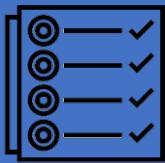


**Implement non-relational data stores**

# Implement non-relational data stores



## Learning Objectives

### NoSQL offerings by Azure

#### Azure Storage

- Demo: Provision
- Replication
- Azure Blob Storage

#### Data Lake

- How Data Lake evolved?
- Data Lake vs Blob
- Security options

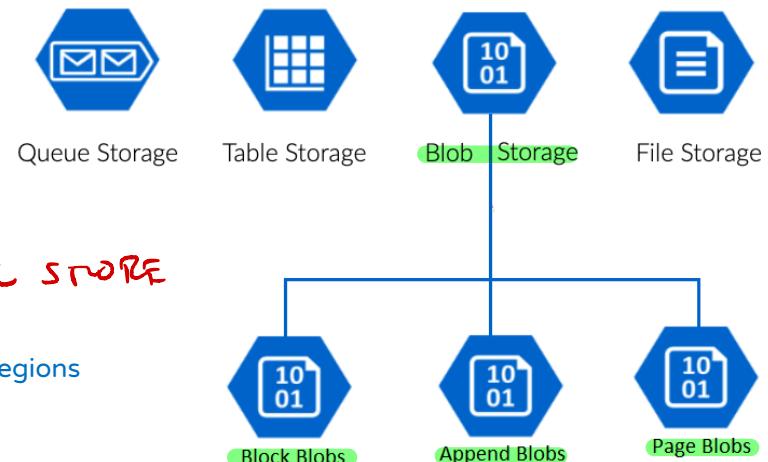
### Cosmos DB

- Features
- Multi-Model and 5 Consistency levels
- Database and Containers
- Throughput and request units (RUs)
- Partitioning and Scaling
- Global Distribution, Multi-master and Failover
- Time to Live, CLI and Pricing
- Security options

# Azure Storage Service

STORAGE ACCOUNT

- Diff types of data and requirements
  - Relational, non-relational/No-SQL, datasheets, images, videos, backups
  - Storage, access, security, availability, latency, processing, backup
- Diff types of Data Service **(4 STORAGE TYPES)**
  - Azure Blobs: Text and binary data
  - Azure Files: Managed file shares (**SMB Protocol**)
  - Azure Queues: Messaging
  - **Azure Tables: NoSQL store**
    - **COSMOS DB PREFERRED NO-SQL STORE**
- Features
  - **Durable and highly available** – redundancy across datacenters or regions
  - Secure – all data **encrypted by default**
  - Scalable – massively scalable
  - **Managed** - Azure handles hardware maintenance, updates, and critical issues for you.
  - **Accessible** - accessible from **anywhere in the world** over HTTP or HTTPS.
    - Clients libraries are available in all languages
    - Support scripting in PowerShell or Azure CLI



# Azure Data Redundancy

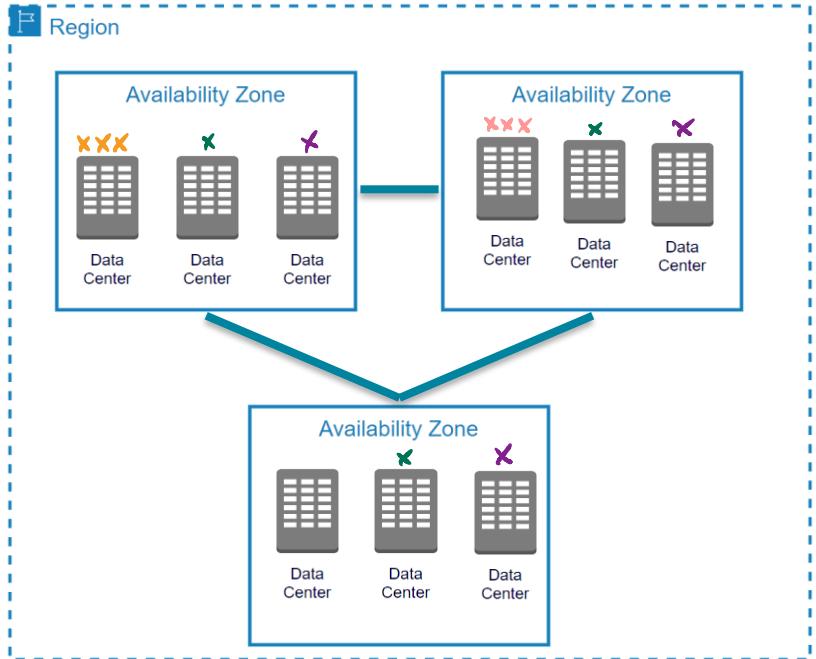
- Protect your data from hardware failures, network or power outages, and massive natural disasters.
- Even in the event of a failure, redundancy ensures your storage account's availability and durability.
- Tradeoffs between lower costs and higher availability
- Redundancy in the primary region
  - Locally redundant storage (LRS) – Three synchronous copies in same data center
  - Zone-redundant storage (ZRS) – Three synchronous copies in three availability zones (AZs)
- Redundancy in a secondary region **READ ONLY**
  - Geo-redundant storage (GRS) – LRS + Asynchronous copy to secondary region ()
  - Geo-zone-redundant storage (GZRS)
- With GRS or GZRS, the data in the secondary region isn't available for read or write access unless there is a failover to the secondary region. — **BYPASS w/ RA-GRS OR RA-GZRS**
- For read access to the secondary region, configure your storage account to use
  - Read-access geo-redundant storage (RA-GRS)
  - Read-access geo-zone-redundant storage (RA-GZRS).

AVAILABILITY ZONES ARE IN  
DIFFERENT PHYSICAL LOCATIONS  
  
REGIONS ARE  
SPREAD OUT  
GLOBALLY

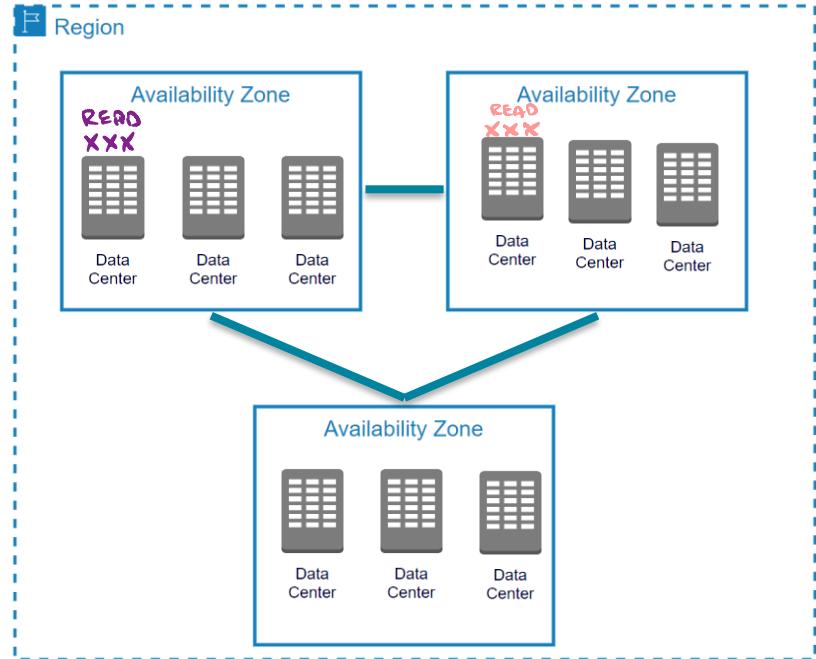
# Azure Storage Redundancy

ZONES ARE GEO DISTANT

EAST US (EXAMPLE)



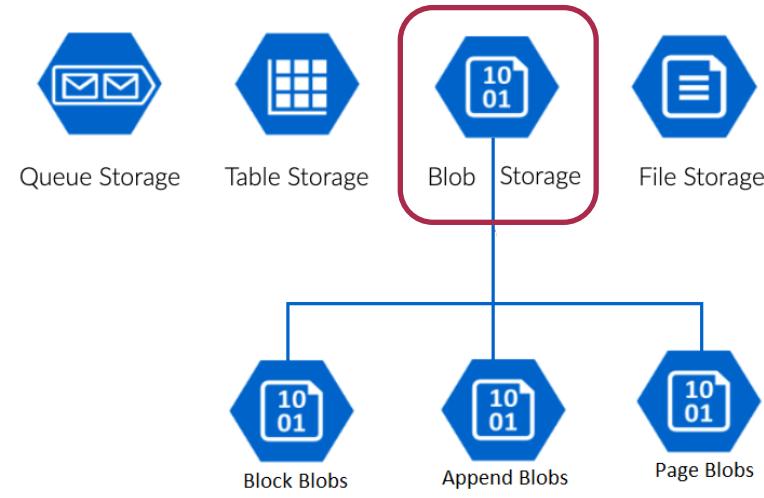
WEST US (EXAMPLE)



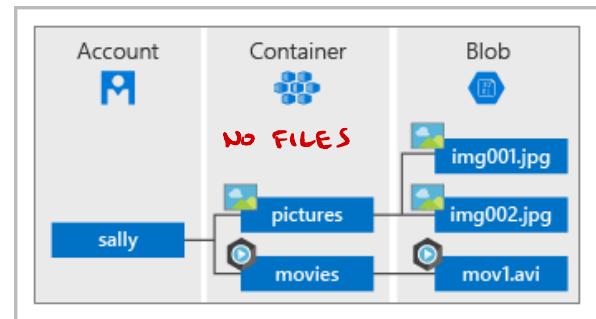
- ■ Locally redundant storage (LRS) – Three synchronous copies in same data center
- ■ Zone-redundant storage (ZRS) – Three synchronous copies in three availability zones (AZs)
- ■ Geo-redundant storage (GRS) - LRS + Asynchronous copy to secondary region (three more copies using LRS) – Read only access
  - Read-access geo-redundant storage (RA-GRS) – Read Access on GRS
- ■ Geo-zone-redundant storage (GZRS) – ZRS + Asynchronous copy to secondary region (three more copies using LRS) – Read only access
  - Read-access geo-zone-redundant storage (RA-GZRS) – Read Access on GZRS

# Blob Storage

- Blob - Binary Large Object
  - Any type or format
  - Text, Images, audio, video, excel, backup files
- Use cases:
  - Storing files for shared access — ACCESS LEVEL
  - Video and audio streaming
  - Storing data for analysis (Data Lake Gen2)
  - Writing to the log file
  - Storing data for disaster recovery, backup, and archiving
- Flat structure SA > CONTAINERS > FILES (8 LOTS)
- Provides a unique namespace in Azure for your data.
  - <http://mystorageaccount.blob.core.windows.net>



FOR HIERARCHICAL STORAGE (CONTAINERS  
WITHIN CONTAINERS) CONVERT STORAGE  
ACCT TO ADLS GEN2 WITH CREATING



# Three types of Blob Storage

## ➤ **Block Blobs:**

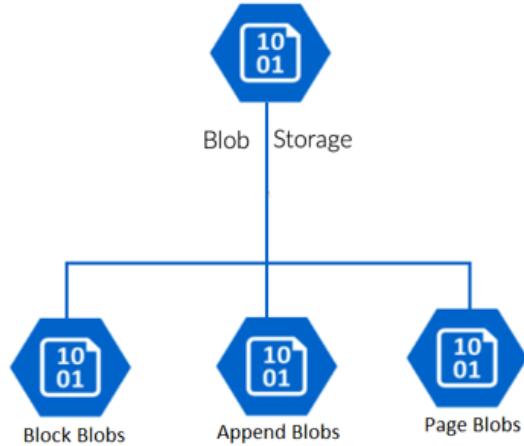
- For large objects that doesn't use random read and write operations, files that are read from beginning to end
- Such as media files or image files for websites.

## ➤ **Page Blobs:**

- Optimized for random read and write operations.
- Provide durable disks for Azure Virtual Machines (Azure VMs)

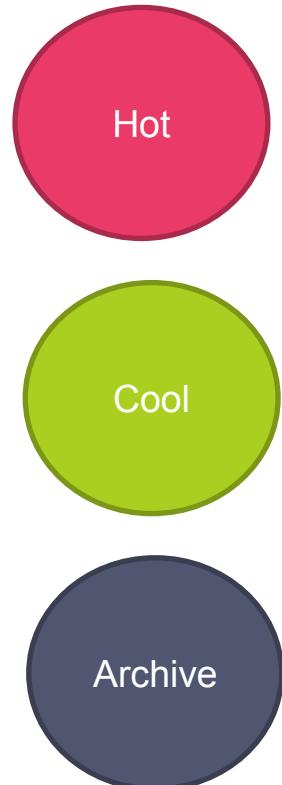
## ➤ **Append Blobs:**

- Optimized for append operations. e. g. Logs
- When you modify an append blob, blocks are added to the end of the blob only
- Updating or deleting of existing blocks is not supported
- For example, you might write all of your trace logging to the same append blob for an application running on multiple VMs



# Storage Access Tiers

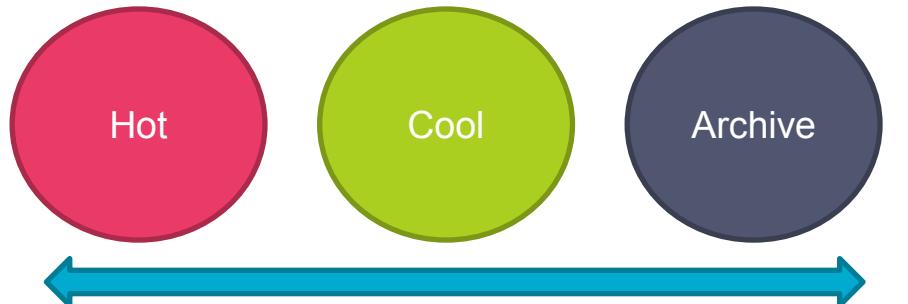
- Data stored in the cloud can be different based on how it's generated, processed, and accessed over its lifetime.
- **Pricing**
  - The **volume** of data stored/month
  - **Types of operations** performed
  - **Number of operations** performed
  - **Data transfer cost**, if any
  - The selected data **redundancy** option
- Organize your data based on **attributes** like frequency of access and planned retention period.
- **Blob access tiers**
  - **Hot** access tier
  - **Cool** access tier
  - **Archive** access tier



# Storage Access Tiers

- **Hot**
  - Frequently accessed data
    - Example - images for your website
  - Low latency
  - Higher access cost
- **Cool**
  - Infrequent accessed data
    - Example - invoices for your customers
  - High latency
  - Lower cost
  - Stored for at least 30 days
- **Archive**
  - Rarely accessed data
    - Example - long-term backups
  - Highest access times and access cost
  - Latency in hours
  - Stored for at least 180 days
  - Use Case: Business policy mandated Data Archiving, long term retention like healthcare data

COOL / ARCHIVE : ADDITIONAL COST FOR REMOVING PRIOR TO MIN STORAGE DURATION



Fast Access  
Higher Cost

Slow Access  
Lower Cost

LATENCY

AUTOMATE DATA  
MOVEMENT WITH  
LIFECYCLE RULES

# Azure Table Storage

- **NoSQL key-value Storage** **NON-RELATIONAL, SEMI-STRUCTURED**
- Items are referred to as rows, and fields are known as columns
- All rows in a table must have a key
- **No concept of relationships, stored procedures, secondary indexes, or foreign keys**
- Data will usually be **denormalized**
- To help ensure fast access, Azure Table Storage splits a table into **partitions**
- Support very large volume of Data
- Consider Cosmos DB for new development
- **Advantages**
  - It's simpler to scale
  - A table can hold semi-structured data
  - No complex relationships
  - Data insertion and retrieval is fast
- Good to use for:
  - Storing TBs of structured data capable of serving web scale applications
  - Storing datasets that don't require complex joins, foreign keys, or stored procedures, and that can be denormalized for fast access.
  - Capturing event logging and performance monitoring data.

Key	COLUMN		
	Value (fields)		
AA	Data for AA	...	...
BB	Data for BB	...	...
CC	Data for CC	...	...
...			

Row —

ZZ	Data for ZZ	...	...
----	-------------	-----	-----

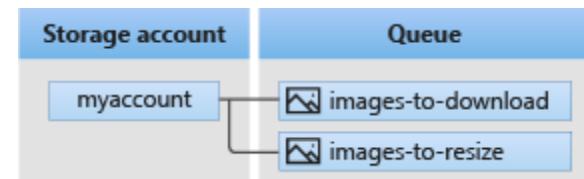
Key (Customer ID)	Value (Customer Data)				
C1	AAAAA	BBB	101 Block Street	YY 999 888	
C2	MM	NN	21 A Street	5 B Avenue	
C3	DDD	EEE	FFF	111 222	66 C Road

# Azure Queue Storage

- Store large numbers of messages.
- Access messages via authenticated calls using [HTTP or HTTPS](#).
- May contain millions of messages, up to the total capacity limit of a storage account.
- Queues are commonly used to create a backlog of work to process asynchronously.

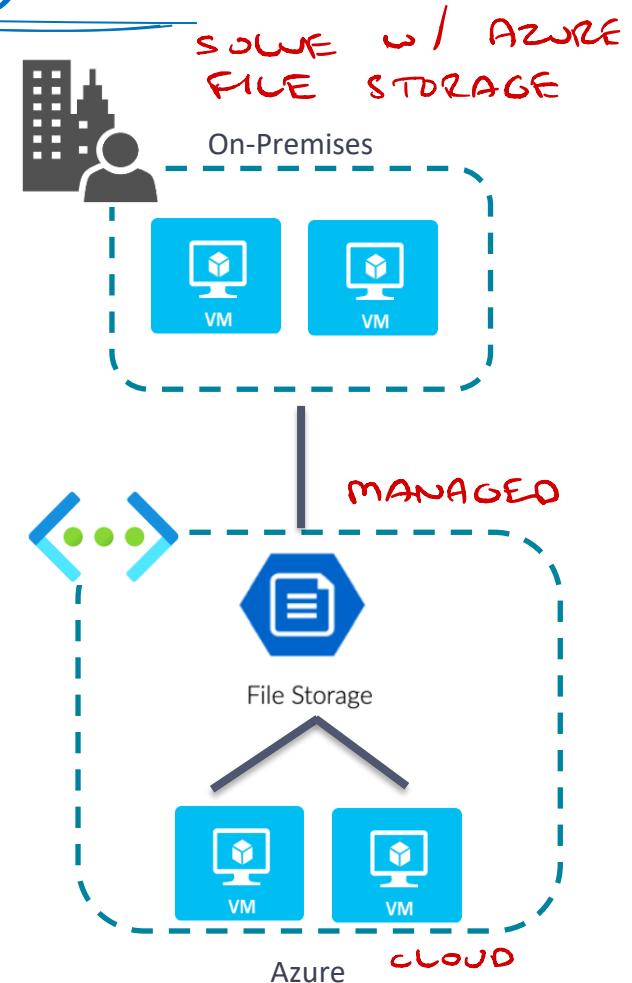
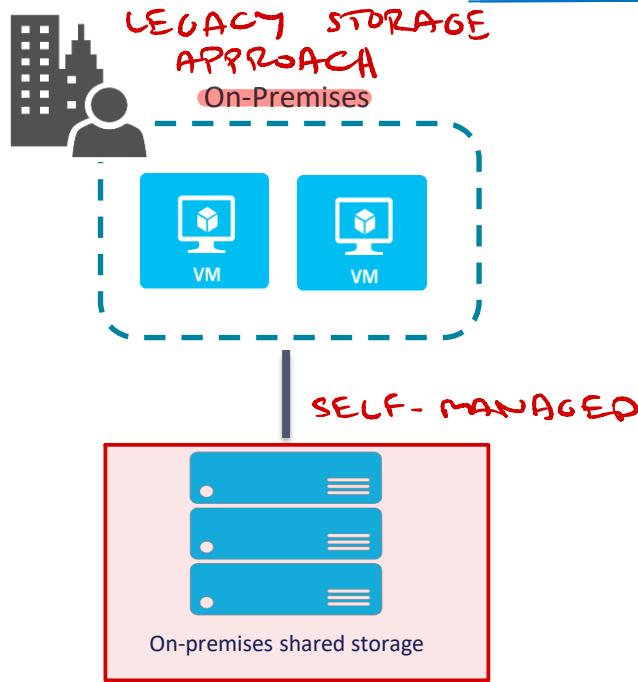
APP<sub>1</sub> →  → APP<sub>2</sub>

MESSAGE Q  
WAITING FOR  
MESSAGES TO  
BE PICKED UP  
BY APP<sub>2</sub>



EACH MESSAGE HAS  
AN EXPIRATION DATE

# Azure File Storage

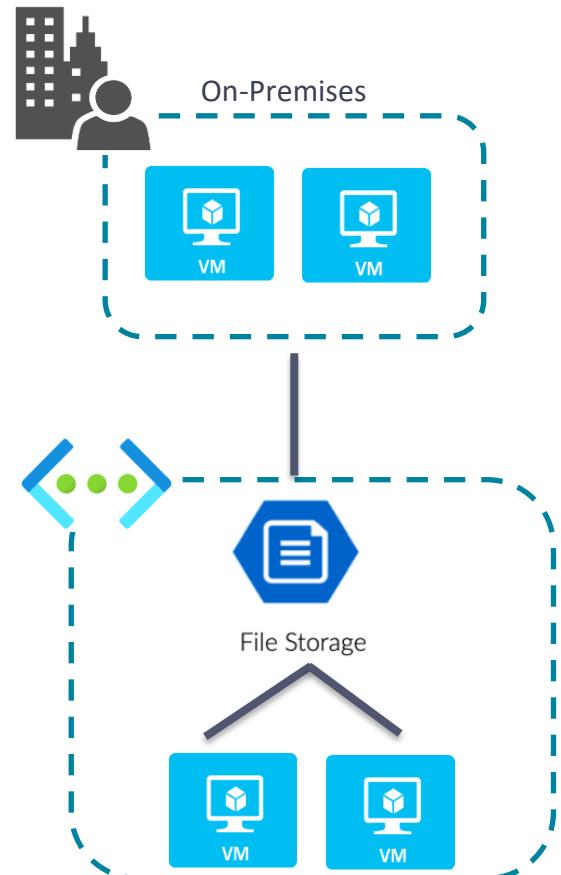


## Challenges

- Limited Amount of Storage
- Maintenance (hardware and OS)
- Schedule Backups
- Security
- Difficult to share files across Datacenters

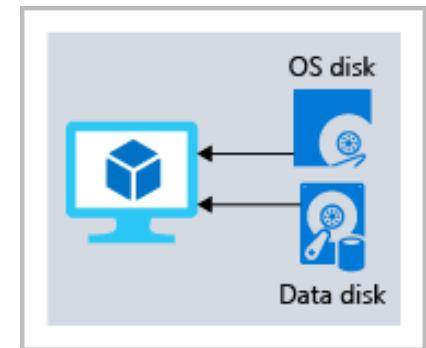
# Azure File Storage

- Enables you to create files shares in the cloud, and access these file shares from anywhere with an internet connection
- Mounted concurrently by cloud or on-premises deployments.
- Accessible from Windows, Linux, and macOS clients.
- Accessible Server Message Block (**SMB**) protocol or Network File System (**NFS**) protocol
- Azure Files ensures the data is encrypted at rest, and the SMB protocol ensures the data is encrypted in transit
- Use Cases
  - Replace or supplement on-premises file servers
  - Share application settings
  - Dev/Test/Debug
- Key Benefits
  - Shared access: Replace on-premises file shares with Azure file shares without application compatibility issues
  - Fully managed: Azure will manage hardware or an OS
  - Resiliency: you don't have to deal with local power and network issues.

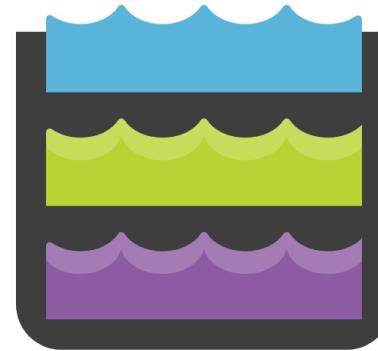


# Azure Disk Storage

- VM uses disks as a place to store an operating system, applications, and data in Azure.
- One virtual machine can have one OS disk and multiple Data disk but one data disk can only be link with one VM.
- Both the OS disk and the data disk are virtual hard disks (VHDs) stored in an Azure storage account. PAGE BLOBS
- The VHDs used in Azure is vhd files stored as page blobs in a standard or premium storage account in Azure.
- **Unmanaged disks:** We can create a storage account and specify it when we create the disk.
  - Not recommended, previous unmanaged disks should migrate to managed disk
- **Managed disk**
  - Azure creates and manages storage accounts in the background.
  - We don't have to worry about scalability issues.
  - Azure creates and manages the disk for us based on the size and performance tier we specify.
- **Managed Disk types:** OS DISK
  - Standard HDD: Backup, non-critical, infrequent access
  - Standard SSD: lightly used production applications or dev/test environments
  - Premium SSD disks: Super fast and high performance, very low latency, recommended for production and performance sensitive workloads
  - Ultra disks (SSD): for most demanding IO-intensive workloads such as SAP HANA, top tier databases (for example, SQL, Oracle), and other transaction-heavy workloads



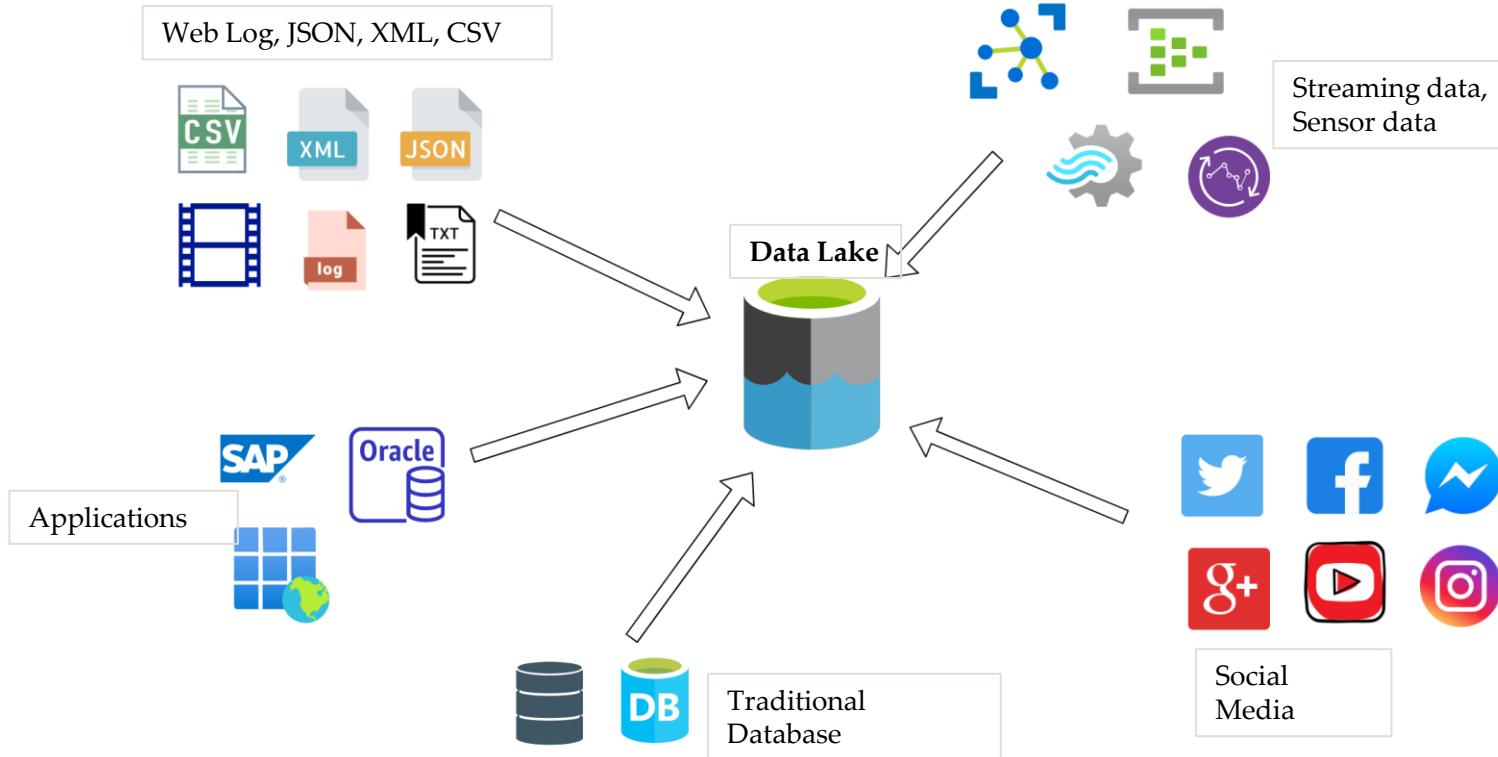
DATA DISK OPTIONS : STANDARD HDD/SSD + PREMIUM SSD

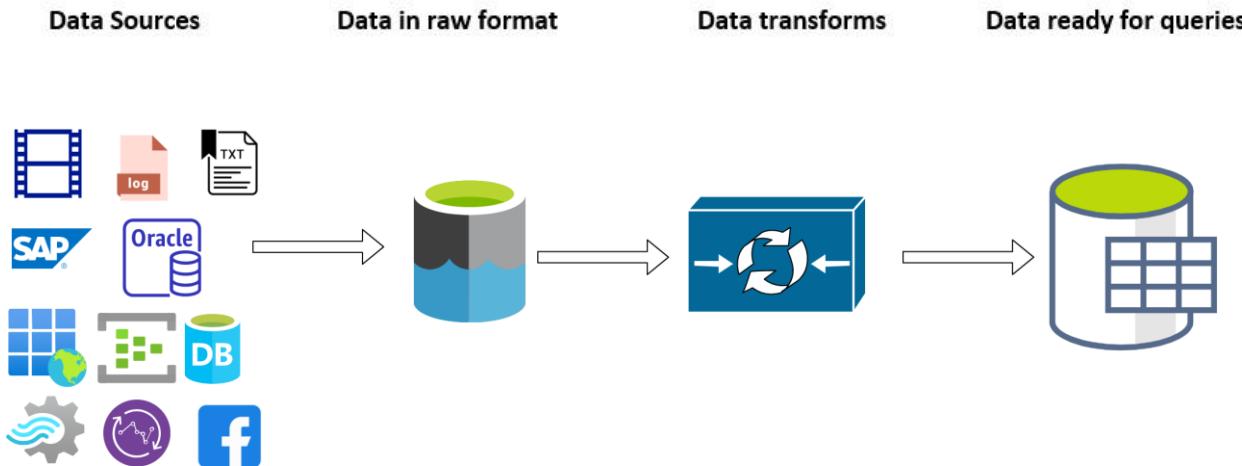


Data Lake is a big container to store data.

NO LIMIT TO STORAGE CAPABILITIES

# Data Lake Sources





# What is Data Lake?

"If you think of a DataMart as a store of bottled water – clean and packaged and structured for easy consumption – the data lake is a large body of water in a more natural state. The contents of the data lake stream in from a source to fill the lake, and various users of the lake can come to examine, dive in, or take samples."



James Dixon  
CTO, Pentaho



Data Warehouse

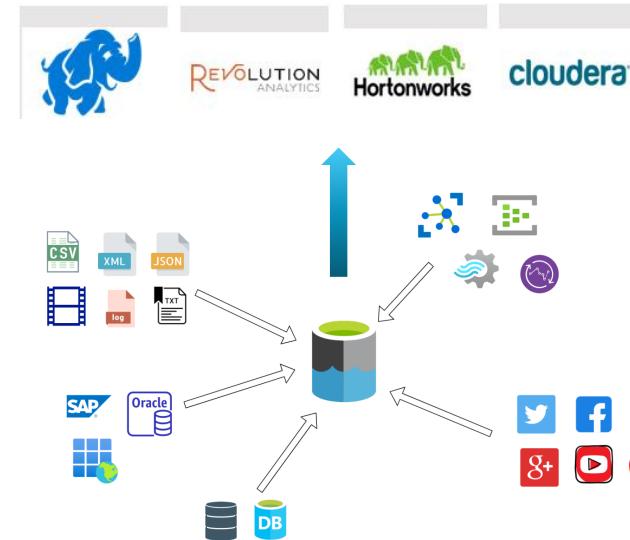
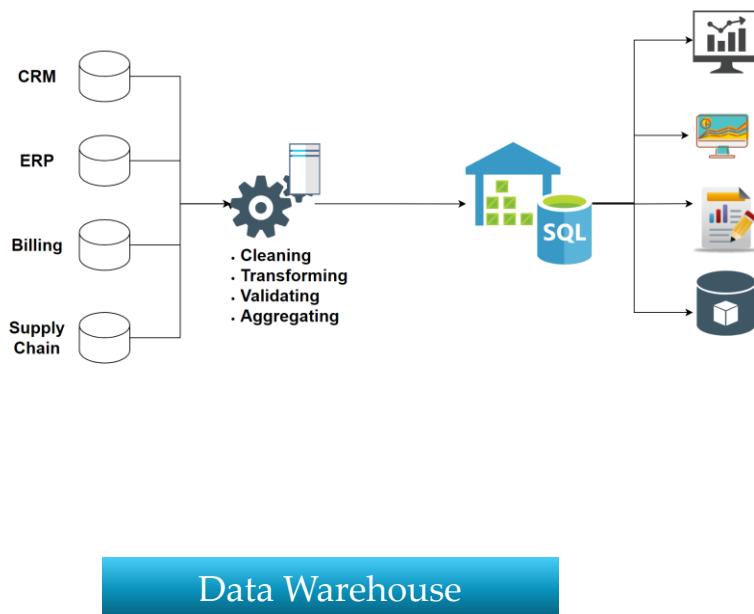
CLEAN



Data Lake

RAW

# Data Warehouse vs Data Lake



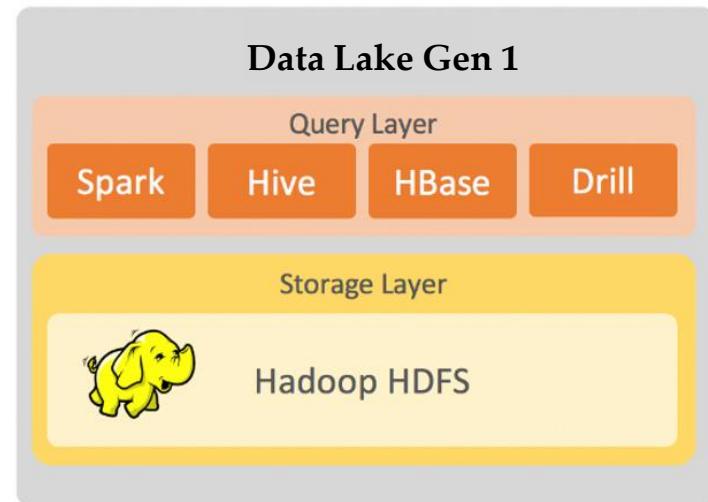
Load now, transform later

Data Lake

# Azure Data Lake Gen1 evolution



- Fault tolerant file system
- Runs on commodity hardware
- MapReduce, Pig, Hive, Spark etc.
- HDFS in Cloud -> Data Lake Storage Gen1



# Cloud storage challenge



## Processing

- Easy to optimize processing by increasing vCPU and Ram



## Storage

- Different requirements
- **No direct solution**

# Azure Blob Storage

## STORAGE ACCOUNT

- Large object storage in cloud
- Optimized for storing massive amounts of unstructured data
  - Text or Binary Data
- General purpose object storage
- Cost efficient
- Provide multiple Tiers

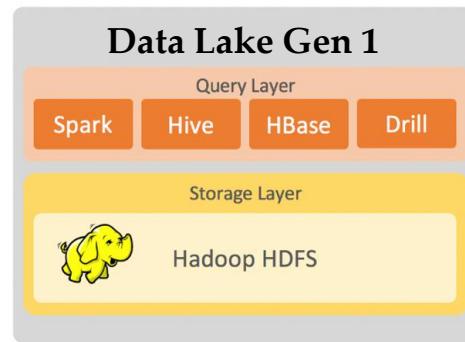
Microsoft Azure  
Blob Storage



# Azure Data lake Gen 2



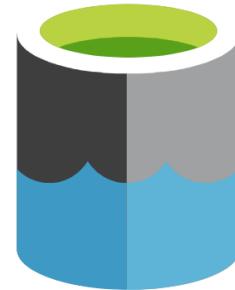
Blob Storage



Azure Data Lake Storage Gen2

# MICRSOFT RECOMMENDS

Data Lake Storage Gen2  
for your big data storage  
needs.



Azure Data Lake Storage Gen2

**Note:** USQL currently not supported in Gen 2

# Blob Storage vs Data Lake Storage

FLAT =  
CONTAINER > FILES

HIERARCHICAL =  
CONTAINER > CONTAINERS > FILES

## Azure Blob Storage

- General purpose data storage
- Container based object storage
- Available in every Azure region
- Local and global redundancy
- Processing performance limit

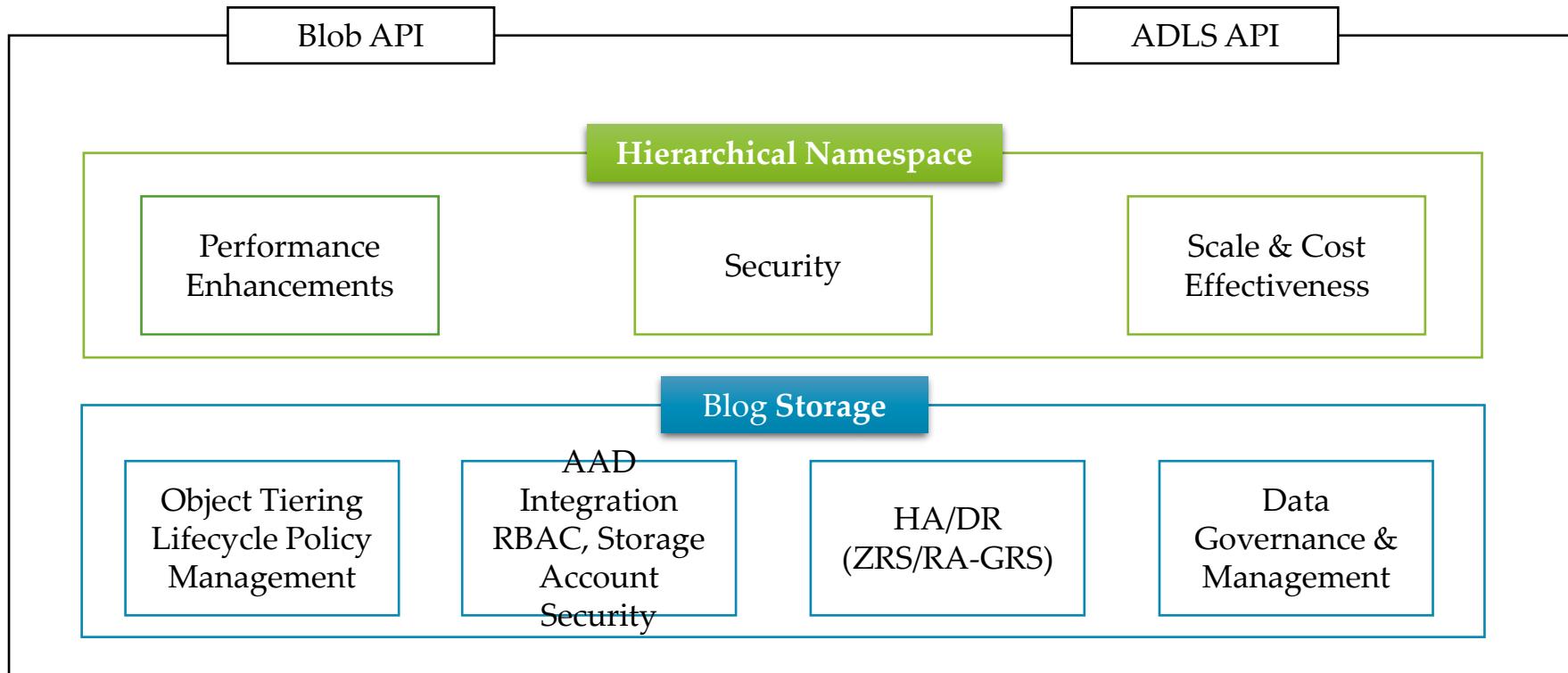
NOT OPTIMIZED FOR BIG DATA

## Azure Data Lake Storage (Gen 2)

- Optimized for big data analytics
- Hierarchical namespace on Blob Storage
- Available in every Azure region
- Local and global redundancy
- Supports a subset of Blob storage features
- Supports multiple Azure integrations
- Compatible with Hadoop



# Data Lake Architecture



# Learning objective



- **Authentication**
  - Storage Account keys **NOT RECOMMENDED**
  - Shared access signature (SAS)
  - Azure Active Directory (Azure AD)
- **Access Control**
  - Role based access control (RBAC)
  - Access control list (ACL)
- **Network access**
  - Firewall and virtual network
- **Data Protection**
  - Data encryption in transit
  - Data encryption at rest
- **Advanced threat Protection**

# Shared Access Signature (SAS)



Shared Access Signature

Security token string

“**SAS Token**”

Contains **permission** like **start and end time**

Azure doesn't track SAS after creation

To invalidate, regenerate storage account  
key used to sign SAS

# Stored access policy



Stored access policy

Reused by multiple SAS

Defined on a resource container

Permissions + validity period

Service level SAS only

Stored access policy can be revoked

# Azure Active Directory (AD)

- Grant access to Azure Active directory (AD) **Identities**
- AD is an enterprise identity provider, **Identity as a Service** (IDaaS)
- Globally available from virtually any device
- **Identities – user, group or application principle**
- **Assign role at Subscription, RG, Storage account, container level.**
- No longer need to store credentials with application config files
- Similar to IIS Application pool identity approach

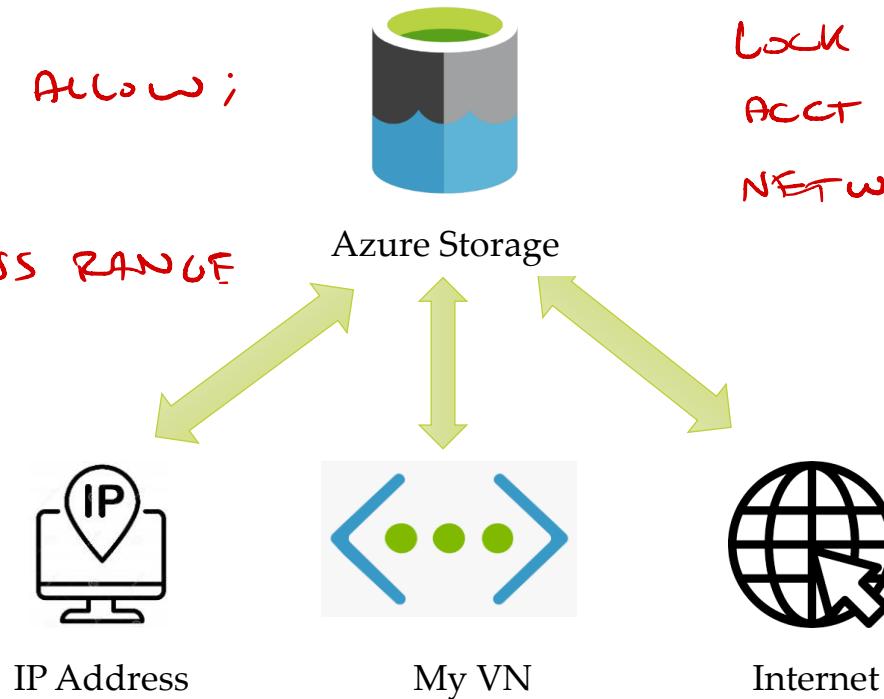


**Azure Active Directory**

# Firewalls and Virtual Networks

FIREWALL RULES ALLOW;  
IP ADDRESSES  
OR IP ADDRESS RANGE

LOCK DOWN STORAGE  
ACCT w/ SELECTED  
NETWORKS



# Data Continuity and Availability

///

## High Availability

- Making a service available within a region
- No expected data loss

DATA REPLICATED SYNCHRONOUSLY

RPO  
RTO

---

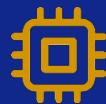
## Disaster Recovery

- Recovery when whole region goes down
- Typically some data loss

CAN INITIATE PLANNED  
FAILOVER + ELIMINATE  
DATA LOSS

# What can go wrong?

Disaster Recovery



Hardware



Software



Connectivity

CPU, memory, controllers,  
disk, server, rack, AC, power

Application, OS, bugs, logical  
corruptions

Redundant network  
connectivity

# What can go wrong?

Disaster Recovery



Entire site



Entire region



Human error

Lose power, lose water, bad weather

Flooding, earthquake

People make mistakes

LRS : LOCAL REDUNDANT STORAGE

ZRS : ZONE REDUNDANT STORAGE

GRS : GEO REDUNDANT STORAGE

## Durability and availability by outage scenario

The following table indicates whether your data is durable and available in a given scenario, depending on which type of redundancy is in effect for your storage account:

Outage scenario	LRS	ZRS	GRS/RA-GRS	GZRS/RA-GZRS
A node within a data center becomes unavailable	Yes	Yes	Yes	Yes
An entire data center (zonal or non-zonal) becomes unavailable	No	Yes	Yes <sup>1</sup>	Yes
A region-wide outage occurs in the primary region	No	No	Yes <sup>1</sup>	Yes <sup>1</sup>
Read access to the secondary region is available if the primary region becomes unavailable	No	No	Yes (with RA-GRS)	Yes (with RA-GZRS)

AS OF NOW, THERE IS NO AUTOMATIC FAILOVER  
USER MUST MANUALLY INITIATE A FAILOVER



# Azure Storage Outages

**Detection:** Subscribe to Azure Service health dashboard.

## LRS or ZRS

- Wait for recovery

## GRS or RA-GRS or GZRS or RA-GZRS

- Manual failover
- Copy data from secondary to some other region
  - Use tools such as AzCopy, Azure PowerShell, and the Azure Data Movement library