**Azure Storage Services & Data Lake Integration**

**1. Azure Storage Account Types**

A **storage account** is the fundamental container in Azure that holds all data objects (blobs, files, queues, tables, disks).

**1.1 Types**

1. **General Purpose v1 (GPv1)**
   * Legacy account type.
   * Supports blobs, files, queues, and tables.
   * Lower cost but limited features.
2. **General Purpose v2 (GPv2)**
   * Recommended for most scenarios.
   * Supports **all storage services**.
   * Advanced features: **tiering, lifecycle management, Azure Data Lake Gen2**.
3. **Blob Storage Accounts**
   * Optimized for blob workloads (large unstructured data).
   * Storage tiers: **Hot, Cool, Archive**.
   * Cost-efficient for massive object storage.

**2. Blob Storage Concepts**

* **Container** → Logical grouping of blobs (like folders).
* **Blob Types**:
  1. **Block Blob** – Stores large amounts of unstructured text/binary (media files, documents).
  2. **Append Blob** – Optimized for log data where new data is always appended.
  3. **Page Blob** – Optimized for random read/write (used by VHD files for Azure VM disks).

**3. Azure Data Lake Storage Gen2**

* Built on **Blob Storage** with **Hierarchical Namespace (HNS)**.
* Features:
  + Directory & folder structure (POSIX-compliant).
  + Fine-grained **ACLs** at file/folder level.
  + Optimized for big data analytics with **ADF, Databricks, Synapse**.

**4. Creating Containers & Folder Structures in ADLS**

* **Containers** act as root-level containers.
* Inside containers:
  + Create **folders** for structured organization.
  + Use **directory paths** for hierarchy.
* Example:
* /raw/sales/2025/jan/data.csv
* /processed/sales/2025/jan/summary.parquet

**5. Table Storage & Queue Storage Basics**

* **Table Storage**
  + NoSQL key-value store.
  + Fast access to structured, non-relational data.
  + Schema-less, good for metadata.
* **Queue Storage**
  + Stores **messages** in a queue for asynchronous processing.
  + Useful for decoupling applications (e.g., order system, background jobs).

**6. Azure Files**

* Fully managed **file share** in the cloud.
* Protocols: **SMB (Windows/Linux)**, **NFS (Linux)**.
* Can mount on-premises or cloud VMs.
* Useful for **lift & shift** scenarios.

**7. Authentication & Authorization**

1. **Shared Key** – Access with account name & key. (not recommended).
2. **SAS (Shared Access Signature)** – Temporary, restricted access (time, IP, permissions).
3. **Azure AD Authentication** – Modern, secure, role-based.

**8. RBAC Roles for Storage**

* **Storage Blob Data Reader** – Read-only access.
* **Storage Blob Data Contributor** – Read/write/delete blobs.
* **Storage Blob Data Owner** – Full access including role assignments.
* Use **least privilege principle** when assigning.

**9. Lifecycle Management**

* Define **rules** for cost optimization:
  + Move infrequently accessed data to **Cool/Archive** tiers.
  + Delete old/unused files.
* Example:
  + If blob not accessed for 30 days → move to Cool.
  + If blob not accessed for 365 days → Archive.

**10. Data Redundancy Options**

* **LRS (Locally Redundant Storage)** – 3 copies within one data center.
* **ZRS (Zone Redundant Storage)** – 3 copies across zones in same region.
* **GRS (Geo-Redundant Storage)** – 6 copies (3 local + 3 in paired region).
* **RA-GRS (Read-Access GRS)** – Same as GRS but allows read from secondary.

**11. Monitoring Azure Storage**

* **Metrics**: Availability, transactions, latency.
* **Diagnostics Logs**: Operations performed on storage.
* **Azure Monitor**: Alerts, dashboards, performance tracking.

**12. Integration with Analytics Services**

* **Azure Data Factory (ADF)**
  + Data ingestion pipelines.
  + Can read/write from ADLS Gen2.
* **Azure Databricks**
  + Data processing, ML, analytics on top of ADLS.
  + Mount ADLS containers for seamless access.
* **Best Practice**: Use **managed identities** for secure authentication.