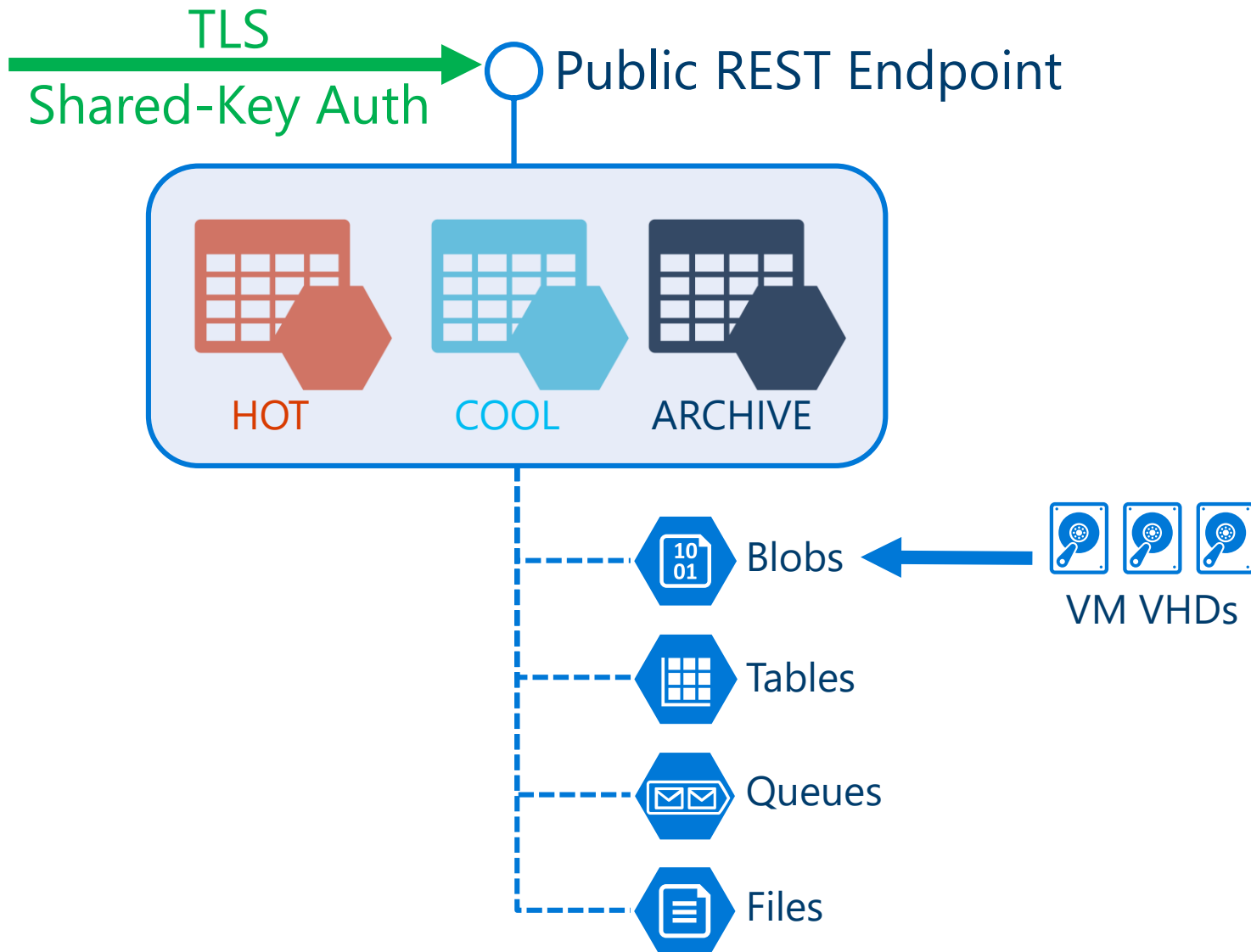


# Design Data Implementation



# Design for Azure Storage solutions

# Azure Storage Account General Properties



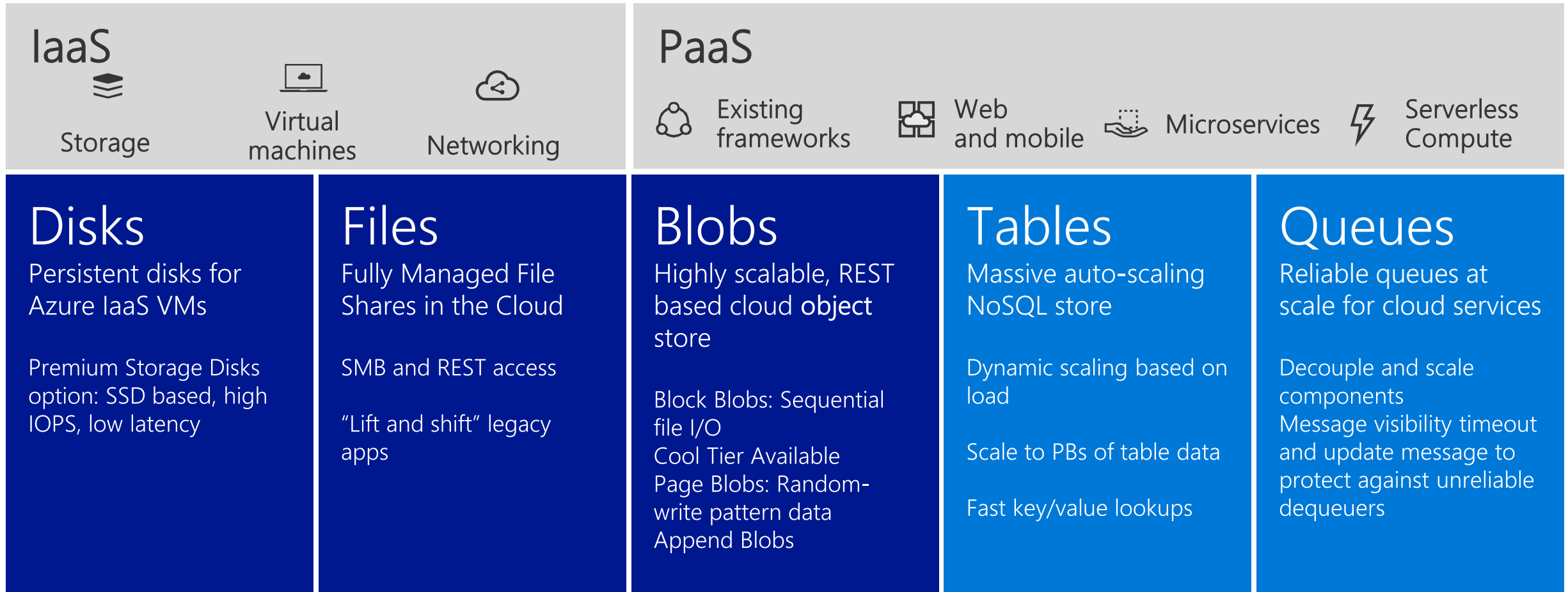
## Scalability

- 500 TB per account
- 20k IOPS per service type
- 10-20 Gbps ingress
- 20-30 Gbps egress

## Features

- Standard (HDD) or Premium (SDD)
- Anonymous access Blob Containers
- Shared Access Signatures
- Server Side Encryption (SSE)
- Geo-redundant storage
- Hot, cold, archive tiers use same API

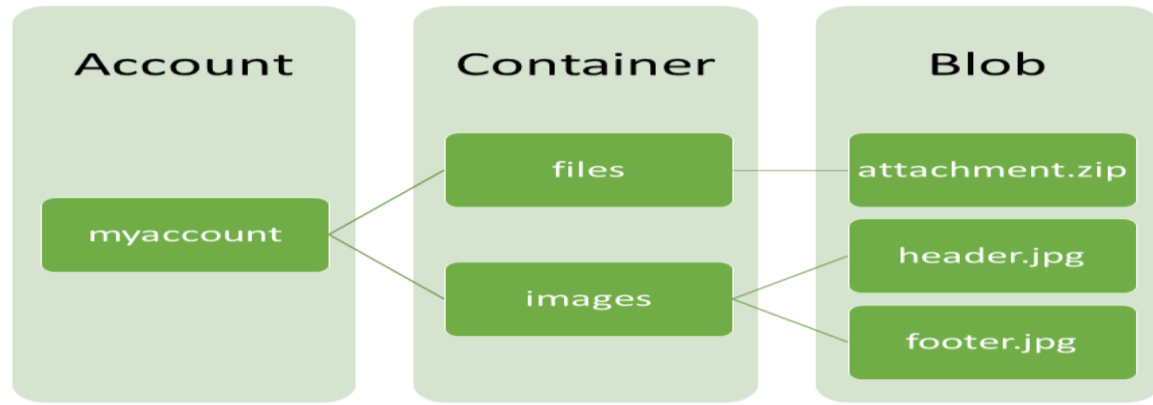
# Azure Storage Account Services



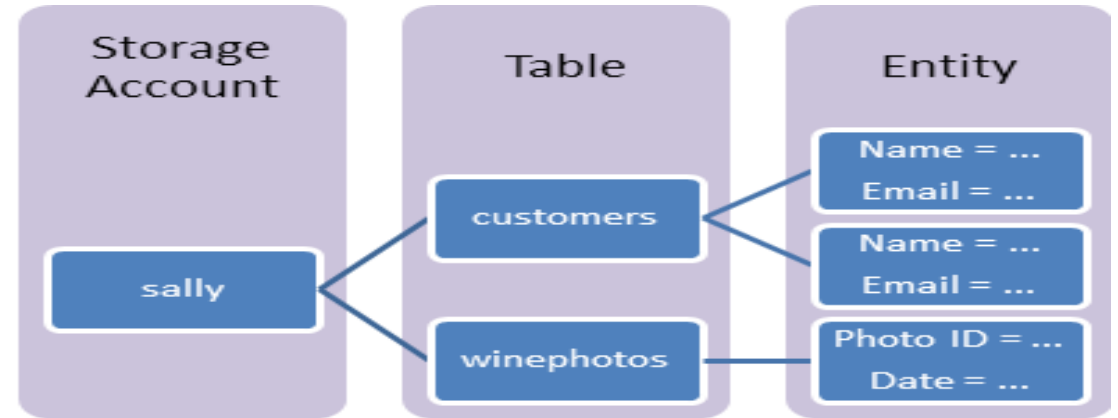
Built on a unified Distributed Storage System

Durability, Encryption at Rest, Strongly Consistent Replication, Fault Tolerance, Auto Load-Balancing

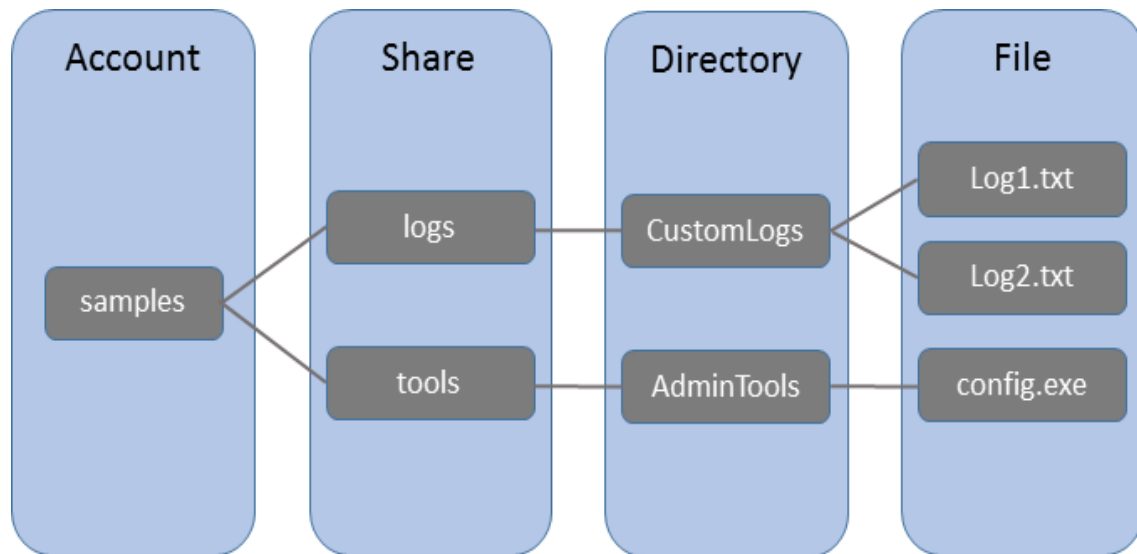
# Azure Storage Service Profiles



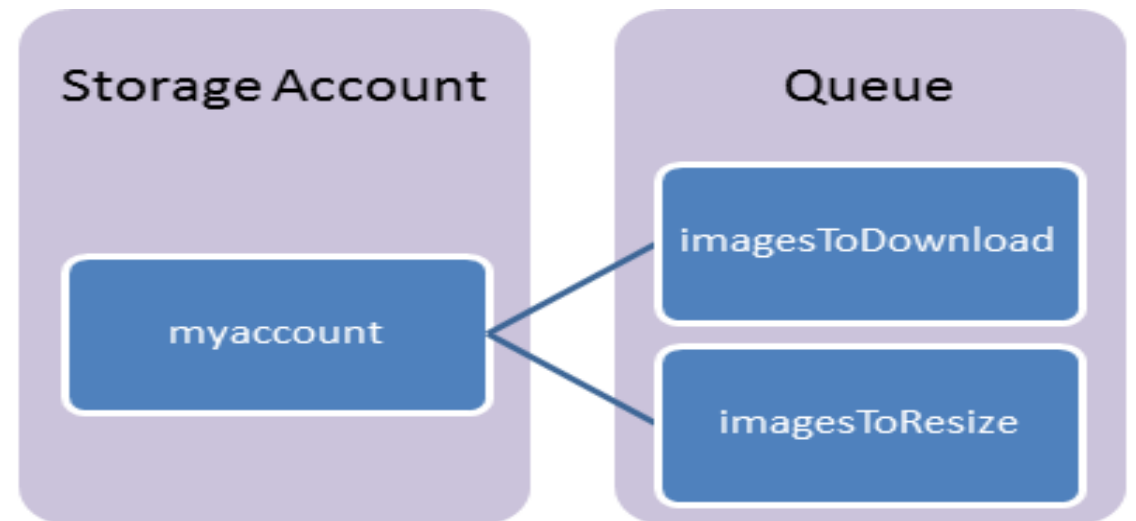
**Blob Storage**



**Azure Table**



**File Storage**



**Azure Queue**

# Azure Storage Account Types

- **General-purpose v1:**
  - Only storage type that works with classic deployment model
  - Supports use of Blobs, Tables, Queues, and files
  - Only uses hot storage method
  - Upgrade path to v2 via PowerShell/CLI
- **Blob Storage Accounts:**
  - Only supports use of Block Blobs
  - Aligns to capabilities of general-purpose v2
  - Blob Storage Types:
    - Block Blobs: block level storage ideal for storing text or binary files
    - Append Blobs: similar to block blobs, they're made of blocks but optimized for append ops and ideal for logging scenarios
    - Page Blobs: are more efficient for frequent read/write operations and are the type used to store VHDs for Azure VMs
- **General-purpose v2:**
  - Also supports use of Blobs, Tables, Queues, and files
  - Enables cool and archive access tier
  - Each account has default access tier that can be hot or cool
  - Each blob can change access tier to hot, cool or archive
  - Changing an access tier can have one-time cost implications

# EXAM TIP!

Know the various storage types and their uses. For example, many times you can use Queues to decouple components of a system.

[aka.ms/azure/storage](https://aka.ms/azure/storage), how they are alike & how they are different

# Storage Queues vs Service Bus Queues

Comparison Criteria	Storage queues	Service Bus Queues
Maximum queue size	<b>500 TB</b>  (limited to a <a href="#">single storage account capacity</a> )	<b>1 GB to 80 GB</b>  (defined upon creation of a queue and <a href="#">enabling partitioning</a> – see the “Additional Information” section)
Maximum message size	<b>64 KB</b>  (48 KB when using <b>Base64</b> encoding)  Azure supports large messages by combining queues and blobs – at which point you can enqueue up to 200 GB for a single item.	<b>256 KB or 1 MB</b>  (including both header and body, maximum header size: 64 KB).  Depends on the <a href="#">service tier</a> .
Maximum message TTL	<b>7 days</b>	<b>TimeSpan.Max</b>
Maximum number of queues	<b>Unlimited</b>	<b>10,000</b>  (per service namespace)
Maximum number of concurrent clients	<b>Unlimited</b>	<b>Unlimited</b>  (100 concurrent connection limit only applies to TCP protocol-based communication)



# Storage Queues vs Service Bus Queues MORE

Comparison Criteria	Storage queues	Service Bus Queues
Ordering guarantee	<b>No</b>  For more information, see the first note in the “Additional Information” section.	<b>Yes - First-In-First-Out (FIFO)</b> (through the use of messaging sessions)
Delivery guarantee	<b>At-Least-Once</b>	<b>At-Least-Once, At-Most-Once</b>
Atomic operation support	<b>No</b>	<b>Yes</b>
Receive behavior	<b>Non-blocking</b>  (completes immediately if no new message is found)	<b>Blocking with/without timeout</b> (offers long polling, or the <a href="#">“Comet technique”</a> )  <b>Non-blocking</b> (through the use of .NET managed API only)
Push-style API	<b>No</b>	<b>Yes</b> <a href="#">OnMessage</a> and <b>OnMessage</b> sessions .NET API.
Receive mode	<b>Peek &amp; Lease</b>	<b>Peek &amp; Lock, Receive &amp; Delete</b>
Exclusive access mode	<b>Lease-based</b>	<b>Lock-based</b>
Lease/Lock duration	<b>30 seconds (default), 7 days (maximum)</b>	<b>60 seconds (default)</b> You can renew a message lock using the <a href="#">RenewLock</a>
Lease/Lock precision	<b>Message level</b>	<b>Queue level</b>
Batched receive	<b>Yes</b>  (explicitly specifying message count when retrieving messages, up to a maximum of 32 messages)	<b>Yes</b>  (implicitly enabling a pre-fetch property or explicitly through the use of transactions)
Batched send	<b>No</b>	<b>Yes</b>  (through the use of transactions or client-side batching)

# Azure Storage Account Security

- **Shared Key Authentication:**
  - Two storage account keys that are 512-bit strings that give access to data for all Storage Account services
  - Blob containers can be set to give anonymous read access for individual blobs or for the whole container
  - Shared Access Signatures (SAS):
    - Contain access rules and an account key in an encrypted query string included with REST calls
    - Service-level SAS is scoped to specific resources within a storage account
    - Account-level SAS is scoped to access anything in the storage account
    - SAS policies can be used on Blob containers, a Table, a Queue, or a file share and used as basis for SAS URIs; allows revocation
- **Network Firewalls:**
  - Azure VNETs can use service endpoints and service tags to reference Azure Storage service IP ranges for direct routing and ACLs
  - Azure Storage accounts can use firewall rules to restrict network access by IP range
- **Data Encryption:**
  - Data in transit is encrypted using TLS security; HTTP access is also allowed but can be turned off
  - Azure files supports encryption for data in transit with SMB 3.0 on Windows only; SMB 2.0 or File Shares on Linux are not encrypted
  - Storage Service Encryption (SSE) encrypts all data at rest in an Azure Storage Account using MSFT managed keys only
  - Client-side Encryption is enabled using an SDK to programmatically encrypt data as its being stored in a Storage Account
  - Azure Disk Encryption:
    - Encrypt VHDs using BitLocker for Windows and DM-Crypt for Linux
    - SSE only encrypts new writes so its good for VM data disks, but marketplace images are unencrypted until new writes occur

# EXAM TIP!

## Read

Azure Storage | [Shared Access Signature](#) - [SAS](#)  
SQL Database – [same model](#) as on-premises

## Download and Use:

Microsoft Azure Storage Explorer is a standalone app from Microsoft that allows you to easily work with Azure Storage data on Windows, macOS and Linux. <http://storageexplorer.com/>

# EXAM TIP! *Example of SAS token*

<https://storagesample.blob.core.windows.net/sample-container/sampleBlob.txt?sv=2015-07-08&sr=b&sig=39Up9JzHkxhUIhFEjEH9594DJxe7w6cIRCg0V6ICGSo%3D&se=2016-10-18T21%3A51%3A37Z&sp=rcw>

Storage Resource URI

SAS Token

## 3.1.3 Exam Tip!

# Storage for VM disks (Page Blobs)

	Azure Premium Disk	Azure Standard Disk
Disk Type	Solid State Drives (SSD)	Hard Disk Drives (HDD)
Overview	SSD-based high-performance, low-latency disk support for VMs running IO-intensive workloads or hosting mission critical production environment	HDD-based cost effective disk support for Dev/Test VM scenarios
Scenario	Production and performance sensitive workloads	Dev/Test, non-critical, Infrequent access
Disk Size	P4: 32 GB (Managed Disks only) P6: 64 GB (Managed Disks only) P10: 128 GB P20: 512 GB P30: 1024 GB P40: 2048 GB P50: 4095 GB	Unmanaged Disks: 1 GB – 4 TB (4095 GB)  Managed Disks: S4: 32 GB S6: 64 GB S10: 128 GB S20: 512 GB S30: 1024 GB S40: 2048 GB S50: 4095 GB
Max Throughput per Disk	250 MB/s	60 MB/s
Max IOPS per Disk	7500 IOPS	500 IOPS

# EXAM TIP!

The storage account name should always be lowercase and unique within \*.core.windows.net namespace.

Microsoft Azure New > Storage account - blob, file, table, queue > Create storage account

Create storage account

The cost of your storage account depends on the usage and the options you choose below.  
[Learn more](#)

\* Name ⓘ

Temp .core.windows.net

Deployment model ⓘ

Resource manager Classic

Account kind ⓘ

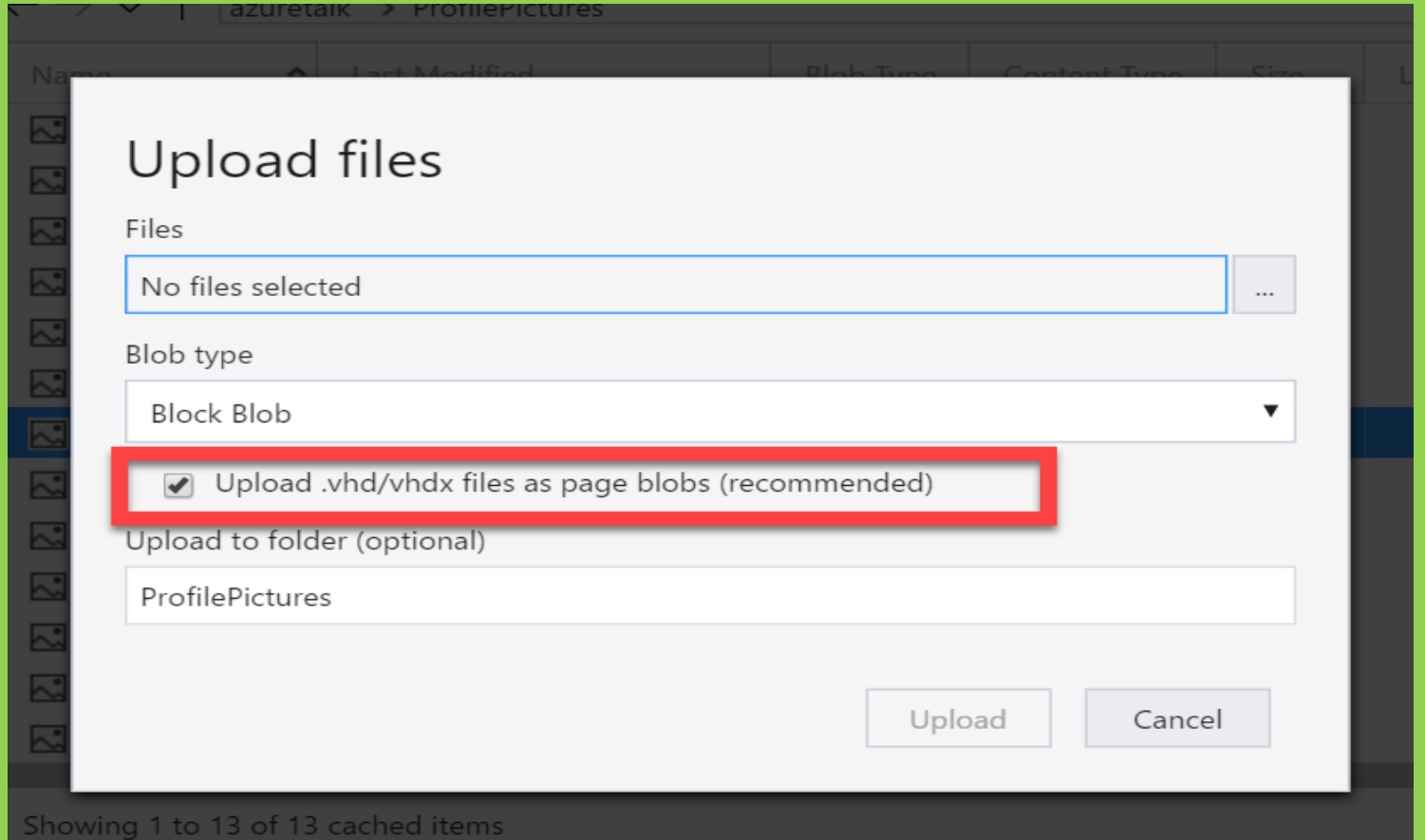
The field can contain only lowercase letters and numbers. Name must be between 3 and 24 characters.

## 3.1.4 Exam Tip!

# EXAM TIP!

Choose appropriate blob type for uploading VHD. If VHD files are uploaded in block blob you can't use those.

## 3.1.5 Exam Tip!



# Storage Account Cost Model

- **Cost Components:**
  - Storage cost: the cost of the provisioned data
  - Data access cost: data retrieval when changing access storage tier from cool to hot
  - Transaction costs: read, write, list, and create container operations
  - Geo-replication data transfer costs: data leaving an Azure region via built-in replication features
  - Outbound data transfer: data leaving an Azure region for any use other than built-in replication features

	Hot storage tier	Cool storage tier	Archive storage tier
Availability	99.9%	99%	N/A
Availability (RA-GRS reads)	99.99%	99.9%	N/A
Usage charges	Higher storage costs, lower access and transaction costs	Lower storage costs, higher access and transaction costs	Lowest storage costs, highest access and transaction costs
Minimum object size	N/A	N/A	N/A
Minimum storage duration	N/A	30 days (GPv2 only)	180 days
Latency (Time to first byte)	milliseconds	milliseconds	< 15 hrs
Scalability and performance targets	Same as general-purpose storage accounts	Same as general-purpose storage accounts	Same as general-purpose storage accounts



# Azure Storage Replication Options

Replication strategy	LRS	ZRS	GRS	RA-GRS
<b>Data is replicated across multiple datacenters.</b>	No	Yes	Yes	Yes
<b>Data can be read from a secondary location as well as the primary location.</b>	No	No	No	Yes
<b>Designed to provide _ durability of objects over a given year.</b>	at least 99.99999999 99% (11 9's)	at least 99.999999999 99% (12 9's)	at least 99.999999999999 9999% (16 9's)	at least 99.999999999999 9999% (16 9's)

# Azure Storage Scalability Targets

Resource	Default Limit
<b>Number of storage accounts per subscription</b>	<b>200</b>
<b>Max storage account capacity</b>	<b>500 TiB</b>
Max number of blob containers, blobs, file shares, tables, queues, entities, or messages per storage account	No limit
<b>Maximum request rate per storage account</b>	<b>20,000 requests per second</b>
Max ingress per storage account (US Regions)	10 Gbps if GRS/ZRS enabled, 20 Gbps for LRS
Max egress per storage account (US Regions)	20 Gbps if RA-GRS/GRS/ZRS enabled, 30 Gbps for LRS
Max ingress per storage account (Non-US regions)	5 Gbps if GRS/ZRS enabled, 10 Gbps for LRS
Max egress per storage account (Non-US regions)	10 Gbps if RA-GRS/GRS/ZRS enabled, 15 Gbps for LRS

# EXAM TIP!

Scalability targets are planned to increase in early 2018 so stay posted on the changes depending on when you take the exam.

- 500 TB → 5 PB storage capacity
- 20k → 50k requests/sec
- 20 Gbps → 50 Gbps bandwidth

# Azure Blob Storage Scale Targets

Resource	Target
Max size of single blob container	500 TiB
Max number of blocks in a block blob or append blob	50,000 blocks
<b>Max size of a block in a block blob</b>	<b>100 MiB</b>
Max size of a block blob	50,000 X 100 MiB (approx. 4.75 TiB)
<b>Max size of a block in an append blob</b>	<b>4 MiB</b>
Max size of an append blob	50,000 x 4 MiB (approx. 195 GiB)
<b>Max size of a page blob</b>	<b>8 TiB</b>
Max number of stored access policies per blob container	5
Target throughput for single blob	Up to 60 MiB per sec, or up to 500 requests per sec

# **EXAM TIP!** *Sufficient bandwidth on VM*

Make sure sufficient bandwidth is available on your VM to drive disk traffic, as described in [Premium Storage-supported VMs](#). Otherwise, your disk throughput and IOPS is constrained to lower values. Maximum throughput and IOPS are based on the VM limits, not on the disk limits described in the preceding table.

How to monitor for disk I/O throttling: <https://aka.ms/VMDiskThrottling>

<https://docs.microsoft.com/en-us/azure/storage/storage-premium-storage#scalability-and-performance-targets>

# **EXAM TIP!** *Premium storage accounts Scalability*

... have the following scalability targets:+

Total account capacity	Total bandwidth
Disk capacity: 35 TB	Locally redundant storage account
Snapshot capacity: 10 TB	Up to 50 gigabits per second
	for <b>inbound</b> <sup>1</sup> + <b>outbound</b> <sup>2</sup>
	<sup>1</sup> All data (requests) that are sent to a storage account+
	<sup>2</sup> All data (responses) that are received from a storage account

<https://docs.microsoft.com/en-us/azure/storage/storage-premium-storage#scalability-and-performance-targets>

# EXAM TIP! *If Using Storage Spaces Striping*

If you stripe premium storage data disks by using [Storage Spaces](#), set up Storage Spaces with 1 column for each disk that you use. Otherwise, overall performance of the striped volume might be lower than expected because of uneven distribution of traffic across the disks. By default, in Server Manager, you can set up columns for up to 8 disks. If you have more than 8 disks, use PowerShell to create the volume. Specify the number of columns manually. Otherwise, the Server Manager UI continues to use 8 columns, even if you have more disks. For example, if you have 32 disks in a single stripe set, specify 32 columns. To specify the number of columns the virtual disk uses, in the [New-VirtualDisk](#) PowerShell cmdlet, use the *NumberOfColumns* parameter. For more information, see [Storage Spaces Overview](#) and [Storage Spaces FAQs](#).

<https://docs.microsoft.com/en-us/azure/storage/storage-premium-storage#scalability-and-performance-targets>

## **EXAM TIP!** *Pay Attention to I/O vs IOPs*

If your disk traffic mostly consists of small I/O sizes, your application likely will hit the IOPS limit before the throughput limit. However, if the disk traffic mostly consists of large I/O sizes, your application likely will hit the throughput limit first, instead of the IOPS limit. You can maximize your application's IOPS and throughput capacity by using optimal I/O sizes. Also, you can limit the number of pending I/O requests for a disk.

<https://docs.microsoft.com/en-us/azure/storage/storage-premium-storage#scalability-and-performance-targets>



# Azure Files Scale Targets

Resource	Target
<b>Max size of a file share</b>	<b>5 TiB</b>
<b>Max size of a file in a file share</b>	<b>1 TiB</b>
Max number of files in a file share	No limit
<b>Max IOPS per share</b>	<b>1000 IOPS</b>
Max number of stored access policies per file share	5
<b>Maximum request rate per storage account</b>	<b>20,000 requests per second for files of any valid size<sup>3</sup></b>
Target throughput for single file share	Up to 60 MiB per second
Maximum open handles for per file	2000 open handles
Maximum number of share snapshots	200 share snapshots

# Azure Queue Storage Scale Targets

Resource	Target
Max size of single queue	500 TiB
<b>Max size of a message in a queue</b>	<b>64 KiB</b>
Max number of stored access policies per queue	5
Maximum request rate per storage account	20,000 messages per second assuming 1 KiB message size
Target throughput for single queue (1 KiB messages)	Up to 2000 messages per second

# Azure Table Storage Scale Targets

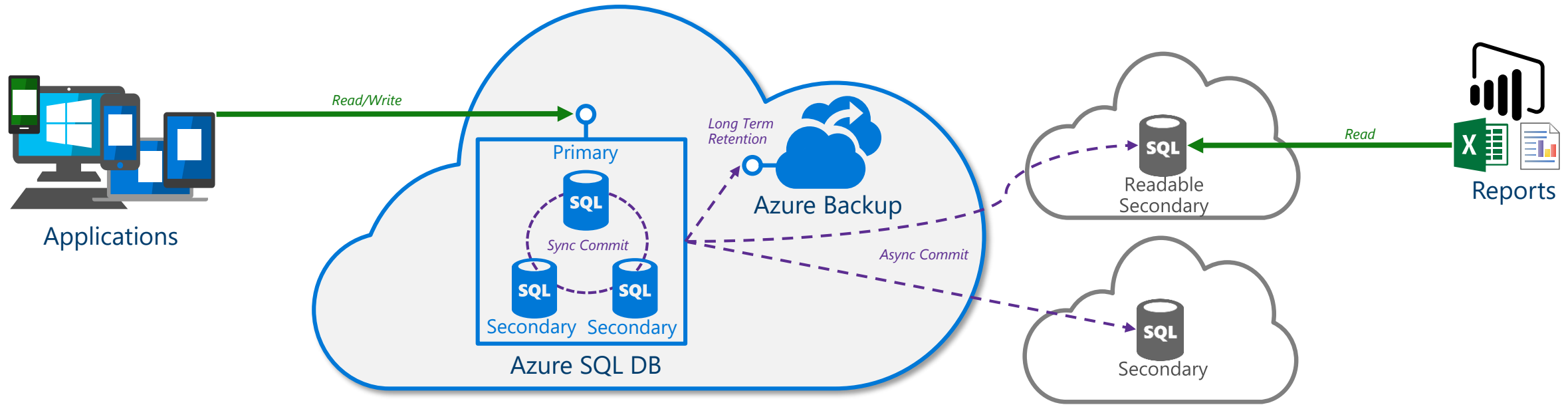
Resource	Target
Max size of single table	500 TiB
Max size of a table entity	1 MiB
Max number of properties in a table entity	252
Max number of stored access policies per table	5
Maximum request rate per storage account	20,000 transactions per second (assuming 1 KiB entity size)
Target throughput for single table partition (1 KiB entities)	Up to 2000 entities per second

# Transferring Data

- Azure API: The Azure REST API or SDK, namely the Azure Storage Data Movement Library, is designed for high-performance copying of data to and from Azure
- AzCopy Utility: Windows command line utility designed for high-performance data transfer
- Azure PowerShell/CLI: Write scripts to push data for any scenario
- Azure Import/Export: Enables data transfer by securely shipping up to 10 TB external drive to Microsoft
- Azure Data Box: Microsoft appliance to securely transfer large amounts of data (100 TB)
- Virtual appliances: Storage virtual appliances from vendors like Nasuni or NetApp
- StorSimple: Azure first party service to tier data from on-prem to Azure
- ExpressRoute: Enabling a high performance network throughput for faster data transfer using above options

# Design for relational database storage

# Database as-a-Service



## Azure SQL Database

- Built-in local **high availability** with **99.99% SLA**
- **Automatic backups** (full, differential, trans log) with optional long term retention
- **Active geo-replication** enables read replicas in a secondary region with a single DNS endpoint
- **Elastic Database Pools** enables resource shared for more efficient cost management
- Azure offers this managed service for **SQL Server, MySQL, PostgreSQL**

# Azure SQL DB Tiers

	Basic	Standard	Premium	<i>Premium RS</i>
Target workload	Development and production	Development and production	Development and production	Workload that can tolerate data loss up to 5-minutes due to service failures
Uptime SLA	99.99%	99.99%	99.99%	N/A
Backup retention	7 days	35 days	35 days	35 days
CPU	Low	Low, Medium, High	Medium, High	Medium
IO throughput	Low	Medium	Order of magnitude higher than Standard	Same as Premium
IO latency	Higher than Premium	Higher than Premium	Lower than Basic and Standard	Same as Premium
Columnstore indexing and in-memory OLTP	N/A	N/A	Supported	Supported

# Choosing DB Performance Level

## Database Transaction Unit – DTU

### Bounding box

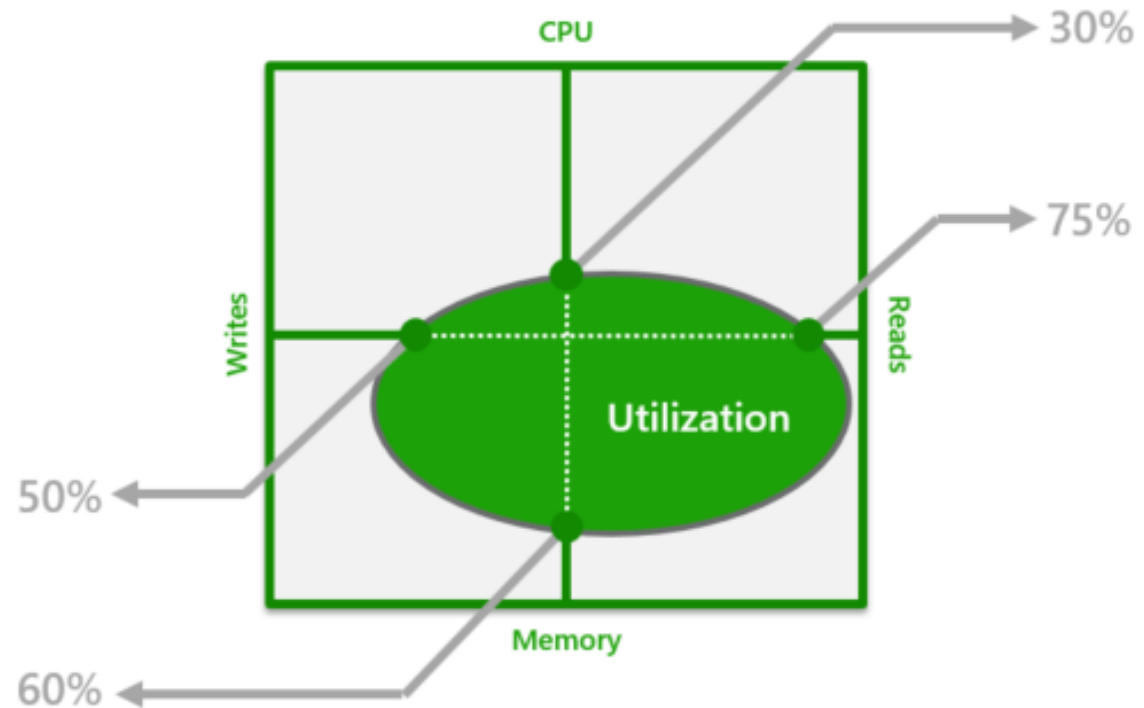
Monitoring database workload utilization within bounding box

Represents the relative power (resources) assigned to the database

Blended measure of CPU, memory, and read-write rates

Compare the power across performance levels

Simplifies talking about performance, think IOPS vs. %



[DTU Calculator](#)



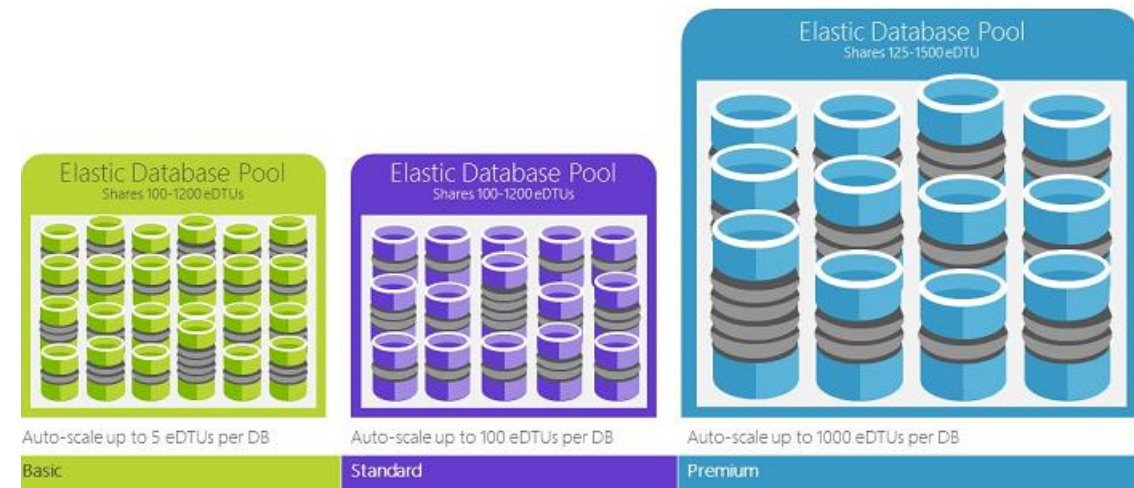
# Database Types

## Single Databases



- Fully managed redundancy, backups, and upgrades
- Wide range of sizes and performance tiers
- Majority of feature parity with SQL Server

## Elastic Database Pools



- A shared pool of eDTUs by multiple DBs enables cost efficiency
- Auto-scale within set parameters
- Simplified management of large number of DBs
- Adjust eDTUs for the pool as needed with no down time

# Performance Level and Storage Size Limits

## Single Databases

	Basic	Standard	Premium	<i>Premium RS</i>
Maximum storage size*	2 GB	1 TB	4 TB	1 TB
Maximum DTUs	5	3000	4000	100

## Elastic Pools

	Basic	Standard	Premium	<i>Premium RS</i>
Maximum storage size per database*	2 GB	1 TB	1 TB	1 TB
Maximum storage size per pool*	156 GB	4 TB	4 TB	1 TB
Maximum eDTUs per database	5	3000	4000	1000
Maximum eDTUs per pool	1600	3000	4000	1000
Maximum number of databases per pool	500	500	100	100

# SQL DB Security

- Connectivity:
  - All connections are over a public endpoint and require TLS encryption for data in transit
  - Firewall rules limit connectivity by originating IP range
  - Service endpoints with an Azure VNET allow direct connectivity
- Authentication:
  - SQL Authentication
  - Azure AD Authentication
- Authorization:
  - Account database role and object level permissions
  - Impersonation
  - Row-level security
  - Data masking
  - Stored procedures
- Security:
  - Transparent Database Encryption (TDE) using Microsoft or customer managed keys
  - Cell-level encryption to encrypt specific columns or cells of data
  - Always Encrypted for data in use
  - SQL database auditing to maintain regulator compliance

# SQL DB Availability

- Long-term data retention: Use Azure Backup to retain beyond 7-35 day window
- Active Geo-Replication: Replicate an asynchronous (readable) secondary to another region
- Failover Groups: Automatic failover of active geo-replicated secondary's using VIPs

Capability	Basic tier	Standard tier	Premium tier
Point in Time Restore from backup	Any restore point within 7 days	Any restore point within 35 days	Any restore point within 35 days
Geo-restore from geo-replicated backups	ERT < 12h, RPO < 1h	ERT < 12h, RPO < 1h	ERT < 12h, RPO < 1h
Restore from Azure Backup Vault	ERT < 12h, RPO < 1 wk	ERT < 12h, RPO < 1 wk	ERT < 12h, RPO < 1 wk
Active geo-replication	ERT < 30s, RPO < 5s	ERT < 30s, RPO < 5s	ERT < 30s, RPO < 5s

# MySQL, PostgreSQL

Pricing tier features	Basic	Standard	Premium (not available)
Maximum Compute Units	100	800	
Maximum total storage	1 TB	1 TB	
Storage IOPS guarantee	N/A	Yes	
Maximum storage IOPS	N/A	3,000	
Database backup retention period	7 days	35 days	
SLA	99.99%	99.99%	
HA	Regional only	Regional only	

# Migrate to Azure SQL DB

- Data Migration Assistant (DMA): copies database, deploys fixes, migrates copy; requires downtime
- Transactional Replication: requires some setup and continuous connectivity but does not require downtime
- Import DB: import from on-prem to Azure using BACPAC file
- Azure SQL Data Sync (Preview): bi-directional connectivity between a group of Azure SQL DB and SQL Server
- Azure DB Migration Service (Preview): migrate from multiple sources or DB types to target DB

3.1.1: A title company needs to provide access to storage for people to upload scans of documents that they signed. There are time limits to when the documents are needed. Which approach should be used to provide access?

- 1) Create a service and set up the valid dates in a database that will be read
- 2) Use an SAS and set the expiration time and date for the user.
- 3) Set up Azure AD with permissions for the user that limit the time in which she can upload documents.
- 4) Open up the Blob storage container to public access so that the user can add the document even if she is late.

# EXAM TIP!

Azure might update to change feature availability. The exam is updated over time, as well, to reflect these changes. However, because of the way Azure is steadily being updated, the newest features might not be on the exams.



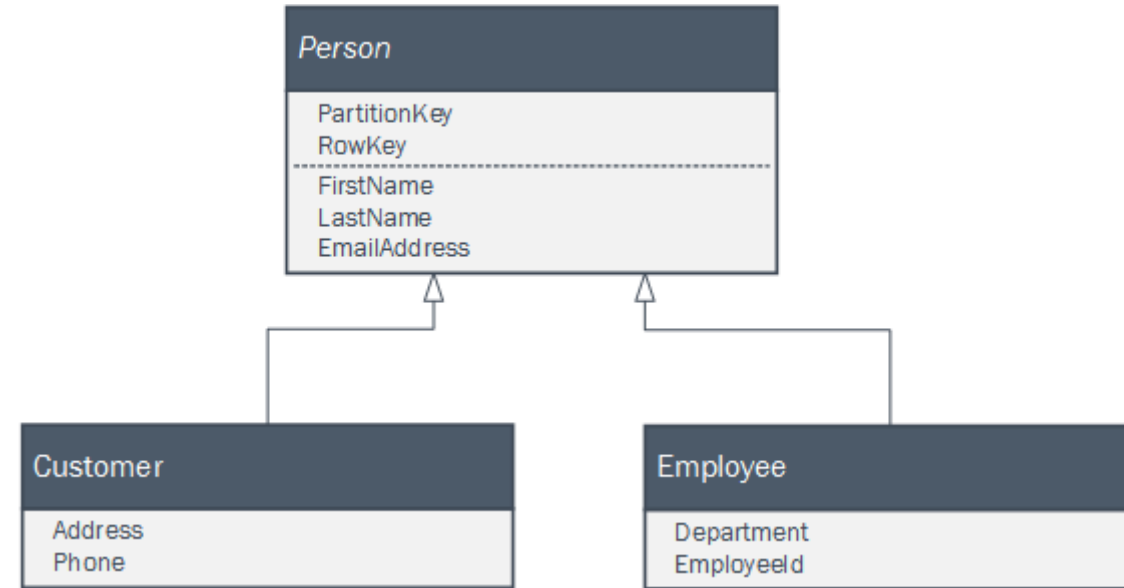
# EXAM TIP!

Performance levels of the database are important to a company, and the ability to change this at times is equally important. For example, the limits of each level of SQL Database can help the architect to determine the minimum level needed to satisfy those needs.

# Design for NoSQL Storage

# Azure Table Storage

- Features:
  - Partition key: determines the partition to use within storage service
  - RowKey: used with partition key to create clustered index, the only index created
- Guidelines for Read Design:
  - Specify both PartitionKey and RowKey in your queries.
  - Table storage is cheap so consider storing the same entity multiple times (with different keys) to enable more efficient queries.
  - Table storage is cheap so consider denormalizing your data.
  - Use compound key values for alternate keyed access paths to entities.
  - Use query projection. You can reduce the amount of data that you transfer over the network by using queries that select just the fields you need
- Guidelines for Write Design:
  - Choose keys that enable you to spread your requests across multiple partitions at any point of time.
  - Avoid spikes in traffic.
  - Don't necessarily create a separate table for each type of entity.
  - Consider the maximum throughput you must achieve.



Customer entity	Employee entity
PartitionKey RowKey	PartitionKey RowKey
PersonType ("Customer") FirstName (string) LastName (string) EmailAddress (string) Address (string) Phone (string)	PersonType ("Employee") FirstName (string) LastName (string) EmailAddress (string) Department (string) EmployeeID (string)

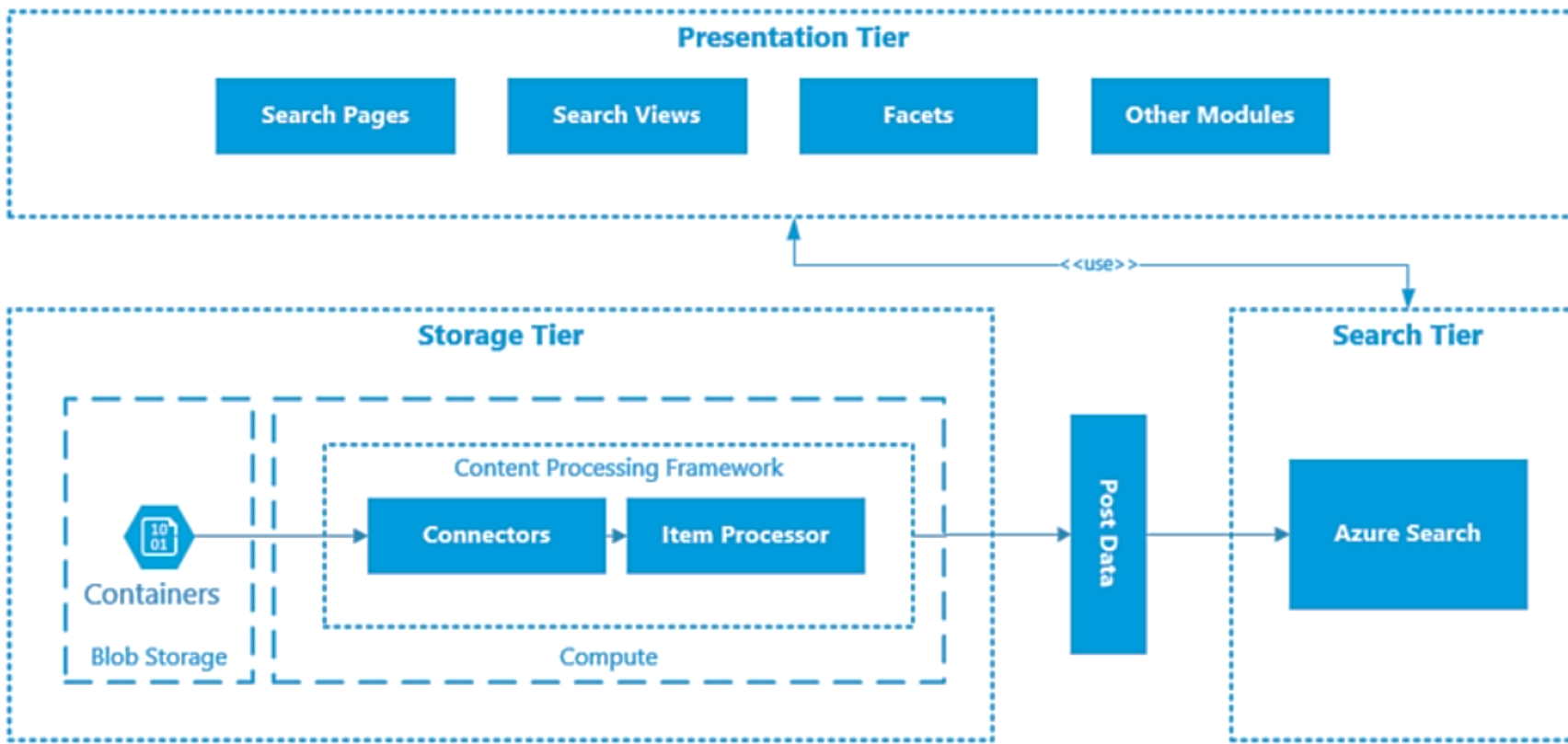
# Redis Cache

	<b>BASIC</b> BASIC CACHE WHICH IS IDEAL FOR DEVELOPMENT/TESTING.	<b>STANDARD</b> PRODUCTION READY CACHE WITH MASTER/SLAVE REPLICATION.	<b>PREMIUM</b> ENTERPRISE READY TIER WHICH CAN BE USED AS A CACHE AND PERSIST DATA. DESIGNED FOR MAXIMUM SCALE AND ENTERPRISE INTEGRATION.
Cache	Yes	Yes	Yes
Replication and failover	-	Yes	Yes
SLA	-	99.9%	99.9%
Configure Redis (keyspace notifications etc.)	-	Yes	Yes
Redis data persistence	-	-	Yes
Redis cluster	-	-	Yes
Scale out to multiple cache units	-	-	Yes
Azure Virtual Network	-	-	Yes
Memory size	250 MB–53 GB	250 MB–53 GB	6 GB–530 GB
Network performance	Low to high	Low to high	Moderate to highest
Maximum number of client connections	256–20,000	256–20,000	7,500–40,000

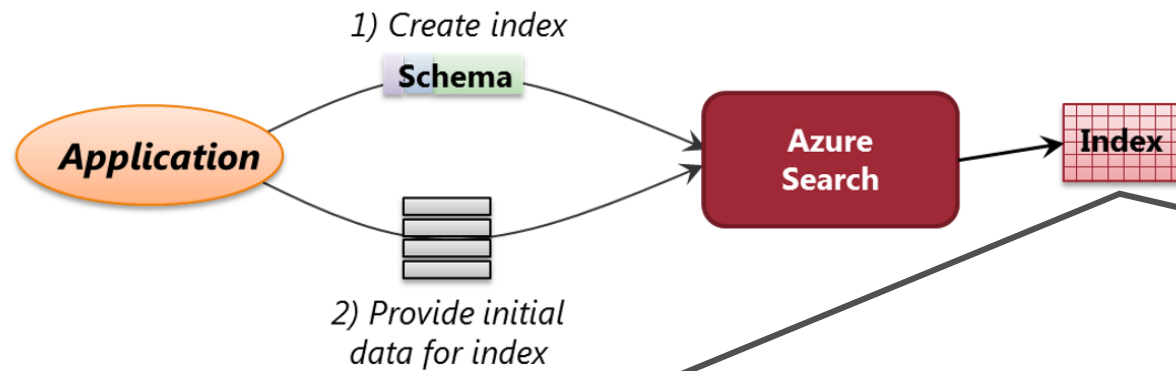
# Azure Search Architecture

## Features

- Query syntax – simple, Lucene
- 56 Languages
- Search suggestions
- Hit highlighting
- Faceted navigation
- Geo-special support
- Filters, sorting, paging
- Fully managed, highly available
- Scoring profiles
- Data connectors
- Testing tools
- Search analytics

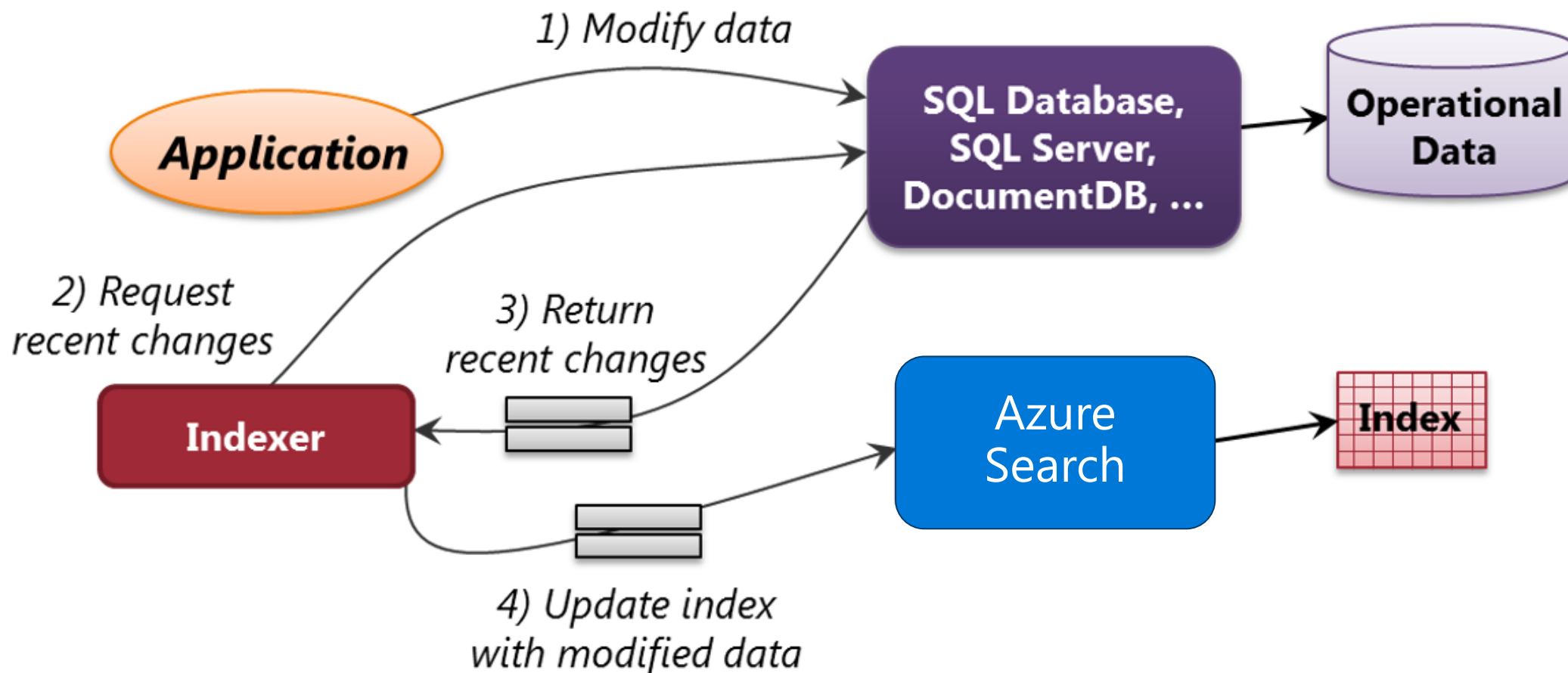


# Create and Populate an Index



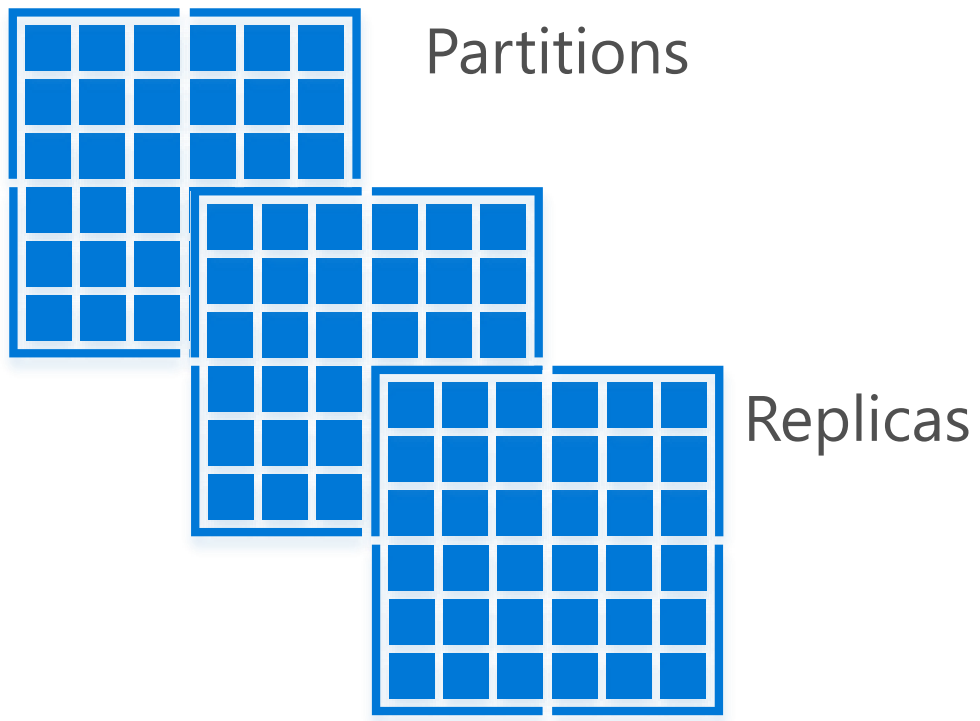
Name	Type	Other Attributes
Category	String	Searchable, Suggestions, Sortable, Retrievable, Filterable
Brand	String	Searchable, Suggestions, Sortable, Retrievable, Filterable
Style	String	Searchable, Suggestions, Sortable, Retrievable, Filterable
Color	Collection(String)	Searchable, Suggestions, Retrievable, Filterable, Facetable
Price	Double	Searchable, Sortable, Retrievable, Filterable, Facetable
Picture	String	Retrievable
Stock	Int32	
Promotion	Boolean	

# Update Indexes















# Cost and Scale

## Search Units



(Replicas x Partitions)

## Service Tiers

F Free		B Basic (Preview)		S Standard	
3	Indexes	5	Indexes	50	Indexes
10K	Documents	1M	Documents	15M	Docs/Partition*
	50 MB Storage		2 GB Storage		25 GB/Partition* Storage
	Shared Resources		Dedicated Resources		Dedicated Resources
	None Scaling		Up to 3 search units Scaling		Up to 36 search units Scaling
			Up to 3 replicas Load Balancing		Up to 12 replicas Load Balancing
					Up to 12 partitions Partitions
0.00		37.57		249.98	
USD/MONTH		USD/MONTH PER UNIT (ESTIMATED)		USD/MONTH PER UNIT (ESTIMATED)	



# Design for CosmosDB Storage

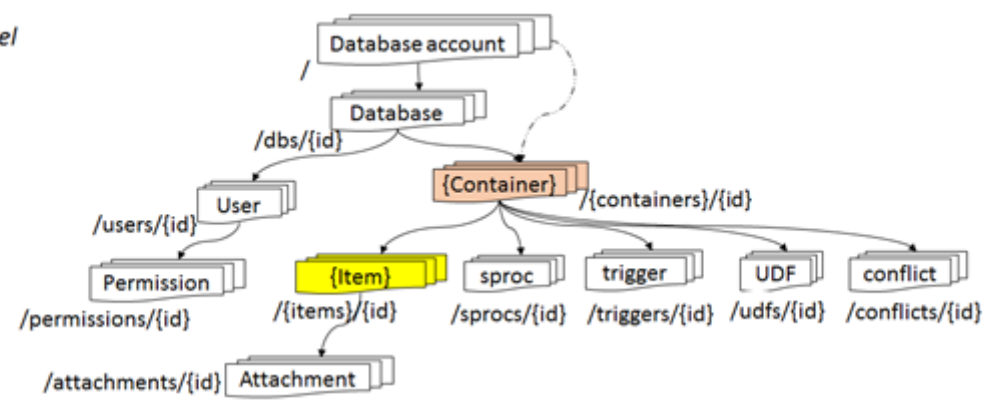
# CosmosDB Key Capabilities

- Enable customers to elastically scale throughput and storage based on demand, globally
- Enable customers to build highly responsive, mission-critical applications
- Ensure that the system is “always on”
- Enable developers to write correct globally distributed applications
- Offer stringent financially-backed comprehensive SLAs
- Relieve the developers from the burden of database schema/index management and versioning
- Natively support multiple data models and popular APIs for accessing data
- Operate at a very low cost

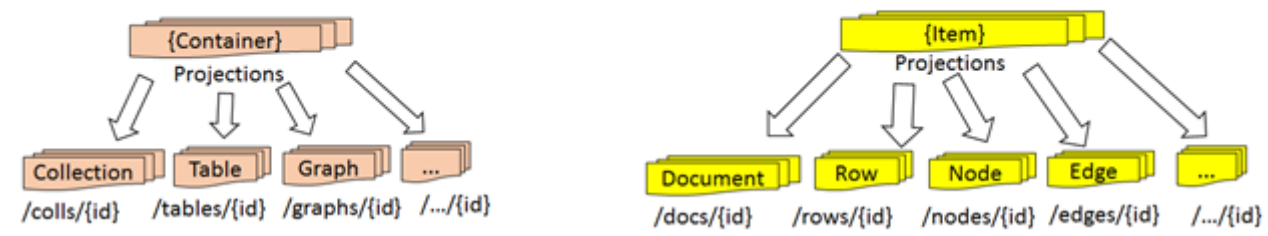
Capabilities	Relational databases	Non-relational (NoSQL) databases	Azure Cosmos DB
Global distribution	No	No	Yes, turnkey distribution in 30+ regions, with multi-homing APIs
Horizontal scale	No	Yes	Yes, you can independently scale storage and throughput
Latency guarantees	No	Yes	Yes, 99% of reads in <10 ms and writes in <15 ms
High availability	No	Yes	Yes, Azure Cosmos DB is always on, has well-defined PACELC tradeoffs, and offers automatic and manual failover options
Data model + API	Relational + SQL	Multi-model + OSS API	Multi-model + SQL + OSS API (more coming soon)
SLAs	Yes	No	Yes, comprehensive SLAs for latency, throughput, consistency, availability

# CosmosDB Resource Model

Resource Model

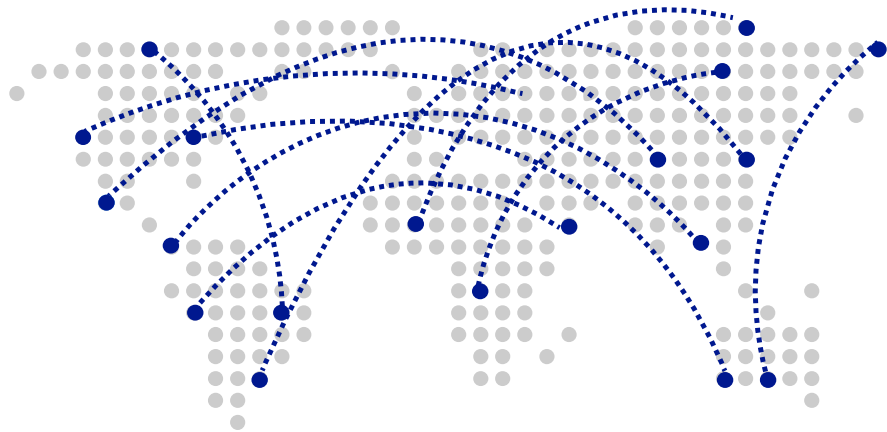


Depending on the API, container and item resources are projected as specialized resource types



API	Container is projected as ...	Item is projected as ...
DocumentDB SQL	Collection	Document
MongoDB	Collection	Document
Cassandra	Table	Entity
Azure Table Storage	Table	Item
Gremlin	Graph	Node and Edge

# CosmosDB



## Turnkey Global Distribution

Worldwide presence as a Foundational Azure service

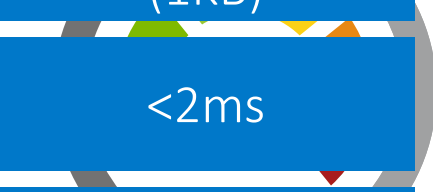
Automatic multi-region replication

Multi-homing APIs

Manual and automatic failovers

Designed for High Availability

# CosmosDB



	Reads (1KB)	Indexed writes (1KB)
P50	<2ms	<6ms
P99	<10ms	<15ms

Guaranteed low latency at P99 (99<sup>th</sup> percentile)

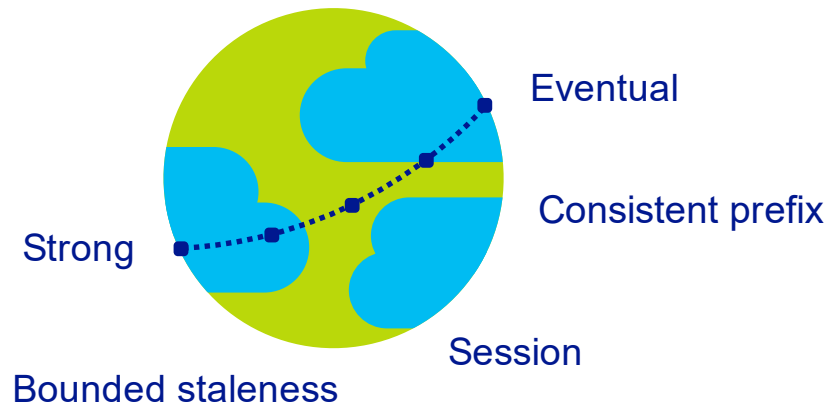
Requests are served from local region

Single-digit millisecond latency worldwide

Write optimized, latch-free database engine designed for SSD

Synchronous automatic indexing at sustained ingestion rates

# CosmosDB



## Multiple, well-defined consistency choices

Global distribution forces us to navigate the CAP theorem

Writing correct distributed applications is hard

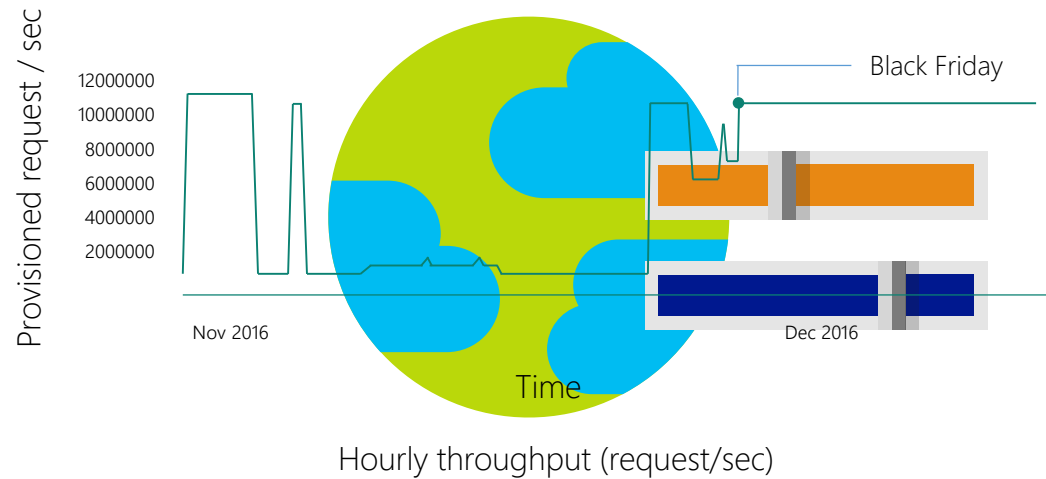
Five well-defined consistency levels

Intuitive and practical with clear PACELC tradeoffs

Programmatically change at anytime

Can be overridden on a per-request basis

# CosmosDB



## Elastically scalable storage and throughput

Single machine is never a bottle neck

Transparent server-side partition management

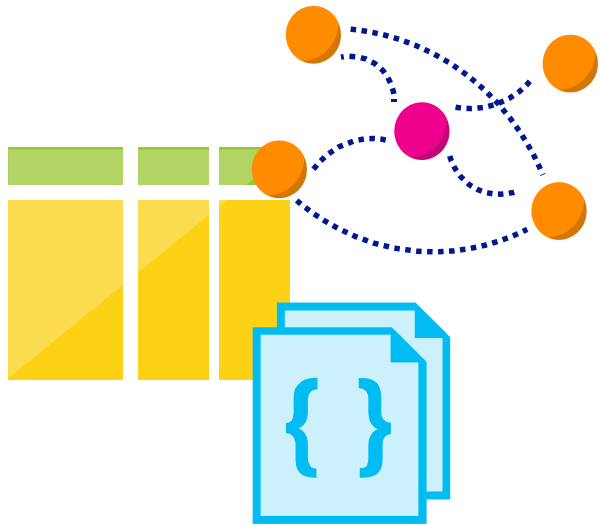
Elastically scale storage (GB to PB) and throughput (100 to 100M req/sec) across many machines and multiple regions

Automatic expiration via policy based TTL

Pay by the hour, change throughput at any time for only what you need

Support for both request per second and requests per minute to handle spikes cost-effectively

# CosmosDB



## Multi-model, multi-API

Database engine operates on Atom-Record-Sequence type system

All data models can be efficiently translated to ARS

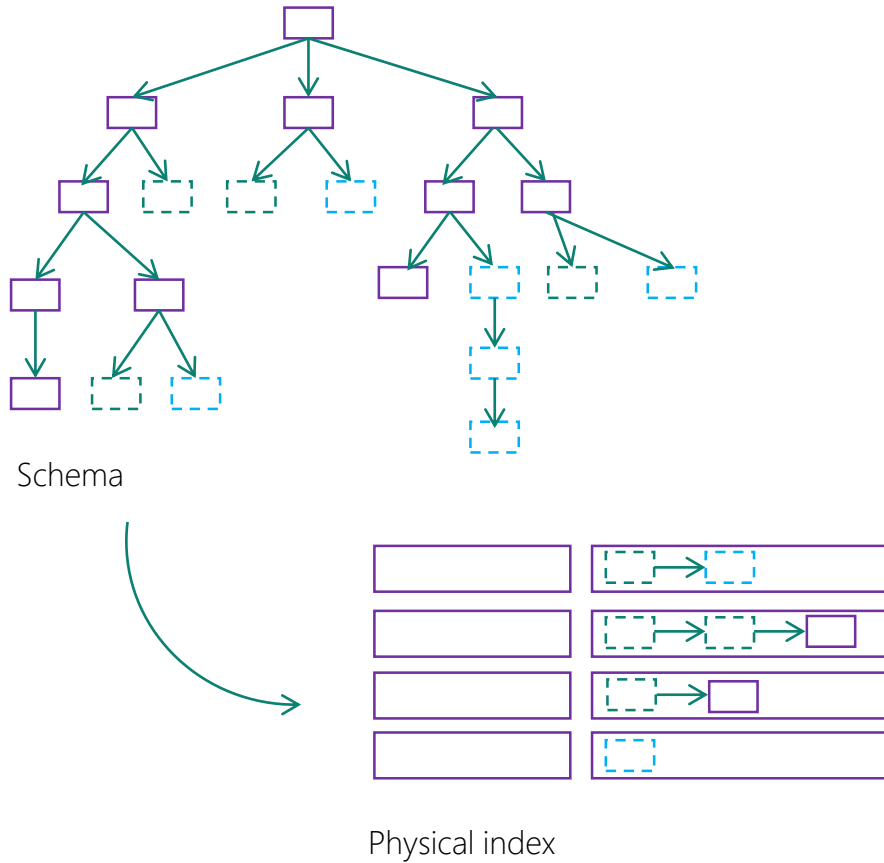
Multi-model: Key-value, Document, and Graph

Multi-API: SQL (DocumentDB), MongoDB, Table, and Gremlin

More data-models and APIs to be added



# CosmosDB



## Schema-agnostic, automatic indexing

At global scale, schema/index management is painful

Automatic and synchronous indexing

Hash, range, and geospatial

Works across every data model

Highly write-optimized database engine

# CosmosDB



## Industry-leading, enterprise-grade SLAs

99.99% availability – even with a single region

Made possible with highly-redundant storage architecture

Guaranteed durability – writes are majority quorum committed

First and only service to offer SLAs on:

- Low-latency
- Consistency
- Throughput

# CosmosDB



## Security & Compliance

Always encrypted at rest and in motion

Fine grained "row level" authorization

Network security with IP firewall rules

Comprehensive Azure compliance certification:

ISO 27001

ISO 27018

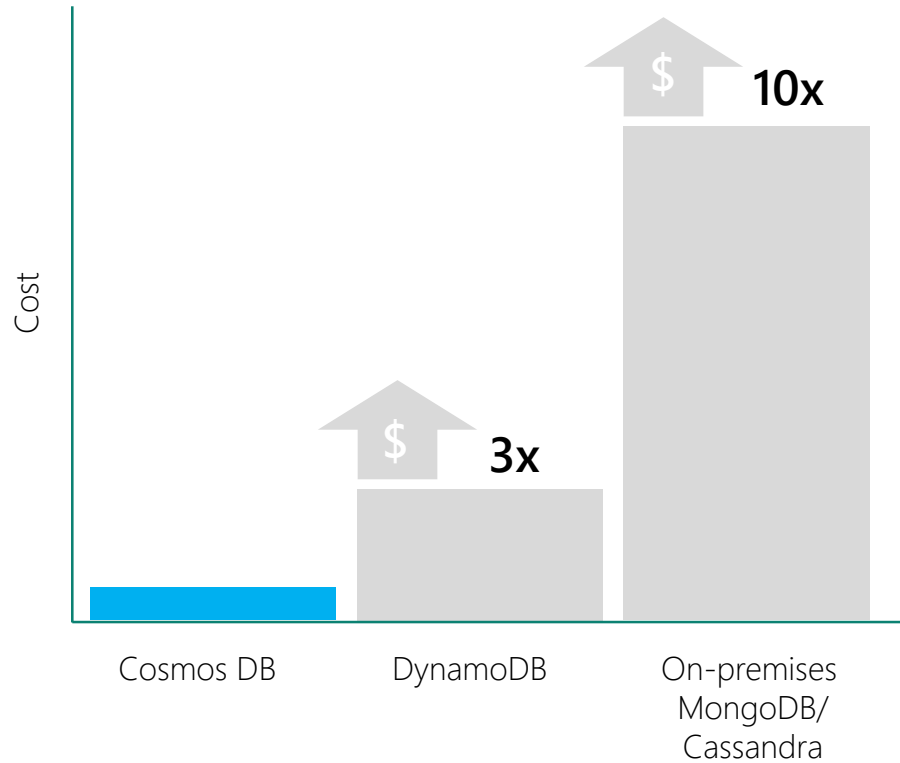
EUMC

HIPAA

PCI

SOC1 and SOC2

# CosmosDB



## Lowest Total Cost of Ownership (TCO)

Deeply exploit cloud core properties and economies of scale

Significantly cheaper than DynamoDB, Cassandra, Cloud Spanner and MongoDB

Designed from the ground up as a multi-tenant service with end-to-end resource governance to provide performance isolation.

Fully managed as a service - much lower operation cost



# Design for Azure Data Services

# Additional Azure Services Included

- [Data Catalog](#)
- [Azure Data Factory](#)
- [SQL Data Warehouse](#)
- [Azure Data Lake Analytics](#)
- [Azure Analysis Services](#)
- [Azure HDInsight](#)

# Hands-on Labs

# Labs

## **Create a storage account**

<https://docs.microsoft.com/en-us/azure/storage/common/storage-quickstart-create-account?toc=%2Fazure%2Fstorage%2Fblobs%2Ftoc.json&tabs=portal>

## **Upload, download, and list blobs using the Azure portal**

<https://docs.microsoft.com/en-us/azure/storage/blobs/storage-quickstart-blobs-portal>

## **Create an Azure SQL database in the Azure portal**

<https://docs.microsoft.com/en-us/azure/sql-database/sql-database-get-started-portal>

## **Use the SQL Query editor to connect and query data**

<https://docs.microsoft.com/en-us/azure/sql-database/sql-database-connect-query-portal>



