Data can be categorized based on its format into three main types: structured, unstructured, and semi-structured. Each type has distinct characteristics and uses.

**1. Structured Data**

**Description**:

* Structured data is organized in a fixed format or structure.
* It is easily searchable and analysable because it resides in predefined fields within a record or file.

**Characteristics**:

* **Format**: Typically, in rows and columns, like a table.
* **Storage**: Stored in relational databases or spreadsheets.
* **Schema**: Has a well-defined schema.
* **Searchability**: Easy to query using structured query languages like SQL.
* **Examples**:
  + Databases (e.g., SQL databases)
  + Spreadsheets (e.g., Excel files)
  + Data from sensors with fixed output format

**Example Use Cases**:

* Financial records
* Customer information in CRM systems
* Inventory management

**2. Unstructured Data**

**Description**:

* Unstructured data lacks a predefined format or organization.
* It may contain a lot of valuable information but is harder to process and analyze using traditional data tools.

**Characteristics**:

* **Format**: Free-form text or multimedia.
* **Storage**: Often stored in non-relational databases or data lakes.
* **Schema**: No predefined schema.
* **Searchability**: More challenging to search and analyze without specialized tools.
* **Examples**:
  + Text documents (e.g., Word files, PDFs)
  + Multimedia files (e.g., images, videos, audio files)
  + Social media posts (e.g., tweets, Facebook updates)
  + Email messages

**Example Use Cases**:

* Content management systems
* Social media analysis
* Multimedia archives

**3. Semi-Structured Data**

**Description**:

* Semi-structured data has some organizational properties but does not conform to a strict schema.
* It includes elements of both structured and unstructured data.

**Characteristics**:

* **Format**: Often contains tags or markers to separate data elements, but lacks the strict tabular structure of structured data.
* **Storage**: Can be stored in non-relational databases or specialized formats.
* **Schema**: Flexible schema; the structure can be irregular or incomplete.
* **Searchability**: More manageable than unstructured data but less straightforward than structured data.
* **Examples**:
  + JSON (JavaScript Object Notation)
  + XML (eXtensible Markup Language)
  + Email metadata (e.g., headers with sender, recipient, and timestamp)
  + NoSQL databases (e.g., MongoDB, Cassandra)

**Example Use Cases**:

* Data exchange between web services (APIs)
* Configuration files
* Log files with consistent but flexible formats

**Summary**

* **Structured Data**: Highly organized, easily searchable, stored in tables.
* **Unstructured Data**: No predefined structure, includes text and multimedia, harder to search.
* **Semi-Structured Data**: Contains organizational elements but is more flexible than structured data, includes formats like JSON and XML.

Understanding the differences between these types of data is crucial for selecting appropriate storage, processing, and analysis techniques.

Why we need cloud rather than Datacenter? Answer: Durability [Chance of not losing the data] is 99.99999999999[11 times]

High-performance, Security and lifecycle.

Storage management in Azure involves various services and tools to manage and optimize data storage in the cloud. Here's an overview of the key aspects and services involved in Azure storage management:

### 1. Azure Storage Services

#### a. Azure Blob Storage

* **Purpose**: Object storage solution for unstructured data.
* **Use Cases**: Storing large amounts of unstructured data like text or binary data, serving images or documents directly to a browser, streaming video and audio, writing log files.

#### b. Azure File Storage

* **Purpose**: Fully managed file shares in the cloud.
* **Use Cases**: Shared storage for legacy applications using the standard SMB protocol, lift and shift on-premises applications to Azure without needing to change code.

#### c. Azure Queue Storage

* **Purpose**: Provides message queuing for large workloads.
* **Use Cases**: Decoupling components of a cloud application to improve scalability and reliability.

#### d. Azure Table Storage

* **Purpose**: A NoSQL key-value store for rapid development.
* **Use Cases**: Storing structured, non-relational data, such as user data for web applications, device information, or other metadata.

#### e. Azure Disk Storage

* **Purpose**: Managed disk storage for Azure Virtual Machines (VMs).
* **Use Cases**: Persistent storage for VM workloads, high-performance storage for critical applications.

### 2. Storage Management Tools

#### a. Azure Portal

* **Description**: Web-based interface for managing Azure resources.
* **Features**: Create, configure, and monitor storage accounts, access and manage stored data, set up alerts and notifications.

#### b. Azure Storage Explorer

* **Description**: Standalone app to manage Azure Storage.
* **Features**: Browse and interact with data in storage accounts, upload and download data, manage access policies.

#### c. Azure CLI and PowerShell

* **Description**: Command-line tools for managing Azure resources.
* **Features**: Automate management tasks, integrate storage management into scripts and DevOps pipelines.

#### d. REST API

* **Description**: Programmatic interface to Azure Storage services.
* **Features**: Access and manage Azure Storage programmatically, integrate with applications and services.

### 3. Storage Management Best Practices

#### a. Security and Access Control

* **Use Role-Based Access Control (RBAC)**: Assign appropriate permissions to users and services.
* **Implement Shared Access Signatures (SAS)**: Provide limited access to resources without exposing account keys.
* **Enable Azure AD Authentication**: Use Azure Active Directory for authentication and access management.

#### b. Data Redundancy and Replication

* **Locally Redundant Storage (LRS)**: Replicates data three times within a single data center.
* **Zone-Redundant Storage (ZRS)**: Replicates data across multiple availability zones.
* **Geo-Redundant Storage (GRS)**: Replicates data to a secondary region hundreds of miles away from the primary location.
* **Read-Access Geo-Redundant Storage (RA-GRS)**: Provides read access to the secondary region.

#### c. Performance Optimization

* **Choose Appropriate Tiers**: Use Hot, Cool, or Archive tiers based on data access patterns.
* **Optimize Blob Storage**: Use Azure Blob Indexer, enable soft delete, and manage lifecycle policies.
* **Monitor Performance**: Use Azure Monitor and Application Insights to track and analyze performance metrics.

#### d. Cost Management

* **Monitor Usage**: Use Azure Cost Management and Billing tools to track storage usage and costs.
* **Set Budgets and Alerts**: Create budgets and set up alerts for cost management.
* **Optimize Storage Tiers**: Regularly review and adjust data storage tiers based on usage patterns.

### 4. Backup and Disaster Recovery

* **Azure Backup**: Simplifies data protection for Azure VMs, on-premises servers, and workloads like SQL and SAP HANA.
* **Azure Site Recovery**: Ensures business continuity by keeping business apps and workloads running during outages.

### 5. Data Migration

* **Azure Migrate**: Assesses and migrates on-premises servers, databases, web applications, and virtual desktops to Azure.
* **Azure Data Box**: Physical device for transferring large amounts of data to Azure.

Managing storage in Azure involves a combination of understanding the various storage services, using the right tools for management, adhering to best practices, and utilizing additional services for backup, disaster recovery, and data migration.

Terms: Encryption, Storage browser, connect option.

Connect option give the command that we can use in file storage.

Connect option give the command that we can use in file storage Managing Top of Form

Bottom of Form