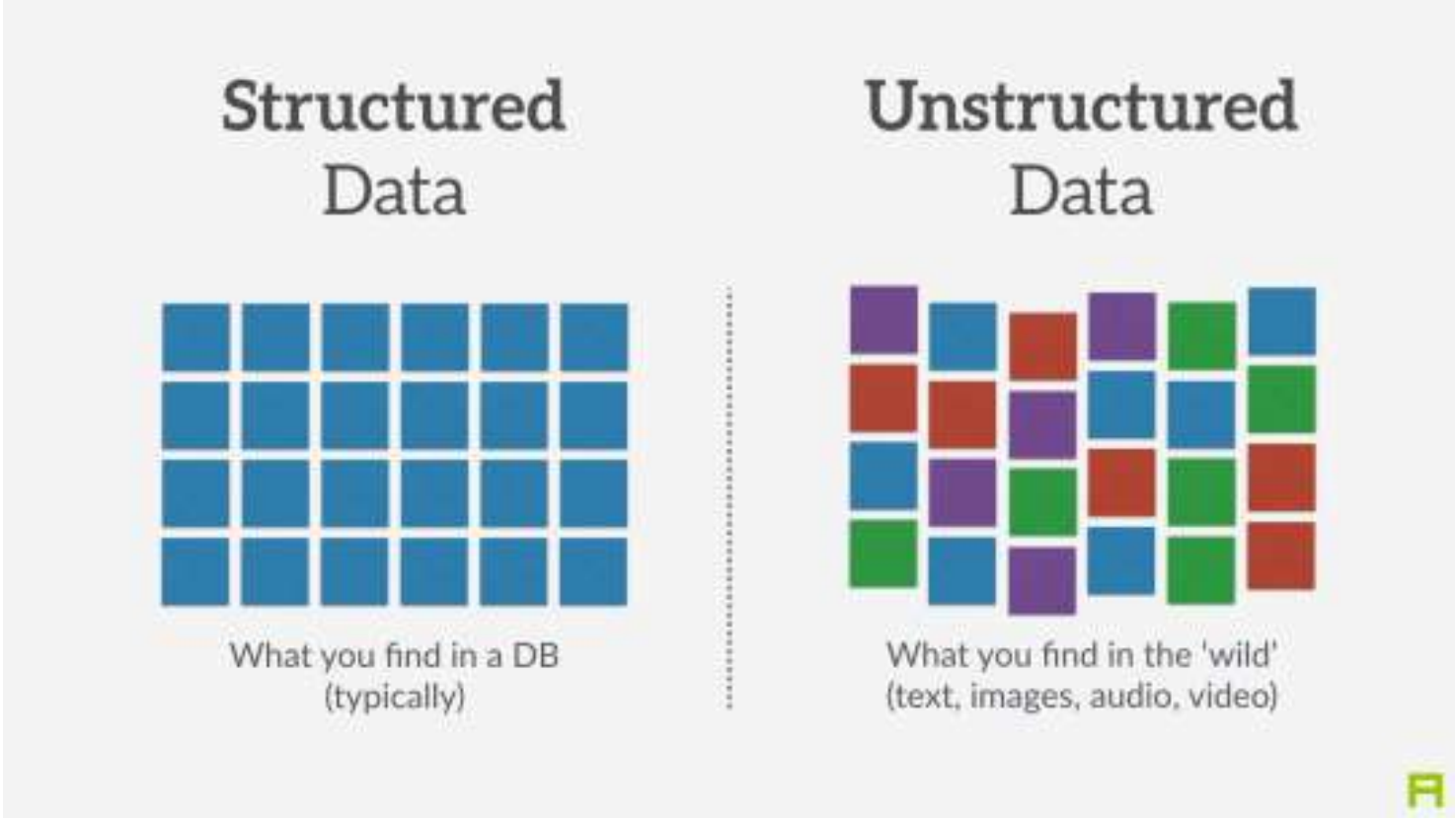
Module 05:
Azure Storage

Azure Storage

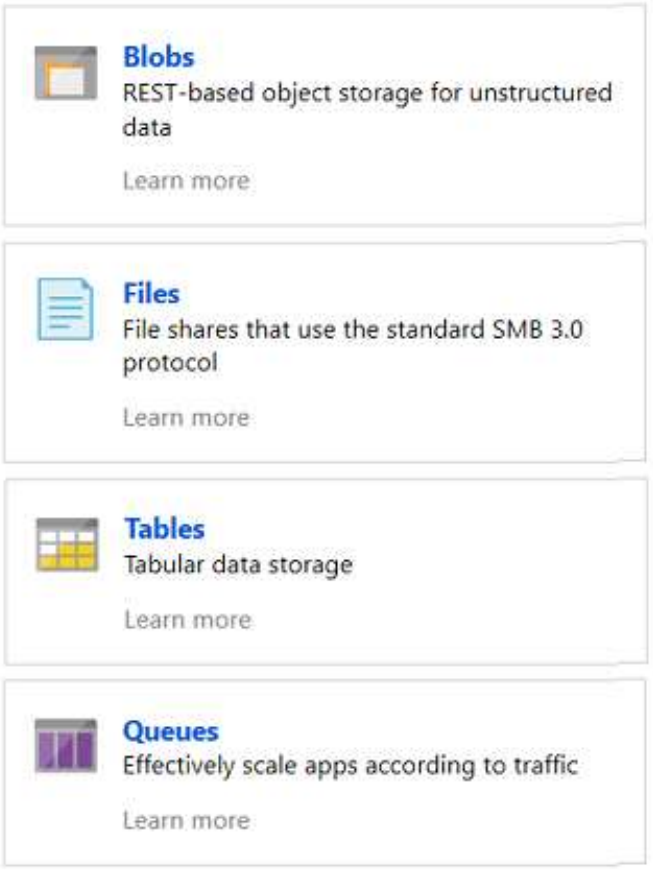
Azure Storage is Microsoft's cloud storage solution for modern data storage scenarios. Azure Storage offers a massively scalable object store for data objects, a file system service for the cloud, a messaging store for reliable messaging, and a NoSQL store.

Azure Storage

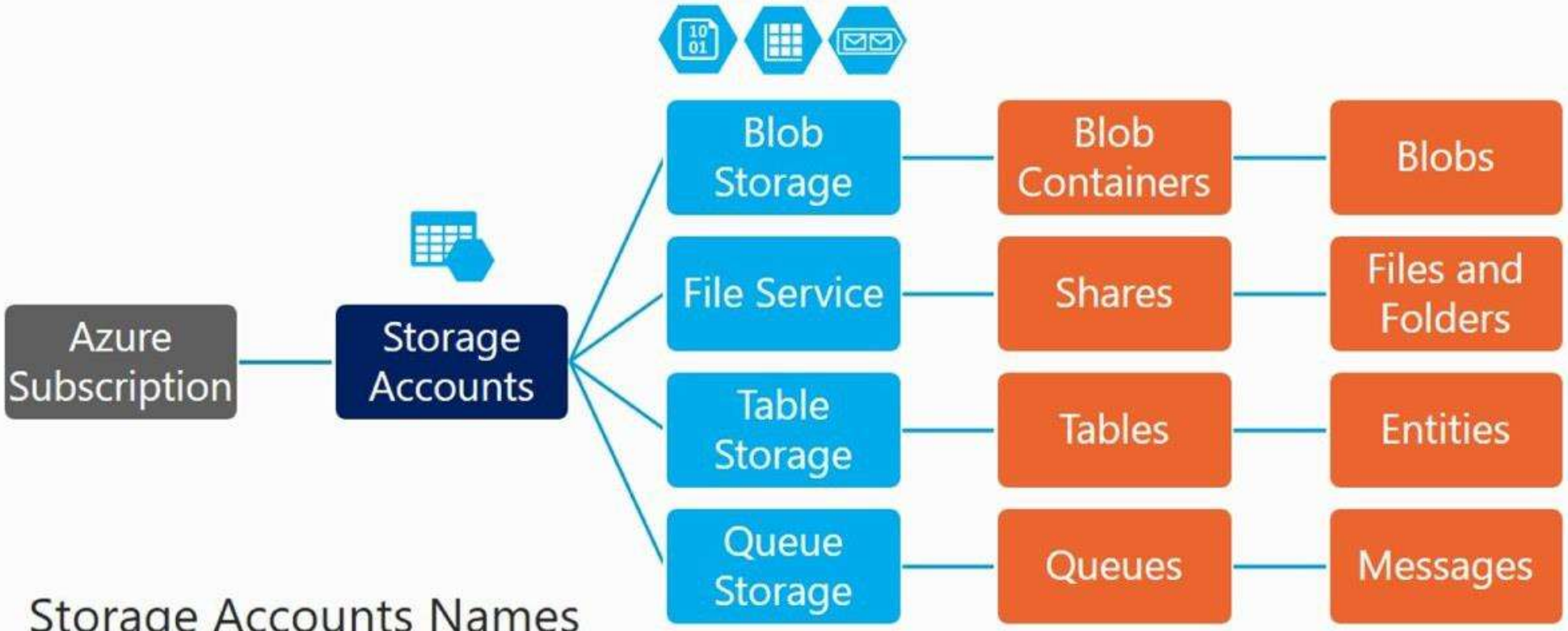


Azure Storage Services

- Azure Blobs: A massively scalable object store for text and binary data
- Azure Files: Managed file shares for cloud or on-premises deployments
- Azure Tables: A NoSQL store for schema less storage of structured data
- Azure Queues: A messaging store for reliable messaging between application components



Container for Storage Services



- Storage Accounts Names
  - 3-24 Characters Alpha-Numeric all lowercase

Components of an Azure Storage Account

- Each Storage Service has its own http:// endpoint
  - https://opsstorage.blob.core.windows.net
  - https://opsstorage.table.core.windows.net
  - https://opsstorage.queue.core.windows.net
  - https://opsstorage.file.core.windows.net

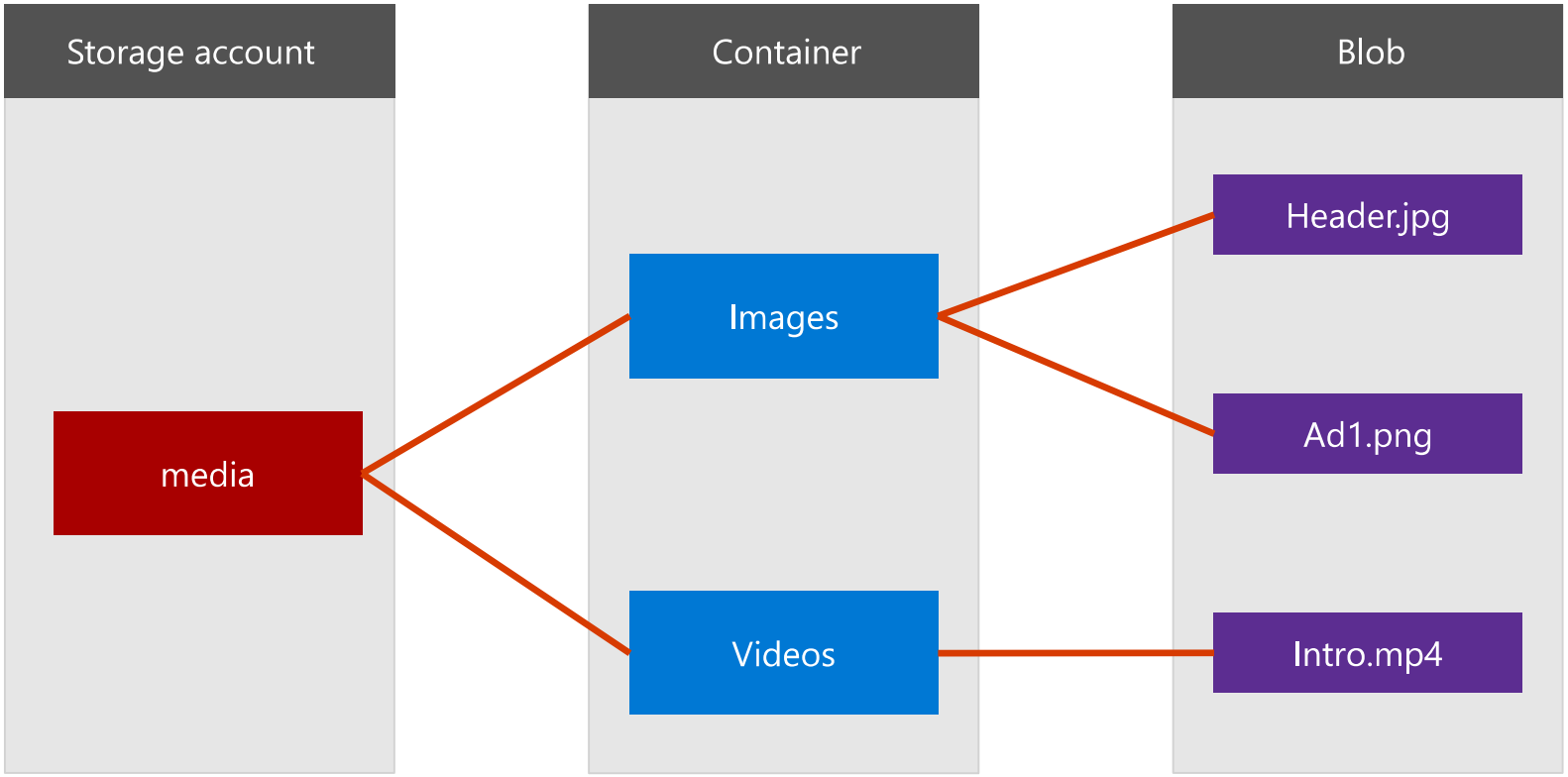
Up to 500 TBs of capacity
20,000 IOPS
Up to 20 Gbps in and 30 out (LRS)
Up to 10 Gbps in and 20 out (GRS)
Security though management certificate or name and authentication key

Blob Storage

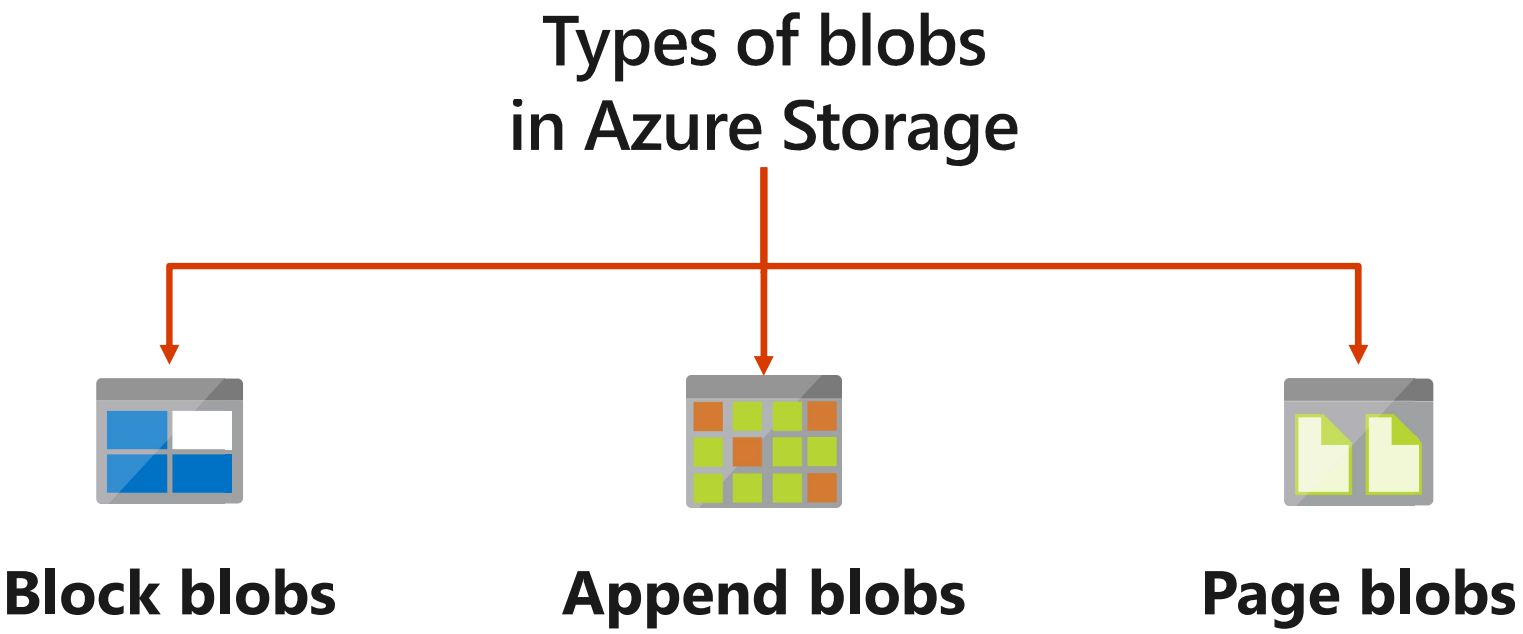
Azure Blob storage

- Object storage solution in the cloud
- Blob storage is designed for:
  - Serving images or documents directly to a browser
  - Storing files for distributed access
  - Streaming video and audio
  - Writing to log files
  - Storing data for backup and restore, disaster recovery, and archiving
  - Storing data for analysis by an on-premises or Azure-hosted service
- Accessible via a HTTP/HTTPS API

Azure Blob storage resource hierarchy

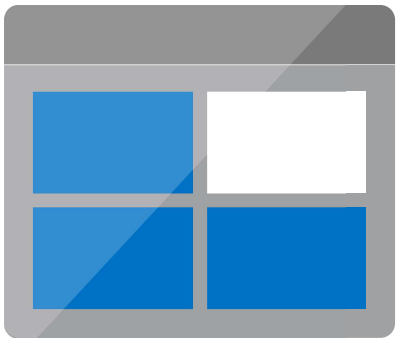


Blob types


Types of blobs
in Azure Storage
Block blobs
Append blobs
Page blobs

Block blobs

- Comprise blocks of data
- Ideal for data that is stored in blocks—up to 100-MB chunks Ex. Images, Videos, Documents, etc
- Simultaneous upload of large blobs with a single write operation
- A single block blob can include up to 50,000 blocks



Append blobs

- Append blobs include the following characteristics:
  - They are composed of blocks
  - They are optimized for append operations
  - They are ideal for performant logging

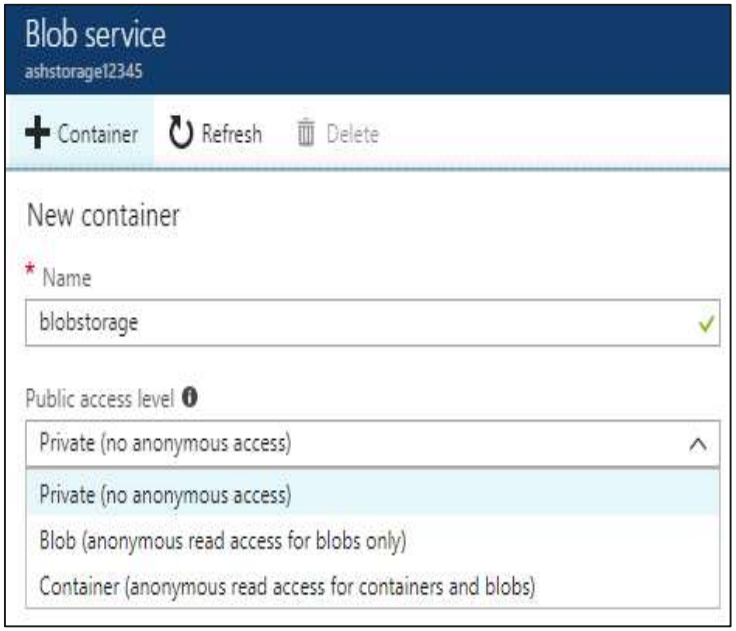
Page blobs

- Composed of 512-byte pages
- Similar to hard disk storage
- Ideal for virtual hard disks
- Pages created by initializing the page blob and specifying the size
- Content to be added within 512-byte page boundaries
- Writes to page blobs commit immediately

Blob Containers

- All blobs must be in a container
- Accounts have unlimited containers
- Containers can have unlimited blobs

- **Private blobs -** no anonymous access
- **Blob access** - anonymous public read access for blobs only
- **Container access** - anonymous public read and list access to the entire container, including the blobs



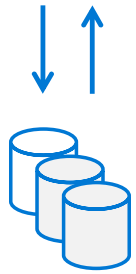
Data Replication Overview

- Locally-redundant Storage
- Zone-redundant Storage
- Geo-redundant Storage
- Geo-zone-redundant Storage
- Comparing Replication Strategies

Locally-redundant Storage

Replication	Copies	Strategy
Locally redundant storage (LRS)	Maintains three copies of your data.	Data is replicated three time within a single facility in a single region.

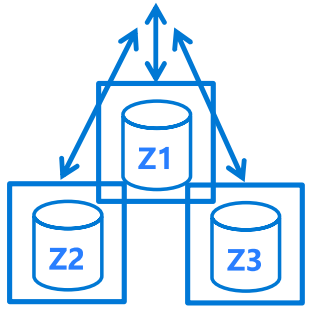
- Lowest cost option
- Use if data can be easily reconstructed
- Use if data is “live” and does not require storage
- Use if there are regional governance requirements



Zone-redundant Storage

Replication	Copies	Strategy
Zone-redundant storage (ZRS)	Maintains three copies of your data.	Data is replicated across three storage clusters in a single region.

- Replicates your data across three storage clusters in a single region
- Each storage cluster is physically separated from the others and resides in its own availability zone
- Each availability zone, and the ZRS cluster within it, is autonomous, with separate utilities and networking capabilities
- Consistency, durability, and high availability scenarios



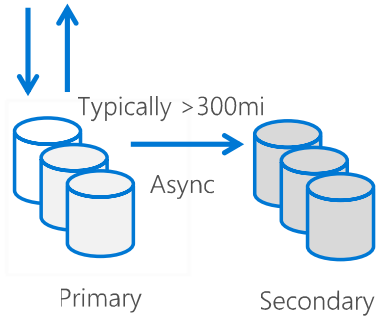
Geo-redundant Storage

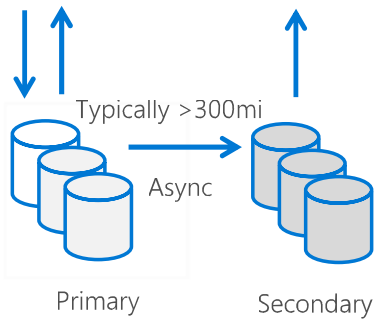
Replication	Copies	Strategy
Geo-redundant storage (GRS)	Maintains six copies of your data.	Data is replicated three times within the primary region and is also replicated three times in a secondary region.
Read access geo-redundant storage (RA-GRS)	Maintains six copies of your data.	Data is replicated to a secondary geographic location and provides read access to your data in the secondary location.

- GRS replicates your data to another data center in a secondary region, but that data is available to be read only during a failure
- RA-GRS is based on GRS and replicates data to another data center in another region. Provides read access from the secondary region, even without a failure
- Consider the Recovery Time Objective and Recovery Point Objective

Geo-redundant Storage

Replication	Copies	Strategy
Geo-redundant storage (GRS)	Maintains six copies of your data.	Data is replicated three times within the primary region and is also replicated three times in a secondary region.
Read access geo-redundant storage (RA-GRS)	Maintains six copies of your data.	Data is replicated to a secondary geographic location and provides read access to your data in the secondary location.





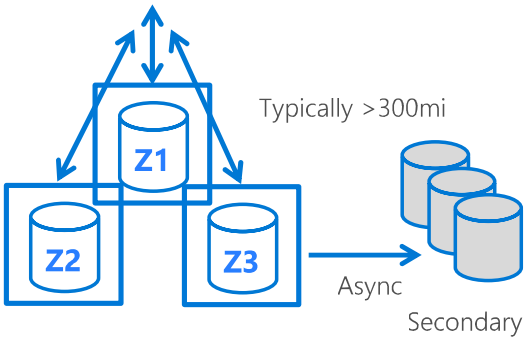
Geo-zone-redundant Storage

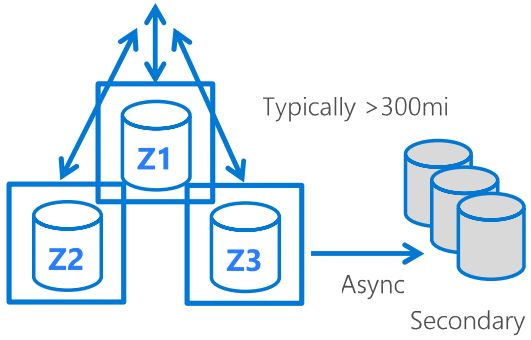
Replication	Strategy
Geo-zone-redundant storage (GZRS)	Data is replicated across three Azure availability zones in the primary region and replicated to a secondary geographic region for protection from regional disasters.
Read access geo-zone-redundant storage (RA-GZRS)	Enables read access to data in the secondary GZRS region.

- Combines the features of ZRS with GRS
- Consistency, durability, high availability, excellent performance, and resilience for disaster recovery

Geo-zone-redundant Storage

Replication	Strategy
Geo-zone-redundant storage (GZRS)	Data is replicated across three Azure availability zones in the primary region and replicated to a secondary geographic region for protection from regional disasters.
Read access geo-zone-redundant storage (RA-GZRS)	Enables read access to data in the secondary GZRS region.





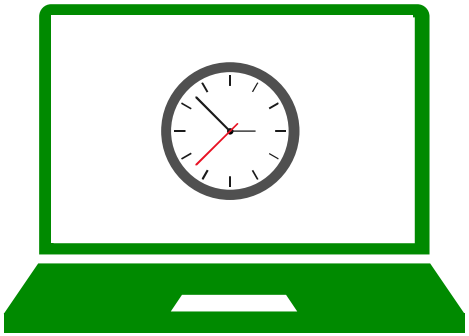
RA-GZRS

Comparing Replication Strategies

Replication Option	LRS	ZRS	GRS/RA-GRS	GZRS/RA-GZRS
Node unavailability within a data center	Yes	Yes	Yes	Yes
An entire data center (zonal or non-zonal) becomes unavailable	No	Yes	Yes	Yes
A region-wide outage	No	No	Yes	Yes
Read access to your data (in a remote, geo-replicated region) for region-wide unavailability	No	No	Yes (with RA-GRS)	Yes (with RA-GZRS)
Available in storage account types	GPv1, GPv2, Blob	GPv2	GPv1, GPv2, Blob	GPv1, GPv2, Blob

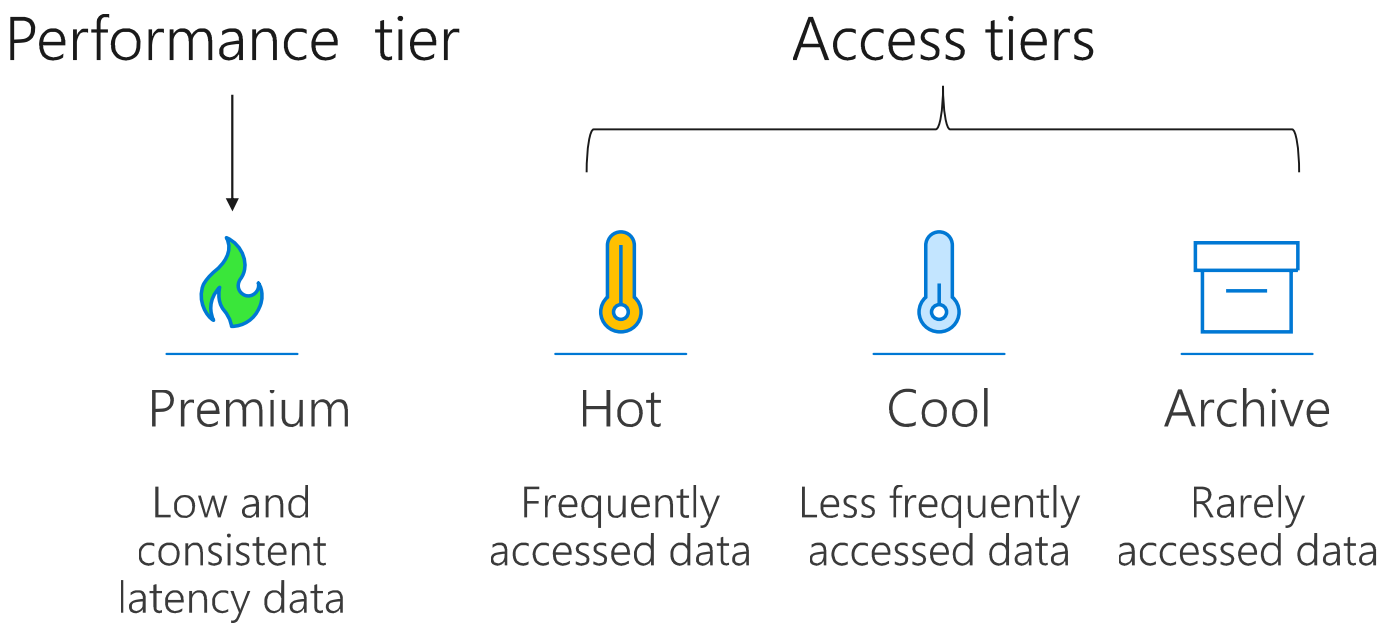
Demo & Lab:

- Create a container
- Upload a block blob
- Download a block blob



Storage tiers

You can use storage tiers to tune performance and cost to a ratio that's ideal for your solution



Storage tier pricing

Premium

Low and consistent latency data

Hot

Frequently accessed data

Cool

Less frequently accessed data

Archive

Rarely accessed data

Storage tier pricing (continued)

Premium

Low and consistent latency data

Hot

Frequently accessed data

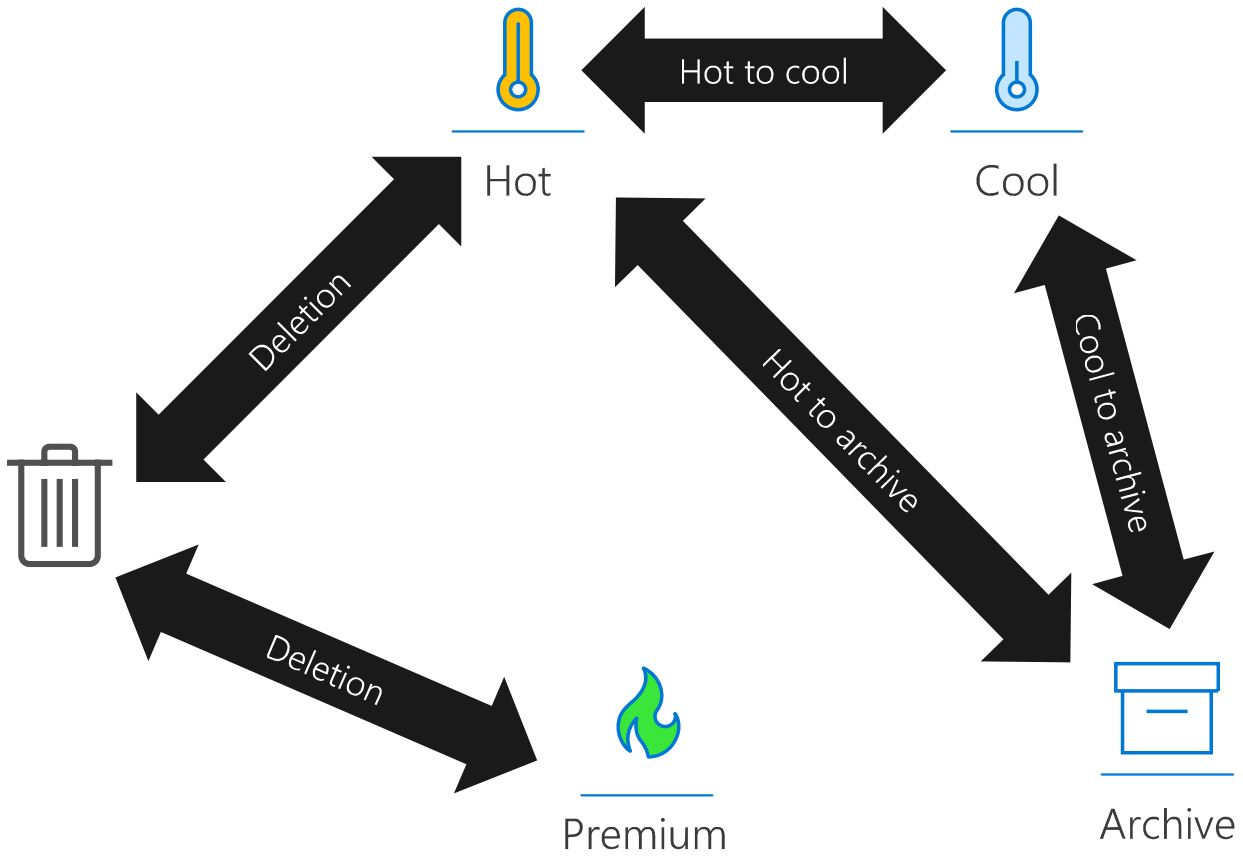
Cool

Less frequently accessed data

Archive

Rarely accessed data

Example of lifecycle management flows



Blob Access Policies

- Provides an additional level of control over server-side SAS
- Groups SAS to provide additional restrictions for signatures bound to the policy
- Supported for Blob containers, File shares, Tables, and Queues

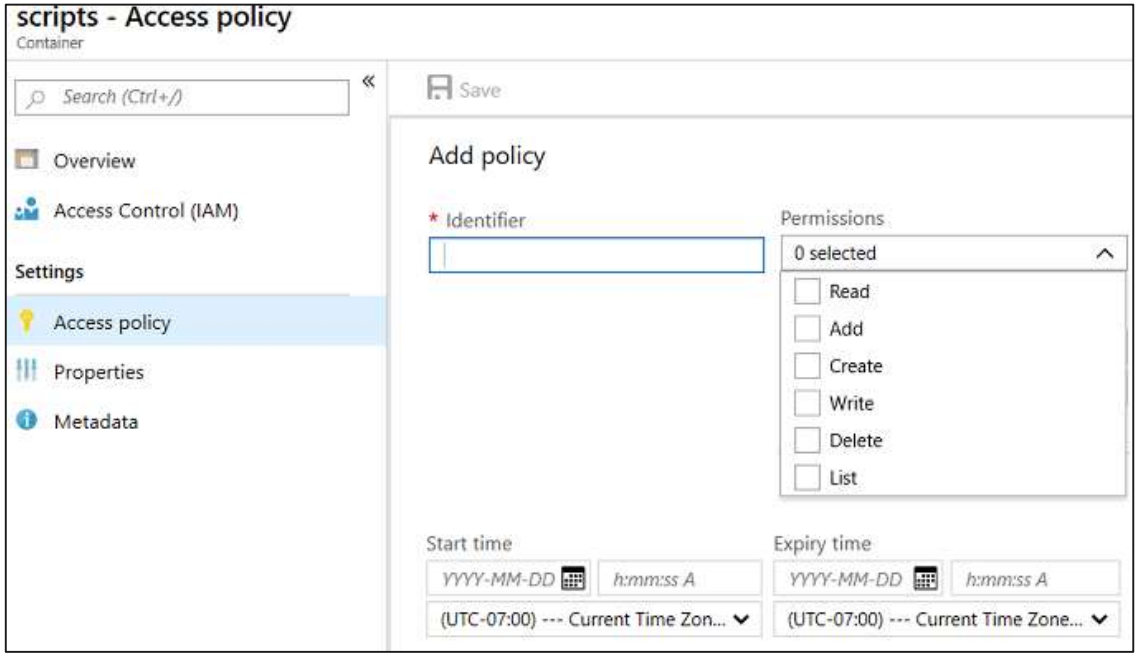


Table Storage

Azure Table storage

- Massive auto-scaling NoSQL Store
- User, device and service metadata, structured data
- Schema-less entities with string consistency
- No limits on number of table rows or table size
- Dynamic load balancing of table regions
- Best for Key/Value lookups on Partition key and Row key
- Entity group transactions for atomic batching

Table Storage Concepts

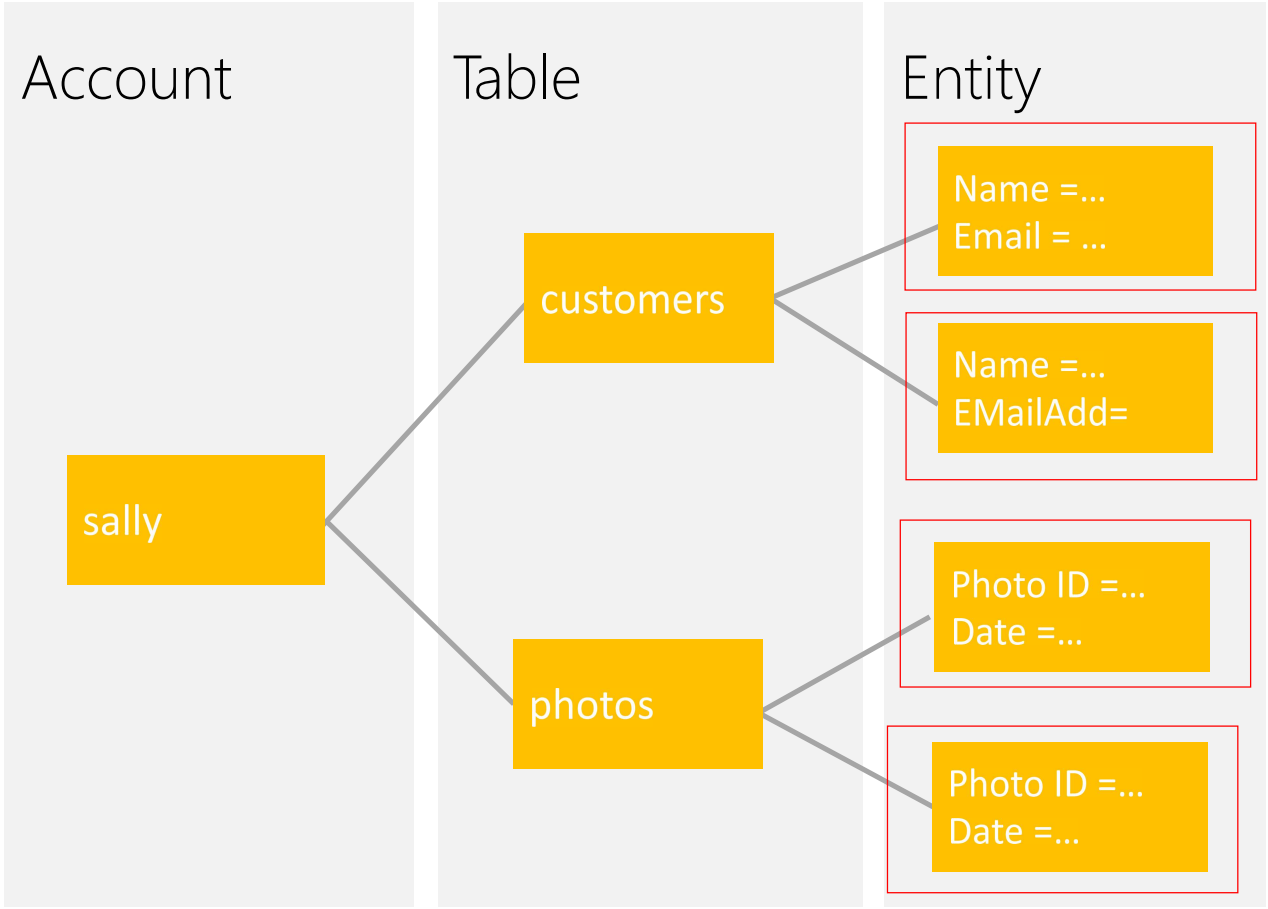


Table Storage

- Store large amounts of structured data
- A NoSQL key-value data store
- Authenticates from inside and outside the Azure cloud.
- Good for web scale applications
- Store datasets for non-complex jobs, foreign keys, or stored procedures
- Quickly query data using a clustered index

Implementing Table Storage

PartitionKey	RowKey	Timestamp	ID	Name
Partition1	Row1	10/10/2016 5:24:54 PM	1	Chris1
Partition1	Row2	10/10/2016 5:23:35 PM	2	Jessie
Partition2	Row1	10/10/2016 5:23:35 PM	3	Christine
Partition2	Row2	10/10/2016 5:23:36 PM	4	Steven

- An entity can have up to 225 properties, including three system properties
- **PartitionKey** and **RowKey** must uniquely identify every entity within a table
- Azure assigns the **Timestamp** and it cannot be modified

No Fixed Schema

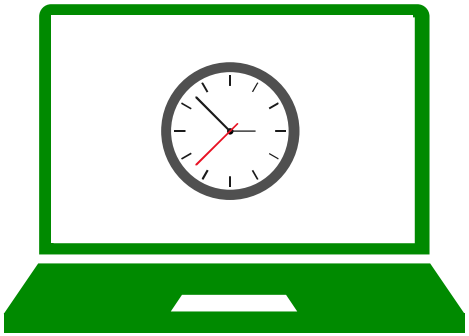
FIRST	LAST	BIRTHDATE	FAV SPORT
Wade	Wegner	2/2/1981	
Nathan	Totten	3/15/1965	Canoeing
Nick	Harris	May 1, 1976	

PartitionKey and RowKey

PARTITIONKEY (CATEGORY)	ROWKEY (TITLE)	TIMESTAMP	MODELYEAR
Bikes	Super Duper Cycle	…	2009
Bikes	Quick Cycle 200 Deluxe	…	2007
…	…	…	…
Canoes	Whitewater	…	2009
Canoes	Flatwater	…	2006
Rafts	14ft Super Tourer	…	1999
…	…	…	…
Skis	Fabrikam Back Trackers	…	2009
…	…	…	…
Tents	Super Palace	…	2008
…	…	…	…
Tents	Super Palace	…	2008

Demo:

Implement Azure Storage Table



Queue Storage

Azure Queue storage

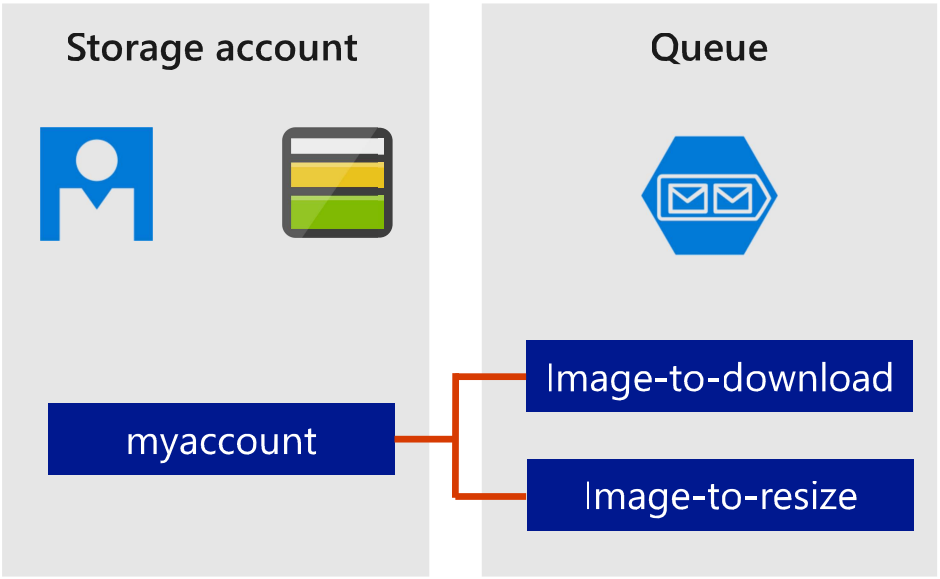
- Service for storing messages in an Azure Storage account
  - Accessed using HTTP or HTTPS
  - Scalable to millions of messages
- Common uses of Queue storage include:
  - Creating a backlog of work to process asynchronously
  - Passing messages from an Azure web role to an Azure worker role
  - Reliable messaging system at scale for cloud services
  - A message can be up to 64 KB in size

Components

- URL format

  `https://<storage-account>.queue.core.windows.net/<queue>`

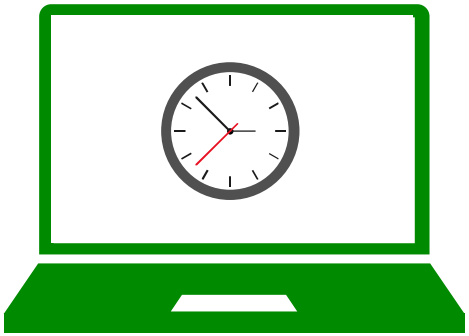
- Storage account
- Queue
- Message



Queue service components

Demo:

Asynchronously processing messages using Azure Storage queues



File Storage

Azure Files

- Managed file shares in the cloud that are accessible via SMB
- Common uses:
  - Replace and supplement
  - Lift and shift
  - Azure File Sync
  - Shared applications
  - Diagnostic data
  - Tools and utilities

Files vs Blobs

Feature	Description	When to use
Azure Files	SMB interface, client libraries, and a REST interface that allows access from anywhere to stored files.	• Lift and shift an application to the cloud. • Store shared data across multiple virtual machines. • Store development and debugging tools that need to be accessed from many virtual machines.
Azure Blobs	Client libraries and a REST interface that allows unstructured data (flat namespace) to be stored and accessed at a massive scale in block blobs.	• Support streaming and random-access scenarios. • Access application data from anywhere.

Creating File Shares

- Portal

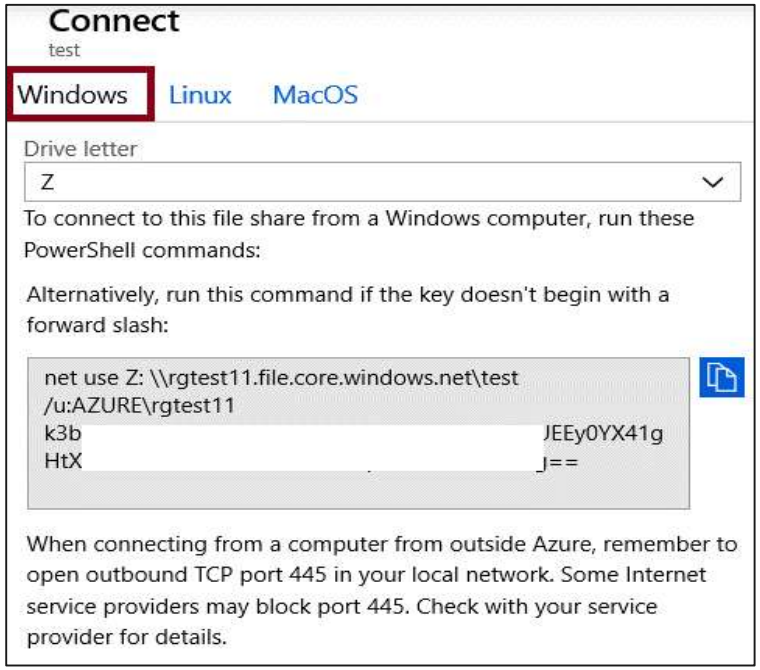
- PowerShell

```
# Retrieve storage account and storage account key
$storageContext = New-AzStorageContext  <storage-account-name>
<storage-account-key>
# Create the file share, in this case “logs”
$share = New-AzStorageShare  logs -Context $storageContext
```

Mapping File Shares (Windows)

- Mapping drive letter
- UNC path
- Account user
- Storage Account Key

✔ Ensure port 445 is open



Mounting File Shares (Linux)

```
sudo mount -t cifs //<storage-account-
name>.file.core.windows.net/<share-name> <mount-point> -o
vers=<smb-version>,username=<storage-account-
name>,password=<storage-account-
key>,dir_mode=0777,file_mode=0777,serverino
```

- Install the cifs-utils package
- Meet the SMB client requirements
- Decide on the directory file chmod permissions
- Create an entry in /etc/fstab to persist the mount

Secure Transfer Required

- Only allows requests by secure connection such as HTTPs
- Portal

- PowerShell

```
Set-AzStorageAccount
   -Name "{StorageAccountName}"
   -ResourceGroupName "{ResourceGroupName}"
   -EnableHttpsTrafficOnly $True
```

File Share Snapshots

- Captures the share state at that point in time
- Is read-only copy of your data
- Snapshot at the file share level
- Retrieve at the individual file level
- Incremental in nature
- Uses:
  - Protection against application error and data corruption.
  - Protection against accidental deletions or unintended changes.
  - General backup purposes.

Demo:

- Create a file share and upload a file
- Manage snapshots
- Create a file share (PowerShell)
- Mount a file share (PowerShell)