**Azure Storage Overview**

Azure Storage is Microsoft's cloud storage solution designed to store massive amounts of data in a highly scalable and durable manner. It provides multiple storage services that cater to various use cases, ranging from storing unstructured data to handling large-scale file systems and managed disks for virtual machines.

Here’s an overview of key Azure Storage services:

**Blob Storage:** Store unstructured data like documents, images, videos, backups, logs, and big data workloads.

**File Storage (Azure Files):** Managed file shares in the cloud that can be accessed via SMB or NFS protocols.

**Queue Storage:** Messaging queue service to store and retrieve messages for communication between distributed applications.

**Table Storage:** NoSQL key-value store for storing large amounts of structured, non-relational data.

**Blob Storage**

Azure Blob Storage is a scalable cloud-based object storage service for large amounts of unstructured data, such as text or binary data. It is ideal for storing any kind of data, from files and images to large datasets used by analytics or machine learning applications.

**Key Concepts in Azure Blob Storage:**

1. **Storage Account**: The top-level container that holds blobs, queues, tables, and files. It's necessary to have a storage account to use Blob Storage.
2. **Containers**: A container organizes a set of blobs, similar to a folder in a file system. All blobs must be stored in a container.
3. **Blobs**: The actual data that is stored. There are three types of blobs:

* **Block blobs**: Ideal for storing text and binary data, commonly used for files and media streaming.
* **Append blobs**: Optimized for append operations, like logging data.
* **Page blobs**: Used for storing random-access files, such as virtual hard disks (VHDs) for Azure virtual machines.

1. **Tiers**:

* **Hot**: For data that is accessed frequently.
* **Cool**: For infrequently accessed data that needs to be stored for at least 30 days.
* **Archive**: For rarely accessed data that needs long-term storage with higher retrieval times.

1. **Data Security**: Supports role-based access control (RBAC), shared access signatures (SAS), and encryption for data at rest and in transit.
2. **Lifecycle Management**: Automates data retention by applying rules to automatically move data to different access tiers or delete it based on age.

I have created a blob storage account and inside that a container named ‘neeraj’ and uploaded some files in that data containers and configured like tiers, blobs types, lifecycle management etc…

**Queue Storage**

Azure Queue Storage is a service that allows you to store large numbers of messages, which can be accessed from anywhere in the world via authenticated calls using HTTP or HTTPS. It is part of the Azure Storage account and provides reliable and scalable message queuing for asynchronous communications between different parts of an application.

**Key Features:**

* **Message Size**: Each message in the queue can be up to 64 KB in size, and a queue can contain millions of messages.
* **Visibility Timeout**: When a message is retrieved from the queue, it becomes invisible to other consumers for a specified period, allowing time for processing. If the message is not deleted after processing within the visibility timeout, it becomes visible again for re-processing.
* **Asynchronous Messaging**: Useful for decoupling different components in a distributed system. For example, it can be used to pass messages between cloud services or between cloud and on-premises applications.
* **TTL (Time-to-Live)**: You can set a time-to-live for messages in the queue, after which they are automatically deleted.

I have worked on scenario of Order Processing Systems where the order message is saved in queue and dequeue the message from queue storage then process the messages.

**Table Storage**

Azure Table Storage is a service provided by Microsoft Azure that allows users to store large amounts of structured, non-relational data. It is a NoSQL data store that supports key-value pairs. Table Storage is part of Azure Storage and is designed for applications that require flexible schema, high availability, and scalability. It can be used for scenarios such as storing metadata, logs, configuration data, user information, and other semi-structured data.

**Data Model**

Table Storage organizes data into the following key components:

1. **Tables**: A collection of entities (analogous to rows in a traditional relational database).
2. **Entities**: An entity is a single record (row) in a table.
3. **Properties**: Each entity has a set of properties. These properties can be different for each entity, allowing schema flexibility.
4. **PartitionKey**: This key is used to partition data into different sections for scaling and load balancing.
5. **RowKey**:
   * This is the unique identifier for each entity within a partition.
   * The combination of the PartitionKey and RowKey uniquely identifies an entity within a table, enabling fast lookups.

I have worked with table storage by creation of two tables (customers, orders) and implemented Insert, update, delete, select functionality

**Azure Files**

Azure Files is a managed file storage service in Azure that provides shared file storage that can be accessed via the Server Message Block (SMB) protocol or the Network File System (NFS) protocol. Azure Files allows you to create file shares in the cloud that can be mounted on-premises and on cloud-based virtual machines or directly accessed via the REST API. This service is particularly useful for scenarios like file sharing, migration, and application file storage.

**Key Features:**

1. **Fully Managed File Shares**: No need to manage infrastructure, Azure manages everything.
2. **Protocol Support**:
   * **SMB** (version 2.1 and 3.0): Allows file shares to be accessed by Windows, macOS, and Linux.
   * **NFS** (version 3.0): Used for Linux-based applications and workloads.
3. **File Sync**: Azure File Sync allows you to replicate your on-premises file shares with Azure Files, offering a hybrid solution.

**Azure Storage Explorer**

Azure Storage Explorer is a tool that allows you to manage and access data stored in your Azure Storage accounts. It provides a user-friendly interface for performing tasks such as uploading, downloading, and managing blobs, files, queues, and tables in Azure. You can use it to:

1. **Manage storage accounts**: Connect to multiple storage accounts, either through your Azure subscription or via connection strings.
2. **Blob storage**: Upload, download, view, and edit blob contents (including block blobs, page blobs, and append blobs).
3. **File storage**: Access and manage files in Azure File shares, including uploading and downloading files.
4. **Table storage**: Create and manage Azure Table entities.
5. **Queue storage**: Manage messages in Azure Queue storage.

It's particularly useful for developers, system administrators, and anyone who needs to interact with Azure Storage directly from a local machine. It works on Windows, macOS, and Linux. You can also use it with local storage emulators for testing and development purposes.

I have worked on managing different storage types in azure storage explorer like creation, updating, deleting, monitoring, inserting data etc…

**Monitoring and Logging**

Azure provides several options for monitoring and logging in Azure Storage. Here's a breakdown of the key components:

1. **Azure Monitor**

Azure Monitor offers insights into Azure Storage by collecting metrics and logs. It helps track performance, diagnose issues, and understand usage patterns.

* **Metrics**: Provides real-time information about storage performance, including:
  + Capacity usage
  + Ingress and egress (data transfer)
  + Availability
  + Latency
  + Success/failure rates for storage operations

1. **Storage Analytics Logs**

Azure Storage Analytics allows you to log data related to storage requests:

* **Request Logs**: Detailed information about read, write, and delete operations.
* **Minute Metrics**: Aggregated data about operations performed in one-minute intervals, including average and total request times, errors, etc.
* **Hourly Metrics**: Similar to minute metrics but aggregated over an hour.

I worked on metrics and alerts by average of storage allocation exceeds 20 then alert should be sent via mail of type of warning message. For Logging I have used storage analytics to created logs for every operation in Azure storage.