Azure Storage – Comprehensive Student Notes

# Overview of Azure Storage

Azure Storage is a cloud-based solution for storing large amounts of data securely, reliably, and cost-effectively. It supports multiple data types and access scenarios, making it ideal for everything from virtual machines to big data analytics.

# Types of Data Stored in Azure

1. **Storage for Virtual Machines**

- Disks: Persistent storage for VM OS and data disks.

- Files: Shared file systems for applications needing file-level access.

Note: VM disks use page blobs, which support random read/write operations.

2. **Unstructured Data**

- Blobs: Store binary data like images, videos, backups.

- Data Lake Store: Optimized for analytics workloads using hierarchical namespace.

Note: Blobs are ideal for storing large media files, logs, and backups.

3. **Structured & Semi-Structured Data**

- Tables: NoSQL key-value store for fast access.

- Cosmos DB: Globally distributed multi-model database.

- Azure SQL DB: Fully managed relational database.

# Storage Tiers

- Standard HDD: Economical, lower performance. Use for backup and infrequent access.

- Premium SSD: High performance, low latency. Use for production workloads and VMs.

Note: Choose tiers based on performance needs and cost constraints.

# Azure Storage Services

- Blob Containers: Store unstructured data.

- Azure Files: Shared file system via SMB.

- Azure Queues: Message queue for decoupled apps.

- Azure Tables: NoSQL store for structured data.

# Storage Account Types

All storage accounts are encrypted at rest using SSE (Storage Service Encryption) with AES 256-bit.

- Standard GPv2: Supports all services, cost-effective.

- Premium Block Blobs: High-performance blob storage.

- Premium File Shares: High-performance file storage.

- Premium Page Blobs: Optimized for VM disks.

# Replication Options

- LRS: Replication within a single datacenter. Cost-effective, less resilient.

- ZRS: Replication across zones in a region. High availability.

- GRS / RA-GRS: Replication across regions. Disaster recovery.

- GZRS / RA-GZRS: Combines zone and geo-redundancy. Maximum resilience.

# Default Storage Endpoints

Each storage account has default endpoints for:

- Blob service

- Table service

- Queue service

- File service

# Custom Domain Mapping

You can map your own domain to Azure Storage using CNAME records.

Direct Mapping:

blobs.contoso.com → contosoblobs.blob.core.windows.net

Intermediary Domain with asverify:

asverify.blobs.contoso.com → asverify.contosoblobs.blob.core.windows.net

blobs.contoso.com → contosoblobs.blob.core.windows.net

# Secure Storage Endpoints

- Use Firewalls and Virtual Networks (VNets) to restrict access.

- Define subnets for granular control.

# Blob Storage Access Levels

- Private: No public access.

- Blob: Public read access to blobs only.

- Container: Public read access to all blobs in container.

# Blob Access Tiers

- Hot: Frequently accessed.

- Cool: Infrequently accessed. >> 30 days

- Cold: Rarely accessed. >> 90 days

- Archive: Long-term archival. >> 180 days

# Blob Types

- Block Blob: For text and binary data.

- Append Blob: Optimized for append operations.

- Page Blob: Used for random read/write (e.g., disks).

# Storage Security

Encryption:

- SSE (Storage Service Encryption) with AES 256-bit

Authentication:

- Azure AD RBAC for identity-based access

Data in Transit:

- Client-side encryption

- HTTPS

- SMB 3.0 for secure file access

Shared Access Signatures (SAS):

- Temporary, scoped access to storage resources

# Authorization Options

- Azure AD: Role-based access

- Shared Key: Full access using account key

- SAS: Scoped access with permissions

- Anonymous: Public access to blobs/containers

# SAS (Shared Access Signatures)

SAS tokens can be customized for:

- Services: Blob, File, Queue, Table

- Time: Start and expiry

- Permissions: Read, Write, Delete

- IP Ranges

- Protocols: HTTPS recommended

# SAS Best Practices

- Always use HTTPS for SAS distribution.

- Use Stored Access Policies for manageability.

- Set short expiration times for temporary access.

- Enable automatic renewal for long-running apps.

- Be cautious with start times to avoid clock skew issues.

- Be specific with resources to minimize exposure.

- Monitor usage with Storage Analytics.

- Validate data written via SAS.

- Don’t assume SAS is always the best option.