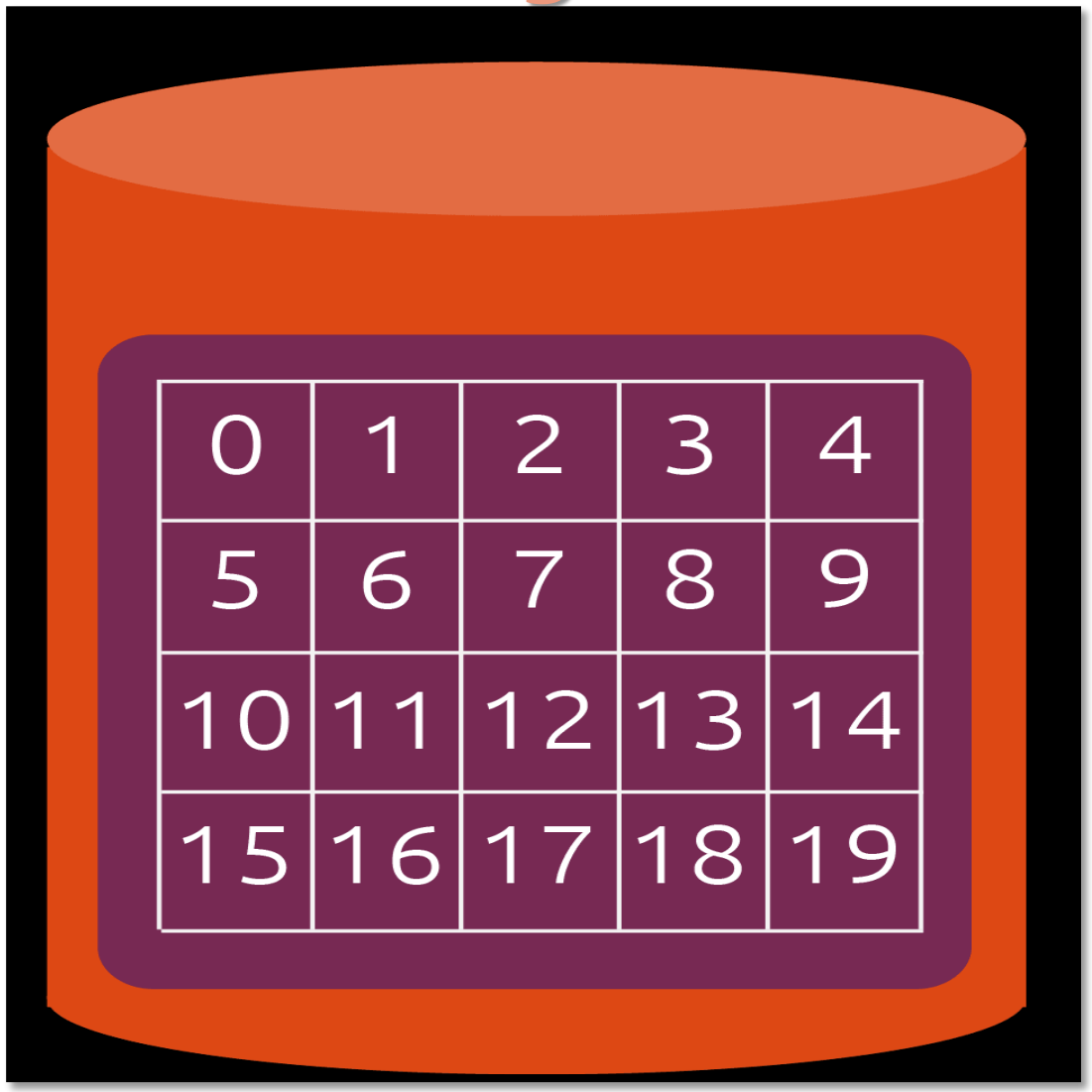
Object Storage

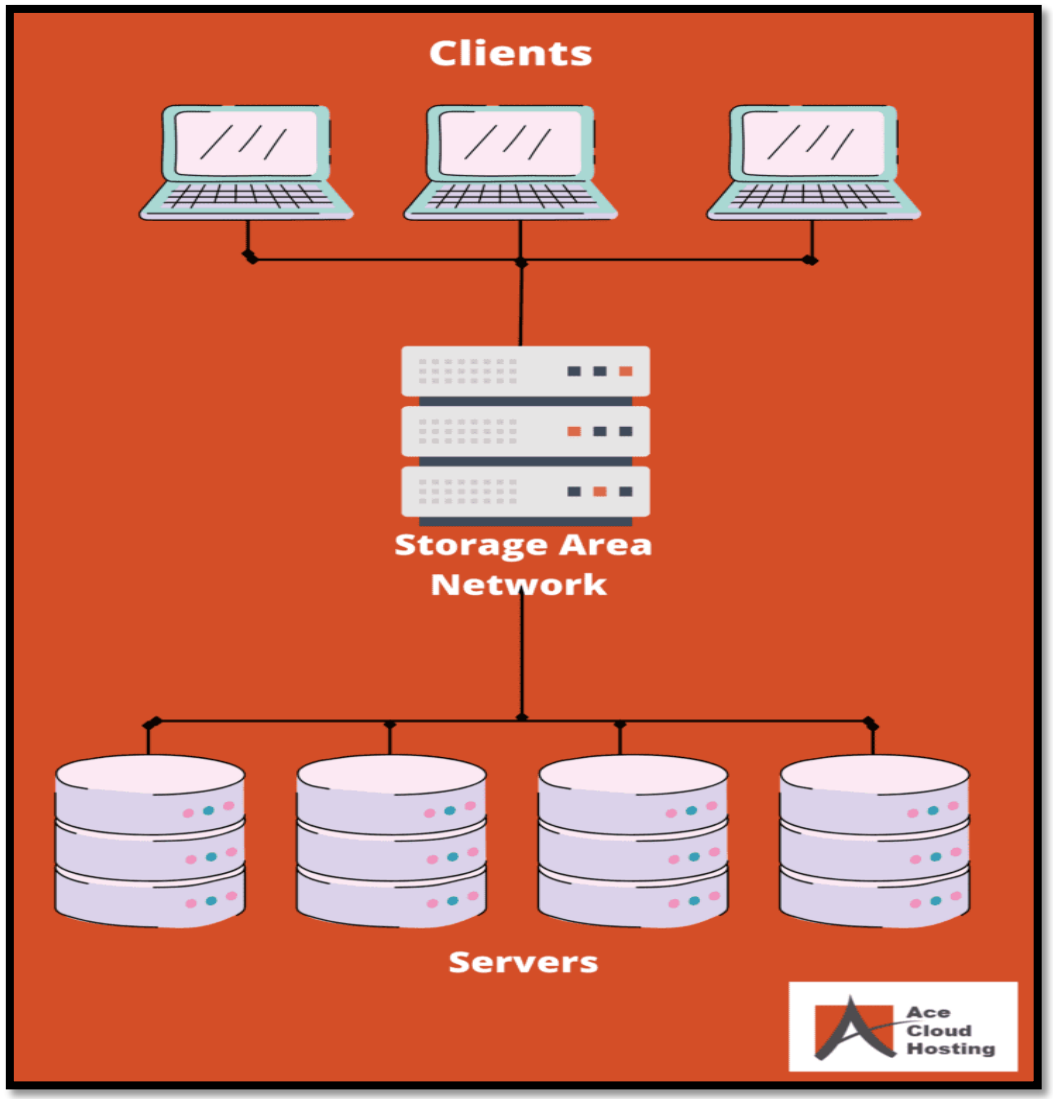
- **Disadvantages of Object Storage:-**
- Storage is tied *to one server* at a time
- Blocks and file systems have *limited metadata* about the blobs of information they're storing (creation date, owner, size). Any additional information about what you're storing will have to be handled at the application and database level, which adds complexity for a developer to handle
- You need to pay for all the block storage space you've allocated, even if you're not using all of it, so it makes it less *cost-efficient.*
- Block storage *requires more work from the developer* to set up as opposed to object storage.

Block Storage

- **Block storage is technology that controls data storage and storage devices.**
- *It takes any data, like a file or database entry, and divides it into blocks of equal sizes.*
- The block storage system then stores the data block on underlying physical storage in a manner that is optimized for *fast access* and *retrieval*.
- Developers often prefer block storage for applications that regularly need to *load data from the backend*.
- Block storage is *fast and scales up* extremely well.
- It also works well with several types of computing and networking models, including container computing, virtual machines, and storage area networks (SANs).

Block Storage





Block Storage

- *Block storage in the cloud refers to a type of storage service that provides raw, unformatted storage volumes that can be attached to virtual machines (VMs) or cloud instances.*
- In a block storage system, you can *break the data* into independent fixed-size blocks or pieces.
- Each block is an individual piece of data storage. A complete piece of information, such as a data file, is stored in multiple, non-sequential blocks.
- The block storage system does not maintain high-level metadata, such as file type, ownership, and timestamp.
- Developers must design a *data lookup table* in the application system to manage the storage of data into respective blocks.
- *Examples of block storages are Amazon Elastic Block Store (EBS), Google Cloud Persistent Disk, Microsoft Azure Managed Disks and IBM Cloud Block Storage.*

Block Storage

- **Working of Block Storage:-**
- **Data write :-** When an application that uses block storage writes data to the block storage database, instead of storing it as one file, it *divides the data into several sections — the "blocks."* These blocks do not have to be stored in any particular order.
- **Unique identifier:-** Each block has a unique identifier number that enables the application to find it later.
- **Data lookup table:-** These unique identifiers are stored in a data lookup table — *a format that allows the application* to easily find where each block is when it is needed.
- **Data read:-** Whenever data stored in the blocks is requested, the application consults the data lookup table to find where the requested data is stored. Usually, the requested data is spread out over multiple blocks. The application uses the identifiers from the table to retrieve the data, and it merges the disparate blocks back into their original form.

Block Storage

- **Advantages of Block Storage:-**

- **High efficiency:** Block storage's high IOPS(Input / Output Operation Per Second) and low latency make it ideal for applications that demand high performance.

- **Compatibility:** Block storage works across different operating systems and file systems, making it compatible for enterprises whatever their configuration and environment.

- **Flexibility:** With block storage, horizontal scaling is extremely flexible. Cluster nodes can be added as needed, allowing for greater overall storage capability.

- **Large file efficiency:** For large files, such as archives and video files, data must be completely overwritten when using file or object storage.

Block Storage

Disadvantages of Block Storage:-

- **Greater cost:** While block storage is easily scalable, it can also be expensive due to the cost of SANs. In addition, managing block storage requires more-specialized training for management and maintenance, increasing the overall expense.
- **Performance limitations:** With block storage, metadata is built in and hierarchical, and it is defined by the file system. Because data is broken up into blocks, searching for a complete file requires the proper identification of all its pieces.
- **Limited accessibility:** Block storage volumes can only be attached to and accessed by a single virtual machine or instance at a time, limiting the ability to share data across multiple instances or applications simultaneously.

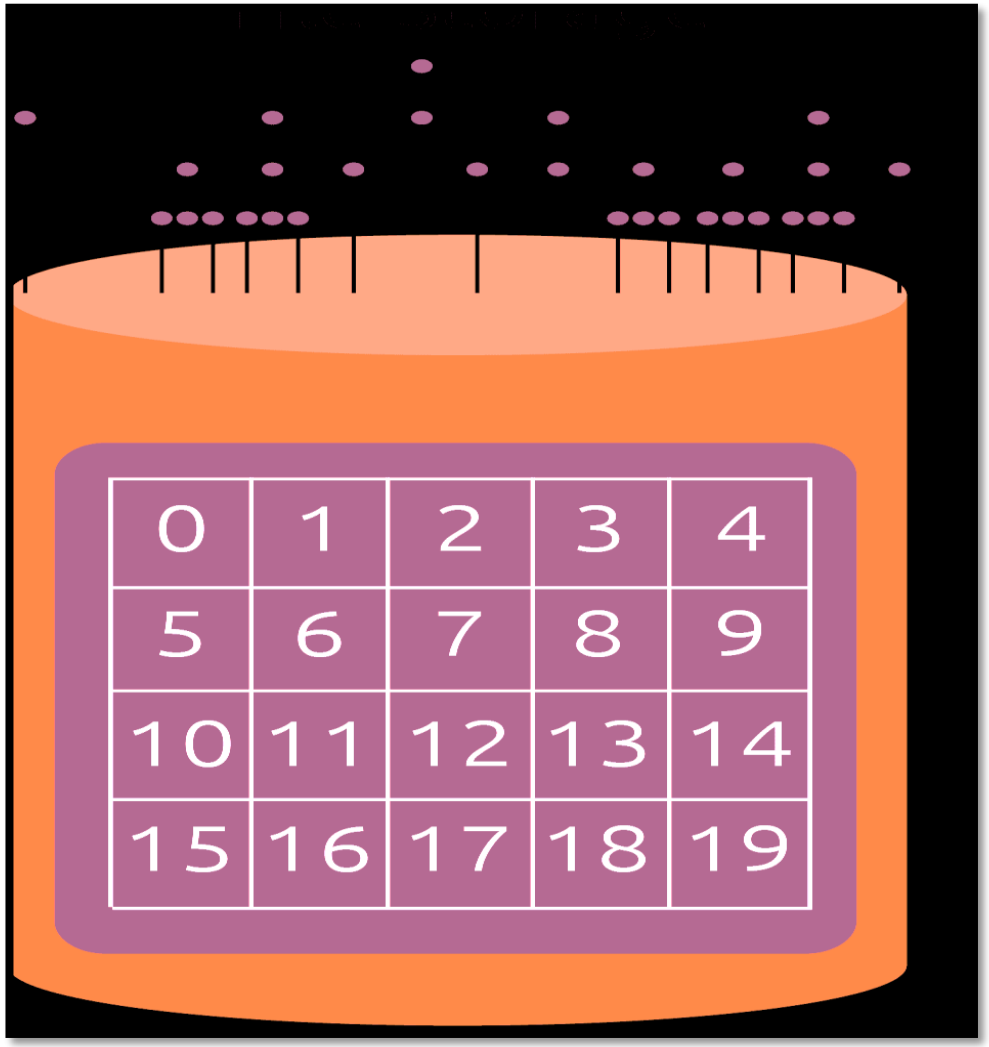
Block Storage

- **Use cases of of Block Storage:-**
- **Containers:** Block storage supports the use of container platforms such as Kubernetes, creating a block volume that enables persistent storage for the entire container. This allows for the clean management and migration of containers as needed.
- **Email servers:** Email servers can take advantage of block storage's flexibility and scalability. In fact, in the case of Microsoft Exchange, block storage is required due to the lack of support for network-attached storage.
- **Databases:** Block storage is fast, efficient, flexible, and scalable, with support for redundant volumes. This allows it to support databases, particularly those that handle a heavy volume of queries and where latency must be minimized.
- **Disaster recovery:** Block storage can be a redundant backup solution for near line storage and quick restoration, with data swiftly moved from backup to production through easy access.

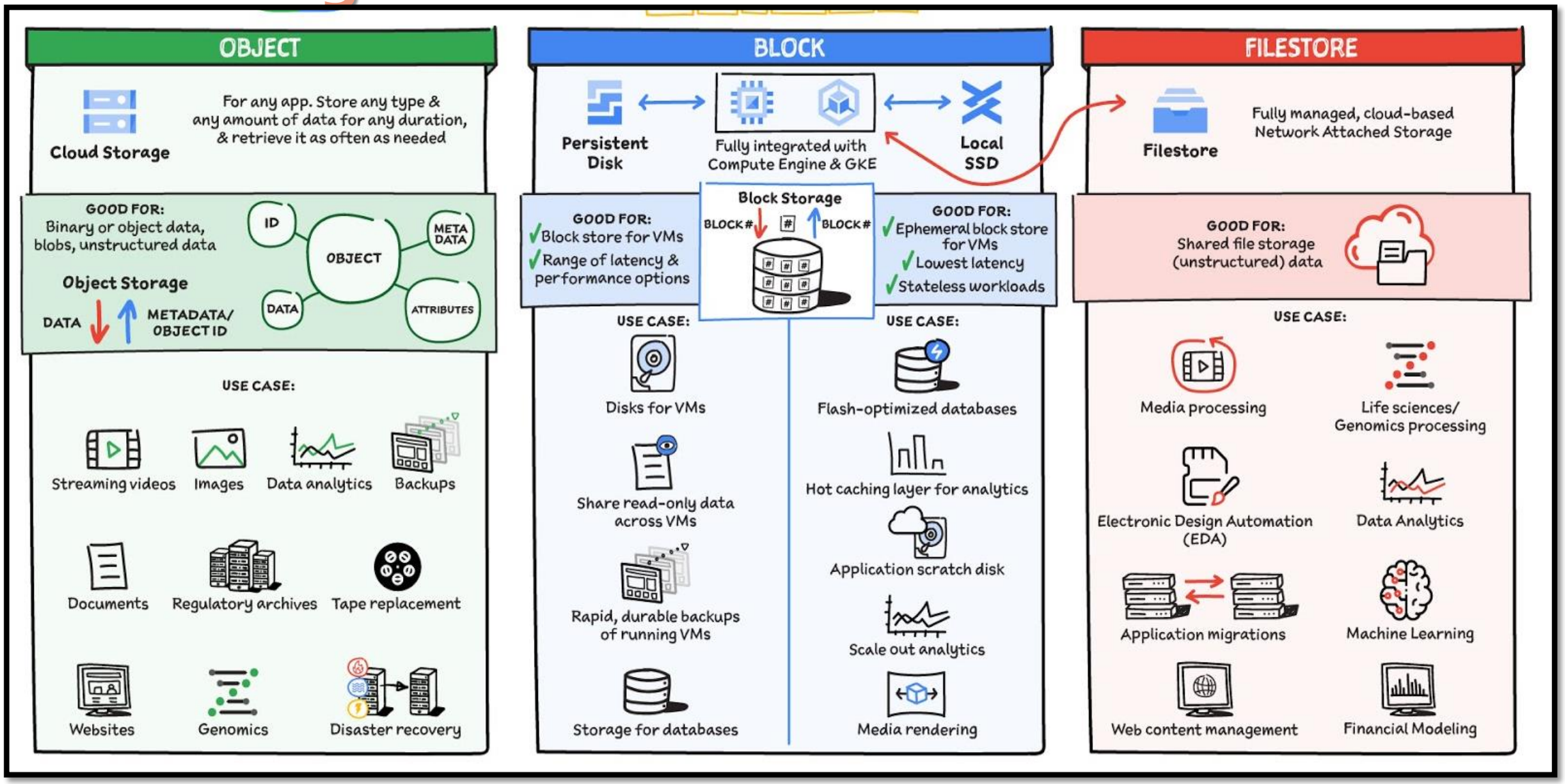
File Storage

- File storage in the cloud, also known as *cloud file storage* or *file sharing services*, is a type of cloud storage service that allows users to store and access files over the internet, similar to traditional network-attached storage (NAS) or file servers.
- ***It is the method of storing data in an ordered hierarchy in the format of file and folders, wherein files are placed inside folders and these folders are further placed in database, which keeps the hierarchy going.***
- This hierarchy can be shared and accessed between multiple servers and networks.
- Each file also has its own path through which it can be located easily and multiple users who have access to the server can access these files anytime.

File Storage



File Storage



File Storage

- ***Cloud file storage*** is a method for storing data in the cloud that provides servers and applications access to data through shared file systems. This compatibility makes cloud file storage ideal for workloads that rely on shared file systems and provides simple integration without code changes.
- **A cloud file system** is a hierarchical storage system in the cloud that provides shared access to file data. Users can create, delete, modify, read, and write files, as well as organize them logically in directory trees for intuitive access.
- **Cloud file sharing** is a service that provides simultaneous access for multiple users to a common set of files stored in the cloud. Security for online file storage is managed with user and group permissions so that administrators can control access to the shared file data.
- **Popular examples of cloud file storage services include Google Drive, Microsoft OneDrive, Dropbox, Box, and Amazon Elastic File System (EFS).**

File Storage

- **Working of File Storage:-**
- **Storage Infrastructure:** Cloud file storage services are provided by cloud service providers, such as Amazon Web Services (AWS), Microsoft Azure, and others. These providers maintain large data centers with vast amounts of storage capacity, numerous servers, hard drives, and other storage hardware.
- **File Upload and Access:** Users can upload their files (documents, media, etc.) to the cloud file storage service using web-based interfaces, desktop applications, mobile apps, or APIs. Once uploaded, the files are stored in the cloud provider's infrastructure.
- **File System Interface:** Cloud file storage services present a file system interface, similar to traditional file servers or network drives. Users can create folders, organize files, and perform operations like renaming, moving, or deleting files using familiar file management techniques.
- **Shared Access and Collaboration:** One of the key advantages of cloud file storage is the ability to share files and folders with other users or collaborate on documents in real-time.

File Storage

- **Advantages of File Storage:-**
- *Shared Access and Collaboration:* File storage services allow multiple users to access and collaborate on files simultaneously, enabling real-time collaboration and sharing of documents, presentations, and other files.
- *Familiar Interface:* File storage services provide a familiar file system interface, similar to traditional network drives or file servers, making it easier for users to manage and organize their files.
- *Cross-Platform Accessibility:* Cloud file storage services can be accessed from various devices and platforms, including desktops, laptops, tablets, and smartphones, enabling users to access their files from anywhere with an internet connection.
- *Versioning and Backup:* Many file storage services offer versioning capabilities, allowing users to access and restore previous versions of files. Additionally, they provide backup and recovery options to protect against data loss.

File Storage

- **Disadvantages of File Storage:-**
- *Limited Performance:* File storage services may not be suitable for applications or workloads that require high-performance or low-latency storage, as they are generally optimized for file sharing and collaboration rather than intensive read/write operations.
- *Internet Dependency:* Cloud file storage services rely on internet connectivity, which means that file access and synchronization may be affected by slow or unreliable internet connections.
- *Potential Security Risks:* Although cloud service providers implement various security measures, storing sensitive data in the cloud may raise concerns about data privacy and potential security breaches.

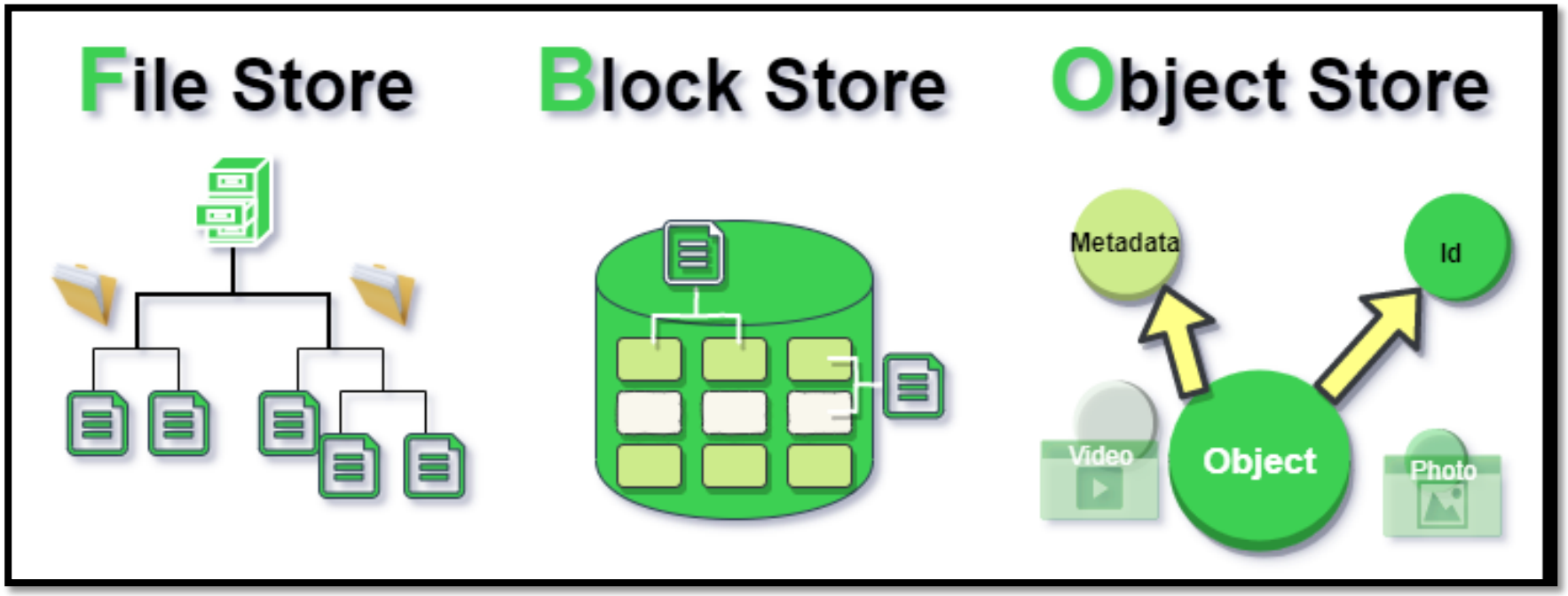
Object Storage Vs. Block Storage Vs. File Storage

Feature	Object Storage	Block Storage	File Storage
Data Structure	Flat namespace with unique identifiers for objects	Raw block-level access to storage volumes	Hierarchical file system with directories and files
Access Method	RESTful APIs, typically HTTP/HTTPS	Block device interface, e.g., iSCSI, Fibre Channel	File system protocols, e.g., NFS, SMB/CIFS
Use Cases	Large unstructured data, backups, archives, static website hosting, media storage	Databases, virtual machine storage, general-purpose storage	User home directories, shared files, application data
Scalability	Highly scalable, designed for cloud-scale storage	Scales vertically by adding more volumes	Scales horizontally by adding more file servers
Examples	Amazon S3, Azure Blob Storage, Google Cloud Storage	Amazon EBS, Azure Managed Disks, Google Persistent Disk	NFS, SMB/CIFS file shares, distributed file systems

Object Storage Vs. Block Storage Vs. File Storage

Feature	Object Storage	Block Storage	File Storage
File management	Store files as objects. Accessing files in object storage with existing applications requires new code and the use of APIs.	Can store files but requires additional budget and management resources to support files on block storage.	Supports common file-level protocols and permissions models. Usable by applications configured to work with shared file storage.
Metadata management	Can store unlimited metadata for any object. Define custom metadata fields.	Uses very little associated metadata.	Stores limited metadata relevant to files only.
Performance	Stores unlimited data with minimal latency.	High-performance, low latency, and rapid data transfer.	Offers high performance for shared file access.
Physical storage	Distributed across multiple storage nodes.	Distributed across SSDs and HDDs.	On-premises NAS servers or over underlying physical block storage.

Object Storage Vs. Block Storage Vs. File Storage



Data consistency and durability

- **Data Consistency:-**
- **Data consistency in the cloud refers to the accuracy and integrity of data stored across multiple distributed systems or nodes within a cloud computing environment.**
- In a cloud setup, data is often replicated and distributed across various locations or nodes to ensure high availability, scalability, and fault tolerance.
- However, maintaining data consistency becomes a crucial challenge in such distributed systems.
- Example:
- In shopping app , if you purchase anything then the app immediately shows the correct information everywhere.

Data consistency and durability

- **Data Consistency:-**
- In cloud computing environments, data consistency is typically achieved through one of two main models:
- *strong consistency*
- *eventual consistency*
- These models represent different approaches to maintaining data integrity and synchronization across distributed systems.
- Real world applications:
- **Financial Transactions**
- **E-commerce Inventory Management**
- **Collaborative Document Editing**

Data consistency and durability

- **Data Consistency:-**
- **Strong Consistency:**
- Strong consistency, also known as strict consistency or synchronous replication, ensures that data is always consistent across all nodes or replicas in the system.
- This means that any read operation will always return the most recent, updated value as seen by the latest write operation, regardless of which node or replica is accessed.
- In a strongly consistent system, write operations are propagated synchronously to all replicas before they are acknowledged as successful.
- This ensures that all replicas have the same view of the data at any given point in time.

Data consistency and durability

- **Data Consistency:-**
- **Eventual Consistency:**
- Eventual consistency is a more relaxed consistency model that prioritizes availability and partition tolerance over strict consistency.
- In an eventually consistent system, changes to data are propagated asynchronously to replicas, which means that different nodes or replicas may have temporarily inconsistent views of the data.
- However, the system is designed to eventually converge to a consistent state, where all replicas will eventually become consistent with each other after the updates have been fully propagated and reconciled.

Data consistency and durability

- **Advantages of Data Consistency:-**
- **Data Integrity:** Ensuring data consistency helps maintain the accuracy and integrity of data across multiple nodes or replicas in the cloud. This is crucial for applications that require reliable and trustworthy data.
- **Improved Data Quality:** Consistent data reduces the risk of data corruption, duplication, or inconsistencies, which can lead to better data quality and decision-making processes.
- **Simplified Data Management:** By maintaining data consistency, organizations can simplify data management processes, such as backups, migrations, and synchronization across different cloud environments or regions.
- **Enhanced User Experience:** Applications that rely on consistent data can provide a better user experience by ensuring that users access the same, up-to-date information regardless of their location or the node they interact with.

Data consistency and durability

- **Disadvantages of Data Consistency:-**
- **Performance Impact:** Achieving strong data consistency can often come at the cost of performance. Techniques like distributed transactions, quorum-based protocols, or synchronous replication can introduce latency and overhead, potentially impacting application performance.
- **Availability Trade-offs:** According to the CAP theorem (Consistency, Availability, Partition Tolerance), in the presence of network partitions, distributed systems must choose between consistency and availability. Prioritizing consistency may sacrifice availability in certain scenarios.
- **Increased Complexity:** Implementing and maintaining data consistency in distributed cloud environments can be complex, requiring specialized techniques, protocols, and tools. This complexity can increase development and operational costs.

Data consistency and durability

- **Data Durability:-**
- **Data durability in the cloud refers to the ability of cloud storage systems to ensure that data remains intact, accessible, and recoverable even in the face of various failures, disruptions, or unforeseen events.**
- Cloud service providers employ various techniques and strategies to achieve data durability and provide reliable storage services to their customers.
- Example:
- IF you save any document in cloud service, it ensures that your document is safe even if there is a power outage or server failure.

Data consistency and durability

- **Data Durability:-**
- **Replication and Redundancy:**
- Cloud storage providers replicate data across multiple storage nodes, data centers, and even geographic regions to ensure redundancy and fault tolerance.
- If one storage node or data center fails, the data can be retrieved from other replicas, ensuring data durability and availability.
- **Continuous Data Protection:**
- Cloud storage systems often employ continuous data protection mechanisms, such as write-ahead logging or journaling, to ensure data consistency and recoverability in the event of system failures or crashes.
- These techniques capture all data modifications and allow for point-in-time recovery or roll-forward recovery, minimizing data loss.

Data consistency and durability

- **Advantages of Data Durability:-**
- **High availability and fault tolerance:** Cloud providers implement redundancy and replication techniques, ensuring data remains accessible even in the event of hardware failures, network outages, or data center disruptions.
- **Disaster recovery and business continuity:** By storing data in multiple geographic locations and providing backup and recovery services, cloud providers enable organizations to recover from disasters and minimize downtime.
- **Scalability and elasticity:** Cloud storage systems can easily scale up or down to accommodate changing data volumes, ensuring data durability without compromising performance or capacity.
- **Automatic data repair and healing:** Cloud storage systems often incorporate self-healing mechanisms that automatically detect and repair corrupted or missing data, reducing the need for manual intervention and minimizing data loss.

Data consistency and durability

- **Disadvantages of Data Durability:-**

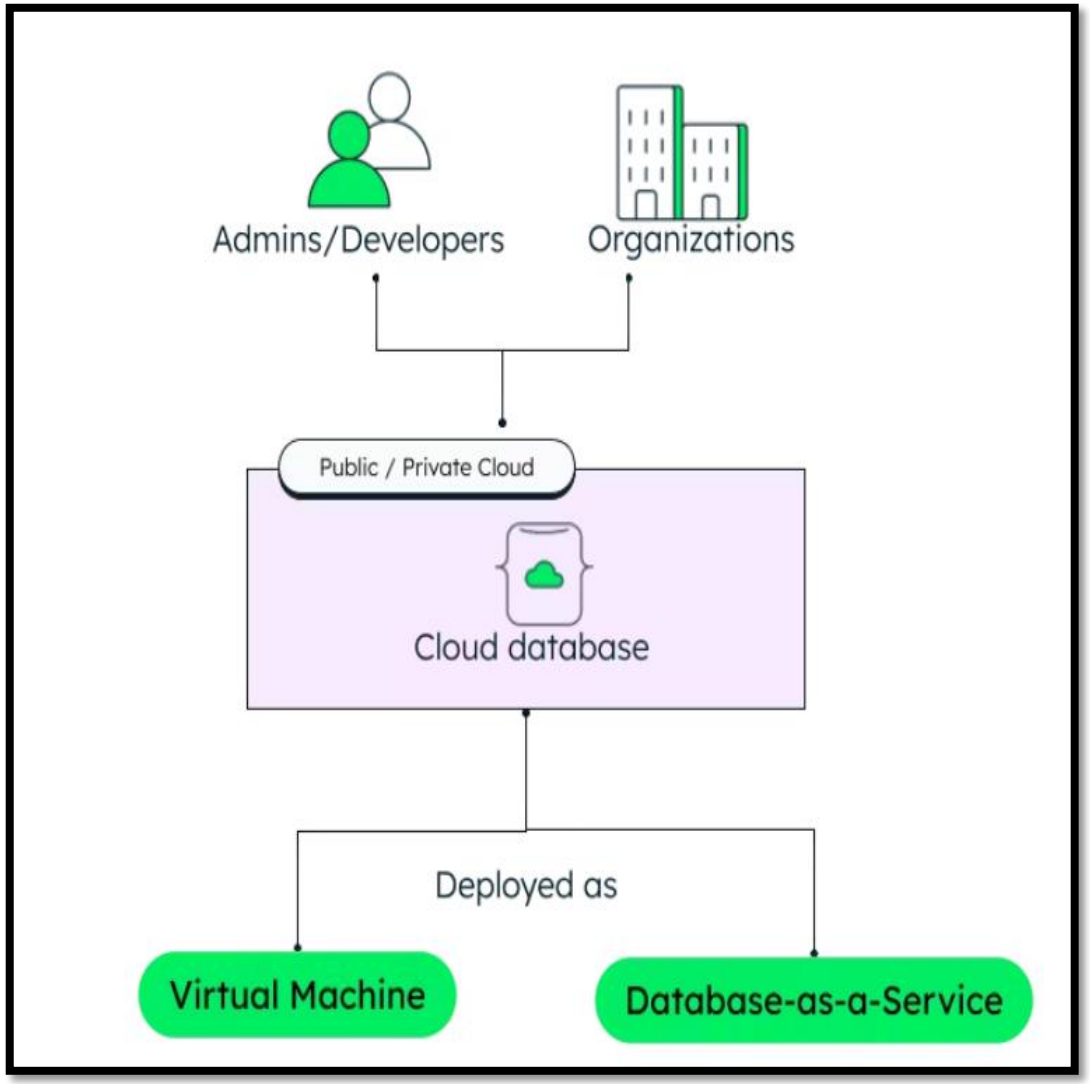
- **Vendor lock-in:** Organizations may become dependent on a specific cloud provider's data durability mechanisms, making it challenging to migrate data or switch providers due to potential compatibility issues or data transfer costs.

- **Potential security risks:** While cloud providers implement robust security measures, the shared nature of cloud infrastructure can introduce potential security risks, such as data breaches or unauthorized access, which could compromise data durability.

- **Network dependencies:** Cloud storage relies heavily on network connectivity, and any network disruptions or latency issues could impact data durability and availability, particularly for synchronous replication or real-time data processing.

Cloud Database

- *A cloud database is a database built to run in a public or hybrid cloud environment to help organize, store, and manage data within an organization.*
- Cloud databases can be offered as a managed *database-as-a-service (DBaaS)* or deployed on a cloud-based *virtual machine (VM)* and self-managed by an in-house IT team.

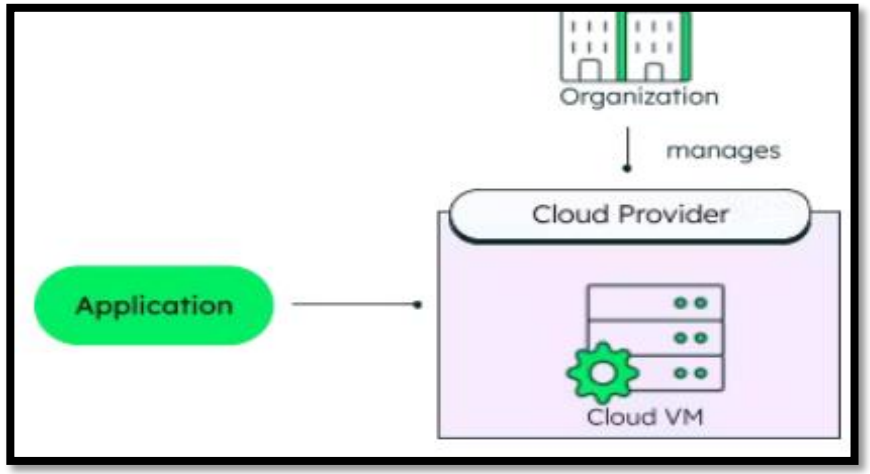


Cloud Database

- *A cloud database is a database service that runs on a cloud computing platform, provided as a cloud service by a third-party cloud provider.*
- Instead of hosting and managing a database on-premises using your own hardware and infrastructure, you can leverage a cloud database offered by companies like Amazon Web Services (AWS), Microsoft Azure, Google Cloud Platform, and others.
- There are two primary cloud database deployment models.
- *Traditional database(Self Managed)*
- *Database as a service (DBaaS)*

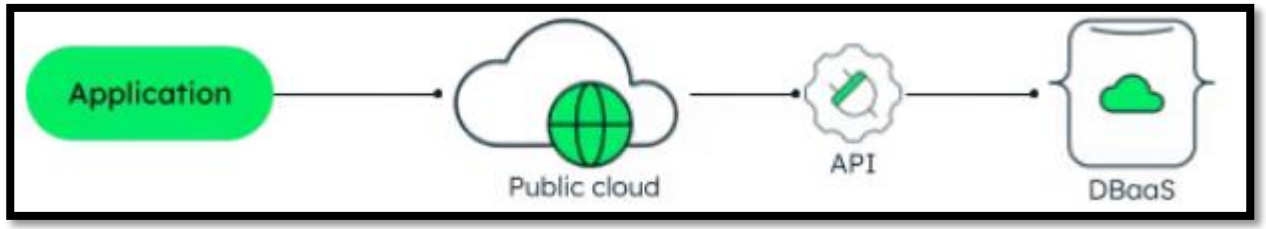
Cloud Database

- *Traditional database(Self Managed):-*
- It is very similar to an onsite, in-house managed database—except for infrastructure provisioning.
- In this case, an organization purchases virtual machine space from a cloud services provider, and the database is deployed to the cloud.
- The organization's developers use a DevOps model or traditional IT staff to control the database.
- The organization is responsible for oversight and database management.

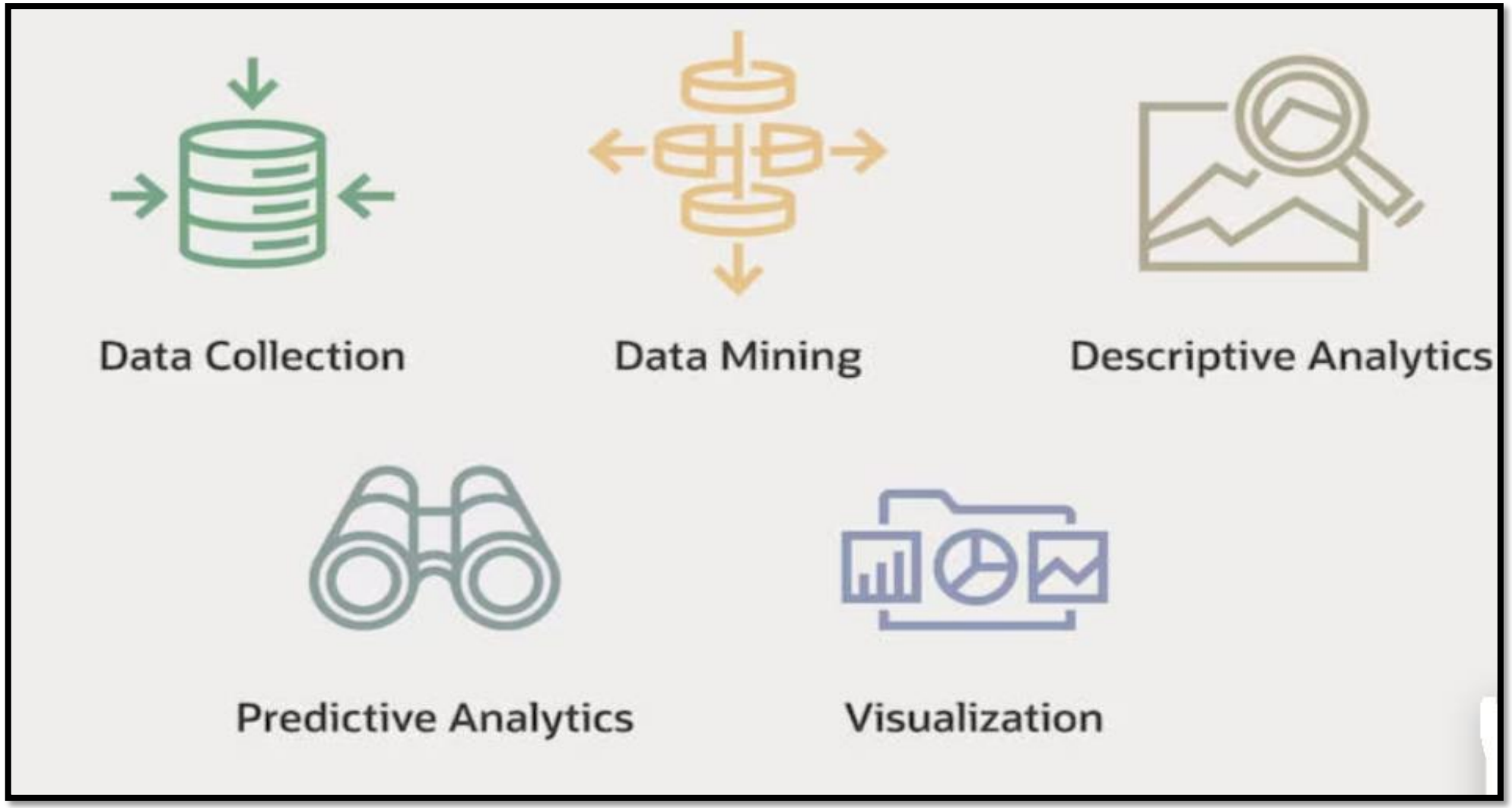


Cloud Database

- *Database as a service (DBaaS):-*
- In which an organization contracts with a cloud services provider through a fee-based subscription service.
- The service provider offers a variety of real-time operational, maintenance, administrative, and database management tasks to the end user.
- The database runs on the service provider's infrastructure.
- This usage model typically includes automation in the areas of provisioning, backup, scaling, high availability, security, patching, and health monitoring.



Cloud Database



Cloud Database

- **Advantages of Cloud Database:-**
- *Access:* Ease of mobile data access is greatly enhanced via cloud access.
- *Scalability:* The rapid scalability of cloud databases can easily accommodate data asset increases and user base growth.
- *Performance:* Automatic alerts to performance issues enable optimization of indexes and access patterns in order to hit performance targets.
- *Reliability:* Cloud databases are usually replicated and backed up automatically, so single-point-of-failure concerns are minimized.

Cloud Database

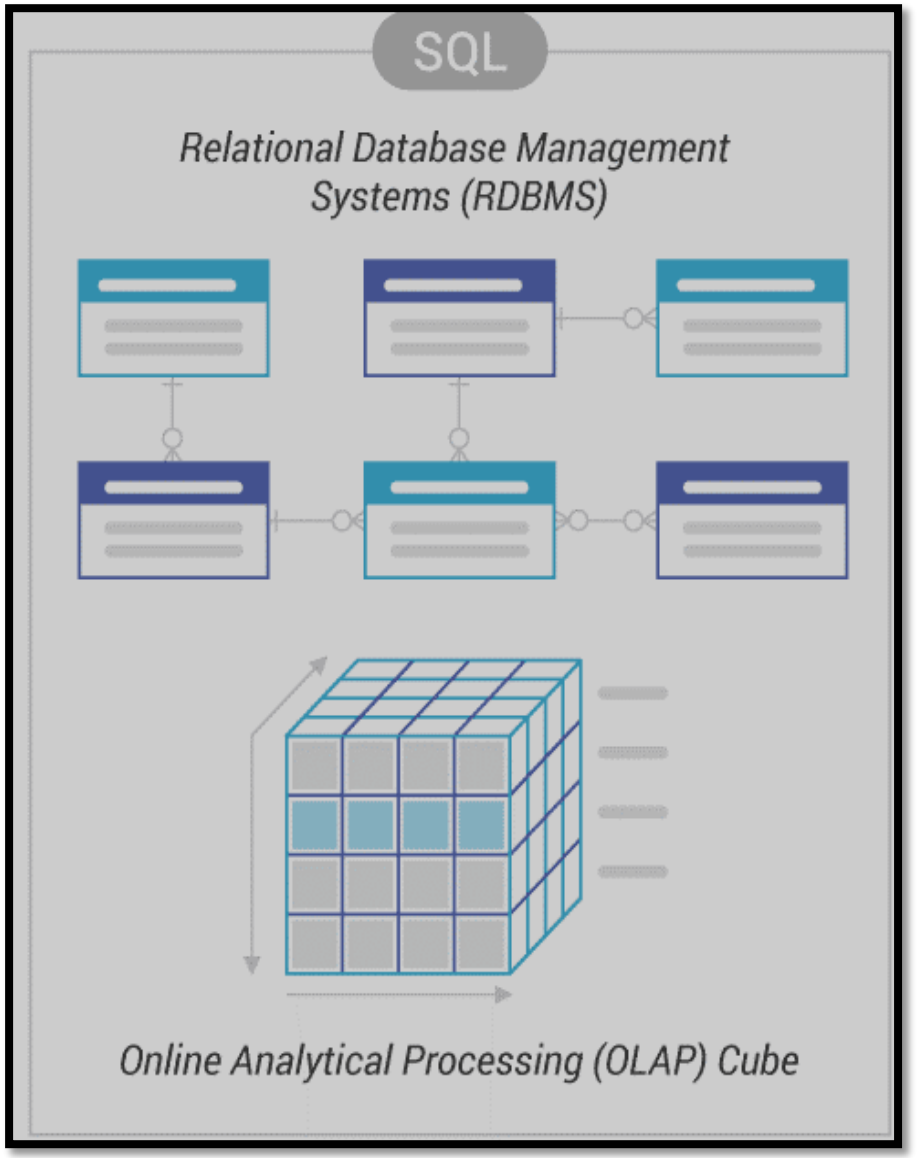
- **Disadvantages of Cloud Database:-**
- *Vendor Lock-In:* Migrating from one cloud provider to another can be challenging due to the lack of standardization and potential vendor lock-in, making it difficult to switch providers if needed.
- *Network Dependency:* Cloud databases rely on a stable and high-speed internet connection to function properly. Network disruptions or latency issues can impact performance and availability.
- *Security Concerns:* While cloud providers offer robust security measures, organizations may have concerns about potential data breaches, unauthorized access, or other security risks associated with storing data in a shared cloud environment.
- *Limited Control and Customization:* Cloud databases are managed by the cloud provider, which may limit organizations' ability to customize or fine-tune the database configurations to their specific needs.

Types of Cloud Databases

- There are 5 types of cloud databases.
- ***Relational Databases (SQL Databases)***
- ***NoSQL Databases***
- ***In-Memory Databases***
- ***Data Warehouses***

Types of Cloud Databases

- ***Relational Databases (SQL Databases):-***
- *SQL cloud databases are based on the relational database model, which organizes data into tables with rows and columns, enforcing relationships and data integrity through schemas and constraints.*
- These databases use *Structured Query Language (SQL)* for defining and manipulating data.
- Relational cloud databases are ideal for structured data, such as retail analytics data related to transactions, inventory, or customer information.



Types of Cloud Databases

- ***Characteristics Relational Databases (SQL Databases):-***
- Structured data model with predefined schemas.
- Support for *ACID (Atomicity, Consistency, Isolation, Durability) properties.*
- Use *SQL* for querying and manipulating data.
- Suitable for complex transactions and relationships.
- Vertically scalable (scale up/down resources for a single instance).
- SQL cloud databases are well-suited for applications that require strict data consistency, complex queries, and transactions, such as e-commerce platforms, banking systems, and enterprise resource planning (ERP) systems.

Types of Cloud Databases

- *Examples of Relational Databases:*
- *Amazon Relational Database Service (RDS)*
- *Microsoft Azure SQL Database*
- *Google Cloud SQL*
- *IBM Db2 on Cloud*

Types of Cloud Databases

- ***Popular Relational Databases (SQL Databases):-***
- **Amazon Relational Database Service (RDS):**
- *Provider: Amazon Web Services (AWS)*
- *Description:-*
- Supports multiple database engines: MySQL, MariaDB, Oracle, and SQL Server.
- Offers features like read replicas, multi-AZ deployments, and automated backups.
- Integrates well with other AWS services like EC2, Lambda, and S3.
- Pricing is based on instance type, storage, and additional features used.

Types of Cloud Databases

- ***Popular Relational Databases (SQL Databases):-***
- **Microsoft Azure SQL Database:**
- *Provider: Microsoft Azure*
- *Description:-*
- Based on the SQL Server engine, with compatibility for on-premises SQL Server databases.
- Offers features like automatic tuning, threat detection, and SQL Data Sync.
- Integrates well with other Azure services like App Service, Functions, and Storage.
- Pricing is based on the service tier (Basic, Standard, or Premium) and the compute resources used.

Types of Cloud Databases

- ***Popular Relational Databases (SQL Databases):-***
- **Google Cloud SQL:**
- *Provider: Google Cloud Platform(GCP)*
- *Description:-*
- Supports MySQL and PostgreSQL database engines.
- Offers features like automated backups, read replicas, and failover replicas.
- Integrates well with other Google Cloud services like App Engine, Cloud Functions, and Cloud Storage.
- Pricing is based on the instance type, storage, and additional features used.

Types of Cloud Databases

- ***Popular Relational Databases (SQL Databases):-***
- **IBM Db2 on Cloud:**
- *Provider: IBM Cloud*
- *Description:-*
- Based on the IBM Db2 database engine, with compatibility for on-premises Db2 databases.
- Offers features like high availability, workload management, and data partitioning.
- Integrates with other IBM Cloud services like Watson Studio and Cloud Object Storage.
- Pricing is based on the deployment model (Virtual Private Cloud or Baremetal), compute resources, and storage used.

Types of Cloud Databases

- ***Advantages of Relational Databases (SQL Databases):-***
- *Data Integrity:* Relational databases enforce data integrity through the use of schemas, constraints, and transactions, ensuring that data remains consistent and accurate.
- *ACID Compliance:* SQL databases adhere to the ACID (Atomicity, Consistency, Isolation, Durability) properties, which guarantee reliable and consistent data operations, even in the event of failures or concurrent transactions.
- *Structured Query Language (SQL):* SQL is a standardized and widely-adopted language for managing and querying data, making it easier to work with and integrate with various applications and tools.
- *Relationships and Joins:* Relational databases allow for defining and querying relationships between data through joins, enabling complex data modeling and analysis.

Types of Cloud Databases

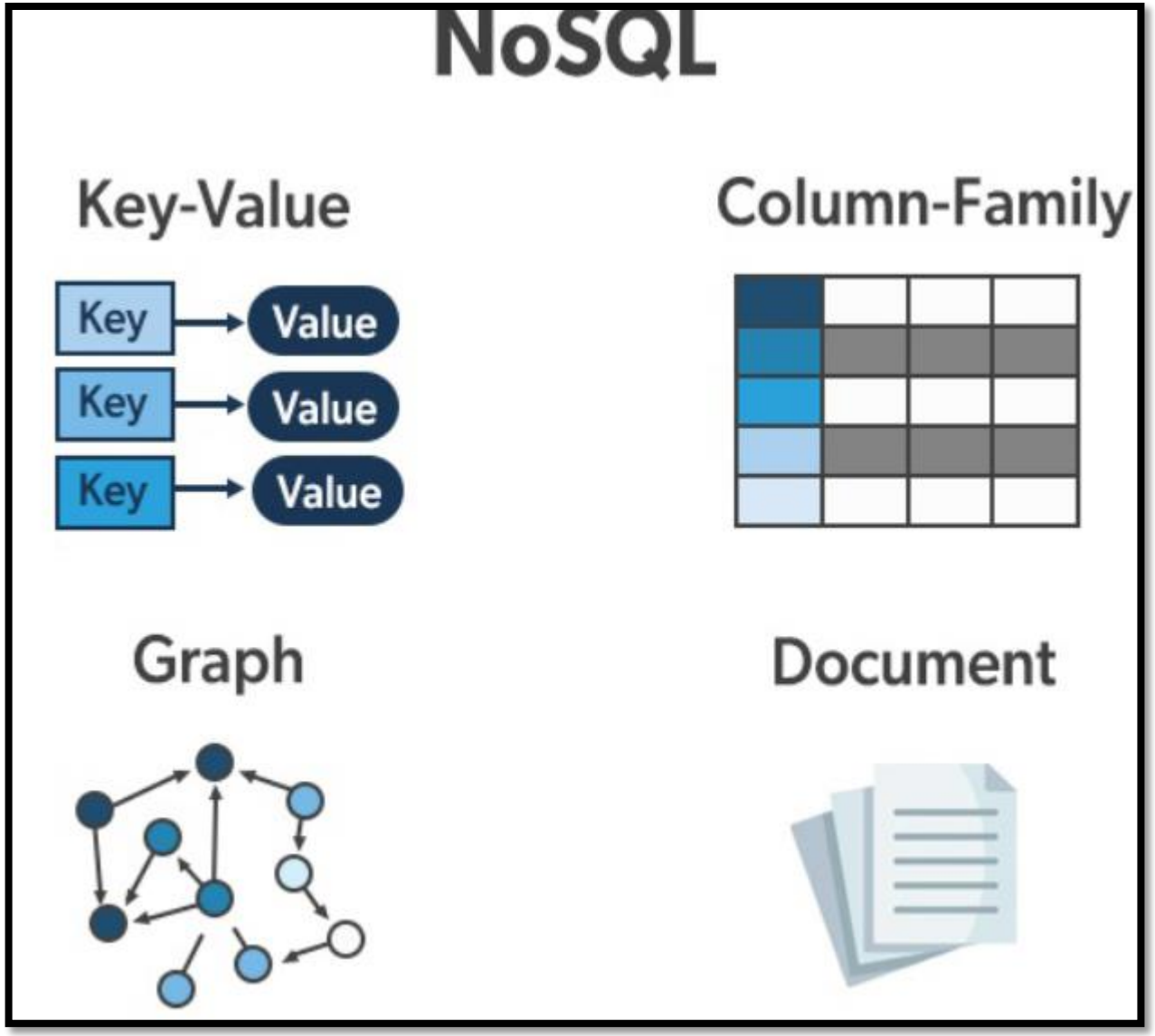
- ***Disadvantages of Relational Databases (SQL Databases):-***
- *Scalability Challenges:* Traditional SQL databases can face scalability challenges when dealing with large volumes of data or high write loads, as they rely on vertical scaling (adding more resources to a single node).
- *Schema Rigidity:* Relational databases require a predefined schema, which can make it difficult to adapt to changing data requirements or handle unstructured or semi-structured data.
- *Complexity:* SQL databases can be complex to set up, configure, and optimize, often requiring specialized database administrators and expertise.
- *Less Suitable for Unstructured Data:* SQL databases are primarily designed for structured data and may not be the best choice for handling large volumes of unstructured or semi-structured data, such as documents, media files, or IoT sensor data.

Types of Cloud Databases

- ***NoSQL Databases:-***
- NoSQL stands for *Not only SQL.*
- NoSQL cloud databases are designed to handle large volumes of *semi-structured or unstructured data*.
- They offer flexible data models, horizontal scalability, and relaxed consistency compared to traditional SQL databases.
- Unlike a relational database, NoSQL databases are non-tabular, meaning they don't store data in relational tables and rows with strict schemas. Because of this flexibility, NoSQL databases are able to store a variety of data types with varying schemas.
- These databases are designed to handle unstructured data, such as social media posts, log files, and user-generated content.

Types of Cloud Databases

- ***Types of NoSQL Database:***
- *Key-value stores*
- *Column-oriented databases*
- *Document-based databases*
- *Graph-based databases*

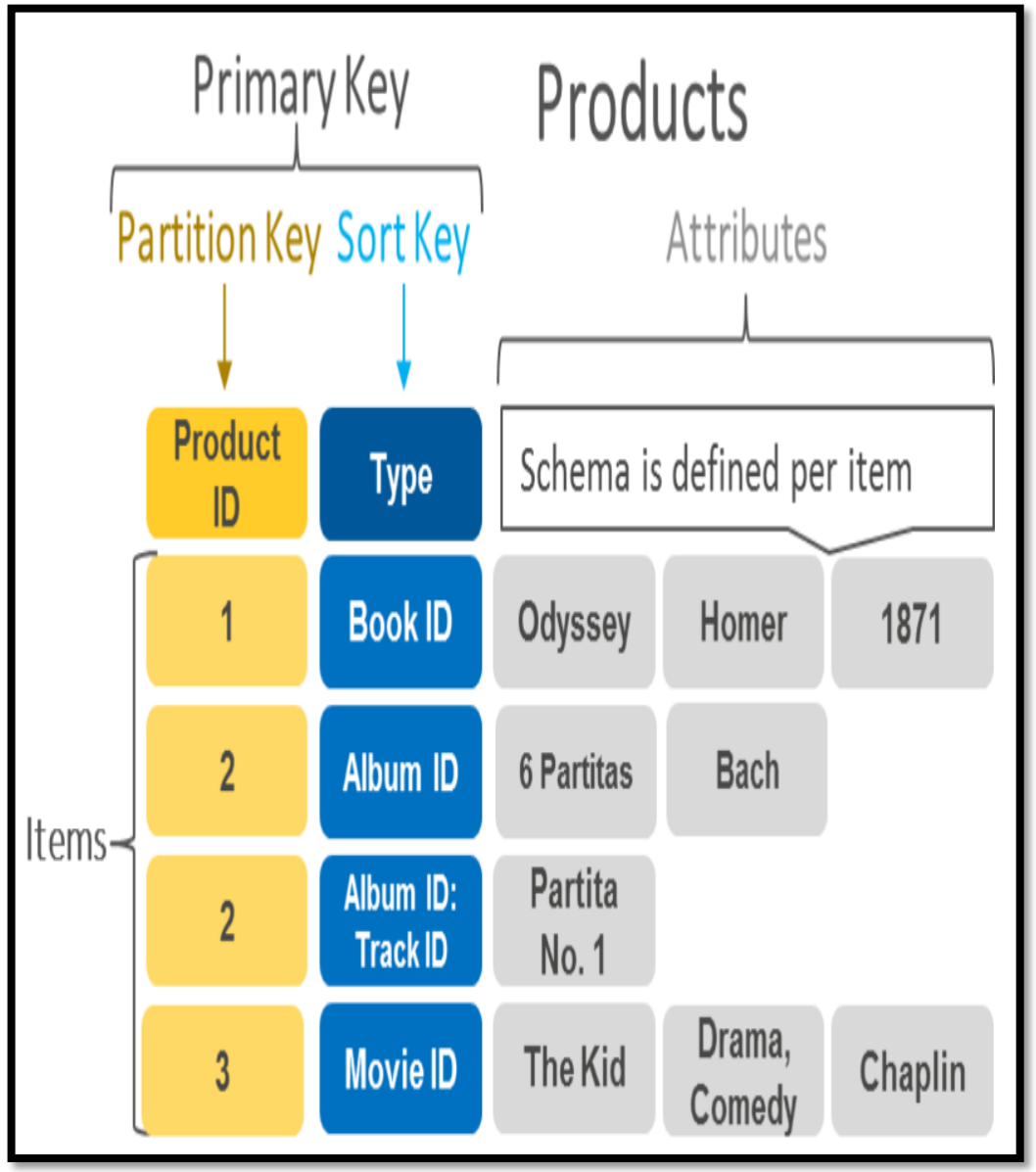


Types of Cloud Databases

- ***Characteristics of NoSQL Databases:-***
- Flexible data models (key-value, document, column-family, graph)
- Horizontally scalable (scale out by adding more nodes)
- Eventual consistency (data consistency achieved over time)
- High availability and partition tolerance
- Optimized for high-throughput and low-latency operations
- **Examples:**
- *Amazon DynamoDB (Key-Value)*
- *Azure Cosmos DB (Multi-model)*
- *Google Cloud Data store (Document-oriented)*
- *Redis (In-memory key-value store)*

Types of Cloud Databases

- ***Key-value stores:-***
- A key-value data store is a type of database that stores data as a collection of key-value pairs.
- In this type of data store, each data item is identified by a unique key, and the value associated with that key can be anything, such as a string, number, object, or even another data structure.
- Key features of the key-value store:
- *Simplicity, Scalability and Speed.*
- Example:-***Amazon DynamoDB***

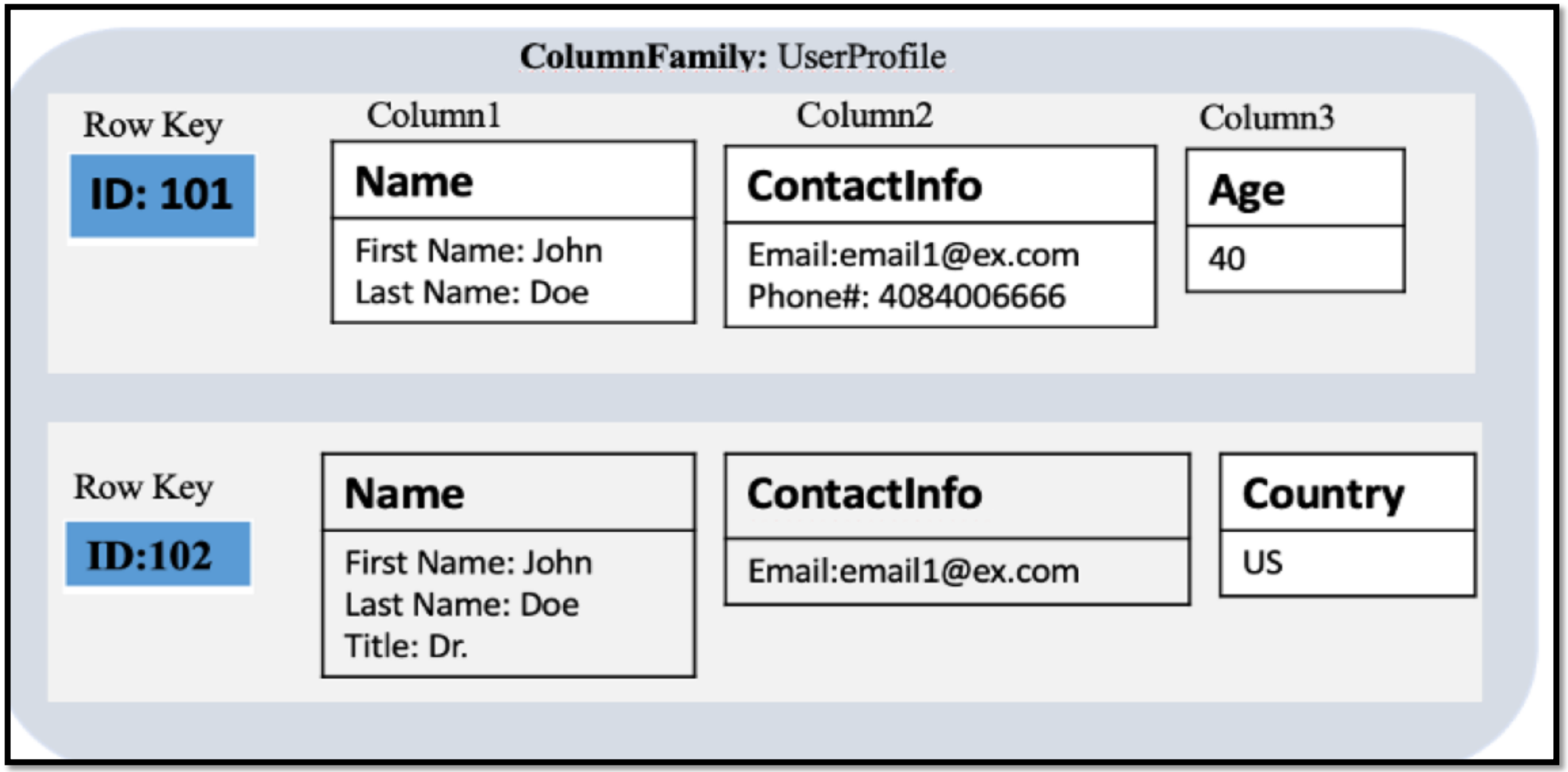


Types of Cloud Databases

- ***Column-oriented databases (Wide Column):-***
- A wide column data store is a type of NoSQL database that stores data in columns rather than rows, making it highly scalable and flexible.
- In a wide column data store, data is organized into column families, which are groups of columns that share the same attributes.
- Each row in a wide column data store is identified by a unique row key, and the columns in that row are further divided into column names and values.
- Key features of columnar oriented database:
- *Scalability.*
- *Compression.*
- *Very responsive.*
- Example:- ***Amazon Keyspaces (for Apache Cassandra)***

Types of Cloud Databases

- ***Column-oriented databases (Wide Column):-***



Types of Cloud Databases

- ***Document-based databases:-***
- In a document database, the data is stored in documents.
- Each document is typically a nested structure of keys and values.
- The values can be atomic data types, or complex elements such as lists, arrays, nested objects.
- A document database stores data in JSON, BSON, or XML documents.
- Documents are retrieved by unique keys.
- It may also be possible to retrieve only parts of a document.
- Example:-***Amazon DocumentDB (with MongoDB compatibility)***

Types of Cloud Databases

- ***Document-based databases:-***

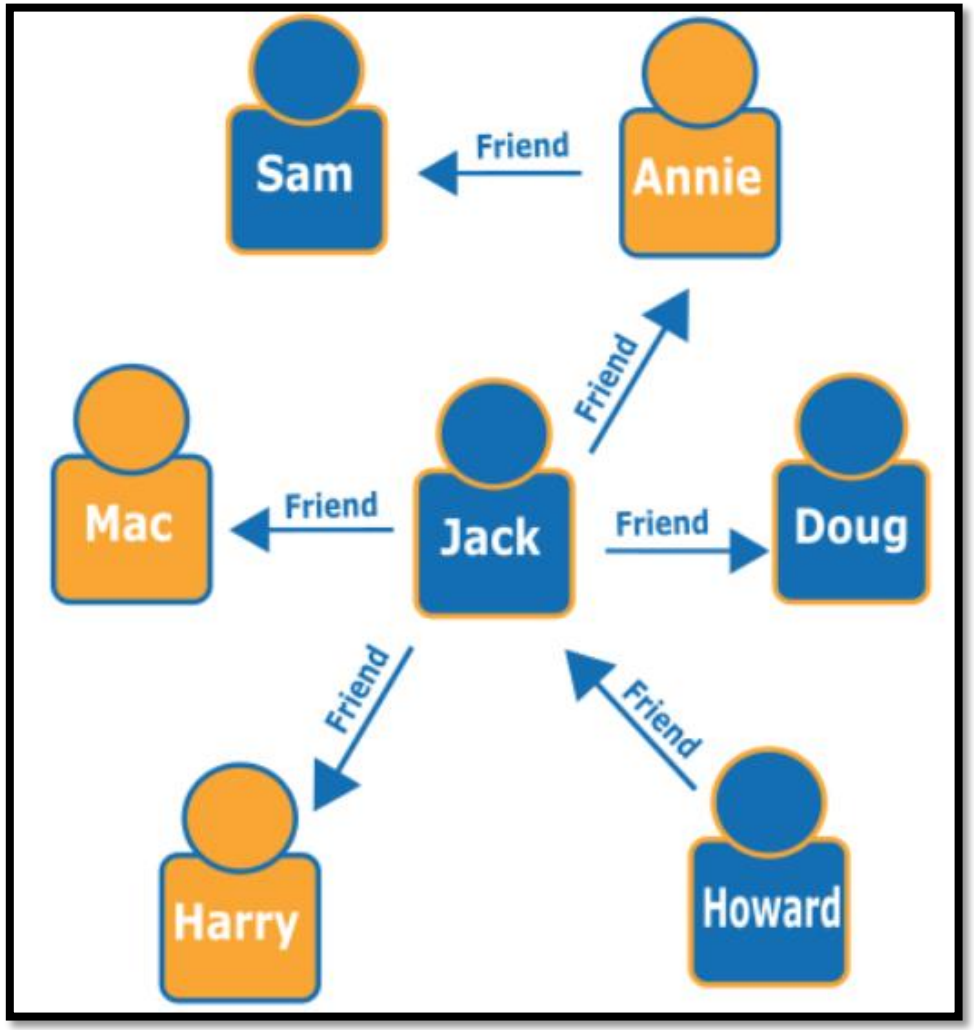
Key	Document
101	{ "ID": "1001", "ItemsOrdered":[{ "ItemID": "1", "Quantity": "2", "cost": "1000", }, { "ItemID": "1001", "Quantity": "2", "cost": "1000", }], "OrderDate": "05/11/2019" }
102	"ID": "1002", "ItemsOrdered":[{ "ItemID": "2890", "Quantity": "11", "cost": "10000", }] "OrderDate": "05/11/2019" }

Types of Cloud Databases

- ***Graph-based databases:-***
- Graph databases are used to store and query highly connected data.
- Data can be modeled in the form of entities (also referred to as nodes, or vertices) and the relationships between those entities (also referred to as edges).
- The strength or nature of the relationships also carry significant meaning in graph databases.
- Users can then traverse the graph structure by starting at a defined set of nodes or edges and travel across the graph, along defined relationship types or strengths, until they reach some defined condition.
- Results can be returned in the form of literals, lists, maps, or graph traversal paths.
- An example of a social network graph
- Example:-***Amazon Neptune***

Types of Cloud Databases

- ***Graph-based databases:-***



Types of Cloud Databases

- ***Popular Non Relational Databases (NoSQL Databases):-***
- **Amazon DynamoDB (Key-Value):**
- *Provider: Amazon Web Services (AWS)*
- *Description:-*
- DynamoDB is a fully managed, proprietary NoSQL database service provided by AWS.
- It is a key-value and document database that delivers single-digit millisecond performance at any scale.
- DynamoDB is designed to handle structured and semi-structured data, making it suitable for various workloads, including mobile, web, gaming, ad-tech, and IoT applications.

Types of Cloud Databases

- ***Popular Non Relational Databases (NoSQL Databases):-***
- **Azure Cosmos DB (Multi-model)**
- *Provider: Microsoft Azure*
- *Description:-*
- Azure Cosmos DB is a globally distributed, multi-model database service provided by Microsoft.
- It supports multiple data models, including key-value, document, wide-column, and graph, allowing developers to choose the right data model for their application.
- Cosmos DB is designed for globally distributed applications that require low latency, high availability, and scalability.

Types of Cloud Databases

- ***Popular Non Relational Databases (NoSQL Databases):-***
- **Google Cloud Data store (Document-oriented)**
- *Provider: Google Cloud Platform (GCP)*
- *Description:-*
- Google Cloud Data store is a highly scalable, fully managed, NoSQL document database service provided by GCP.
- It is designed for storing and querying structured, semi-structured, and unstructured data.
- Cloud Data store automatically handles replication, providing high availability and durability for applications running on GCP or in hybrid environments.

Types of Cloud Databases

- ***Popular Non Relational Databases (NoSQL Databases):-***
- **Amazon DocumentDB**
- *Provider: Amazon Web Services (AWS)*
- *Description:-*
- Amazon DocumentDB is a cloud-native document database service provided by AWS.
- It is designed to be compatible with the MongoDB 3.6 and 4.0 APIs, allowing developers to use the same MongoDB drivers and tools they are familiar with.
- DocumentDB is a separate database engine built for the cloud, providing better performance, scalability, and availability compared to running MongoDB on Amazon EC2 instances.

Types of Cloud Databases

- ***Advantages of Non Relational Databases (NoSQL Databases):-***
- *Scalability:* NoSQL databases are designed to scale horizontally by adding more servers or nodes, making them suitable for handling large volumes of data and high traffic loads.
- *Flexible Data Model:* NoSQL databases use flexible, schema-less data models, allowing for easier storage and retrieval of unstructured or semi-structured data, such as JSON, XML, or key-value pairs.
- *High Performance:* NoSQL databases are often optimized for specific data models and use cases, resulting in faster read and write operations compared to traditional SQL databases for certain workloads.
- *Availability and Partition Tolerance:* Many NoSQL databases prioritize availability and partition tolerance over strict data consistency, which can be beneficial for distributed systems and applications that require high availability.

Types of Cloud Databases

- ***Disadvantages of Non Relational Databases (NoSQL Databases):-***
- *Limited Query Capabilities:* NoSQL databases often lack the advanced querying capabilities and expressiveness of SQL, making it challenging to perform complex queries or join operations across multiple data sets.
- *Potential Data Inconsistency:* Some NoSQL databases sacrifice strong data consistency (ACID) for availability and partition tolerance, which can lead to eventual consistency and potential data inconsistencies in certain scenarios.
- *Lack of Standardization:* NoSQL databases lack a common query language or standard interface, making it more challenging to migrate between different NoSQL solutions or integrate with existing SQL-based systems.
- *Limited Tooling and Support:* NoSQL databases may have less mature tooling, documentation, and community support compared to established SQL databases, which can make development and maintenance more challenging.

SQL VS NoSQL

Aspect	*SQL Databases*	*NoSQL Databases*
Data Model	Tabular, relational	Key-value, document-oriented, column-family, graph
Schema	Predefined, rigid schema	Schema-less or dynamic schema
Query Language	Structured Query Language (SQL)	Various query languages/APIs specific to the database type
Scalability	Vertically scalable (scaling up)	Horizontally scalable (scaling out)
ACID Compliance	Generally ACID compliant (Atomicity, Consistency, Isolation, Durability)	Often follow BASE principles (Basically Available, Soft-state, Eventual consistency)
Use Cases	Complex transactions, queries, strict data integrity (e.g., financial systems, ERP, CRM)	Large volumes of unstructured/semi-structured data, real-time web applications, content management, big data analytics
Examples	MySQL, PostgreSQL, Oracle, SQL Server	MongoDB, Cassandra, Couchbase, Redis, Neo4j

Data Scaling and Replication

- ***Data Scaling:***
- Data scaling refers to the ability of a database to handle growth in data volume and traffic load.
- There are two main types of data scaling:
- *Vertical Scaling*
- *Horizontal Scaling*

Data Scaling and Replication

- ***Data Scaling:***
- *Vertical Scaling:*
- Vertical scaling, also known as scaling up, involves increasing the resources (such as CPU, RAM, and storage) of a single server or instance to handle higher workloads.
- This approach is often limited by the physical constraints of the hardware and can become costly.

Data Scaling and Replication

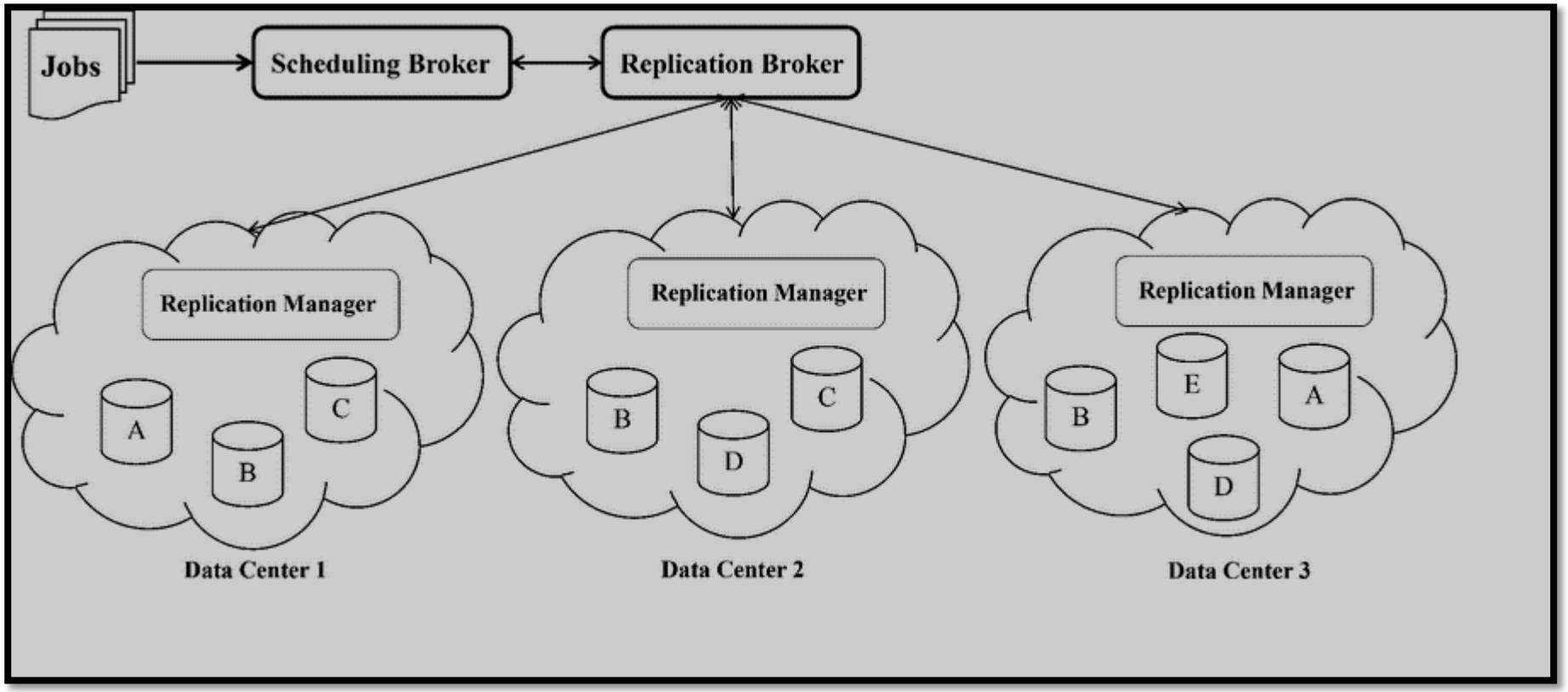
- ***Data Scaling:***
- *Horizontal Scaling:*
- Horizontal scaling, or scaling out, involves adding more servers or instances to a distributed system to share the workload.
- This approach is particularly beneficial for NoSQL databases, which are designed to scale horizontally by distributing data across multiple nodes or shards.
- Horizontal scaling is commonly achieved through techniques like sharding (partitioning data across multiple nodes) and replication (maintaining multiple copies of data on different nodes).

Data Scaling and Replication

- ***Data Replication:***
- Data replication is the process of creating and maintaining multiple copies of data across different nodes or servers in a distributed system.
- Data replication is the process of maintaining redundant copies of primary data.
- This is important for several reasons, including *fault tolerance*, *high availability* and *reduced network latency*.
- Replication can occur synchronously (where write operations are committed only after being replicated to all nodes) or asynchronously (where write operations are committed first, and replication occurs in the background).
- Asynchronous replication is generally faster but may result in temporary data inconsistencies in the event of a failure.

Data Scaling and Replication

- ***Data Replication:***



Data Scaling and Replication

- ***Data Replication:***
- *Fault Tolerance:* Data replication is necessary when applications must preserve data in the case of hardware or network failure due to causes ranging from someone tripping over a power cable to a regional disaster such as an earthquake. Thus, every application needs data replication for resilience and consistency.
- *High Availability:* By having multiple copies of data distributed across different nodes, replication ensures that the data remains accessible even if one or more nodes fail. This redundancy improves the overall availability and fault tolerance of the system.
- *Reduce Latency:* Data replication also helps modern cloud applications run off distributed data in different networks or geographic regions that serve the end user better.

Data Scaling and Replication

- ***Data Replication:***
- There are different replication strategies employed by database systems:
- *Master-Slave Replication:* In this model, one node (the master) serves as the authoritative source for write operations, while one or more slave nodes maintain read-only copies of the data. Changes made to the master are propagated to the slaves.
- *Multi-Master Replication:* In this model, multiple nodes can accept write operations, and changes are propagated to all other nodes in the cluster. This approach provides higher availability for write operations but can be more complex to manage and ensure data consistency.
- *Peer-to-Peer Replication:* In this decentralized model, each node in the cluster can accept both read and write operations, and changes are propagated to all other nodes using a consensus protocol.

Data Scaling and Replication

- ***Advantages of Data Replication:***
- *High Availability:* By maintaining multiple copies of data across different nodes or servers, data replication ensures that the data remains accessible even if one or more nodes fail, providing increased uptime and fault tolerance.
- *Improved Performance:* Replication allows read operations to be distributed across multiple nodes, reducing the load on individual nodes and improving query performance, especially for read-heavy workloads or applications with geographically distributed users.
- *Data Redundancy and Disaster Recovery:* With replicated data, if one copy of the data is lost or corrupted, the system can recover the data from other replicas, providing data redundancy and enabling disaster recovery.

Data Scaling and Replication

- ***Disadvantages of Data Replication:***
- *Increased Complexity:* Implementing and managing data replication can be complex, especially in large-scale distributed systems. Ensuring data consistency across multiple replicas and handling conflicts can be challenging.
- *Potential Data Inconsistency:* Depending on the replication strategy (synchronous or asynchronous) and the timing of updates, temporary data inconsistencies may arise, where different replicas have slightly different versions of the data for a short period.
- *Increased Storage Requirements:* Maintaining multiple copies of data across different nodes or servers results in increased storage requirements, which can lead to higher infrastructure costs.
- *Network Overhead:* Replicating data across a distributed system introduces additional network traffic, which can impact performance if the network infrastructure is not adequate or if the replicas are geographically distant.