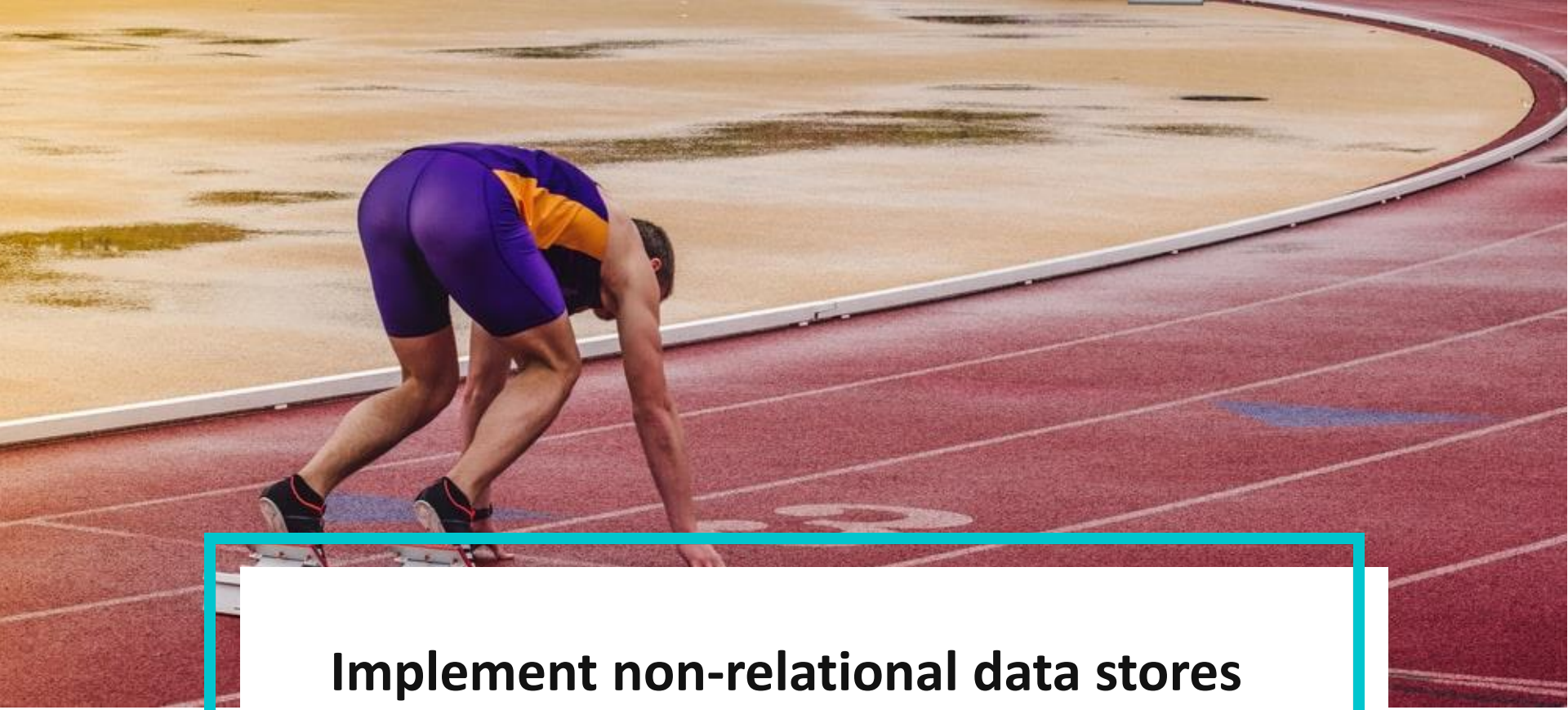


Implement non-relational data stores

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

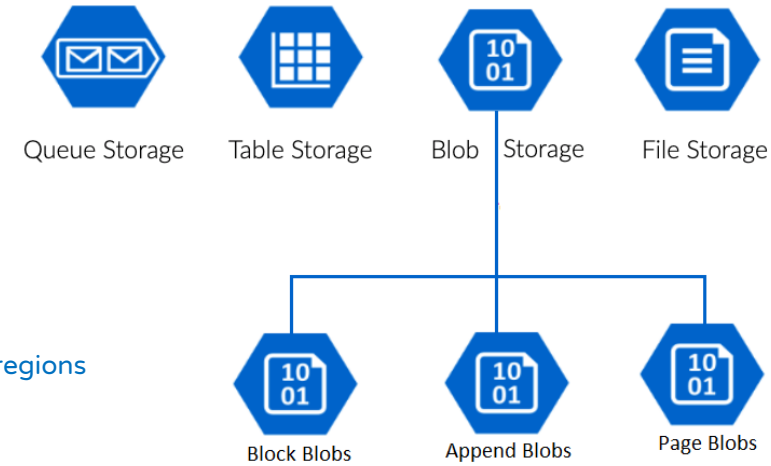
- Features
- Multi-Model and 5 Consistency levels
- Database and Containers
- Throughput and request units (RUs)
- Partitioning and Scaling
- Global Distribution, Multi-master and Failover
- Time to Live, CLI and Pricing
- Security options

Azure Storage Service

Fast, reliable, and private connection to Azure

Azure Storage Service

- 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
  - Azure Blobs: Text and binary data
  - Azure Files: Managed file shares (SMB Protocol)
  - Azure Queues: Messaging
  - Azure Tables: NoSQL 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

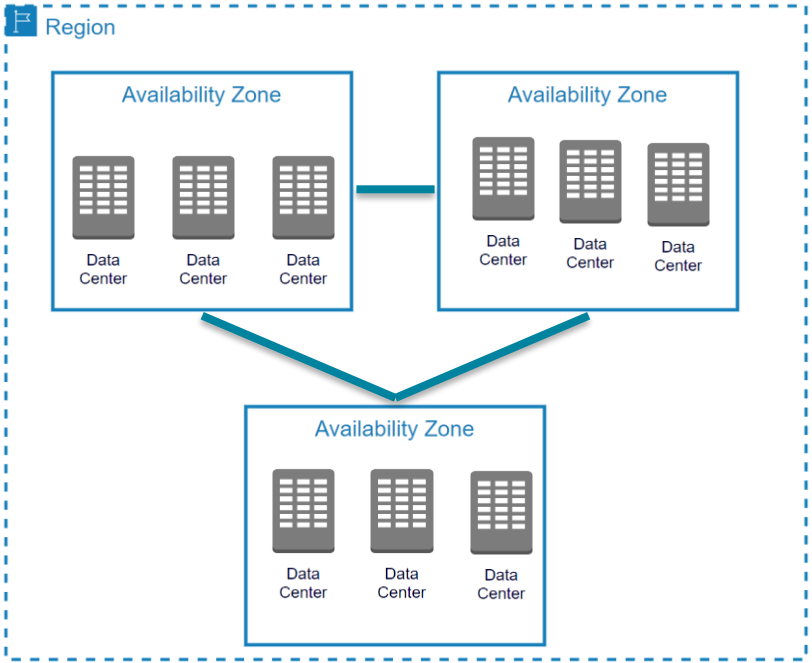
Azure Storage Data Redundancy

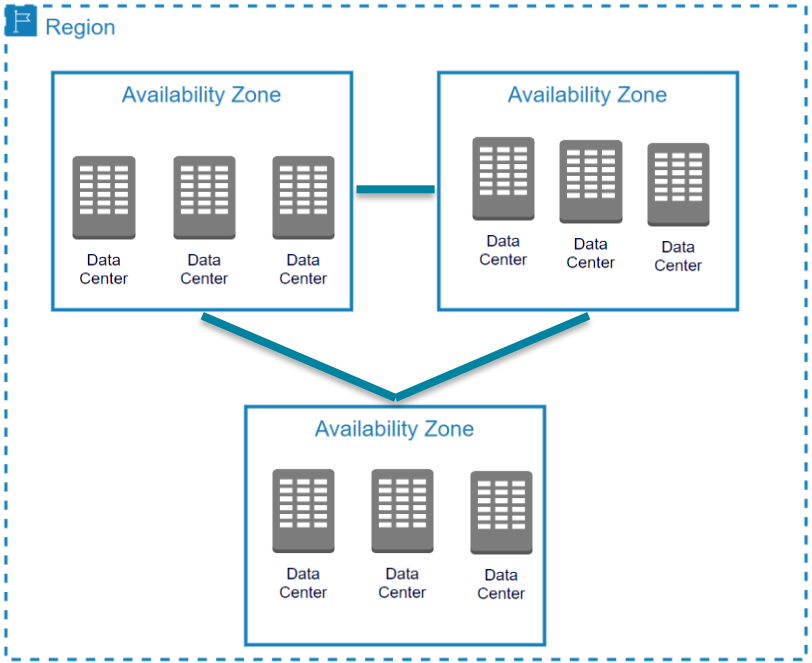
Protect your data from hardware failures, network or power outage, and natural disasters

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
  - 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.
- 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).

Azure Storage Redundancy





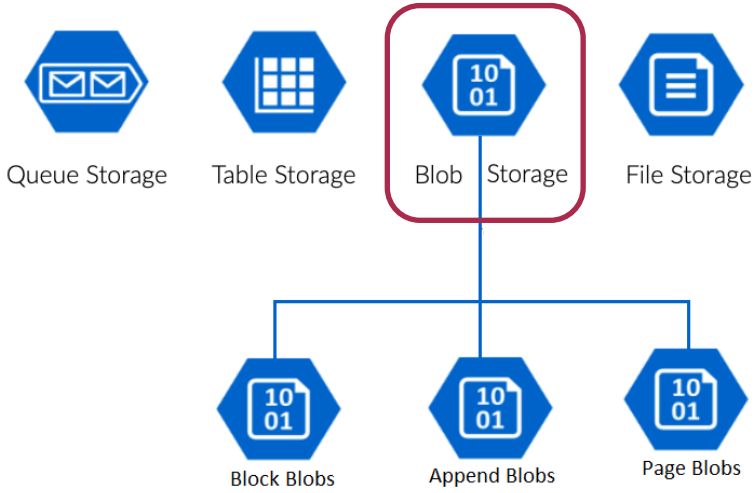
- Locally redundant storage (LRS) – Three synchronous copies in same data center
- Zone-redundant storage (ZRS) – Three synchronous copies in three availability zones (AZs)
- Geo-redundant storage (GRS) - LRS + Asynchronous copy to secondary region (three more copies using LRS) – Read only access
  -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

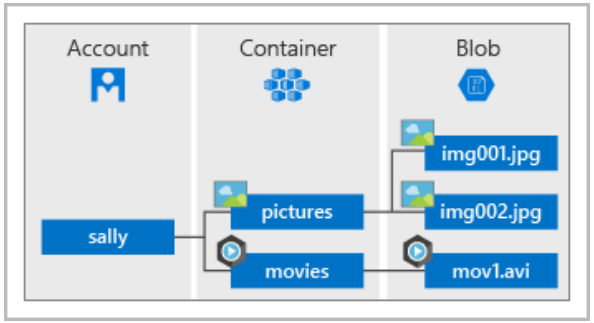
Azure Blob Storage

Binary Large Object

Blob Storage

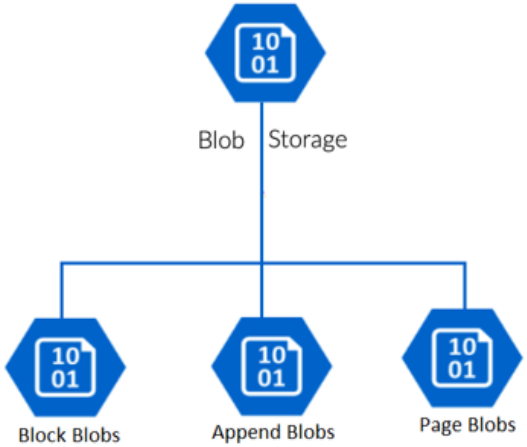
- **Blob** - Binary Large Object
  - Any type or format
  - Text, Images, audio, video, excel, backup files
- **Use cases:**
  - Storing files for shared access
  - Video and audio streaming
  - Storing data for analysis (Data Lake Gen2)
  - Writing to the log file
  - Storing data for disaster recovery,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





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

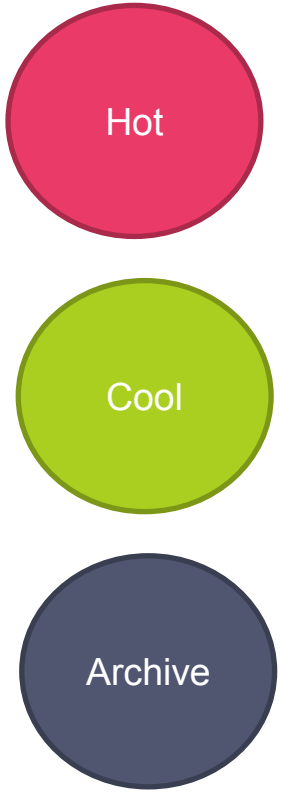


Storage Access Tiers

Organize your data based on attributes like frequency of access and planned retention period.

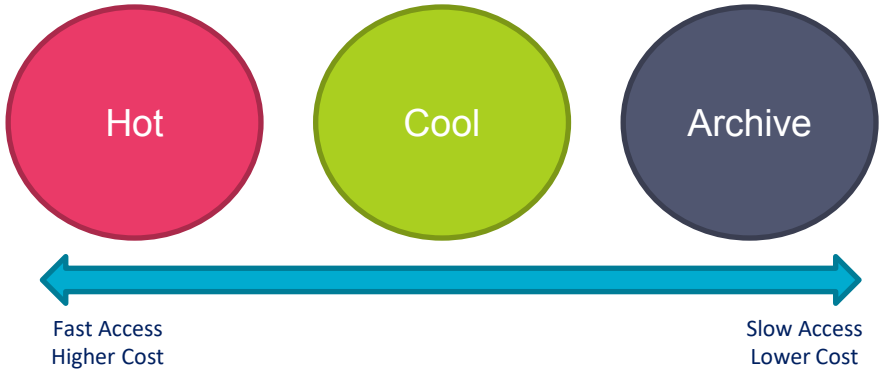
Storage Access Tiers

- 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

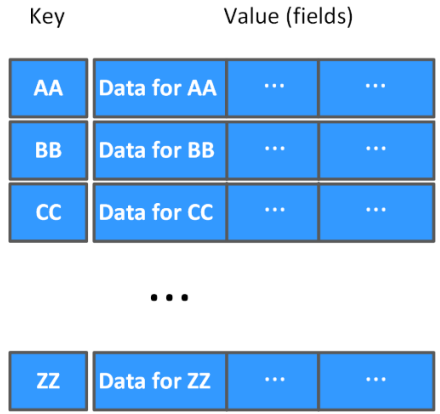


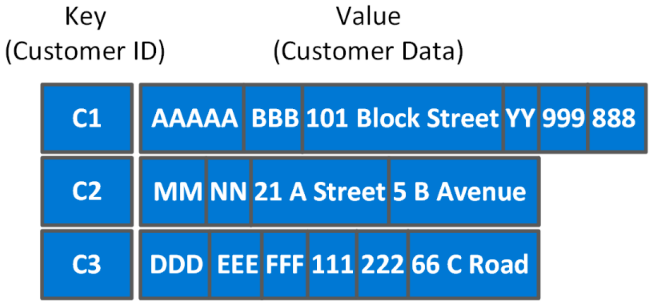
Azure Table Storage

A NoSQL key-value store

Azure Table Storage

- NoSQL key-value Storage
- 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.



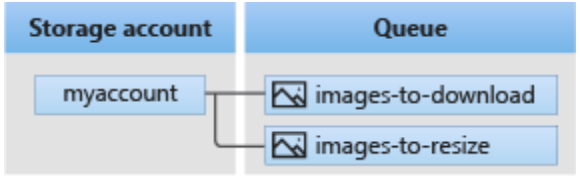


Azure Queue Storage

Message queuing service to store large numbers of messages.

Azure Queue Storage

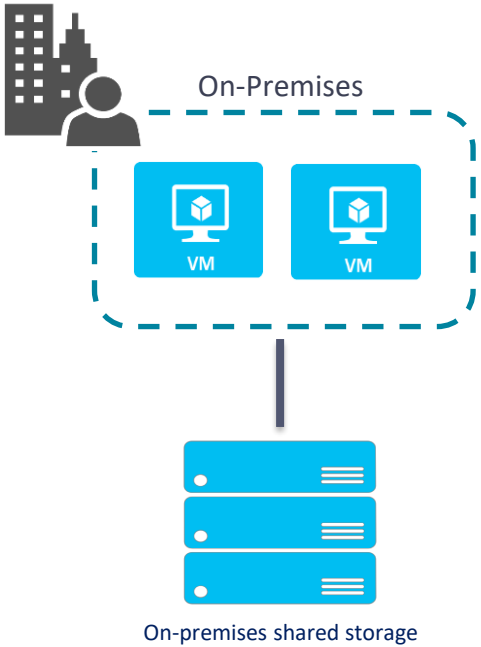
- Store large numbers of messages.
- Access messages via authenticated calls using HTTP or HTTPS.
- May contain millions of messages, up to the total capacity limit of a storage account.
- Queues are commonly used to create a backlog of work to process asynchronously.



Azure File Storage

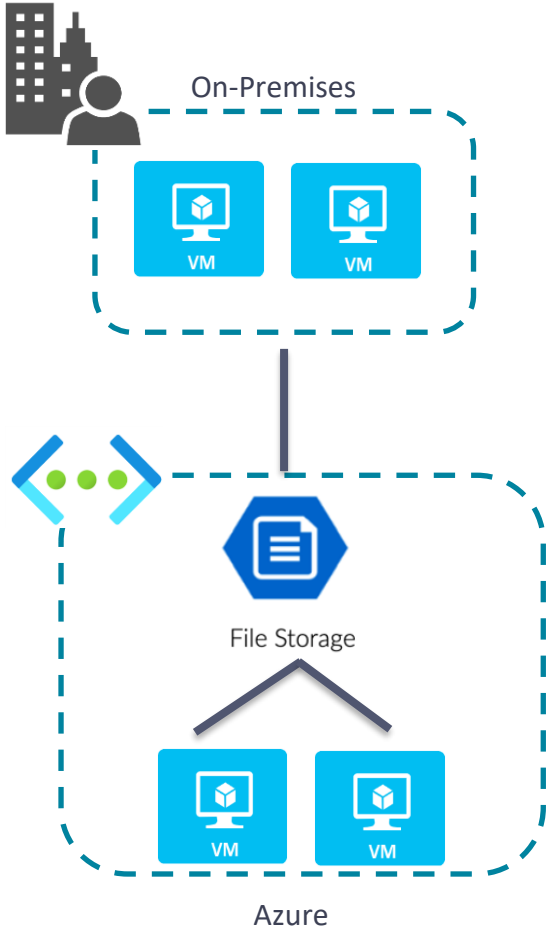
Simple, secure, and serverless enterprise-grade cloud file shares

Azure File Storage



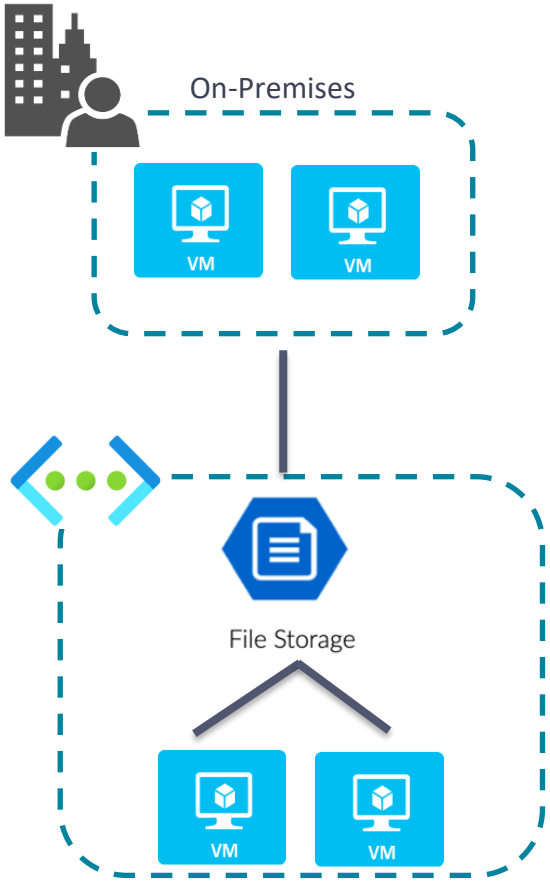
Challenges

- 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.

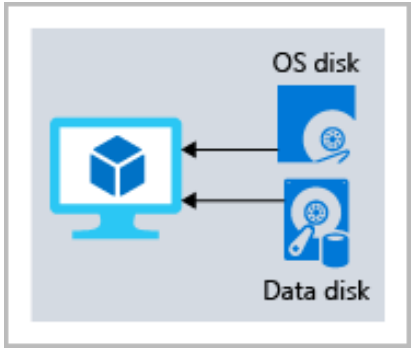


Disk Storage 

High-performance, highly durable block storage for Azure Virtual Machines

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.
- 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:
  - 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



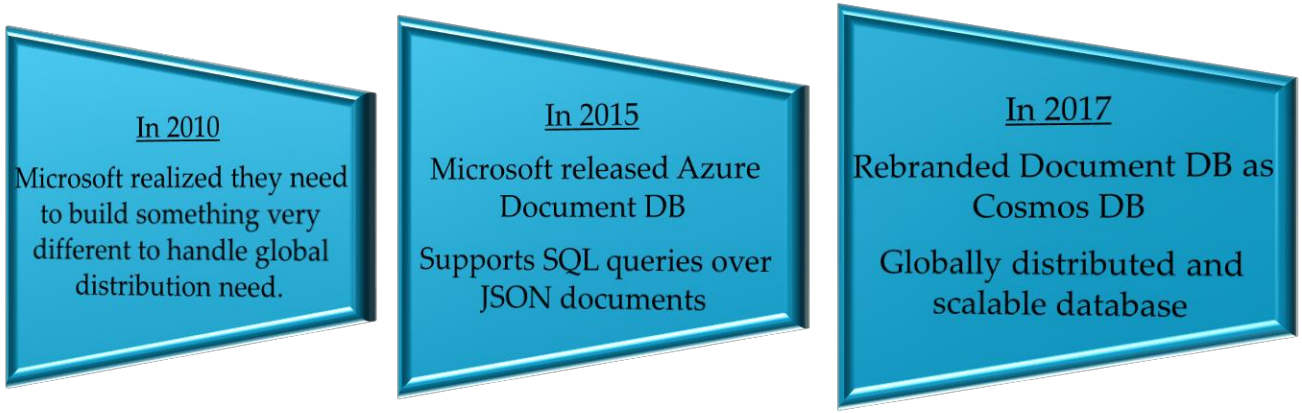
Why Cosmos DB?

What traditional databases were lacking?

Challenges with globally distributed Databases

- Long time
- Lot of effort
- Need own infrastructure
  - Teams
  - Data centers etc.

How Cosmos DB evolved?



Why Cosmos DB?

FULLY MANAGED

- Database as a service (DaaS)
  - Serverless architecture
  - No operational overhead
- No schema or Index management

GLOBALLY DISTRIBUTED

- Turnkey global distribution

CONSISTENCY CHOICES

- Azure Cosmos DB's support for consistency levels like strong, eventual, consistent prefix, session, and bounded-staleness.

SCALABLE

- Unlimited scale for both storage and throughput.

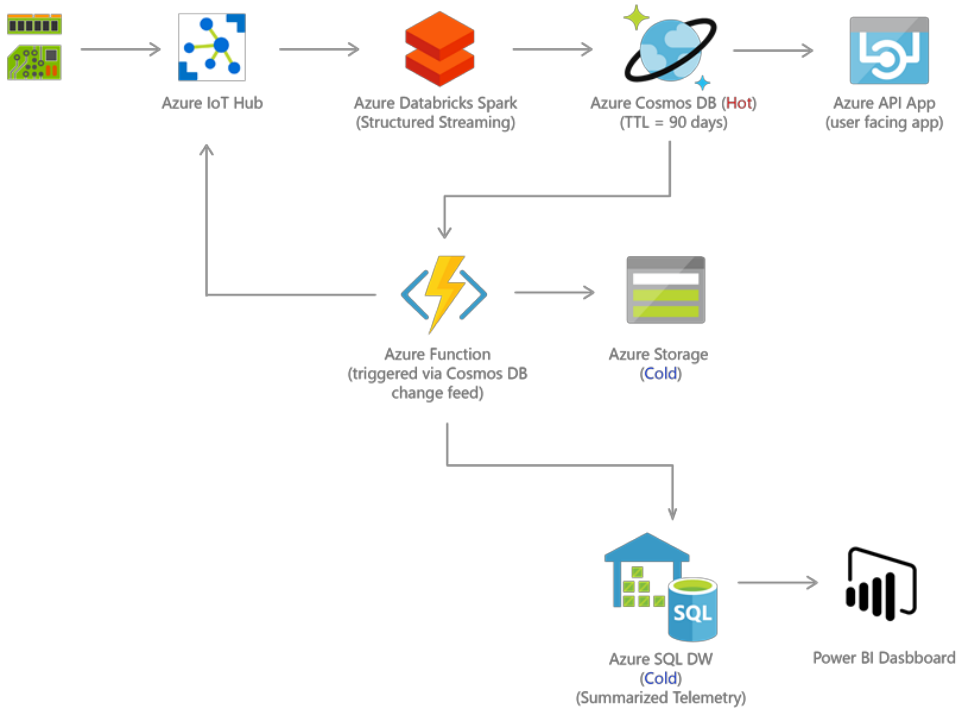
MULTIMODEL & MULTI-LANGUAGE

- Supports Jason documents, table graph and columnar data models
  - Java, .NET, Python, Node.js, JavaScript, etc.

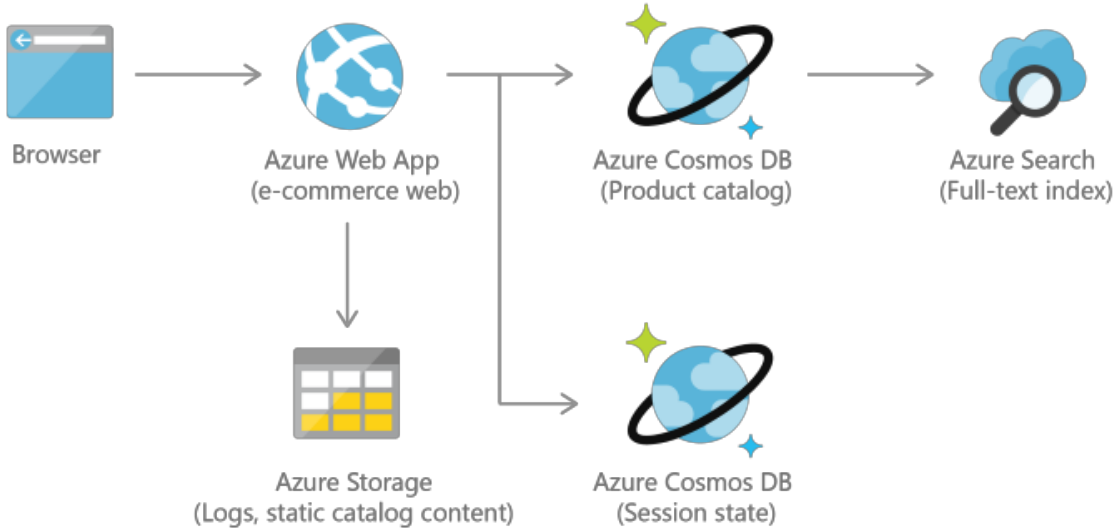
HIGHLY AVAILABLE, RELIABLE & SECURE

- Always on
- 99.999% SLA
- < 10ms latency

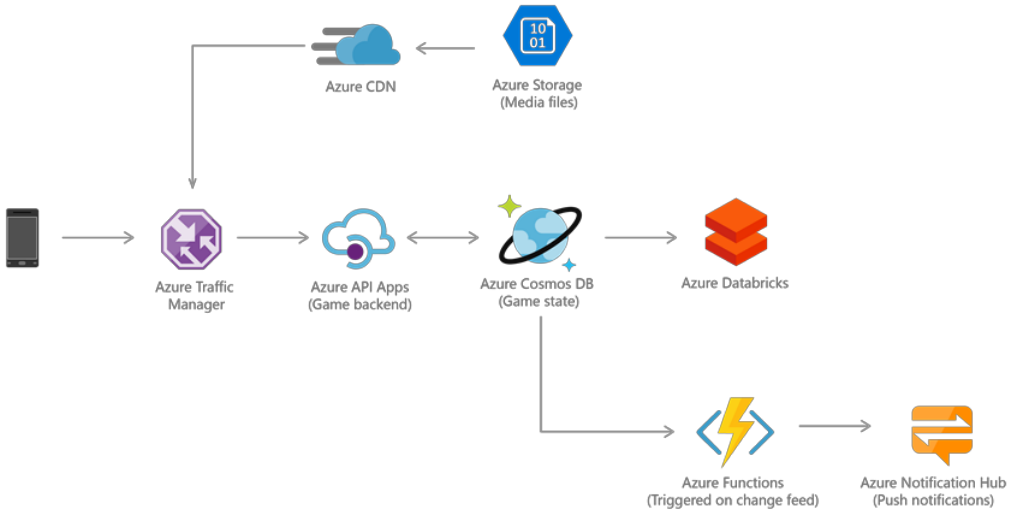
Use case - IOT


Azure IoT Hub
Azure Databricks Spark
(Structured Streaming)
Azure Cosmos DB (Hot)
(TTL = 90 days)
Azure API App
(user facing app)
Azure Function
(triggered via Cosmos DB
change feed)
Azure Storage
(Cold)
SQL
Azure SQL DW
(Cold)
(Summarized Telemetry)
Power BI Dasbboard

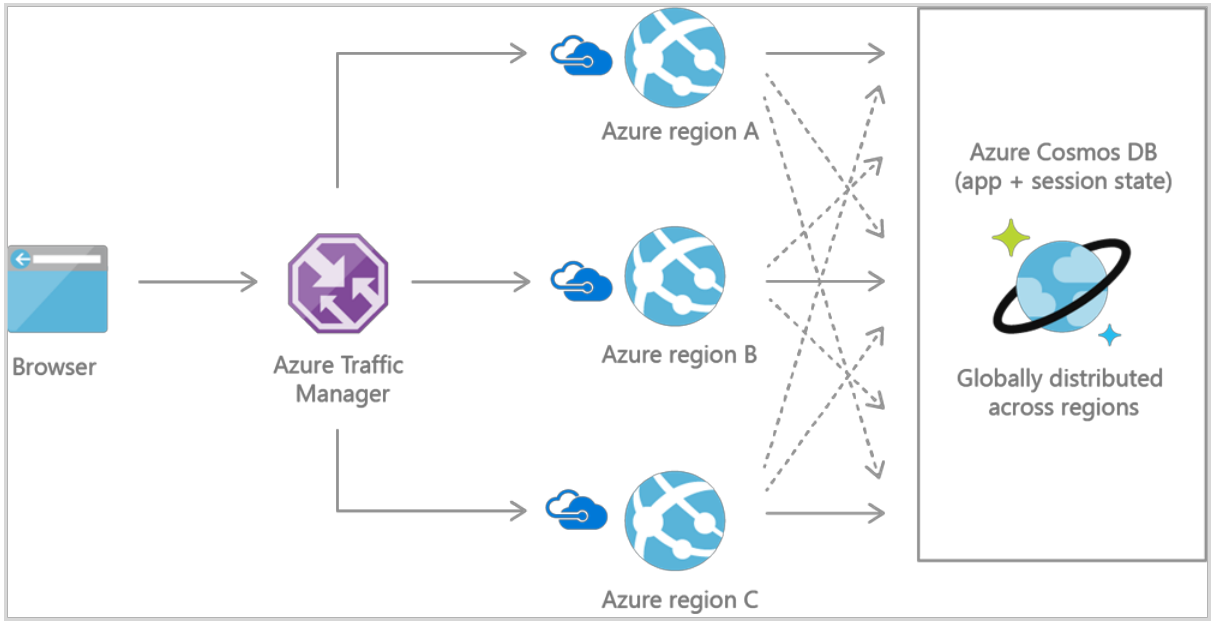
Use case – Retail and Marketing



Use case – Gaming

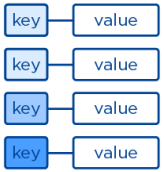


Use case – Web and mobile

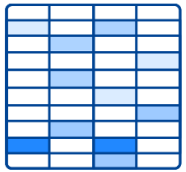


NoSQL

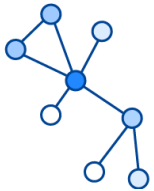
Key-Value



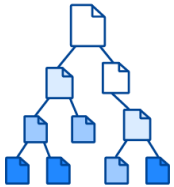
Wide-column



Graph



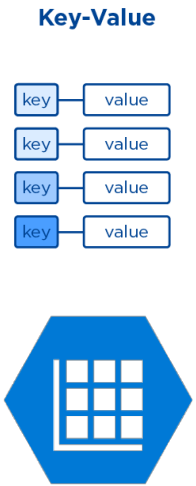
Document



SQL API vs MongoDB API

SQL(CORE) API	MongoDB API
JSON Documents	BSON Documents
Microsoft original Document DB platform Supports server side programming model	Implement Wire protocol Fully compatible with Mongo DB application code
You can use SQL like language to query JSON documents.	Migrate existing Cosmos DB without much change of logic Use SQL(CORE) API for new development

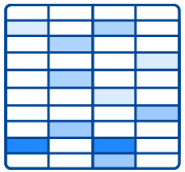
Cosmos DB Table API



- Key-Value store
- Premium offering for Azure Table Storage
- Existing Table Storage customers will migrate to Cosmos DB Table API
- Row value can be simple like number or string
- Row cannot store object

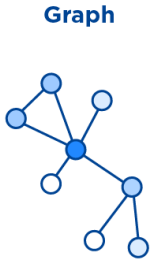
Cosmos DB Cassandra API

Wide-column



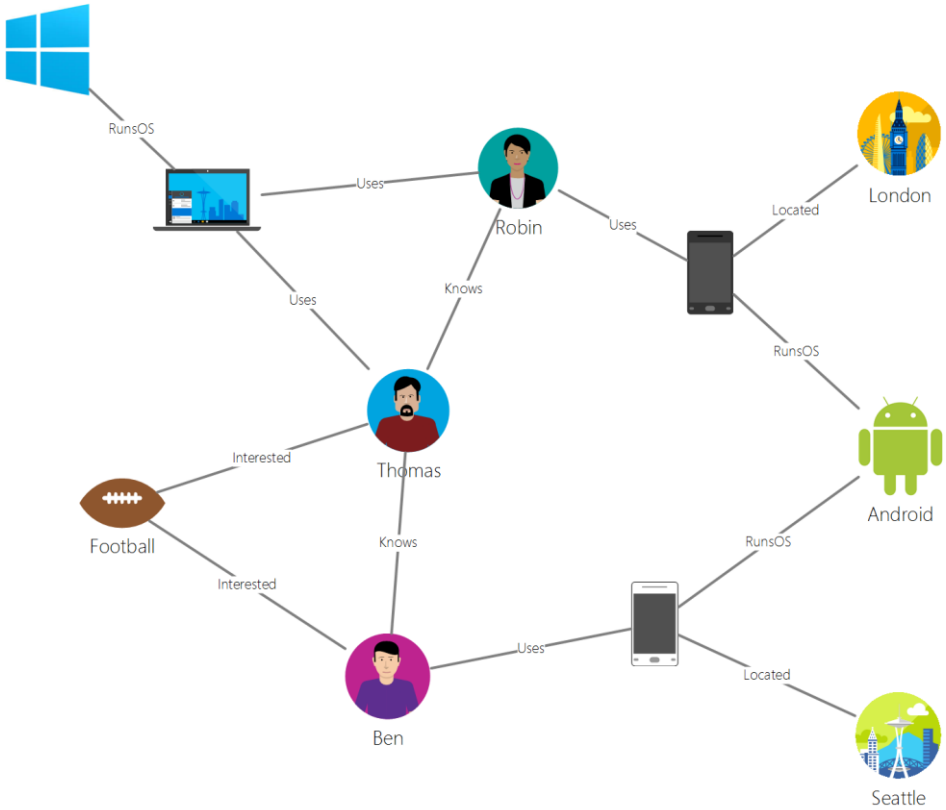
- Wide column No SQL Database
- Name and format of column can vary from row to row.
- Simple migrate your Cassandra application to Cosmos Cassandra API and change connection string.
- Interact
  - Cassandra based tools
  - Data Explorer
  - Programmatically, using SDK (CassandraCSharpdriver)

Cosmos DB Gremlin API

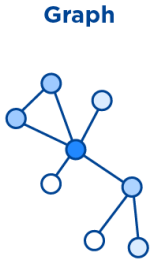


- Graph Data Model
- Real world data connected with each other
- Graph database can persist relationships in the storage layer

Graph Model


RunsOS
Uses
Robin
Uses
London
Located
Uses
Knows
RunsOS
Thomas
Interested
Android
Football
Knows
RunsOS
Interested
Uses
Located
Ben
Seattle

Cosmos DB Gremlin API



- Graph Data Model
- Real world data connected with each other
- Graph database can persist relationships in the storage layer
- Use cases
  - Social networks
  - Recommendation engines
  - Geospatial
  - Internet of things
- Migrate existing apps to Cosmos DB Gremlin API
- Graph traverse a language

Analyze the decision criteria

	Core (SQL)	MongoDB	Cassandra	Azure Table	Gremlin
New projects being created from scratch	✓				
Existing MongoDB, Cassandra, Azure Table, or Gremlin data		✓	✓	✓	✓
Analysis of the relationships between data					✓
All other scenarios	✓				

Azure Table storage vs Cosmos DB Table API

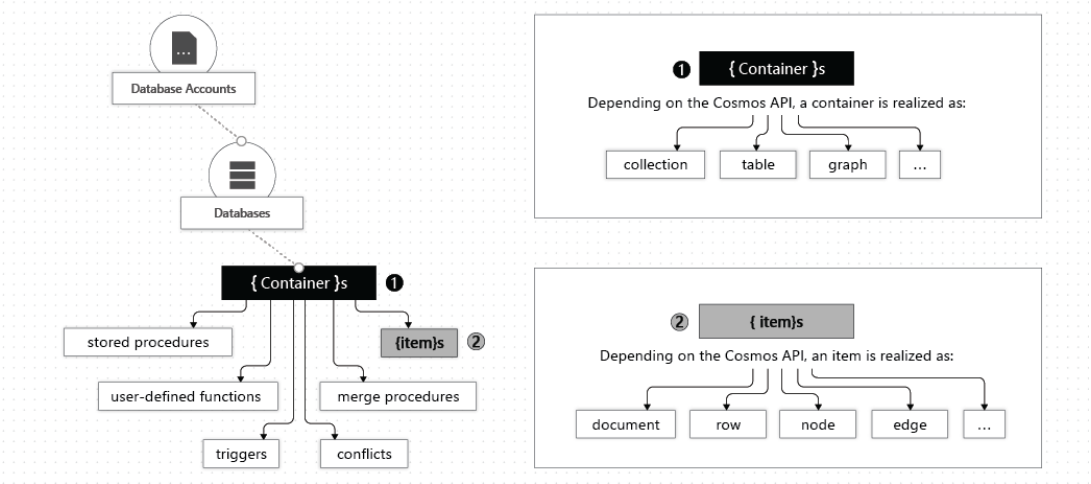
Azure Table Storage

- Geo replication is restricted
  - Only 1 additional pair region
- Support for primary key lookups only
- Price optimized for cold storage
- Lower performance
  - Throughput is capped
  - Latency is higher
- No consistency options

Cosmos DB Table API

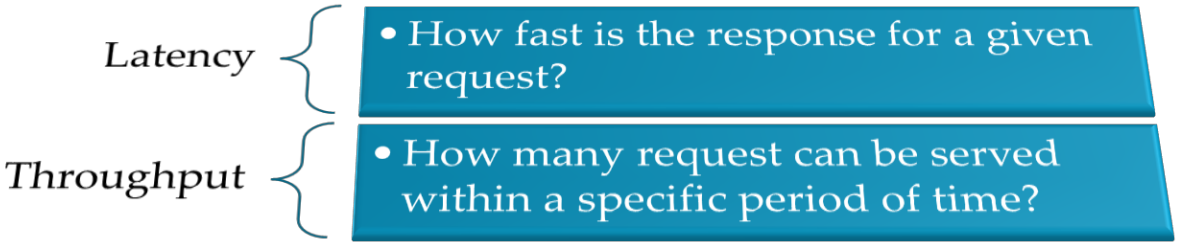
- Geo replication across your choice of any number of regions
- Secondary index support for lookups across multiple dimensions
- Better performance
  - Unlimited and predictable throughput
  - latency is lower
- 5 consistency options

Database Containers and Items

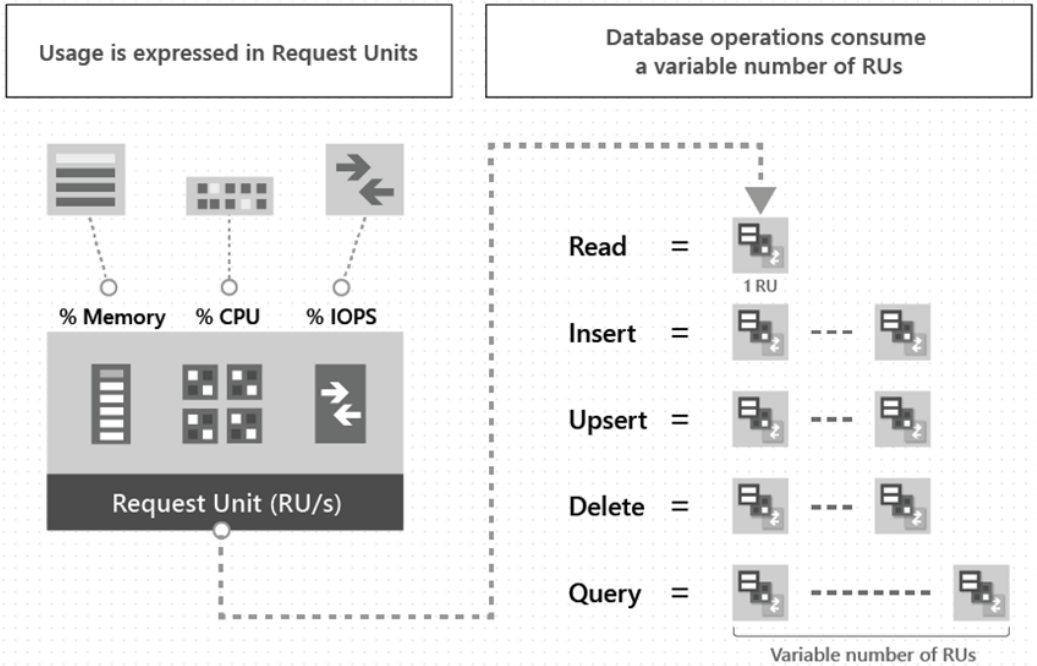


Azure Cosmos entity	SQL API	Cassandra API	MongoDB API	Gremlin API	Table API
Azure Cosmos database	Database	Keyspace	Database	Database	NA
Azure Cosmos container	Container	Table	Collection	Graph	Table
Azure Cosmos item	Document	Row	Document	Node or edge	Item

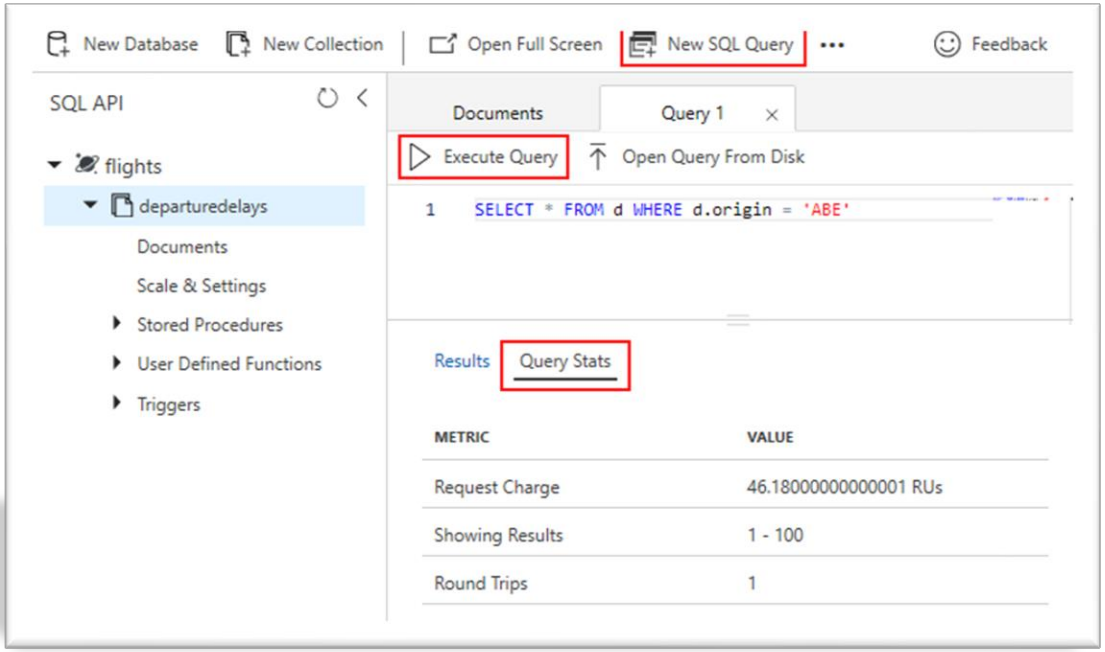
Measuring Performance



Introducing Request Units

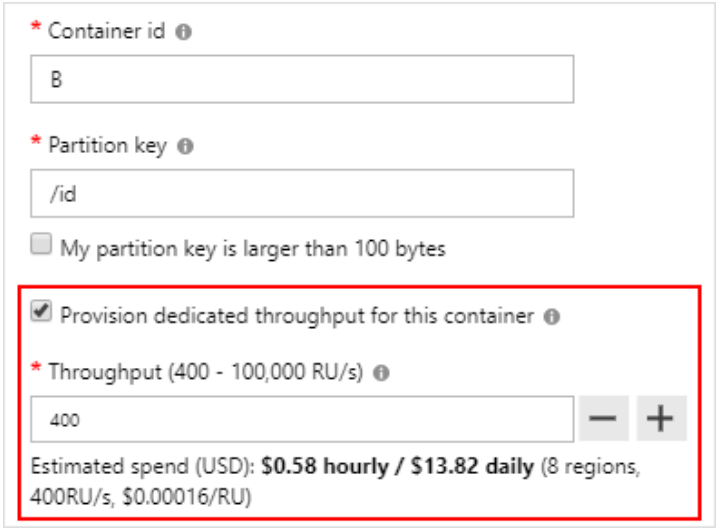


Introducing Request Units

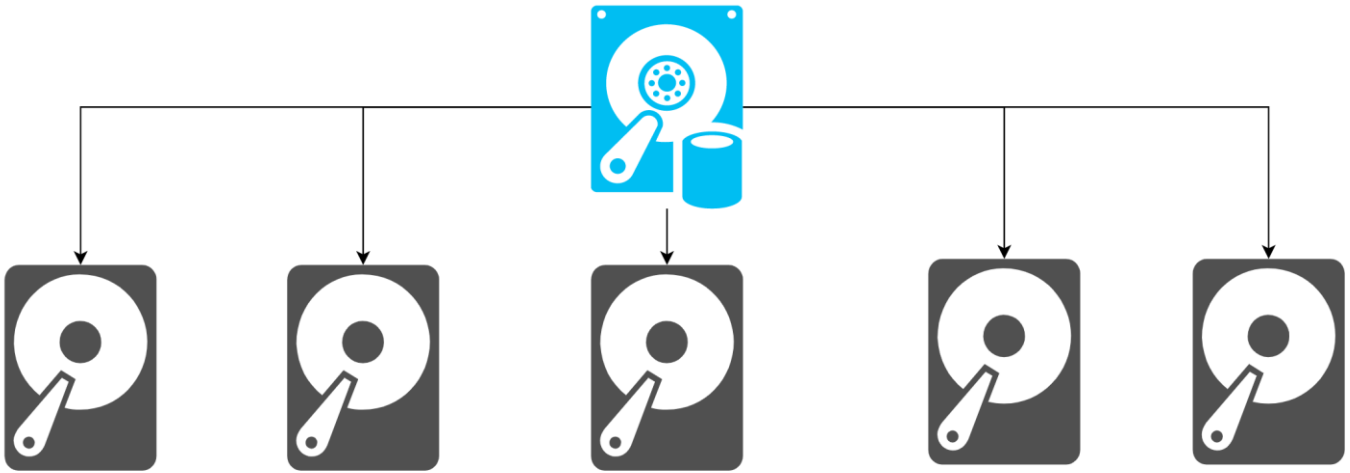


Reserving requests units

- Provision Request units per second (RU/s)
  - How many request units (not requests) per second are available to your application
- Exceeding reserved throughput limits
  - Requests are “throttled” (HTTP 429)



Horizontally Scalable

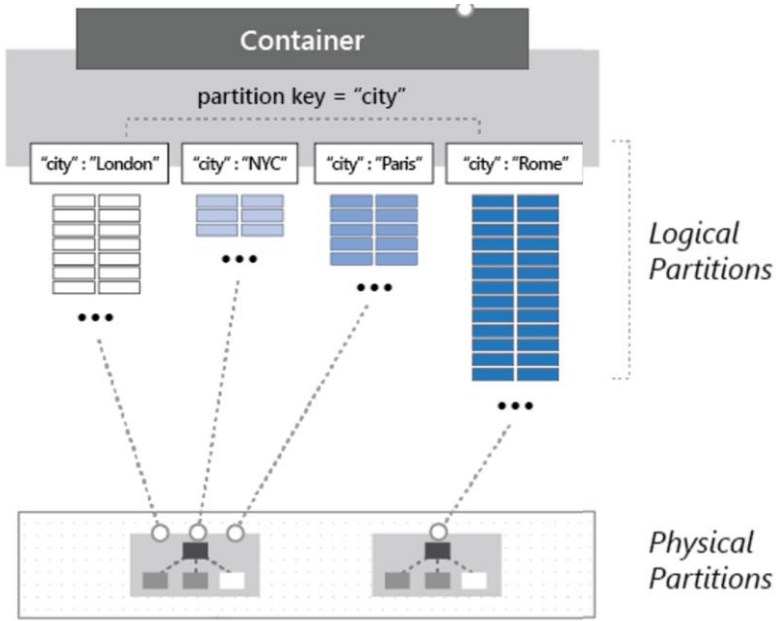


Unlimited Storage

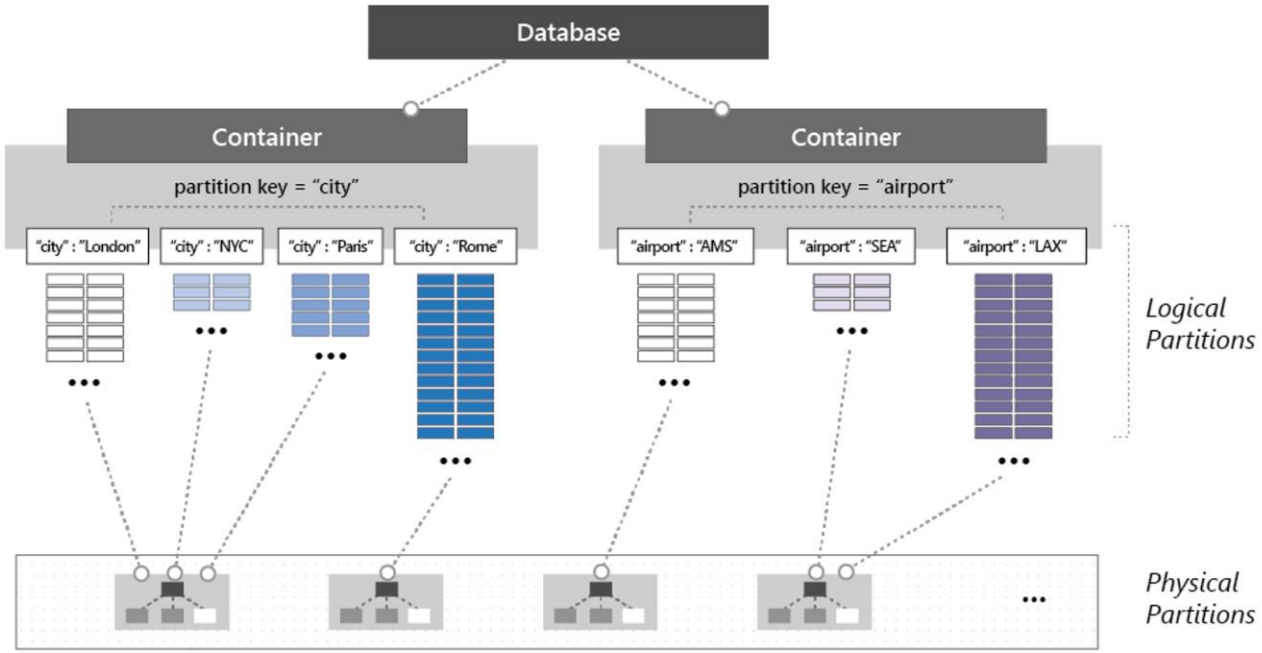
Unlimited Throughput

Partitioning

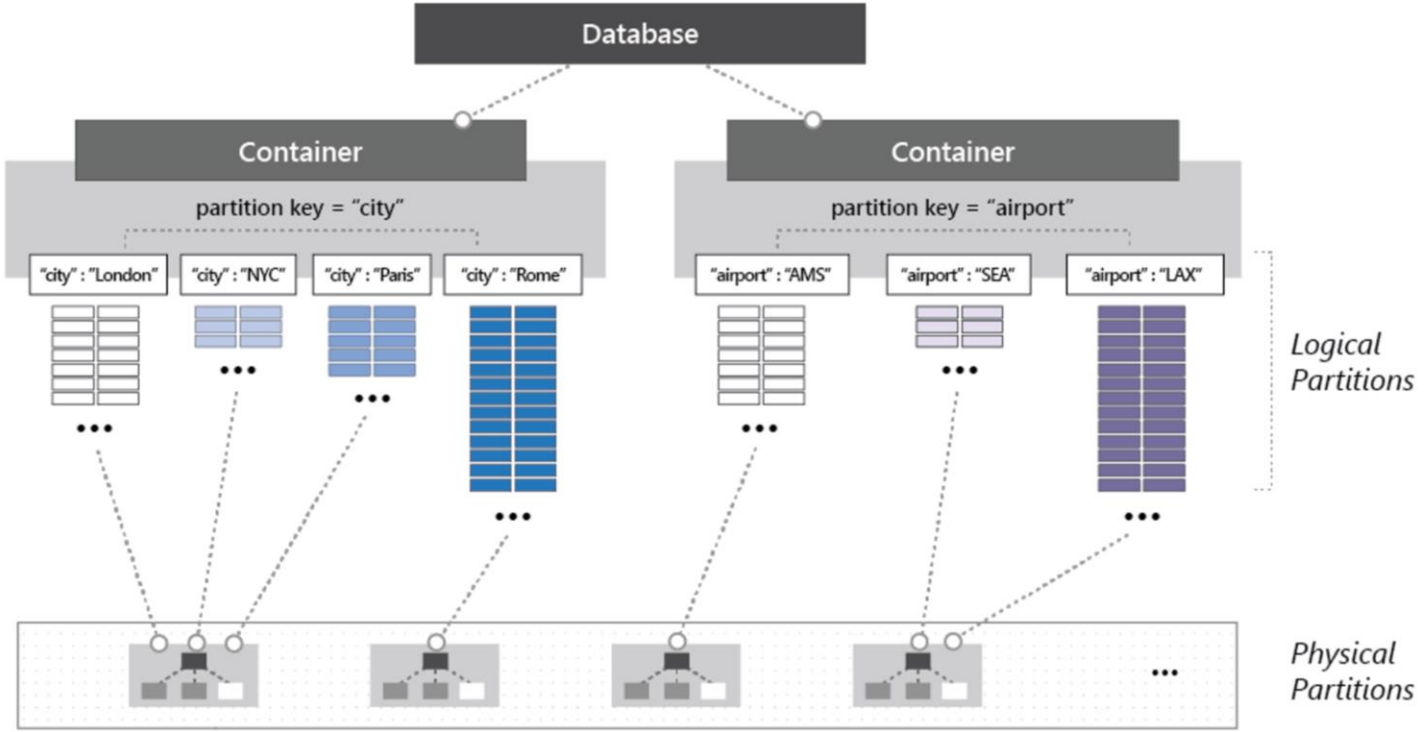
- **Partitioning:** the items in a container are divided into distinct subsets called logical partitions.
- **Partition key** is the value by which Azure organizes your data into logical divisions.
- **Logical partitions** are formed based on the value of a partition key that is associated with each item in a container.
- **Physical partitions:** Internally, one or more logical partitions are mapped to a single physical partition.



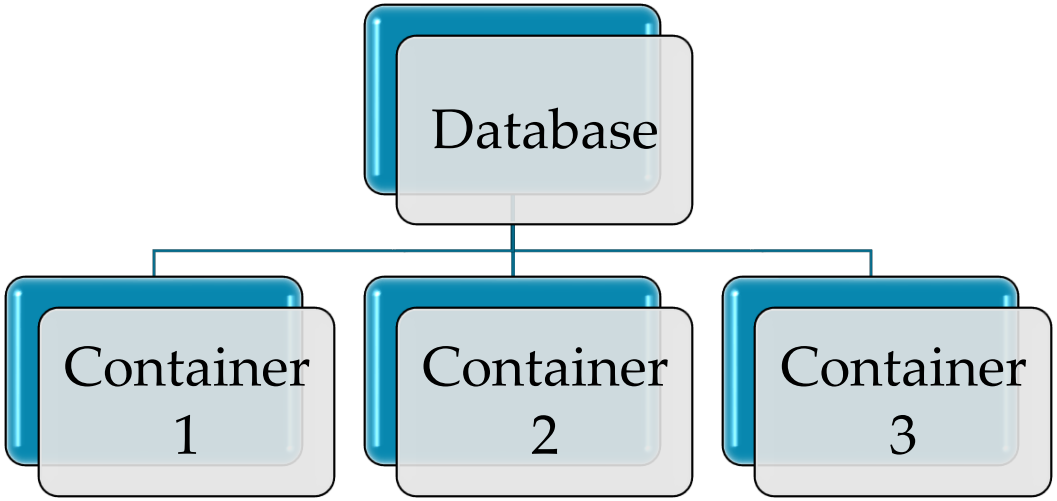
Partitioning



Dedicated vs Shared throughput

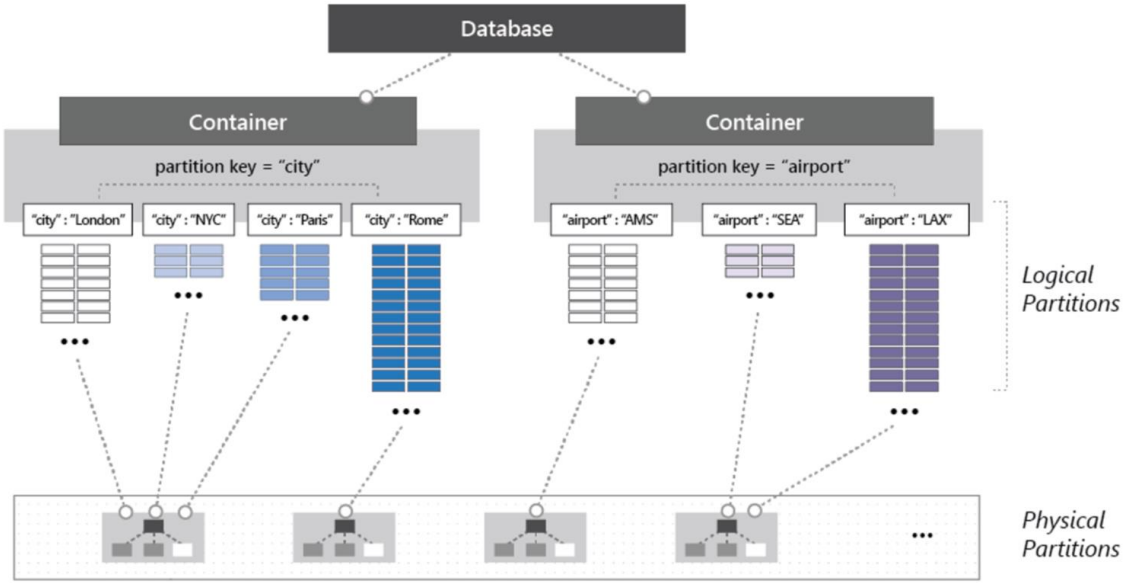


Dedicated vs Shared throughput

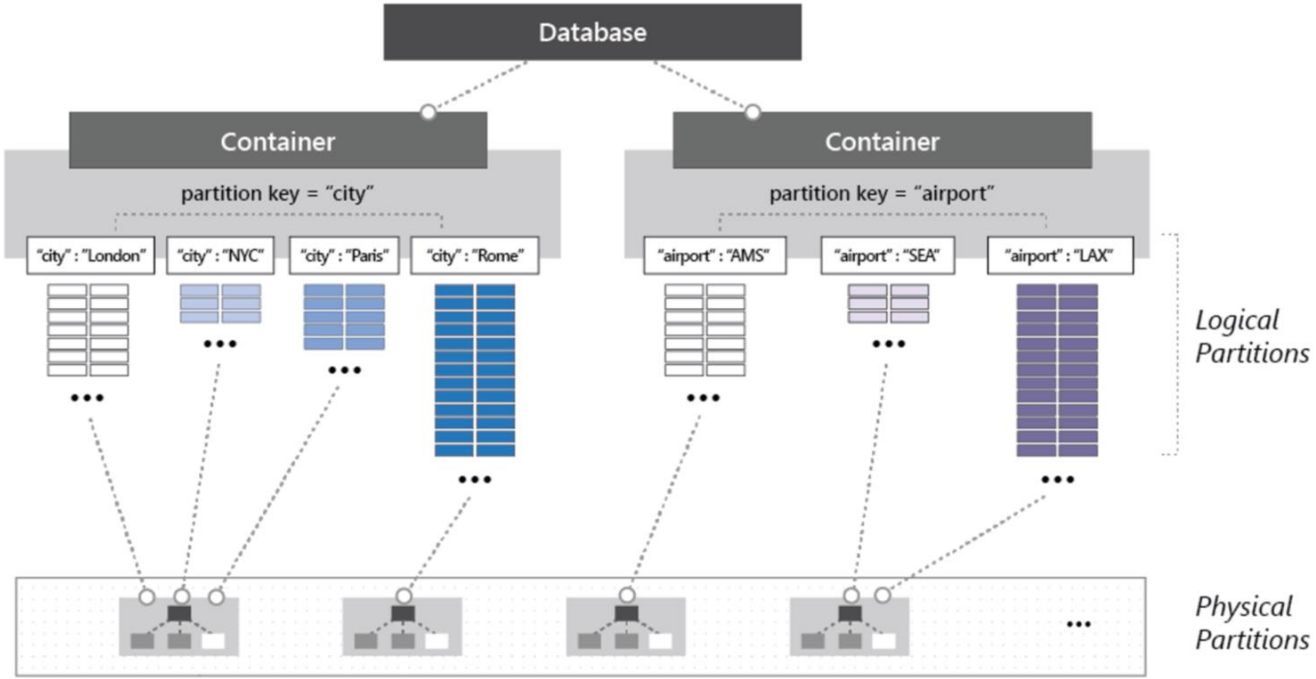


Dedicated vs Shared throughput

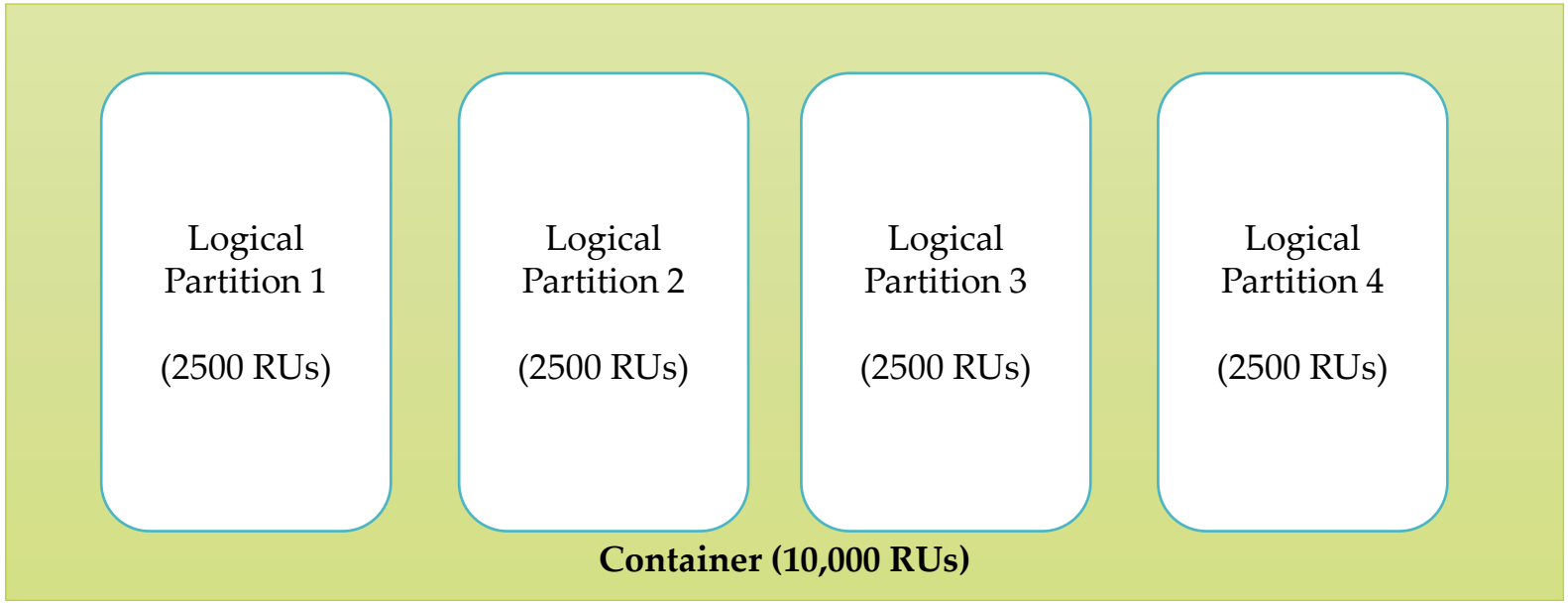
- You can set throughput at:
  - Database level – Shared throughput
  - Container level – Dedicated throughput
  - It is recommend to set throughput at container level.
- Rate-Limited
- Choose at the time of creation



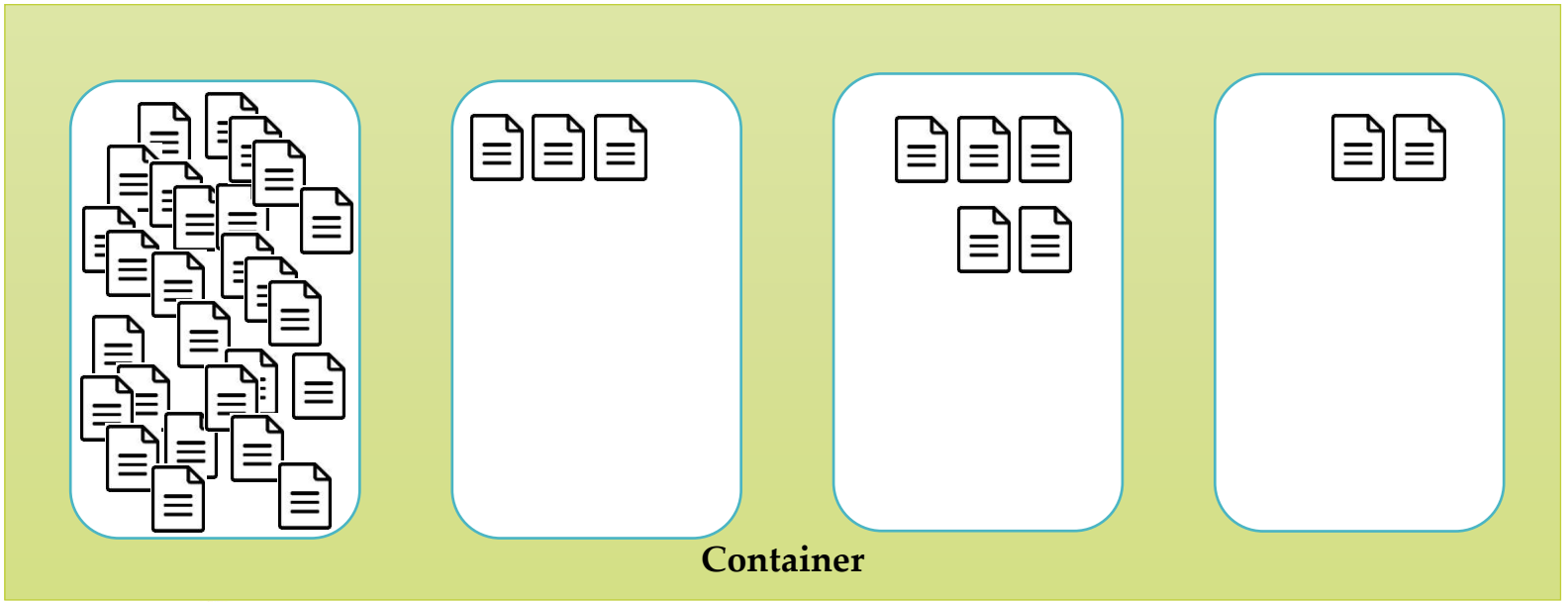
Avoiding hot partition



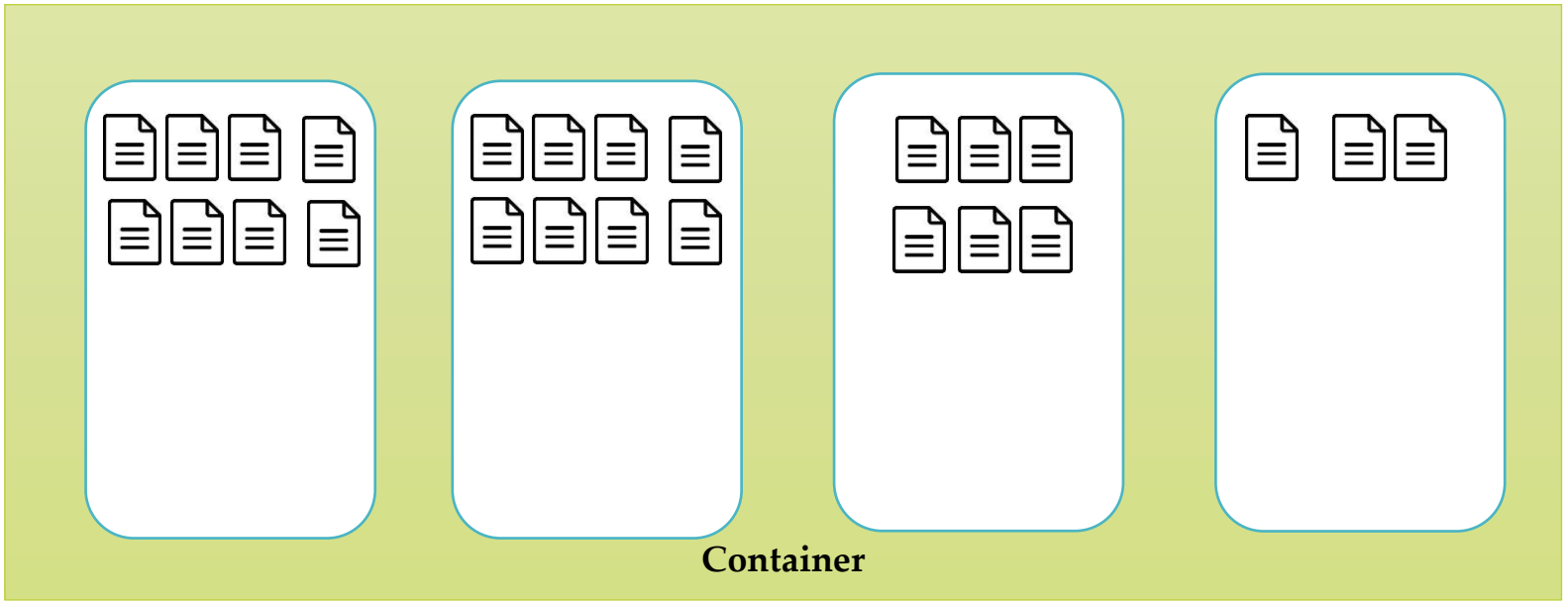
Avoiding Hot Partitions



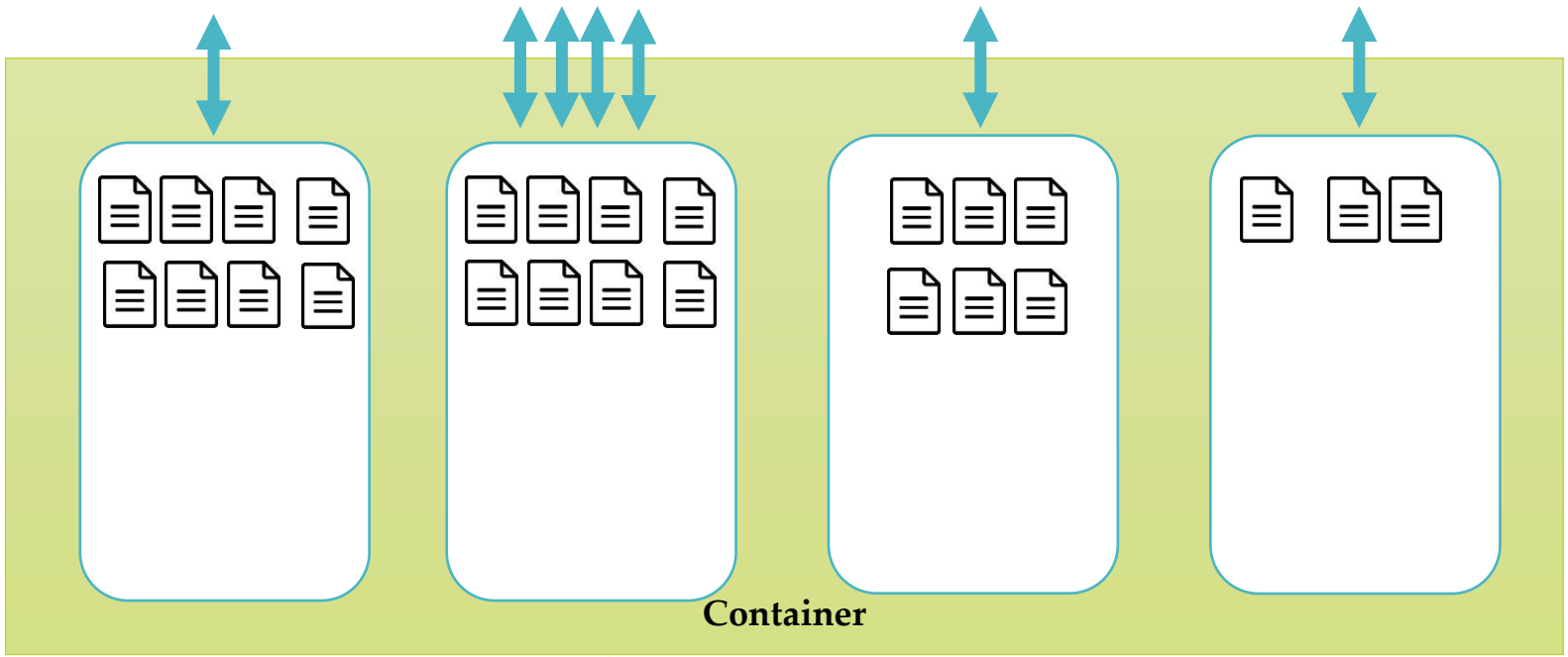
Avoid Hot partitions on storage



Avoiding Hot Partitions at store

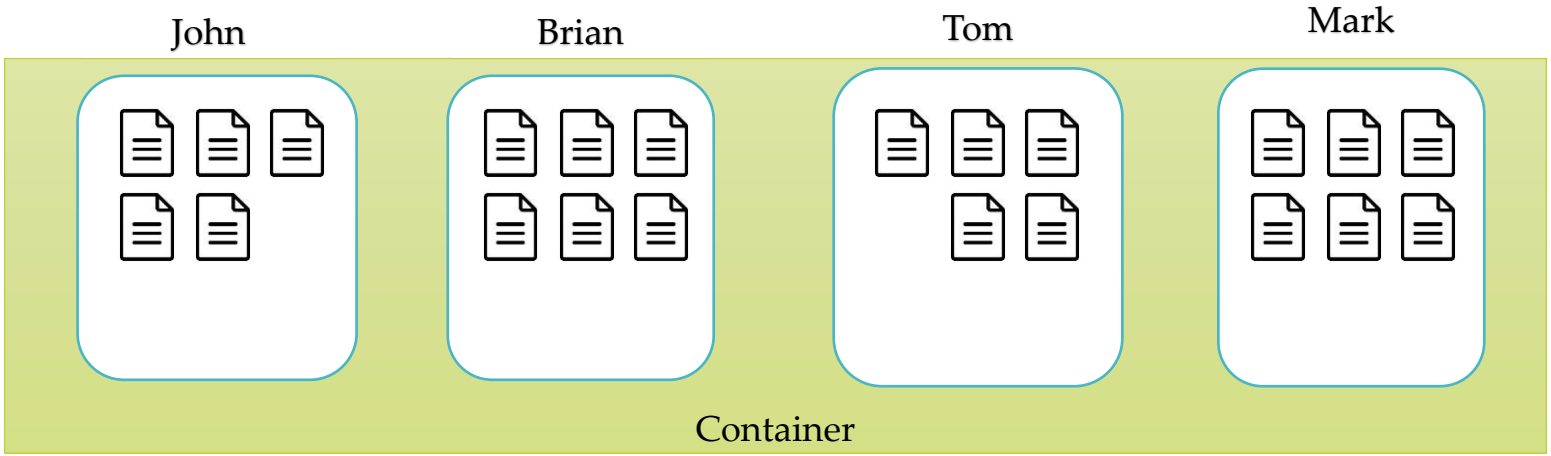


Avoid Hot partitions on throughput



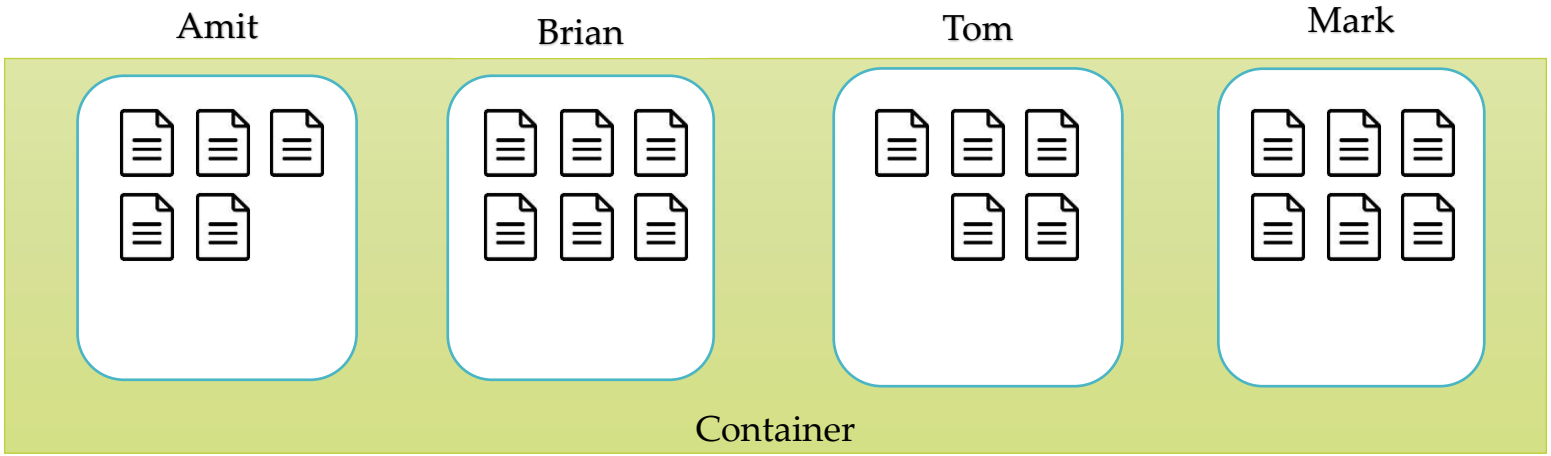
- **Partition key Bad choice:** Current time
- **Partition key Good choices:** User ID, Product ID

Single partition Query



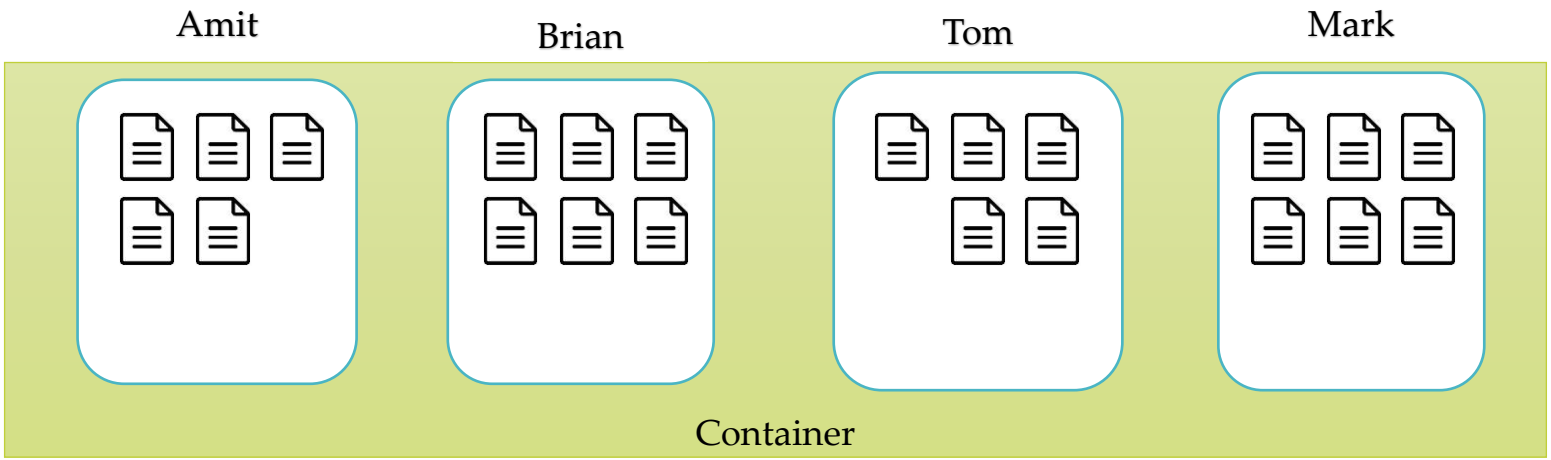
SELECT * FROM c WHERE c.username = ‘Brian’

Cross partition Queries (fan out queries)



SELECT * FROM c WHERE c.favoritecolor= ‘Blue’

Composite Key



Composite Key: CustomerName-mmddyyyy

Choosing a Partition key

- Evenly distribute storage
  - Make sure you pick your partition key that doesn't result in hot spots within your applications
  - Have a high cardinality
  - Don’t be afraid of choosing a partition key that has a large number of values
  - Example User Id & Product Id
- Evenly distribute requests.
  - RUs evenly distribute across all partitions.
  - Review where clause of top queries
- Consider document and partition limit while designing partition key.
  - Max document size – 2 MB
  - Max logical partition size – 20 GB

Choosing a Partition key

Question: Your organization is planning to use Azure Cosmos DB to store vehicle telemetry data generated from millions of vehicles every second. Which of the following options for your Partition Key will optimize storage distribution?

Answer choices:

1. Vehicle model
2. Vehicle Identification Number (VIN) which looks like WDDEJ9EB6DA032037

Choosing a Partition key

Question: Your organization is planning to use Azure Cosmos DB to store vehicle telemetry data generated from millions of vehicles every second. Which of the following options for your Partition Key will optimize storage distribution?

Answer choices:

1. Vehicle model

   Most auto manufactures only have a couple dozen models. This option is potentially the least granular, will create a fixed number of logical partitions, and may not distribute data evenly across all physical partitions.

2. **Vehicle Identification Number (VIN) which looks like WDDEJ9EB6DA032037**

   Auto manufacturers have transactions occurring throughout the year. This option will create a more balanced distribution of storage across partition key values.

Automatic Indexing

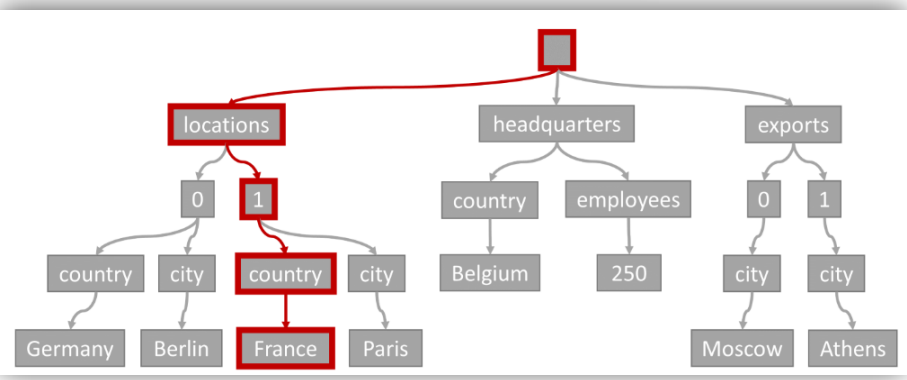
- Index all data without requiring Index management
- Every property of every record automatically index
- Index update synchronously as you create, update or delete items
- Not specific for SQL, but available for all APIs

Automatic Indexing

JSON

```
{
    "locations": [
        { "country": "Germany", "city": "Berlin" },
        { "country": "France", "city": "Paris" }
    ],
    "headquarters": { "country": "Belgium", "employees": 250 },
    "exports": [
        { "city": "Moscow" },
        { "city": "Athens" }
    ]
}
```

```
SELECT location
FROM location IN company.locations
WHERE location.country = 'France'
```




```
/locations/0/country: "Germany"
/locations/0/city: "Berlin"
/locations/1/country: "France"
/locations/1/city: "Paris"
/headquarters/country: "Belgium"
/headquarters/employees: 250
/exports/0/city: "Moscow"
/exports/1/city: "Athens"
```

Time-to-Live

Time to Live (TTL)

- You can set the expiry time for Cosmos DB data items
- Time to live value is configured in seconds.
- The system will automatically delete the expired items based on the TTL value
- Consume only leftover Request units
- Data deletion delay if not enough Request units
  - Though the data deletion is delayed, data is not returned by any queries (by any API) after the TTL has expired.

Global Distribution benefits

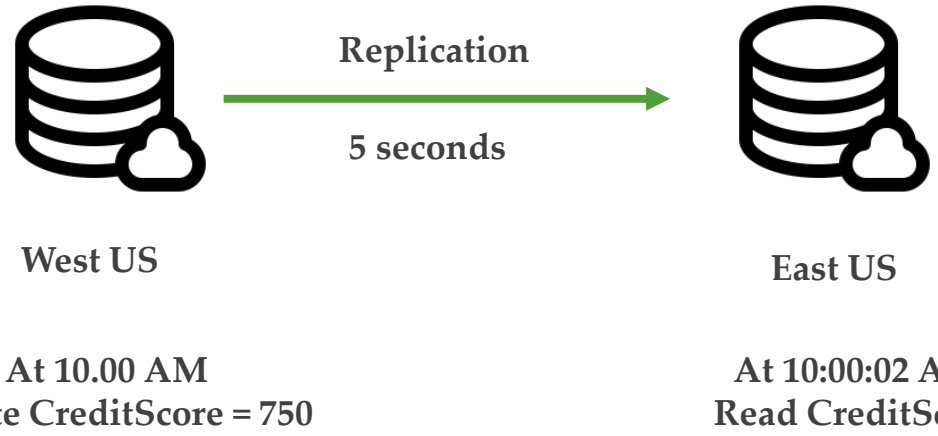
Performance

- Ensures high availability within a region
- Across regions, brings data closer to the consumer.

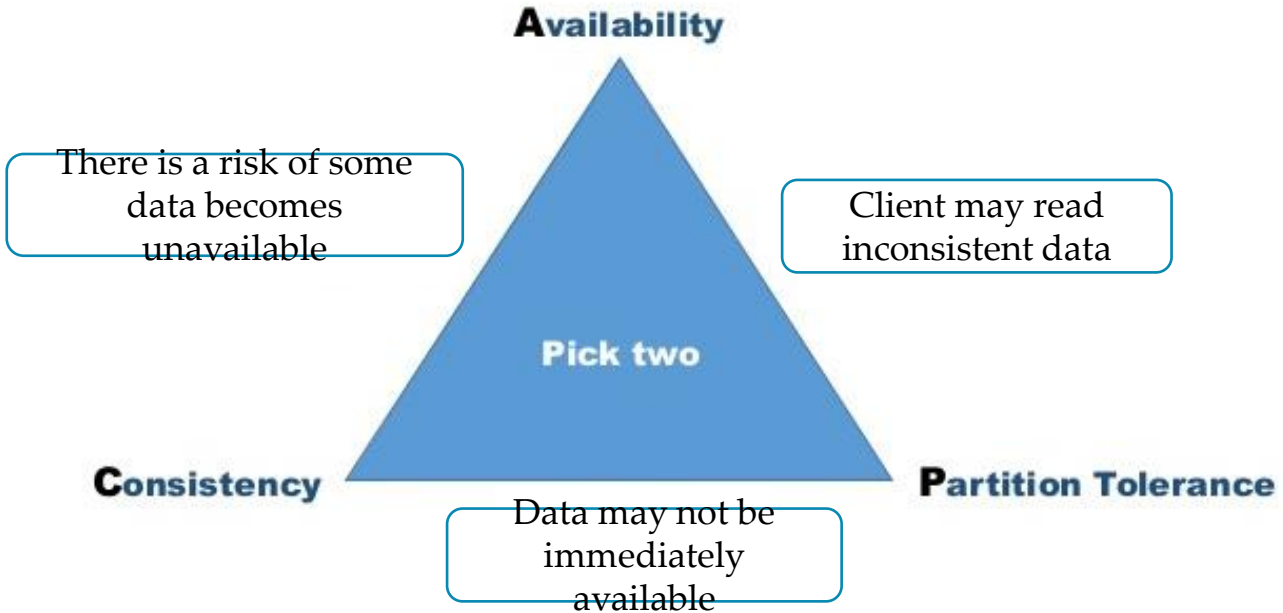
Business continuity

- In the event of major failure or natural disaster

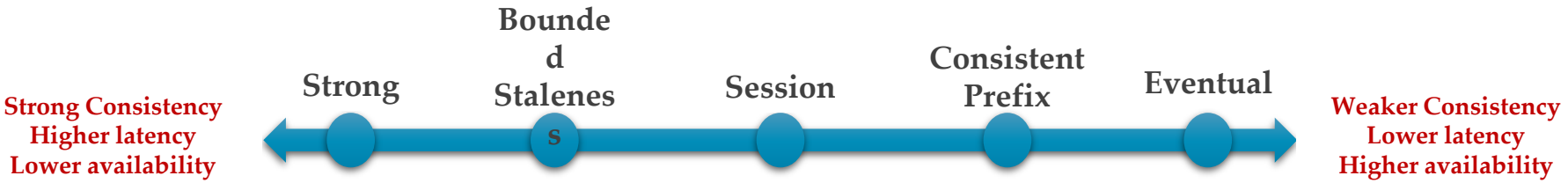
Data consistency



CAP Theorem


Availability
There is a risk of some data becomes unavailable
Client may read inconsistent data
Pick two
Consistency
Partition Tolerance
Data may not be immediately available

Five consistency Levels



Strong: No dirty reads, high latency, cost highest, closest to RDBMS

Bounded staleness: Dirty reads possible, bounded by time and updates

Session: No dirty reads for writers (within same session), dirty read possible for other users

Consistency prefix: Dirty reads possible but sequence maintain, reads never see out-of-order writes

Eventual: No guaranteed order, but eventually everything gets in order

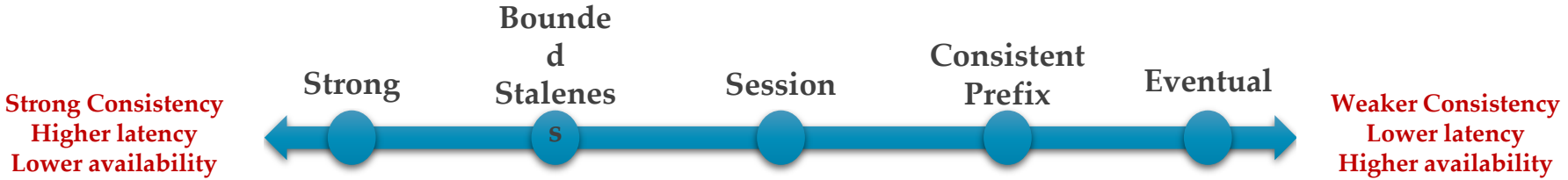
Setting the consistency level

Set default for entire account

Can be changed any time

Override at request level

Any request can weaken the default consistency level



Azure CLI

```
az cosmosdb create --name
                   --resource-group
                   [--capabilities]
                   [--default-consistency-level {BoundedStaleness, ConsistentPrefix, Eventual, Session, Strong}]
                   [--enable-automatic-failover {false, true}]
                   [--enable-multiple-write-locations {false, true}]
                   [--enable-virtual-network {false, true}]
                   [--ip-range-filter]
                   [--kind {GlobalDocumentDB, MongoDB, Parse}]
                   [--locations]
                   [--max-interval]
                   [--max-staleness-prefix]
                   [--subscription]
                   [--tags]
                   [--virtual-network-rules]
```

Azure CLI – Create Database example

```
reate a SQL API database and container

# Generate a unique 10 character alphanumeric string to ensure unique resource names
uniqueId=$(env LC_CTYPE=C tr -dc 'a-z0-9' < /dev/urandom | fold -w 10 | head -n 1)

# Variables for SQL API resources
resourceGroupName="Group-$uniqueId"
location='westus2'
accountName="cosmos-$uniqueId" #needs to be lower case
databaseName='database1'
containerName='container1'

# Create a resource group
az group create -n $resourceGroupName -l $location

# Create a Cosmos account for SQL API
az cosmosdb create \
    -n $accountName \
    -g $resourceGroupName \
    --default-consistency-level Eventual \
    --locations regionName='West US 2' failoverPriority=0 isZoneRedundant=False \
    --locations regionName='East US 2' failoverPriority=1 isZoneRedundant=False

# Create a SQL API database
az cosmosdb sql database create \
    -a $accountName \
    -g $resourceGroupName \
    -n $databaseName
```

Azure CLI – Example question

You are a data engineer for your company. You use the following Azure CLI commands to create an Azure Cosmos DB account. You plan to use this account to store sales data.

```
az cosmosdb create --resource-group 'sales-rg' --name 'sales' --kind GlobalDocumentDB \
--locations regionName="South Central US" failoverPriority=0 \
--locations regionName="North Central US" failoverPriority=1 \
--default-consistency-level "Strong" --enable-multiple-write-locations true
```

You need to answer questions regarding sales data queries and updates.

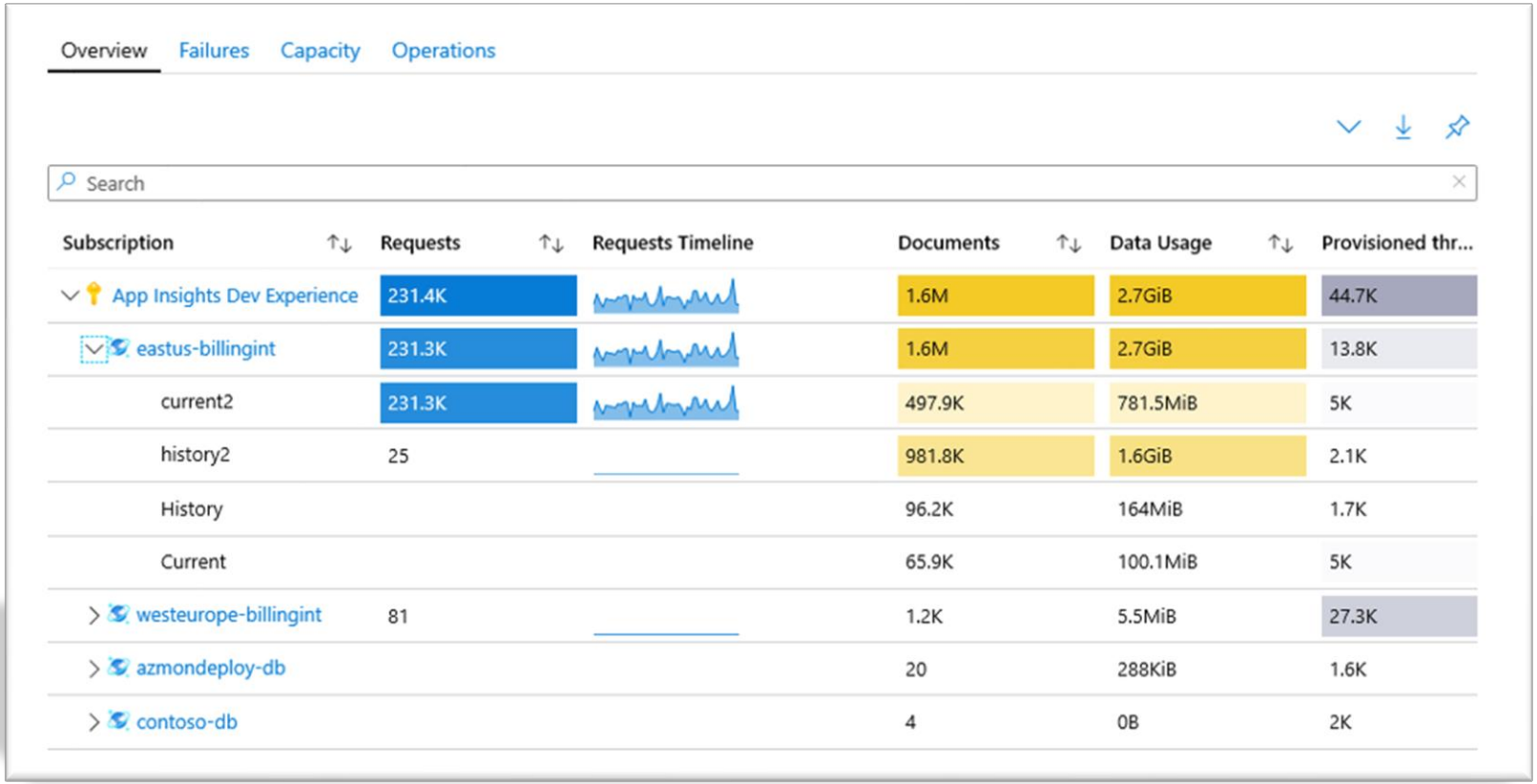
For each of the following statements, select Yes if the statement is true. Otherwise, select No.

	Statement
✗	You can query data by using Gremlin API.
✗	A client can see partial writes of a sales data record by default.
✓	A client can set the consistency level to Eventual Consistency at connection time.
✓	A client can set a different consistency level during each request to sales data.

Cosmos DB

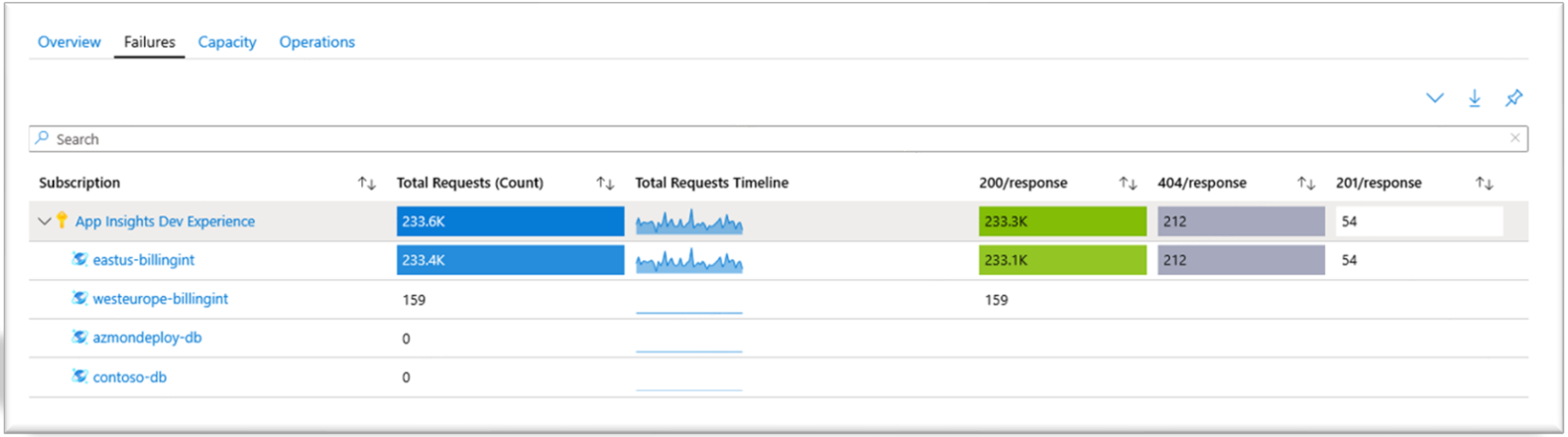
- **Cosmos DB was build as a Cloud Database – Highly Managed**
- **What we can monitor?**
  - Monitor Partitions and Partitioning key performance
  - Monitor RU's consumption
- **Azure Monitor Service**
  - Insights
  - Metrics
- **Monitoring from Cosmos DB Account**
  - Metrics
  - Alerts
  - Diagnostic Settings
  - Logs

Cosmos DB - Monitor through Insights

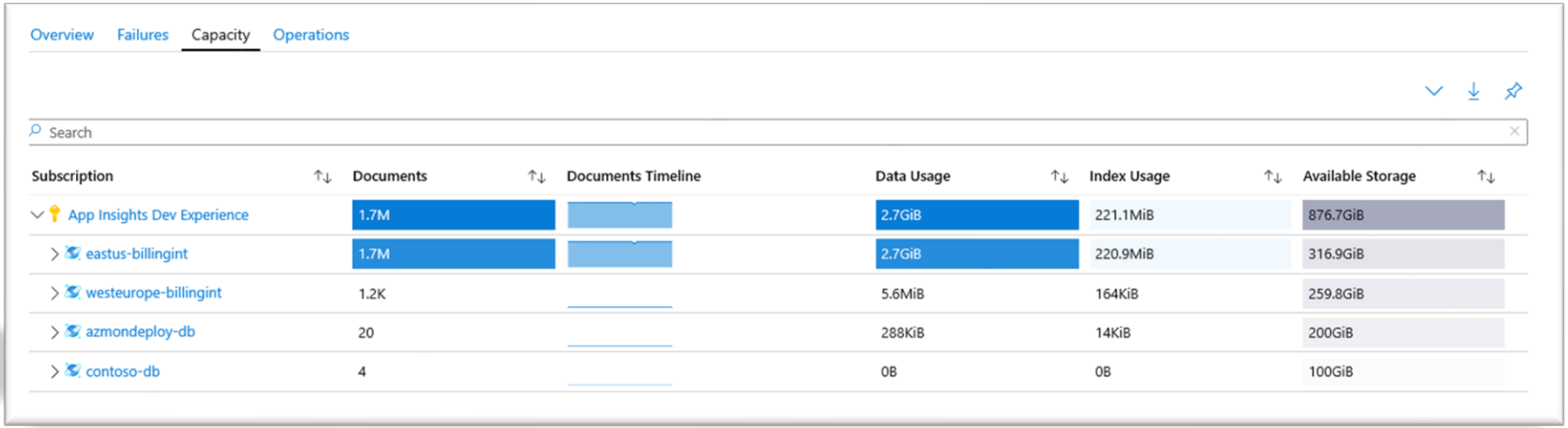


Subscription	Requests	Requests Timeline	Documents	Data Usage	Provisioned thr...
App Insights Dev Experience	231.4K		1.6M	2.7GiB	44.7K
eastus-billingint	231.3K		1.6M	2.7GiB	13.8K
current2	231.3K		497.9K	781.5MiB	5K
history2	25		981.8K	1.6GiB	2.1K
History			96.2K	164MiB	1.7K
Current			65.9K	100.1MiB	5K
westeurope-billingint	81		1.2K	5.5MiB	27.3K
azmondeploy-db			20	288KiB	1.6K
contoso-db			4	0B	2K

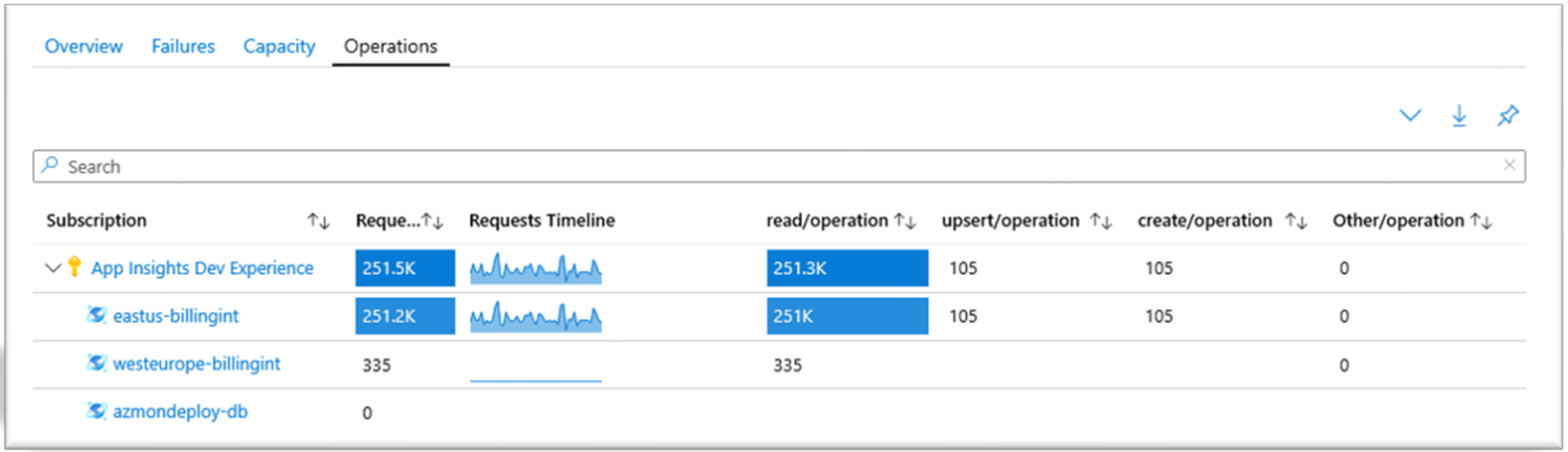
Cosmos DB - Monitor through Insights



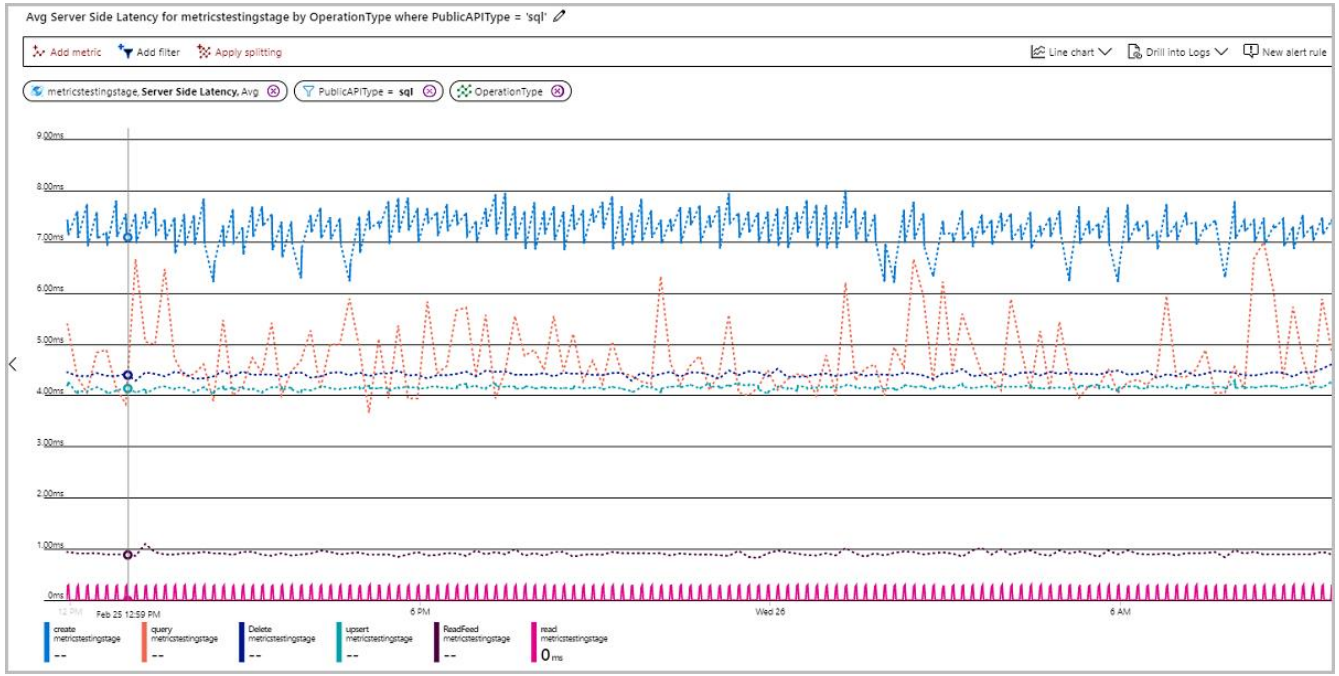
Cosmos DB - Monitor through Insights



Cosmos DB - Monitor through Insights



Cosmos DB - Monitor Service Metrics

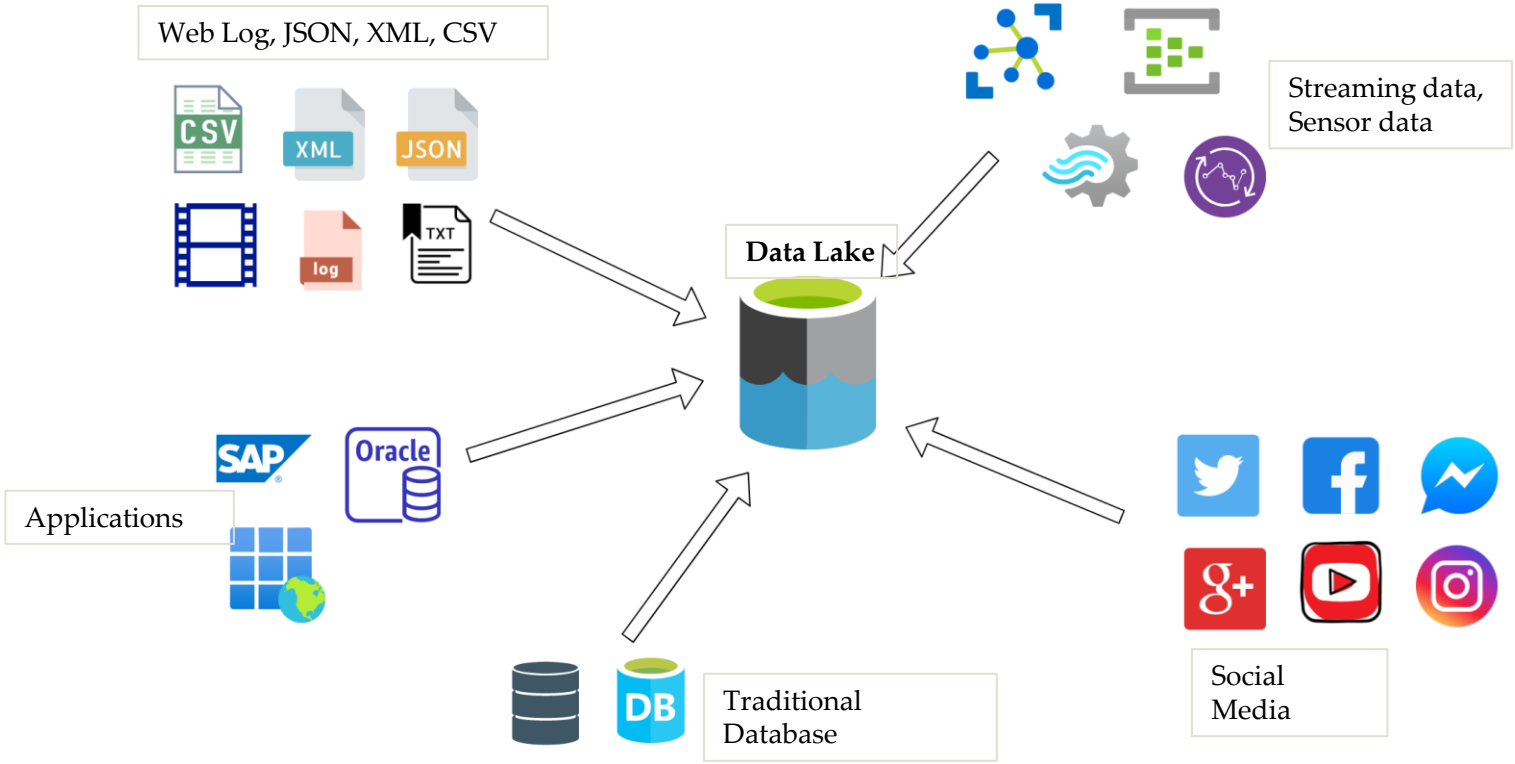


Cosmos DB

- Role based access control (RBAC)
- Network security
- Access Security Keys
- CORS (Cross-Origin Resource Sharing)
- Azure Private Endpoint
- Advance Security Option

Data Lake is a big container to store data.

Data Lake Sources

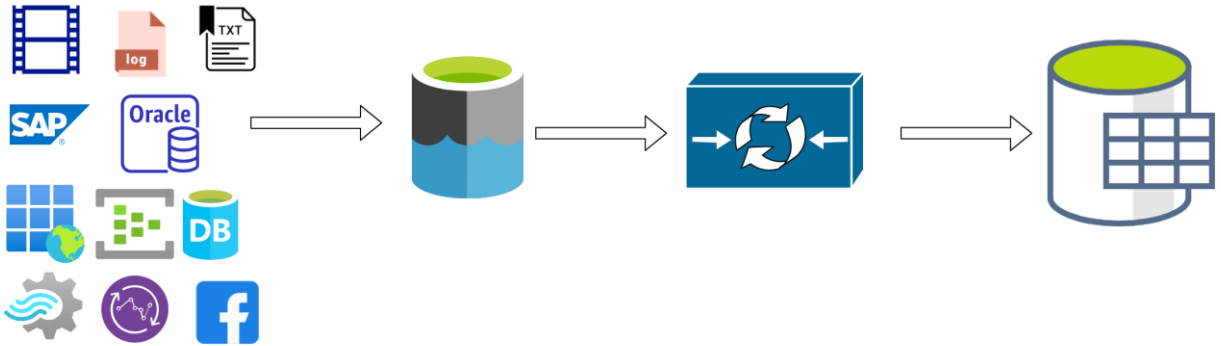


Data Sources

Data in raw format

Data transforms

Data ready for queries



What is Data Lake?

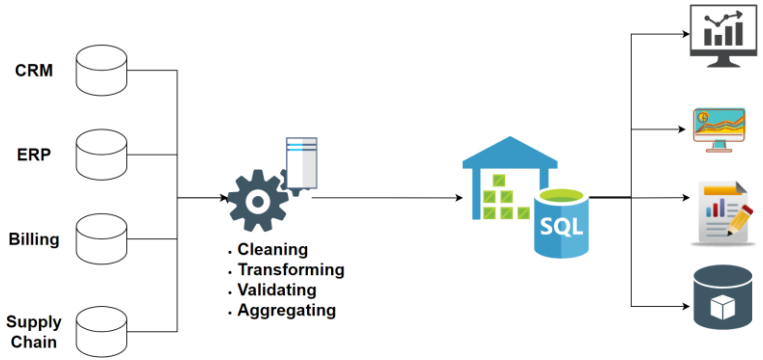
"If you think of a DataMart as a store of bottled water – clean and packaged and structured for easy consumption – the data lake is a large body of water in a more natural state. The contents of the data lake stream in from a source to fill the lake, and various users of the lake can come to examine,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

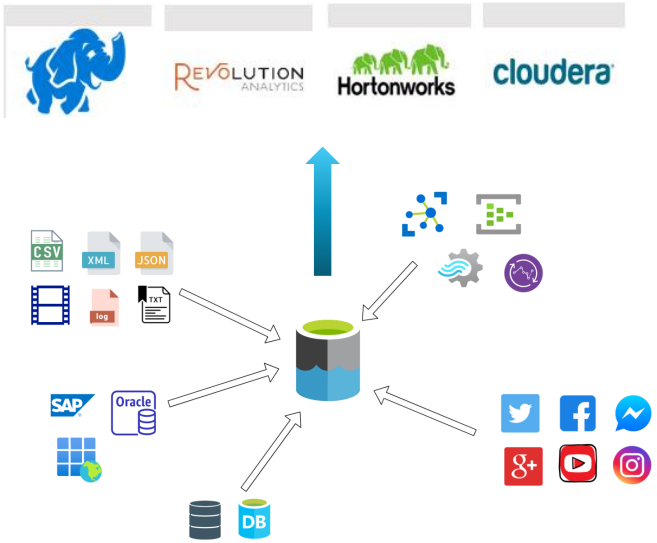
James Dixon
CTO, Pentaho

Data Warehouse

Data Lake

Data Warehouse vs Data Lake





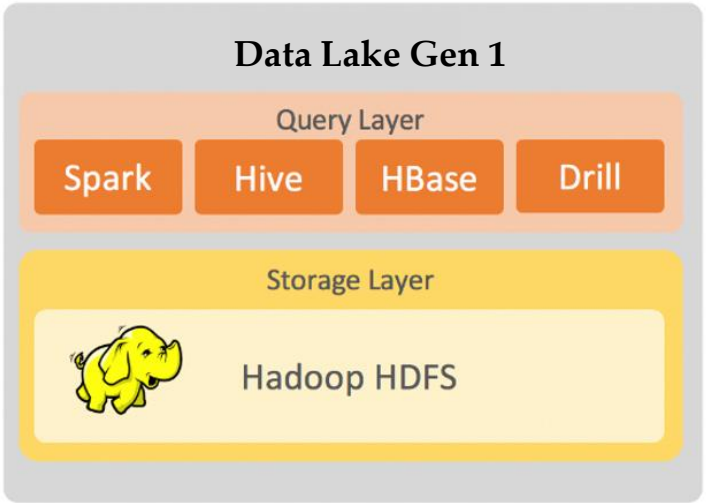
Data Warehouse

Data Lake

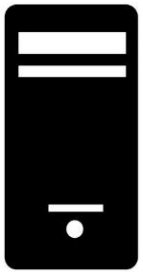
LearnCloud.Info

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

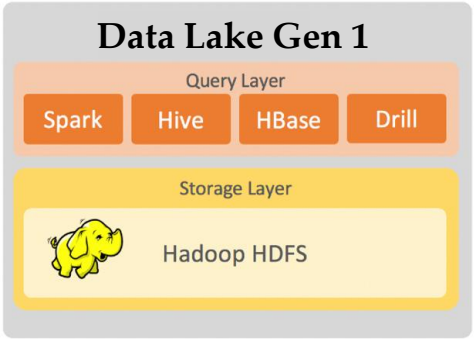
- Different requirements
- No direct solution

Azure Blob Storage

- 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

Azure Data lake Gen 2

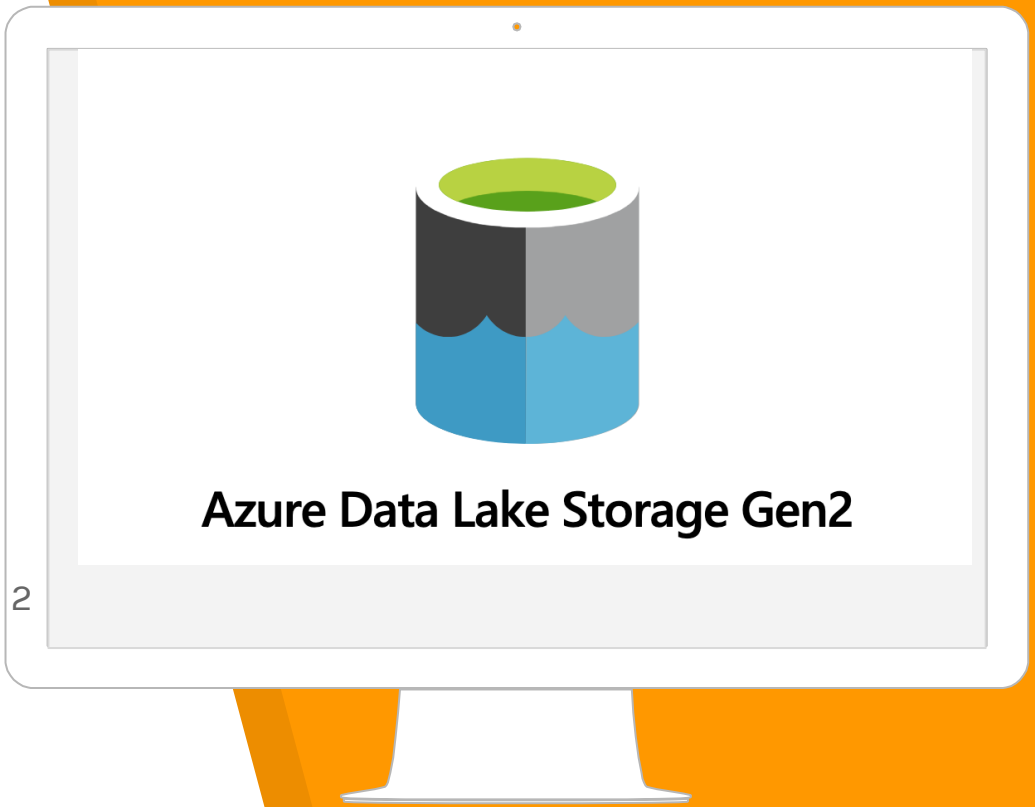
Blob Storage



Azure Data Lake Storage Gen2

MICRSOFT RECOMMENDS

Data Lake Storage Gen2 for your big data storage needs.



Note: USQL currently not supported in Gen 2

Blob Storage vs Data Lake Storage

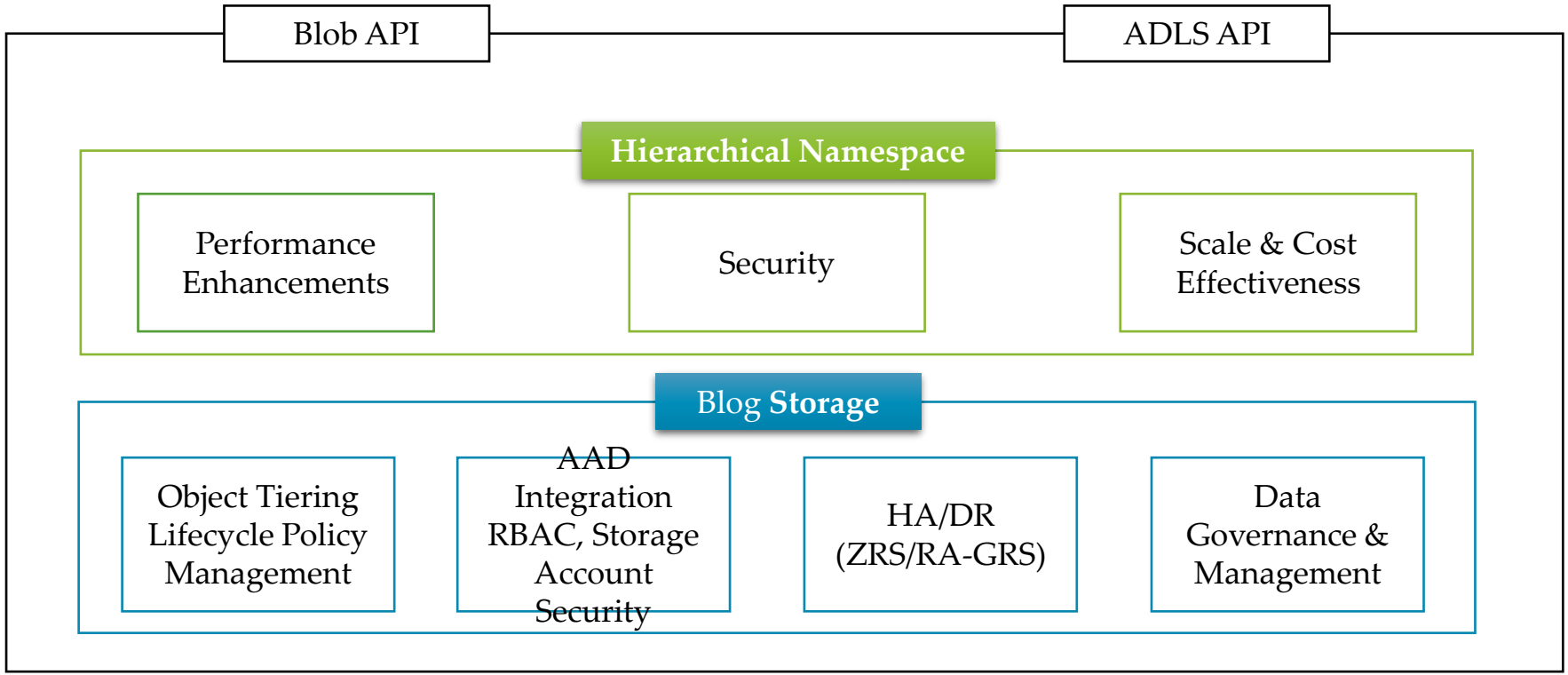
Azure Blob Storage

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

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
  - 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**

Storage Account Access Keys

Authentication

Shared Access Signature (SAS)

Authentication

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

Authentication

Azure Active Directory

Azure Active Directory

Azure Active Directory (AD)

- Grand 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

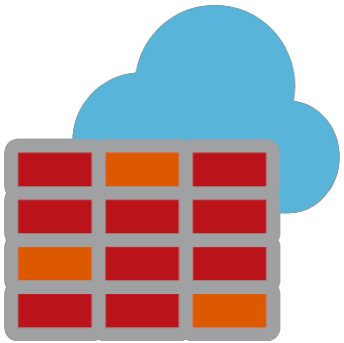
Role based access control (RBAC)

Access Control

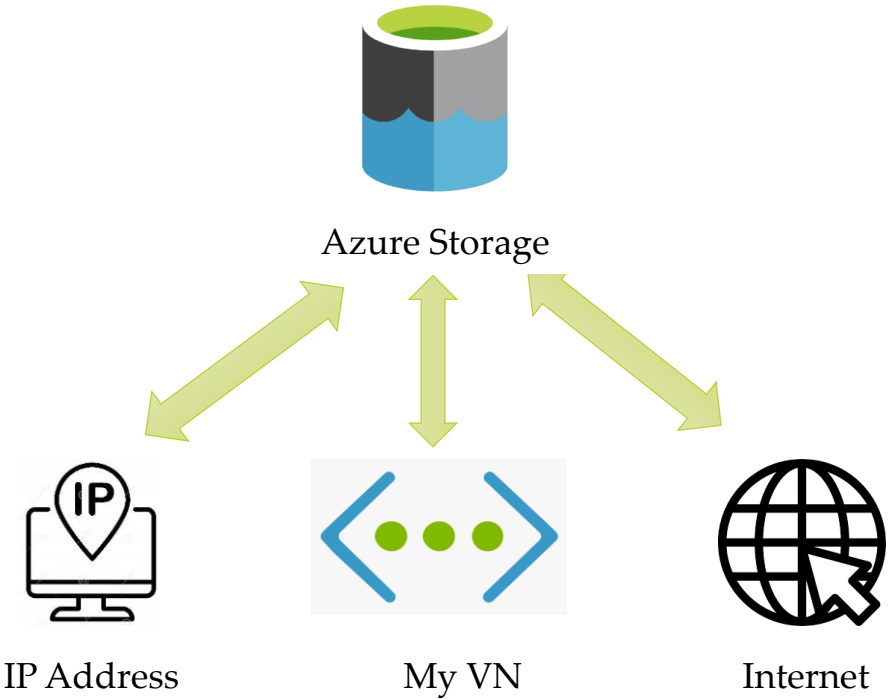
Role based access control (RBAC)

Access control

Firewalls and Virtual Networks



Firewalls and Virtual Networks



Data Continuity and Availability

High Availability

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

Disaster Recovery

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

What can go wrong?

Disaster Recovery

Hardware

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

Software

Application, OS, bugs, logical corruptions

Connectivity

Redundant network connectivity

What can go wrong?

Disaster Recovery

Entire site

Lose power, lose water, bad weather

Entire region

Flooding, earthquake

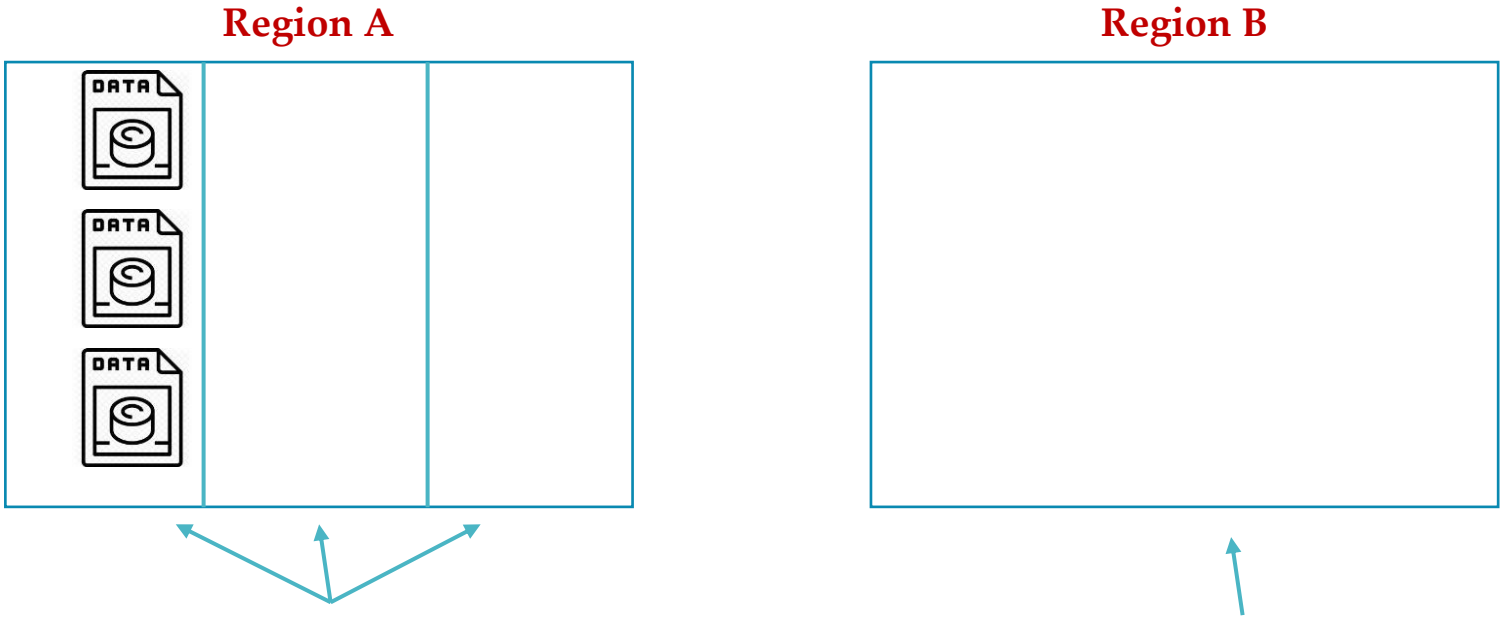
Human error

People make mistakes

Azure Storage

High Availability and Disaster recovery options

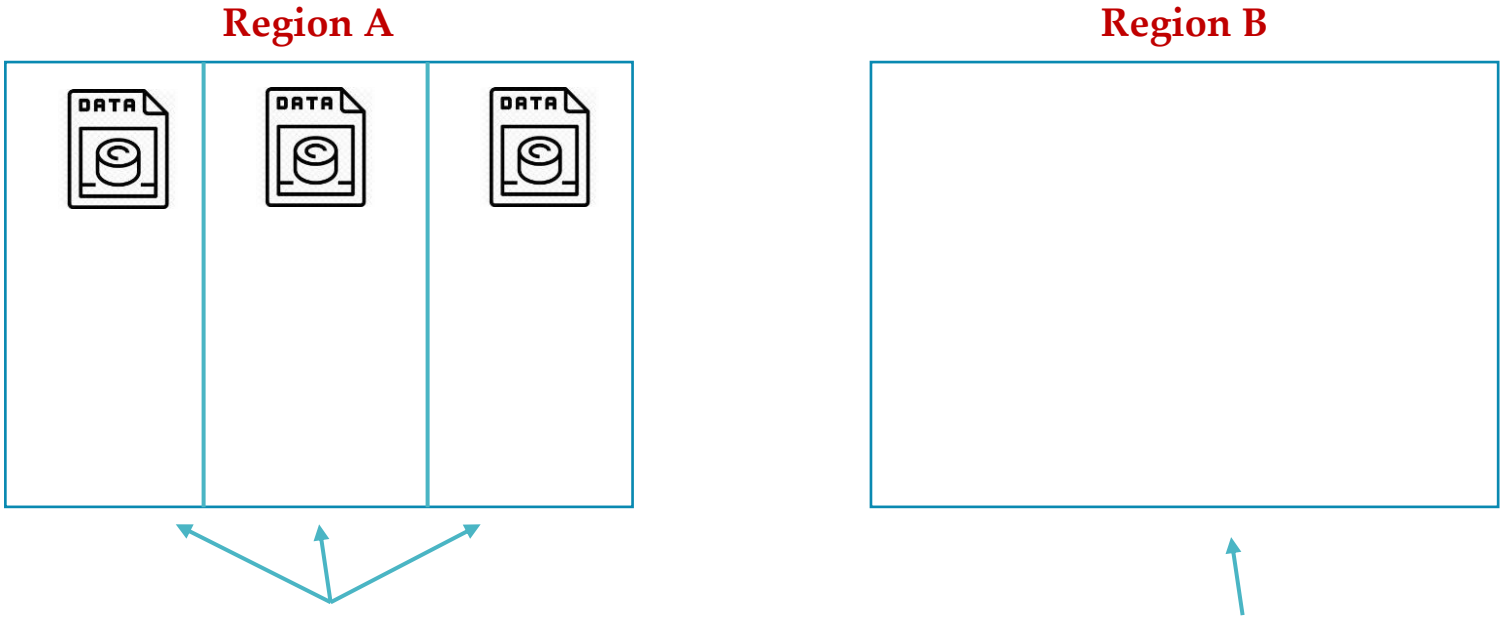
Locally Redundant Storage (LRS)



Storage Clusters
Each cluster is physically separate in what's called an availability zone, with its own separate utilities and networking.

Region B
Hundreds of miles away from the primary region to prevent data loss in the event of a natural disaster.

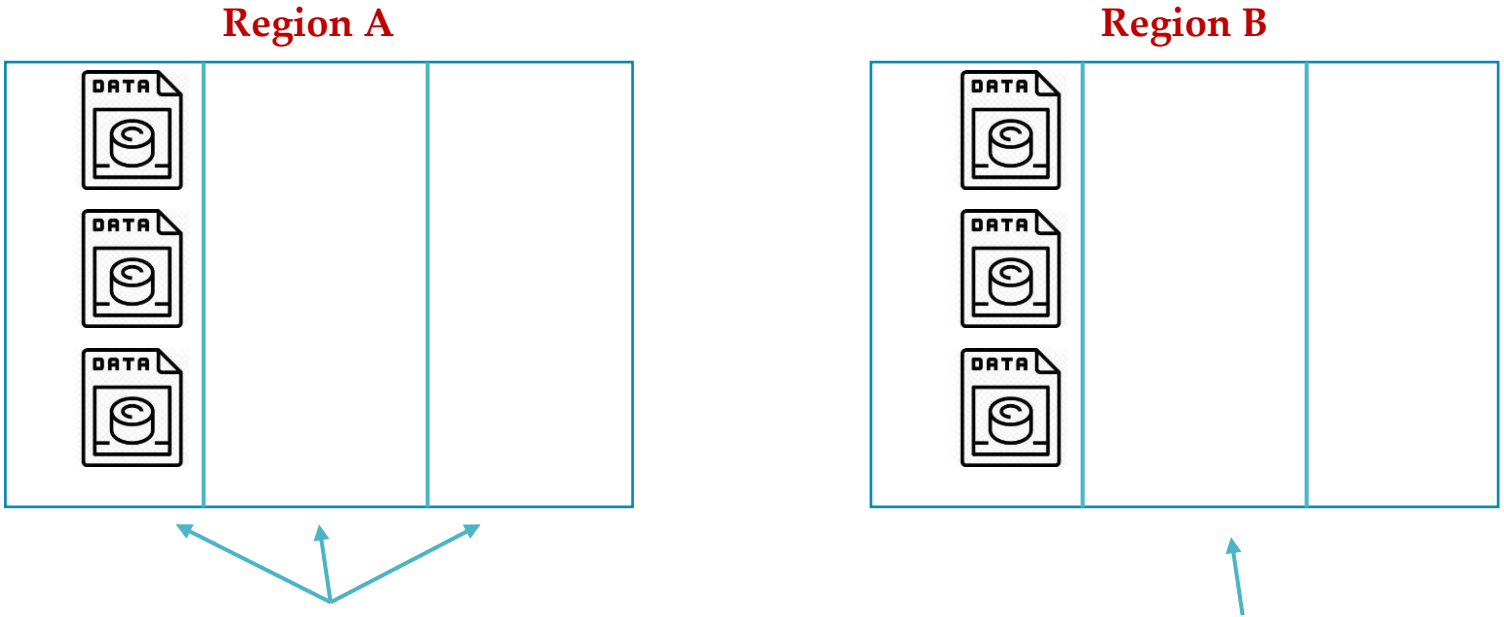
Zone Redundant Storage (ZRS)



Storage Clusters
Each cluster is physically separate in what's called an availability zone, with its own separate utilities and networking.

Region B
Hundreds of miles away from the primary region to prevent data loss in the event of a natural disaster.

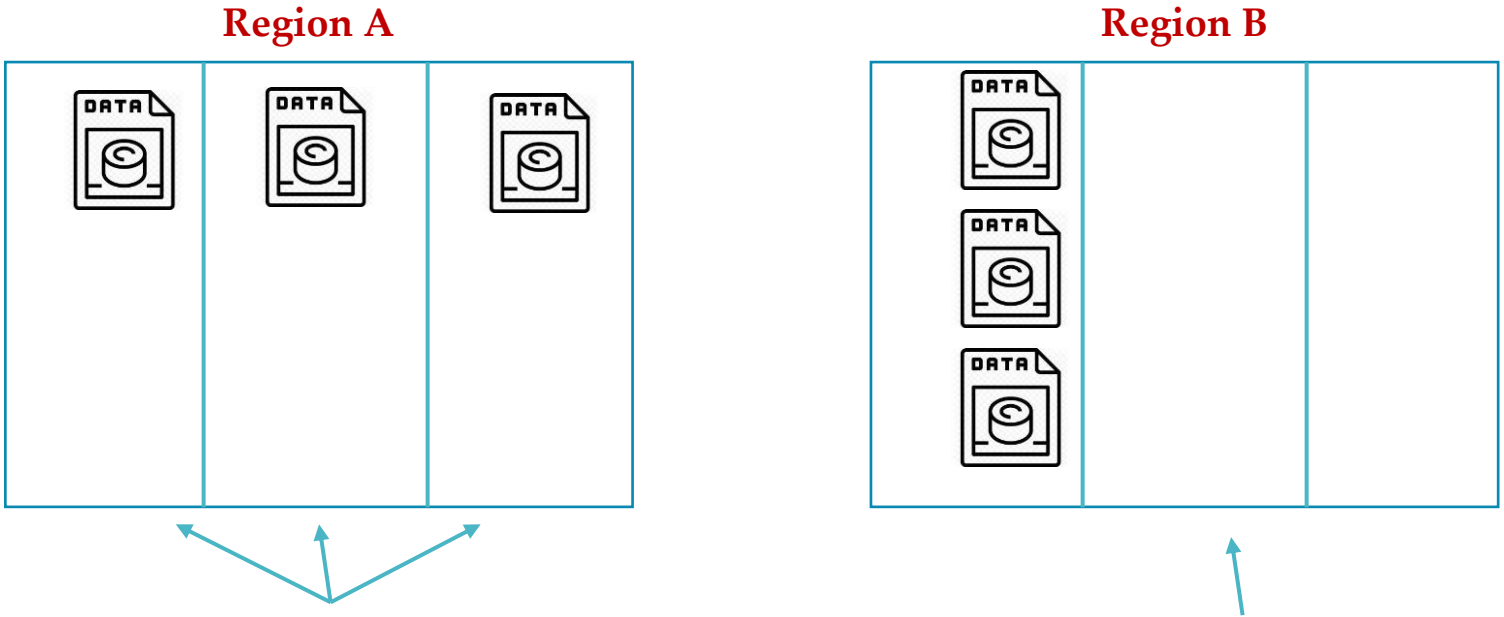
Geo Redundant Storage (GRS)



Storage Clusters
Each cluster is physically separate in what's called an availability zone, with its own separate utilities and networking.

Region B
Hundreds of miles away from the primary region to prevent data loss in the event of a natural disaster.

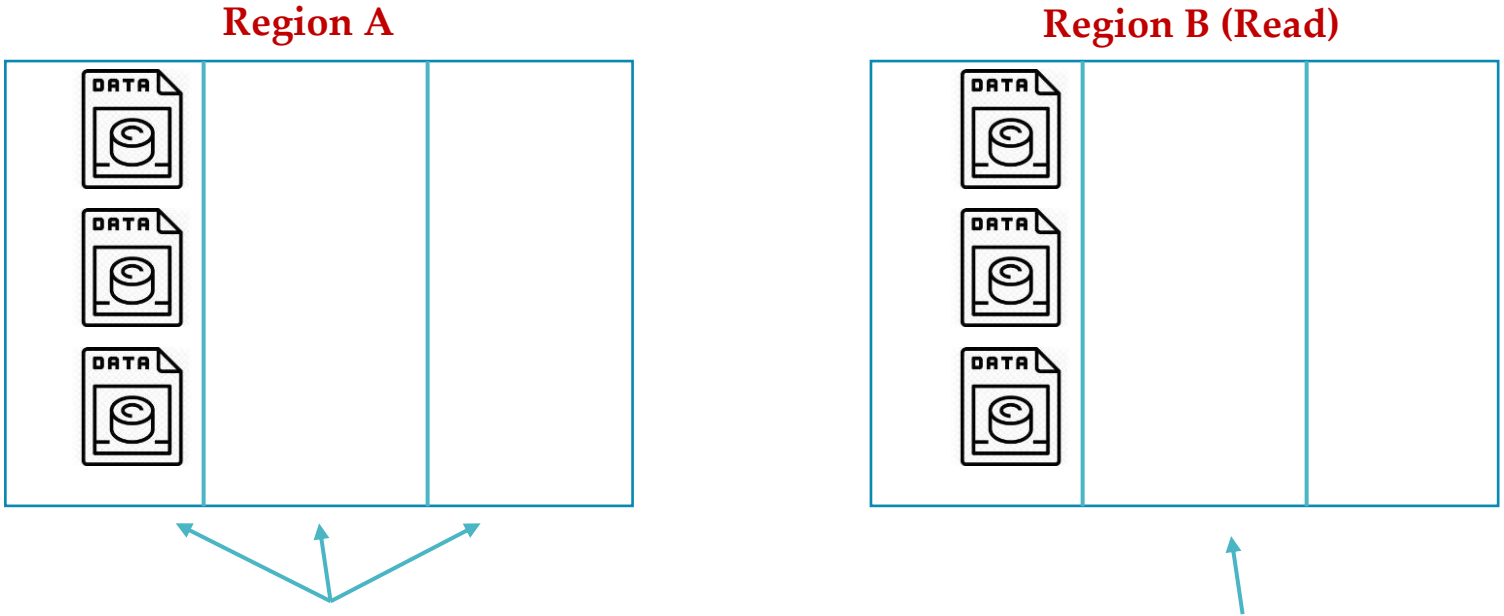
Geo Zone Redundant Storage (GZRS)



Storage Clusters
Each cluster is physically separate in what's called an availability zone, with its own separate utilities and networking.

Region B
Hundreds of miles away from the primary region to prevent data loss in the event of a natural disaster.

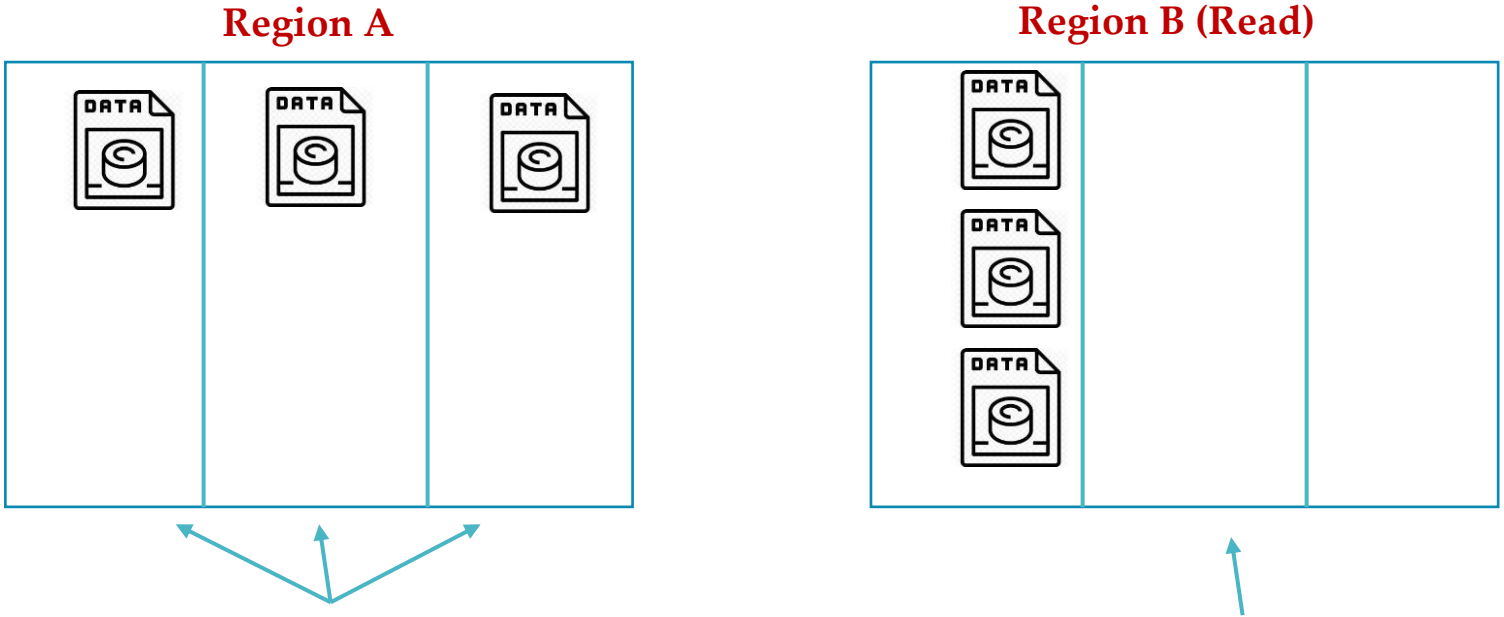
Read access geo Redundant Storage (RA-GRS)



Storage Clusters
Each cluster is physically separate in what's called an availability zone, with its own separate utilities and networking.

Region B
Hundreds of miles away from the primary region to prevent data loss in the event of a natural disaster.

Read access Geo Zone Redundant Storage (RA-GZRS)



Storage Clusters

Each cluster is physically separate in what's called an availability zone, with its own separate utilities and networking.

Region B

Hundreds of miles away from the primary region to prevent data loss in the event of a natural disaster.

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[1]	Yes
A region-wide outage occurs in the primary region	No	No	Yes[1]	Yes[1]
Read access to the secondary region is available if the primary region becomes unavailable	No	No	Yes (with RA-GRS)	Yes (with RA-GZRS)

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

Cosmos DB – HA and DR Options

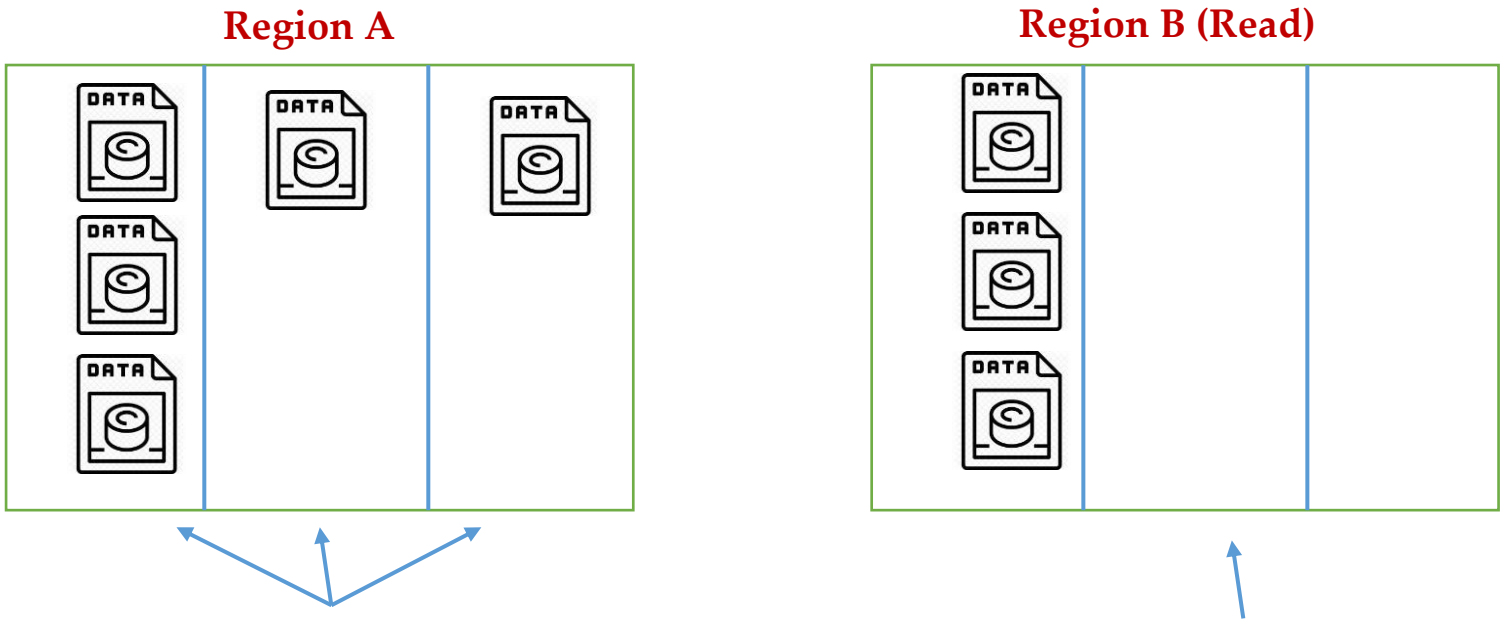
Agenda

- Cosmos DB – High availability and Disaster recovery option
  - Local/Zone/Global – Replication
  - Backup and Restore of Cosmos DB

Prerequisite

- Global distribution
- Multi-master
- Manual vs Automatic Failover

Cosmos DB – HA and DR Options



Storage Clusters

Each cluster is physically separate in what's called an availability zone, with its own separate utilities and networking.

Region B

Hundreds of miles away from the primary region to prevent data loss in the event of a natural disaster.

Cosmos DB – HA and DR Options

Single Region

- Data within a container is durably committed by a majority of replica members within the replica set
- Availability Zone support - replicas are placed across multiple zones within a given region

Multi Region – Multi write

- Regional failovers are instantaneous and don't require any changes from the application.

Multi Region – Single write (read region outage)

- No changes are required in your application code
- The impacted region is automatically disconnected and will be marked offline.
- The Azure Cosmos DB SDKs will redirect read calls to the next available region
- When the impacted read region is back online it will automatically sync with the current write region and will be available again to serve read requests.

Multi Region – Single write (write region outage)

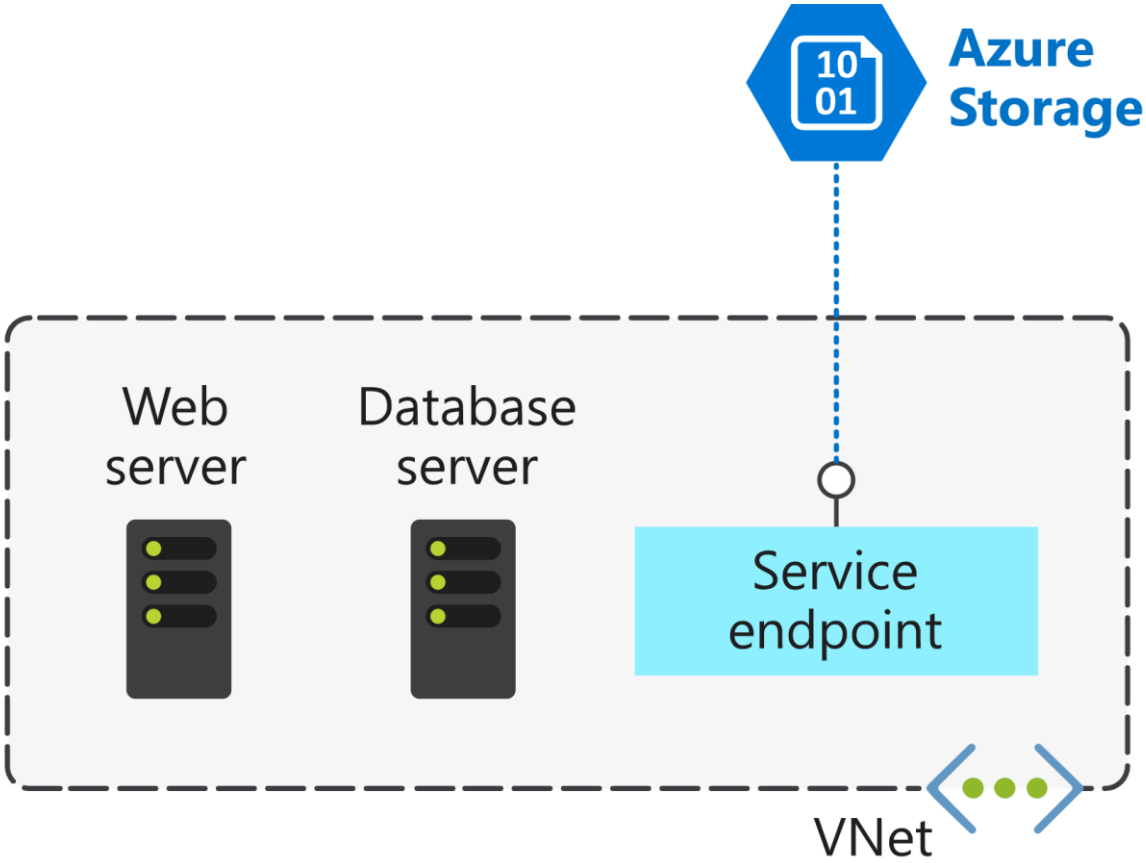
- Manual failover
- Automatic Failover enable – automatically promote a secondary region (no action required)
- When impacted region back only – Lost data can be recovered using conflict feed
- you can switch back to the recovered region as the write region using PowerShell, Azure CLI or Azure portal.

Cosmos DB Backup and Restore

- Accidentally deleted or update data? – use backups
- Backups are completely automated
- No performance impacted, No additional RUs or cost
- Default:
  - Interval: every 4 hours
  - Retention: 8 hours
  - Interval and Retention can be changed but max 2 backups are possible
- Stored separately in a blob storage service
- Backups stored in same region, and also replicated to paired region
- How to Restore? - Raise a ticket to support team

  Custom data backup – solutions can be implemented via Azure Data Factory or via the Cosmos DB change feed
- Accidently deleted your data? – You have 8 hours

Virtual network service endpoints



Service endpoints