Azure Storage

- Vm Disks
  - Data disks
    - Azure managed disks
    - Max capacity 37767 GB
    - Used for Database files, static content
  - Support SMB & NFS
  - Readable by REST & SDKs
    - Content of apps
    - Good for Logs, crash dumps
- BLob storage & Data lake storage (Accessible by NFS)
  - highly scalable REST based object storo
  - Azure Datalake storage (ADLS) is HDFS servic
- Structured storage (Table storage)
  - Azure Table storage — non-relational Data
  - Azure Cosmos DB — Globally distributed database service
  - Azure SQL database (SQL server)
- Azure Queue storage
  - Each message upto 64KB
  - Million of messages
  - Async access
  - One usecase with Az functions

Std - HDD
Premium - SSD

Std storage cannot be converted to Premium storage

Data Replication

- Locally redundant (LRS)
  - within in same zone to another files
  - → redundancy for rack or device failure
  - → 3 copies in same zone
- Zone redundant (ZRS)
  - → Data backed up in another zone within same region ~~zone~~
  - → 3 copies in same zone ~~and 1 copy in another zone~~ but independent clusters
  - →
- Geo redundant (GRS) - 16 9's durability
  - → 3 copies in same region and 1 async copy in another region. Only primary regions serves data unless failove triggers
  - → Read Access GRS (RAGRS) - to be configured if secondary region - to provide RO access before failover
- Geo zone redundant (GZRS) -
  - ↳ 3 copies in Each AZ in same region and 3 copies in secondary region.

Azure storage
No HTTPS with custom domain
CDN tobe Used insteed

Dircet mapping in custom domain mapping does not have downtime
Intermediary domain mapping has downtime. Users "asverify"

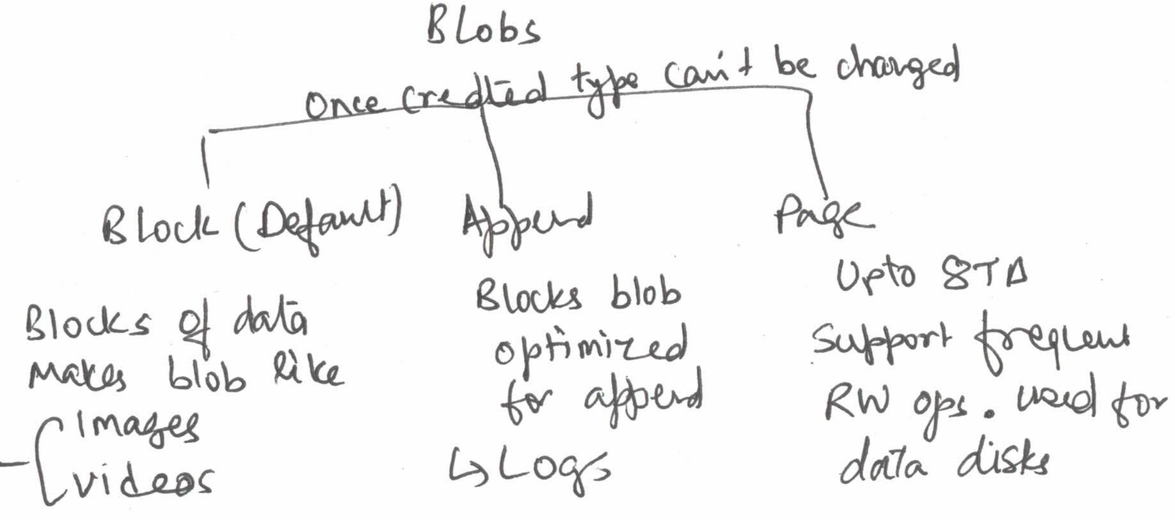
Blob storage life cycle management policy rules

1 ~~modi~~ Transition blob to another tier
2 Delete blob at end of lifecycle
3 Define rule based condition for per day at A/c level
4 Apply rule based conditions to container or blobs.

- Azure Blob storage (Unstructured data text, binary images, video app installers) — Storage in blob container (name to be unique across Azure globally)
  - Private — No anonymous access
  - Blob — anonymous public RO to blobs only
  - Container — anonymous public RO to both container & Blobs.
- Tiers
  - HOT — frequent RW — Default creation mode
  - COLD — Data stays for atleast 90 days.
  - COOL — Infrequently accessed — Data stays for atleast 30 days. Costlier to access ~~log~~ but cheaper to keep data.
  - Archive — Data stays for atleast 180 days. time to first byte is hours

Access tier for a/c can be changed anytime.

blob objects can be replicated from one tier to another tier.



Blobs

Once created type can't be changed

Block (Default)

Blocks of data

Makes blob like

{ Images

videos

Append

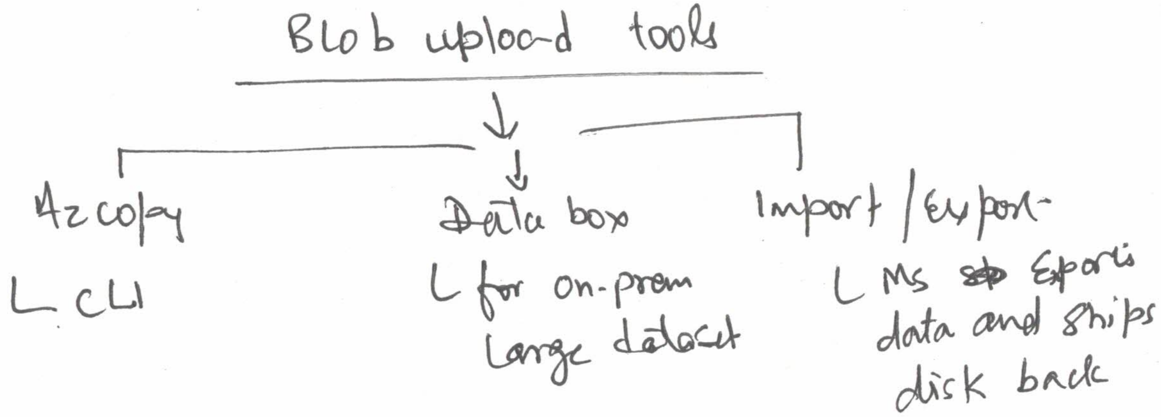
Blocks blob optimized for append

↳ Logs

Page

Upto 8TB

Support frequent RW ops. used for data disks



Blob upload tools

Azcopy

∟ CLI

Data box

∟ for on-prem Large dataset

Import / export

∟ MS ~~sp~~ Exports data and ships disk back

Characteristics of a SAS.

- An account-level SAS can delegate access to multiple Azure Storage services, such as blobs, files, queues, and tables.
- You can specify the time interval for which a SAS is valid, including the start time and the expiration time.
- You specify the permissions granted by the SAS. A SAS for a blob might grant read and write permissions to that blob, but not delete permissions.
- SAS provides account-level and service-level control.
    - Account-level SAS delegates access to resources in one or more Azure Storage services.
    - Service-level SAS delegates access to a resource in only one Azure Storage service.

Note

A stored access policy can provide another level of control when you use a service-level SAS on the server side. You can group SASs and provide other restrictions by using a stored access policy.

- There are optional SAS configuration settings:
    - IP addresses. You can identify an IP address or range of IP addresses from which Azure Storage accepts the SAS. Configure this option to specify a range of IP addresses that belong to your organization.
    - Protocols. You can specify the protocol over which Azure Storage accepts the SAS. Configure this option to restrict access to clients by using HTTPS.

SAS Url has below fields-

Resource URI, Storage version, Storage service, Start time(optional) ,Expiry time,Resource,Permissions,IP range,Protocol,Signature

SAS in your application, there can be potential risks.

- If a SAS is compromised, it can be used by anyone who obtains it, including a malicious user.
- If a SAS provided to a client application expires and the application is unable to retrieve a new SAS from your service, the application functionality might be hindered.

Azure Files (file shares)

Azure Files provides the SMB and NFS protocols, client libraries, and a REST interface that allows access from anywhere to stored files.

- Files in an Azure Files share are true directory objects.
- Data in Azure Files is accessed through file shares across multiple virtual machines.

***Azure Files** is ideal to lift and shift an application to the cloud that already uses the native file system APIs. Share data between the app and other applications running in Azure.*

Azure Files is a good option when you want to store development and debugging tools that need to be accessed from many virtual machines.

Azure Blob Storage (blobs)

Azure Blob Storage provides client libraries and a REST interface that allows unstructured data to be stored and accessed at a massive scale in block blobs.

- Blobs in Azure Blob Storage are a flat namespace.
- Blob data in Azure Blob Storage is accessed through a container.

***Azure Blob Storage** is ideal for applications that need to support streaming and random-access scenarios.*

Azure Blob Storage is a good option when you want to be able to access application data from anywhere.

Azure Disks (page blobs)

Azure Disks is similar to Azure Blob Storage. Azure Disks provides a REST interface to store and access index-based or structured data in page blobs.

- Page blobs in Azure Disks are stored as 512-byte pages.
- Page blob data is exclusive to a single virtual machine.

***Azure Disks** solutions are ideal when your applications run frequent random read/write operations.*

Azure Disks is a good option when you want to store operating system and data disks in Azure Virtual Machines.

Replicate Azure file shares to Windows Servers by using Azure File Sync.

Azure file shares don't support both the SMB and NFS protocols on the same file share, although you can create SMB and NFS Azure file shares within the same storage account.

Standard (HDD) file shares can be used with SMB and REST protocols only, while premium (SSD) file shares can be used with SMB, NFS, and REST protocols.

You can easily switch between hot, cool, and transaction optimized tiers of standard file shares, but you can't switch from premium file shares to any of the standard tiers.

Two important settings that you need to be aware of when creating and configuring SMB Azure file shares.

- **Open port 445**. Azure Files uses the SMB protocol. SMB communicates over TCP port 445. Be sure port 445 is open. Also, make sure your firewall isn't blocking TCP port 445 from the client machine. If you can't unblock port 445, then a VPN or ExpressRoute connection from on-premises to your Azure network is required, with Azure Files exposed on your internal network using private endpoints.

- **Enable secure transfer**. The Secure transfer required setting enhances the security of your storage account by limiting requests to your storage account from secure connections only. Consider the scenario where you use REST APIs to access your storage account. If you attempt to connect, and secure transfer required is enabled, you must connect by using HTTPS. If you try to connect to your account by using HTTP, and secure transfer required is enabled, the connection is rejected.

Azure file shares can be mounted in Linux distributions by using the CIFS kernel client.

Characteristics of Azure file share snapshots.

- The Azure Files share **snapshot capability is provided at the file share level.**
- **Share snapshots are incremental in nature.** Only data changed since the most recent share snapshot is saved.
- Incremental snapshots minimize the time required to create share snapshots and saves on storage costs.
- Even though share snapshots are saved incrementally, you only need to retain the most recent share snapshot to restore the share.
- You can retrieve a share snapshot for an individual file. This level of support helps with restoring individual files rather than having to restore to the entire file share.
- If you delete a file share that has share snapshots, all of its snapshots will be deleted along with the share.

Characteristics of soft delete for Azure Files.

- Soft delete for file shares is enabled at the file share level.
- Soft delete transitions content to a soft deleted state instead of being permanently erased.

- Soft delete lets you configure the retention period. The retention period is the amount of time that soft deleted file shares are stored and available for recovery.
- Soft delete provides a retention period between 1 and 365 days.
- Soft delete can be enabled on either new or existing file shares.

Cloud tiering

Cloud tiering is an optional feature of Azure File Sync. Frequently accessed files are cached locally on the server while all other files are tiered to Azure Files based on policy settings.

- When a file is tiered, Azure File Sync replaces the file locally with a pointer. A pointer is commonly referred to as a *reparse point*. The reparse point represents a URL to the file in Azure Files.
- When a user opens a tiered file, Azure File Sync seamlessly recalls the file data from Azure Files without the user needing to know that the file is stored in Azure.
- Cloud tiering files have greyed icons with an offline O file attribute to let the user know when the file is only in Azure

Azure groups four of these data services together under the name *Azure Storage*. The four services are:

- Azure Blobs
- Azure Files
- Azure Queues
- Azure Tables

A *storage account* is a container that groups a set of Azure Storage services together. Only data services from Azure Storage can be included in a storage account (Azure Blobs, Azure Files, Azure Queues, and Azure Tables). It has globally unique name.

A storage account is an Azure resource and is part of a resource group.

Azure data services, such as Azure SQL and Azure Cosmos DB, are managed as independent Azure resources and can't be included in a storage account.

A storage account represents a collection of settings like location, replication strategy, and subscription owner. You need one storage account for each group of settings that you want to apply to your data.

Azure Storage supports three types of shared access signatures (SAS):

- **User delegation SAS**: A user delegation SAS is secured with Microsoft Entra credentials, because the OAuth 2.0 token used to sign the SAS is requested on behalf of the user. It can only be used for Blob storage.
- **Service SAS**: A service SAS is secured using a storage account key. A service SAS delegates access to a resource in any one of four Azure Storage services: Blob, Queue, Table, or File.
- **Account SAS**: An account SAS is secured with a storage account key. An account SAS has the same controls as a service SAS, but can also control access to service-level operations, such as Get Service Stats.

Service SAS can be associated a stored access policy. A stored access policy can be associated with up to five active SASs. You can control access and expiration at the stored access policy level.

Storage Explorer supports two emulators: Azure Storage Emulator and Azurite.

- Azure Storage Emulator uses a local instance of Microsoft SQL Server 2012 Express LocalDB. It emulates Azure Table, Queue, and Blob storage.
- Azurite, which is based on Node.js, is an open-source emulator that supports most Azure Storage commands through an API.

two permissions to access your Azure Storage account: management and data. However, you can use Storage Explorer with only the data-layer permission.