TYPES OF STORAGES

File Storage (Azure Files)

- **What it is:**
  Think of it like a **shared network drive (like a USB or local folder)** but hosted in the cloud.
  It allows multiple machines or applications to access the same files at the same time.
- **How it works:**
  - Uses **SMB (Server Message Block)** or **NFS (Network File System)** protocols (the same protocols used in Windows or Linux to share folders).
  - Applications can mount it just like a normal drive (Z:\ on Windows or /mnt/share on Linux).
- **Use Cases:**
  - Replacing on-premises file servers.
  - Storing logs, configuration files, or datasets that multiple VMs/apps need.
  - Easy **lift-and-shift migration** of existing apps that expect a normal file system.
- **Example:**
  Suppose you have 5 virtual machines (VMs) running an app. Instead of copying the same file to each VM, you just put it in File Storage and all 5 VMs read from there.

Blob Storage (Binary Large Objects)

- **What it is:**
  Think of it as a **big container for any kind of unstructured data** (images, videos, PDFs, backups, large binary files).

- **How it works:**
    - Stores files as **objects (blobs)** inside containers.
    - Each blob has a unique URL, so you can directly access it via the internet (if allowed).
    - Types of blobs:
        - **Block blobs** (general files, documents, images).
        - **Append blobs** (optimized for logging – only add at the end).
        - **Page blobs** (used for virtual hard disks in Azure VMs).
- **Use Cases:**
    - Hosting website images/videos.
    - Storing large media files, data backups, IoT data.
    - Serving downloadable content.
- **Example:**
  A company app stores all uploaded customer profile pictures in Blob Storage. Each picture can be retrieved using its unique link.

Queue Storage

- **What it is:**
  A **message storage system** where you can send and receive messages reliably.
  Think of it like a **post office queue** – messages wait until an application picks them up.
- **How it works:**
    - Stores small text messages (up to 64 KB each).

    - A producer (sender) adds a message to the queue.
    - A consumer (receiver) reads and processes it.
    - Ensures **asynchronous communication** between app components.
- **Use Cases:**
    - Decoupling app components (if one service is down, messages wait in queue).
    - Order processing (e.g., customer order goes into a queue, processed later).
    - Task scheduling and load balancing.
- **Example:**
  In an e-commerce site: when a customer orders a product, the order details are stored in Queue Storage. The backend system picks it up later and processes payment/shipping.

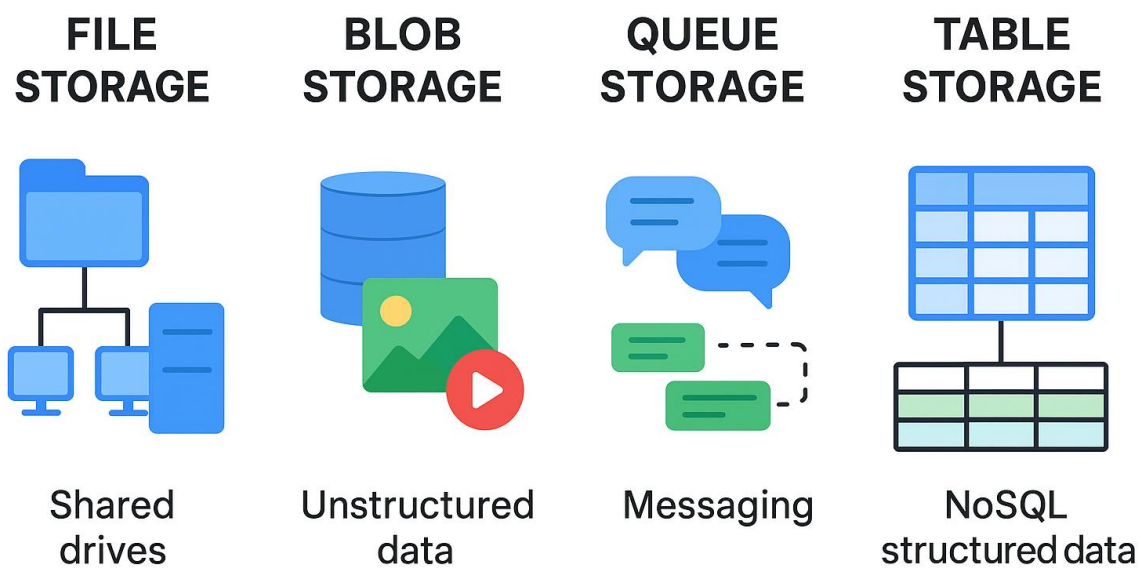
Table Storage (NoSQL Database)

- **What it is:**
  A **NoSQL key-value store** that stores large amounts of structured, non-relational data.
  Think of it like an **Excel sheet without fixed schema**.
- **How it works:**
    - Data is stored in **tables** (but not like SQL).
    - Each row is an **entity**, with flexible properties (columns).
    - Supports **PartitionKey** and **RowKey** for fast access.
    - Highly scalable and cheaper than SQL DB.
- **Use Cases:**

    - Storing large amounts of metadata (user profiles, device info).
    - Storing IoT sensor data.
    - Scenarios where you don't need complex queries or joins.
- **Example:**
  A mobile app stores millions of user profiles in Table Storage. Each profile can have different attributes (some may have "phone_number," some may not), but all can be retrieved quickly using PartitionKey + RowKey.

Quick Summary Table

Storage Type	Best For	Data Type	Example
File Storage	Shared drives	Files, documents	Shared VM folder
Blob Storage	Unstructured data	Images, videos, backups	Store user-uploaded images
Queue Storage	Messaging	Small text messages	Order queue in e-commerce
Table Storage	NoSQL structured data	Key-value entities	User profiles, IoT data



Specific use cases

File Storage (Azure Files)

When to Use:

- **If your application expects a shared drive/folder.**
- **When you want to lift-and-shift legacy apps to the cloud without changing code.**
- **If multiple VMs or apps need to access the same files at the same time.**

Use Case Examples:

- **Centralized storage for configuration files, logs, or reports.**
- **Migrating an on-premises file server to the cloud.**
- **Shared development environment where multiple users access the same files.**

Blob Storage

When to Use:

- **If you need to store huge amounts of unstructured data (images, videos, backups).**
- **When you want files accessible via URLs over the internet.**
- **If your app requires cheap and scalable storage.**

Use Case Examples:

- **Hosting product images, videos, or documents for a website/app.**
- **Storing backups, archives, or log data.**
- **Serving streaming content like video or audio.**
- **Data lake for big data analytics.**

Queue Storage

When to Use:

- **If different parts of your application need to communicate asynchronously.**
- **When you want to decouple services (one service sends a message, another picks it later).**
- **For task scheduling or load balancing.**

Use Case Examples:

- **E-commerce: When an order is placed, add it to a queue for later processing.**
- **Video processing app: Upload video → send a queue message "process this file."**

- **Background tasks (email sending, notification processing, report generation).**

Table Storage

When to Use:

- **If you need to store structured, queryable, but schema-flexible data.**
- **When data is too big for relational DB and doesn't need complex joins.**
- **For metadata storage alongside files in Blob/File storage.**

Use Case Examples:

- **User profiles (name, email, preferences).**
- **IoT sensor data (device ID, temperature, timestamp).**
- **Product catalogs with millions of entries.**
- **Metadata for files (store image info in Table while actual image is in Blob).**

Quick Real-Life Scenario (E-commerce Website):

- **File Storage: Store invoice templates and shared reports used by admins.**
- **Blob Storage: Store product images, videos, and customer-uploaded files.**
- **Queue Storage: Place customer orders into a queue so backend services can process them asynchronously.**
- **Table Storage: Store customer profiles, order history, and product metadata.**

COMPARISON

File Storage vs Blob Storage

- **File Storage Advantages**
  - Easy for apps expecting a normal file system.
  - Supports SMB/NFS → multiple VMs can share the same files.
  - Great for "lift-and-shift" of legacy apps.
- **Blob Storage Advantages**
  - Scales better for massive unstructured data.
  - Direct internet access with unique URLs.
  - Cheaper for storing large backups, media, logs.
- **When to Use**
  - **File Storage:** When you need a shared drive for multiple servers.
  - **Blob Storage:** When you need scalable storage for images, videos, backups, or streaming.

Blob Storage vs Queue Storage

- **Blob Storage Advantages**
  - Stores large unstructured objects (images, videos, documents).
  - Can be accessed globally with URLs.
- **Queue Storage Advantages**
  - Best for communication, not data storage.
  - Helps apps communicate asynchronously.

    - Decouples components (fault-tolerant).
- **When to Use**
    - **Blob Storage:** Store a video uploaded by a user.
    - **Queue Storage:** Store a message saying *"process this video later"*.

Queue Storage vs Table Storage

- **Queue Storage Advantages**
    - Handles **workflow messaging**.
    - Ensures reliability in distributed systems.
- **Table Storage Advantages**
    - Stores **structured, queryable** data.
    - Highly scalable NoSQL store, flexible schema.
    - Cheaper than SQL DB.
- **When to Use**
    - **Queue Storage:** Order placed → add to queue → processed later.
    - **Table Storage:** Store order details (customer ID, amount, status).

File Storage vs Table Storage

- **File Storage Advantages**
    - Looks like a regular shared folder, no code changes needed.
    - Good for traditional apps expecting files.

- **Table Storage Advantages**
  - Better for structured, queryable, large-scale data.
  - High availability and scalability.
- **When to Use**
  - **File Storage:** Multiple apps/VMs accessing shared reports/logs.
  - **Table Storage:** Storing millions of rows of user profiles or IoT data.

Comparison Summary Table

Storage Type	Strengths	Benefits	Best Use Cases
File Storage	Shared drive	Easy migration, multi-VM access	Legacy apps, shared configs/logs
Blob Storage	Unstructured data	Scalable, cheap, URL access	Images, videos, backups
Queue Storage	Messaging system	Decouples apps, reliable	Order queue, task scheduling
Table Storage	NoSQL data	Flexible schema, scalable	User profiles, IoT, metadata