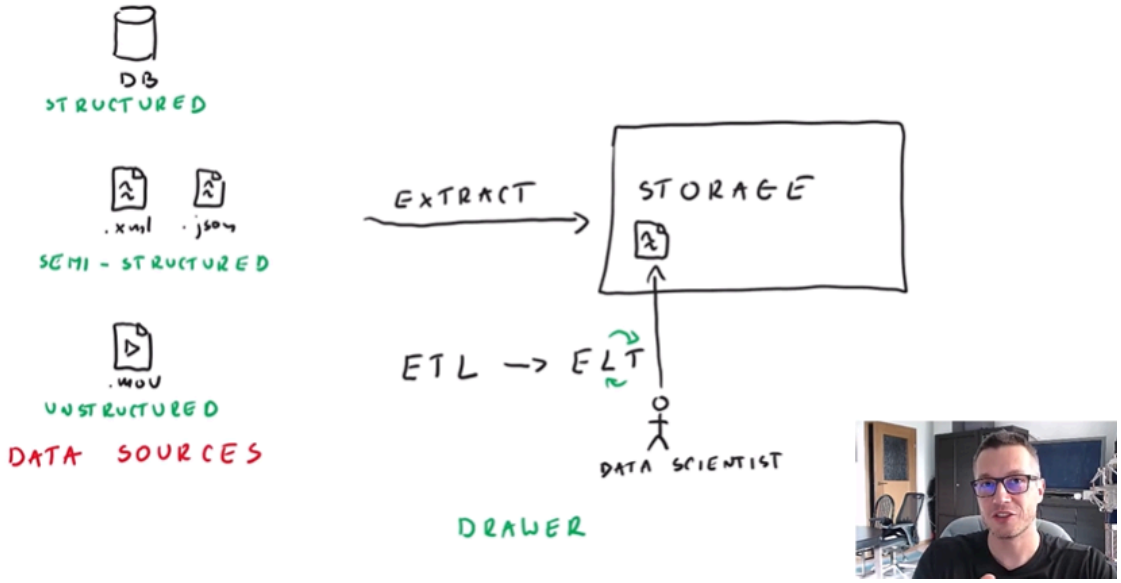
Storage Account Overview

DataLake Revisiting

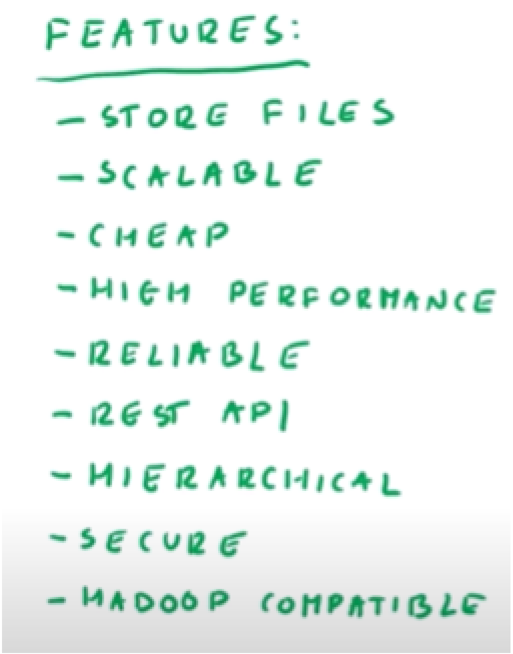


1.
2. Here in the datalake the data will be coming different sources
3. Using data lake we can implement ELT …
4. Here raw data will stored aside and if data team wants to use this raw data…they can come and retrieve this data

DataLake Features

1. We should be able to store files

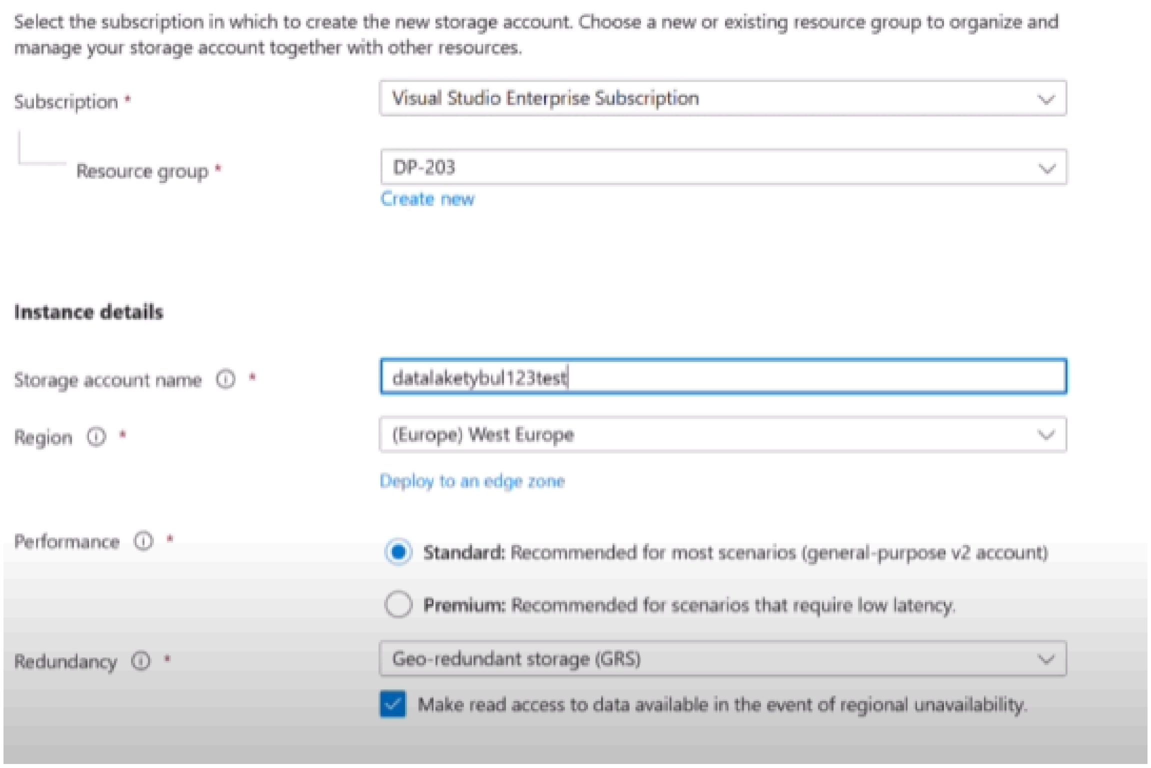
2. It must be scalable to handle large volume of data



FEATURES:

- STORE FILES
- SCALABLE
- CHEAP
- HIGH PERFORMANCE
- RELIABLE
- REST API
- HIERARCHICAL
- SECURE
- HADOOP COMPATIBLE

3. So inside the Azure ..we get this by storage account..
4. We create a storage account with unique name…because we’ll be accessing this using the URL



Select the subscription in which to create the new storage account. Choose a new or existing resource group to organize and manage your storage account together with other resources.

Subscription *	Visual Studio Enterprise Subscription
Resource group *	DP-203 Create new

Instance details

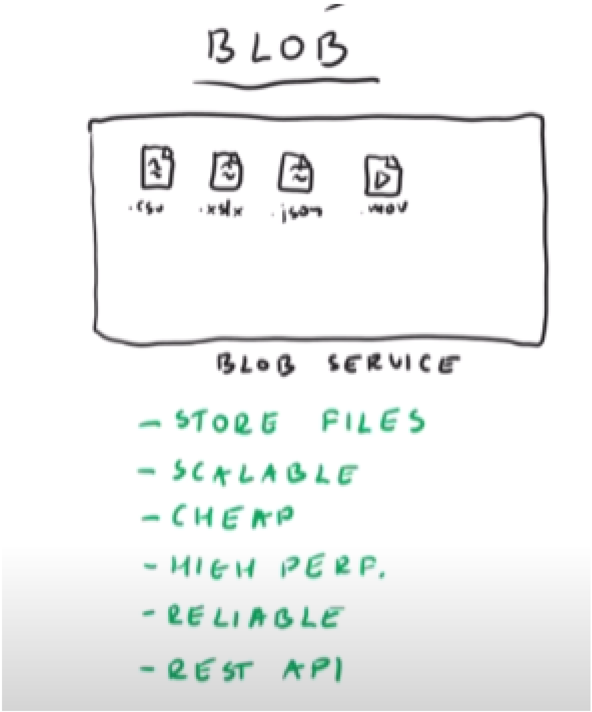
Storage account name ⓘ *	datalaketybul123test
Region ⓘ *	(Europe) West Europe Deploy to an edge zone
Performance ⓘ *	● **Standard:** Recommended for most scenarios (general-purpose v2 account) ○ **Premium:** Recommended for scenarios that require low latency.
Redundancy ⓘ *	Geo-redundant storage (GRS) ☑ Make read access to data available in the event of regional unavailability.

and we’ll go by default setting and create this

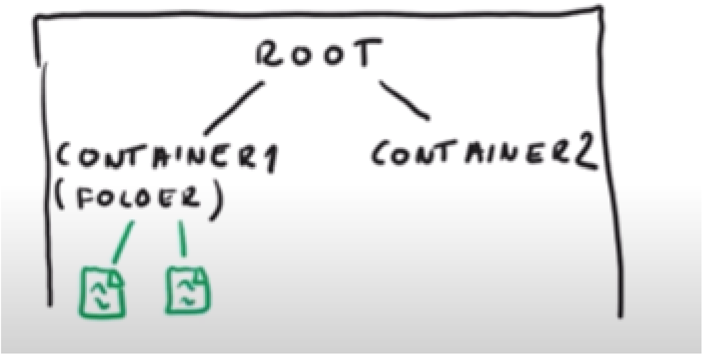
5. Inside storage account we have 4 services Blob Service, file service, Queue Service, Table Service

Blob Service

1. Blob stands for binary large object and It is used to store files(csv,audio, video files)

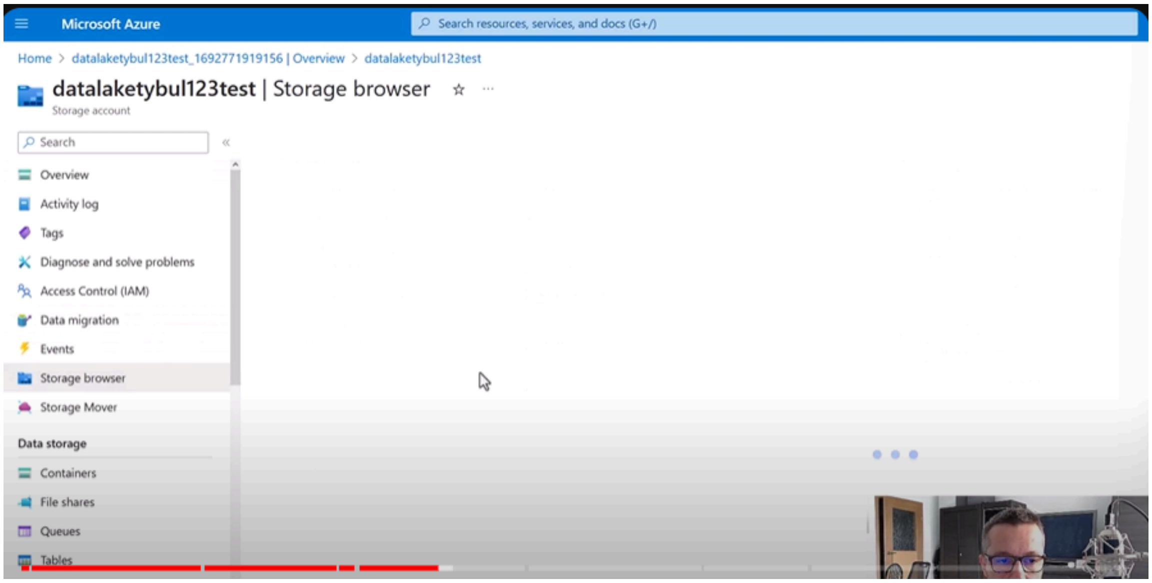


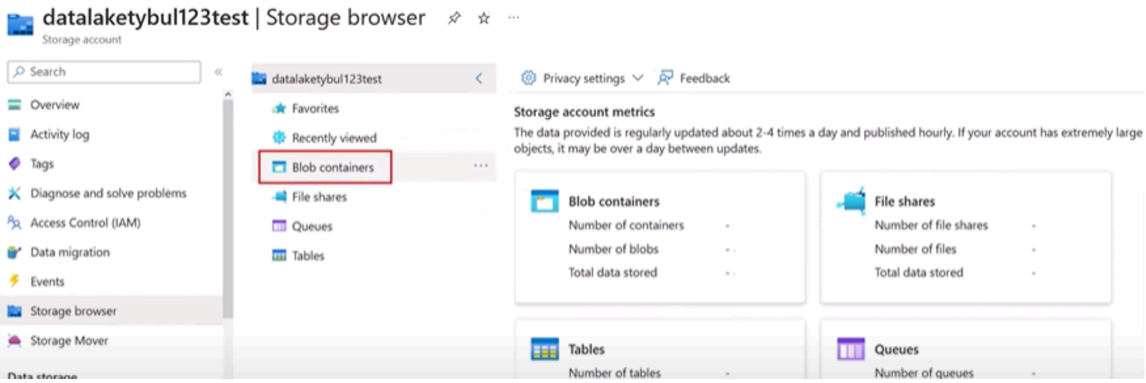
2. Now inside the blob we'll have 2 directories(containers) …and within our container we store some files



3. Lets see this practically

4. Now inside the blob storage we go to storage browser-> Blob containers


Microsoft Azure
Search resources, services, and docs (G+/)
Home > datalaketybul123test_1692771919156 | Overview > datalaketybul123test
datalaketybul123test | Storage browser
Storage account
Search
Overview
Activity log
Tags
Diagnose and solve problems
Access Control (IAM)
Data migration
Events
Storage browser
Storage Mover
Data storage
Containers
File shares
Queues
Tables



Storage account metrics

The data provided is regularly updated about 2-4 times a day and published hourly. If your account has extremely large objects, it may be over a day between updates.

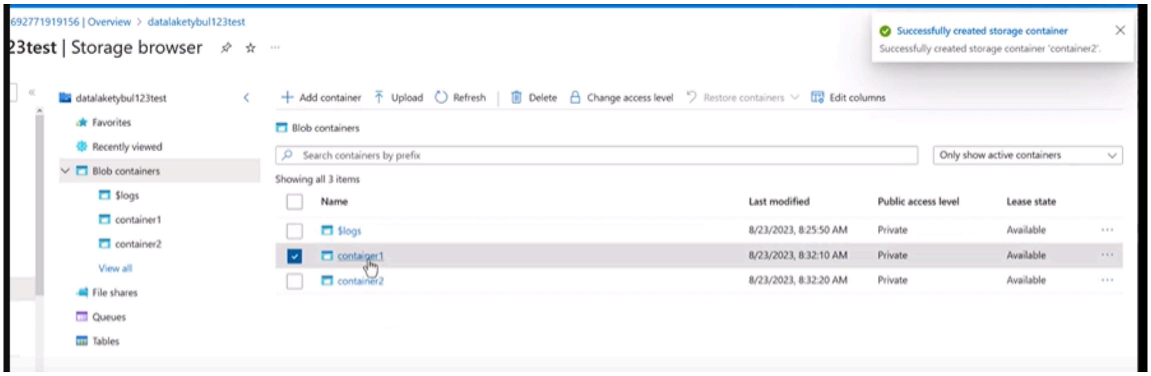
Blob containers	
Number of containers	-
Number of blobs	-
Total data stored	-

File shares	
Number of file shares	-
Number of files	-
Total data stored	-

Tables	
Number of tables	-

Queues	
Number of queues	-

5. Inside the blob container..we'll create two containers



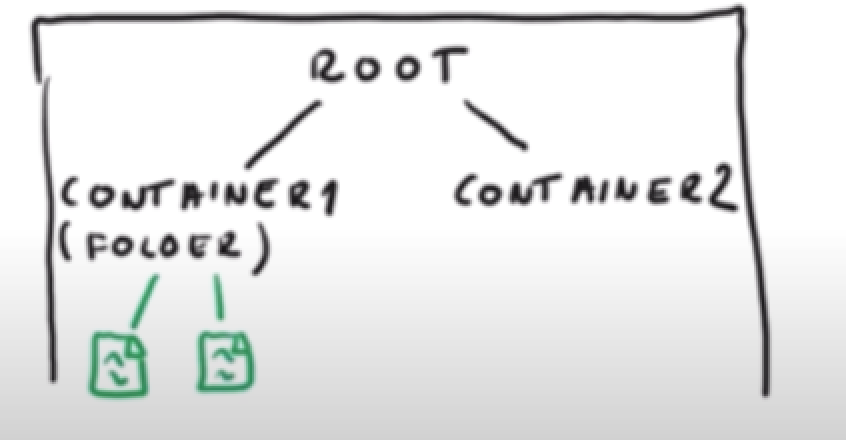
Successfully created storage container

Successfully created storage container 'container2'.

Showing all 3 items

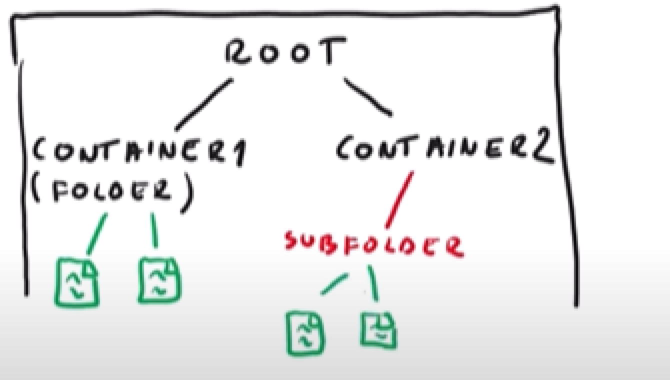
Name	Last modified	Public access level	Lease state
$logs	8/23/2023, 8:25:50 AM	Private	Available
container1	8/23/2023, 8:32:10 AM	Private	Available
container2	8/23/2023, 8:32:20 AM	Private	Available

6. And we will upload some sample files inside the container 1

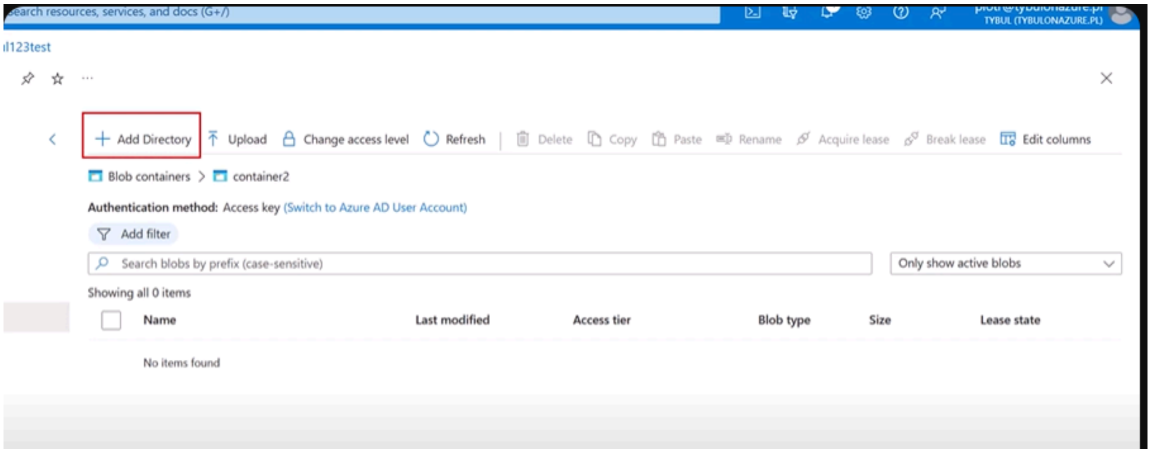


we have implemented this in container 1

7. We can also create subfolders/directories inside the container



8. Lets implement this practically…we'll go to container 2



Authentication method: Access key (Switch to Azure AD User Account)

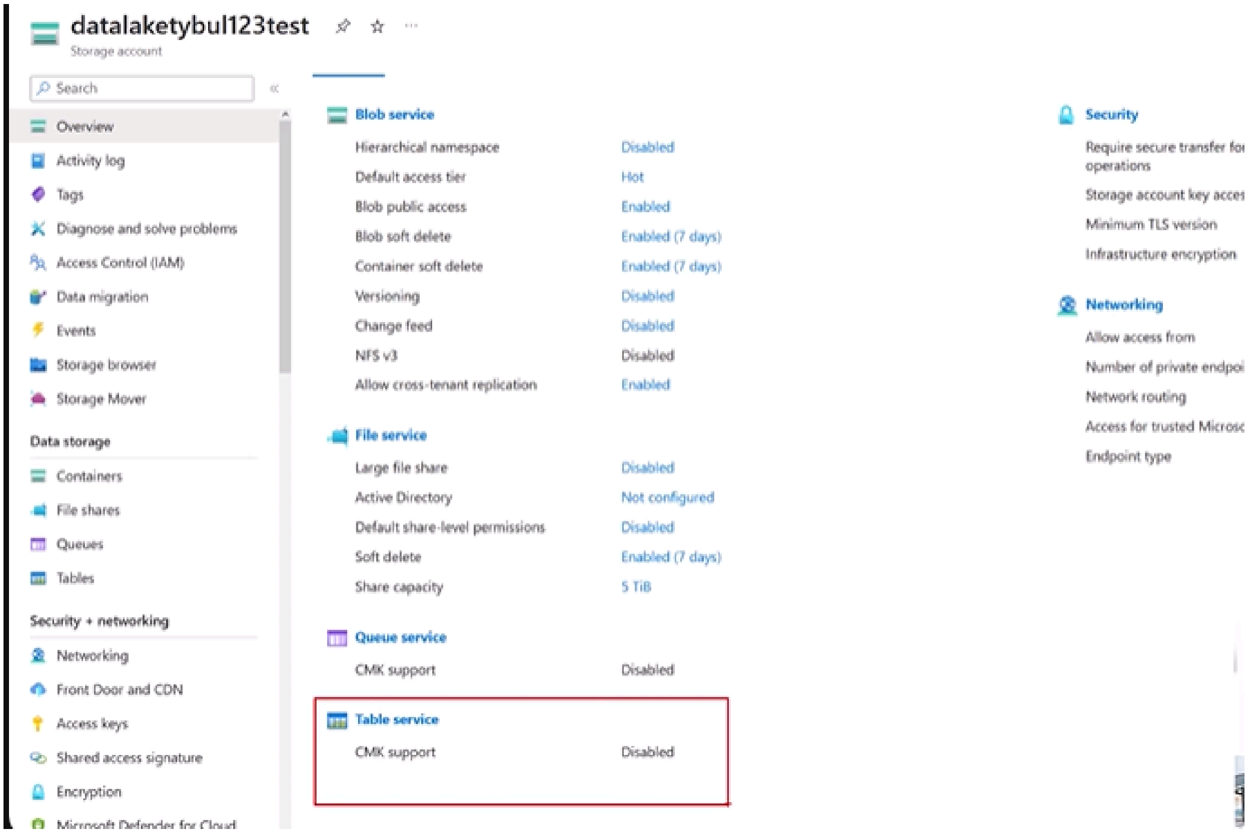
Showing all 0 items

□ Name	Last modified	Access tier	Blob type	Size	Lease state
No items found					

and click on add directory

9. Here if we create a folder(directory)…then in the UI it shows that..it actually created ..but it does not
10. So Blob Storage stores the files in flat storage…so this is the main drawback of blob storage

Storage Account - Table Service



Blob service

Hierarchical namespace	Disabled
Default access tier	Hot
Blob public access	Enabled
Blob soft delete	Enabled (7 days)
Container soft delete	Enabled (7 days)
Versioning	Disabled
Change feed	Disabled
NFS v3	Disabled
Allow cross-tenant replication	Enabled

File service

Large file share	Disabled
Active Directory	Not configured
Default share-level permissions	Disabled
Soft delete	Enabled (7 days)
Share capacity	5 TiB

Queue service

CMK support	Disabled

Table service

CMK support	Disabled

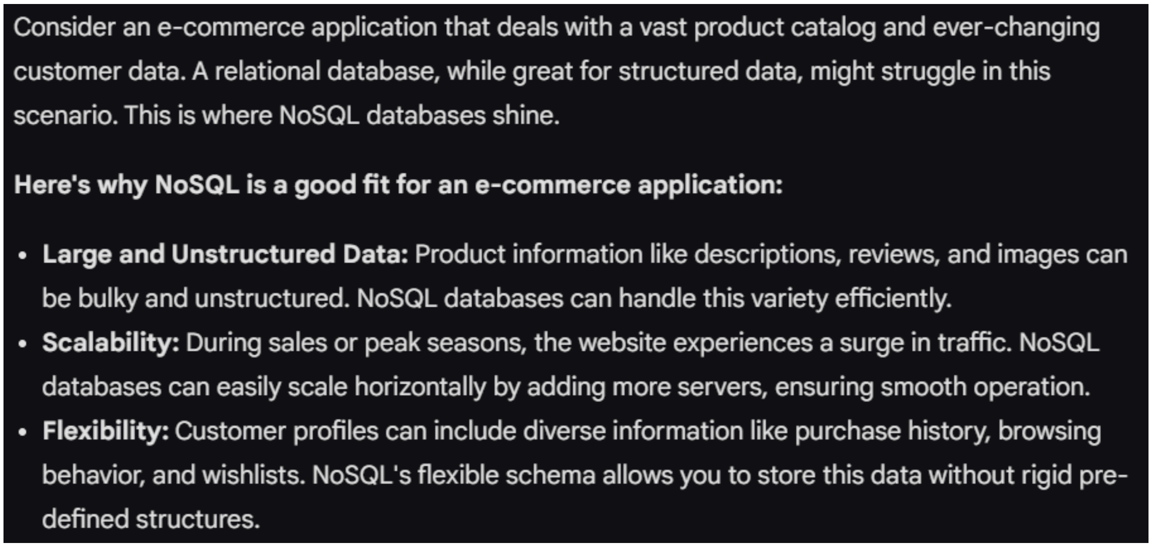
Security

Require secure transfer fo operations
Storage account key acces
Minimum TLS version
Infrastructure encryption

Networking

Allow access from
Number of private endpoi
Network routing
Access for trusted Microsc
Endpoint type

1. Its main usecase if for No-SQL usecase..if data does not have schema
2. So when does No-SQL comes to use?

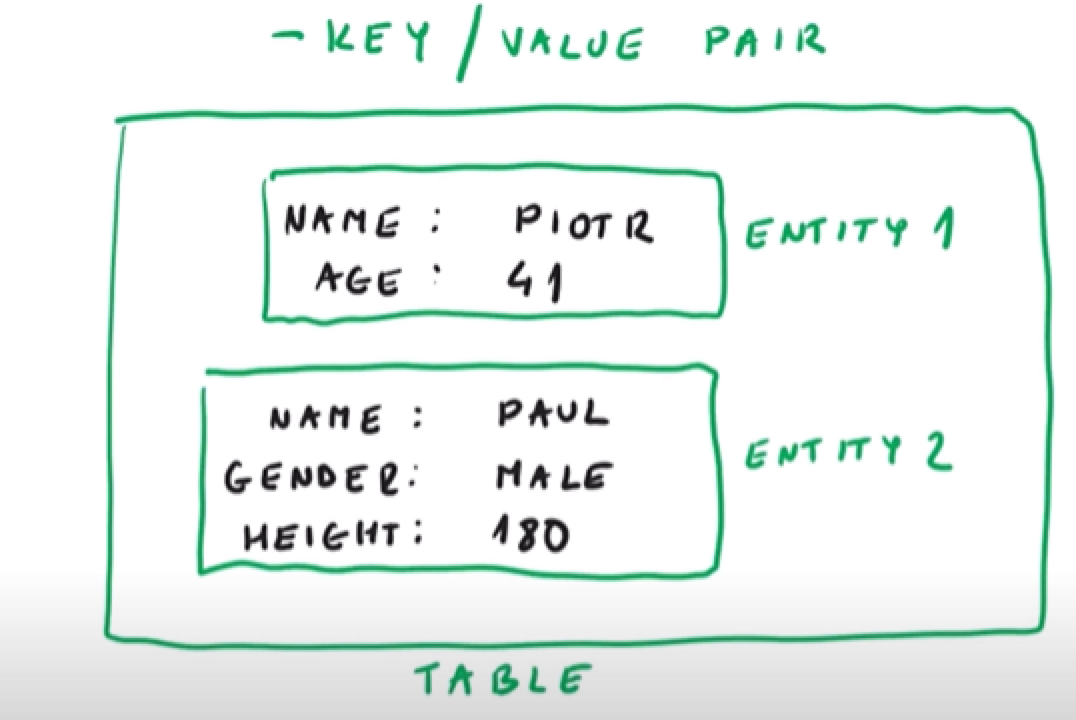


Consider an e-commerce application that deals with a vast product catalog and ever-changing customer data. A relational database, while great for structured data, might struggle in this scenario. This is where NoSQL databases shine.

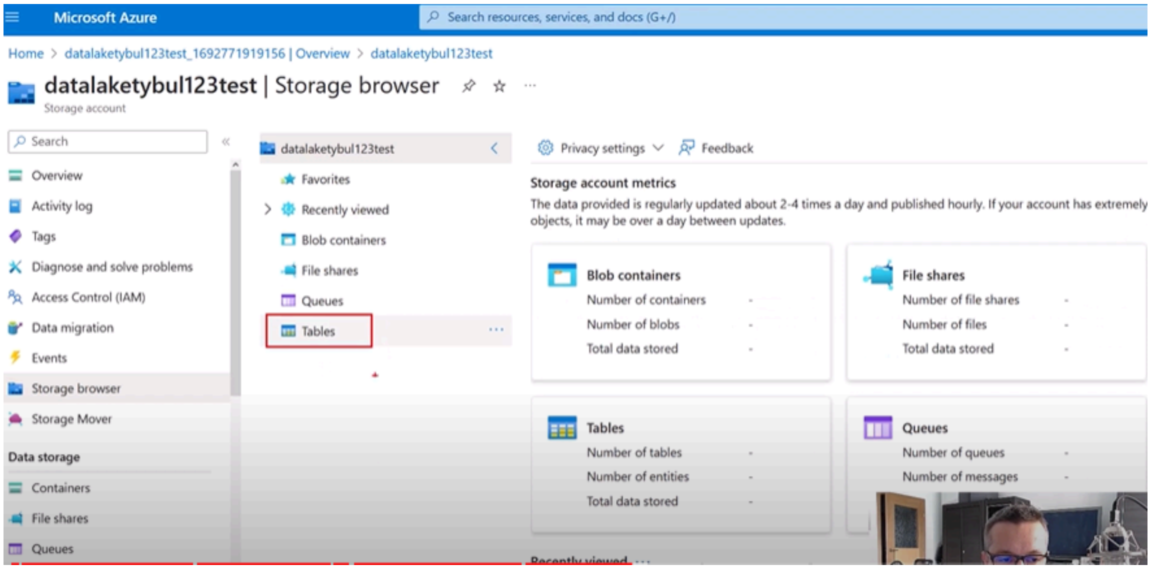
Here's why NoSQL is a good fit for an e-commerce application:

- **Large and Unstructured Data:** Product information like descriptions, reviews, and images can be bulky and unstructured. NoSQL databases can handle this variety efficiently.
- **Scalability:** During sales or peak seasons, the website experiences a surge in traffic. NoSQL databases can easily scale horizontally by adding more servers, ensuring smooth operation.
- **Flexibility:** Customer profiles can include diverse information like purchase history, browsing behavior, and wishlists. NoSQL's flexible schema allows you to store this data without rigid pre-defined structures.

3. It stores the data in key-value pairs



4. Lets see this practically



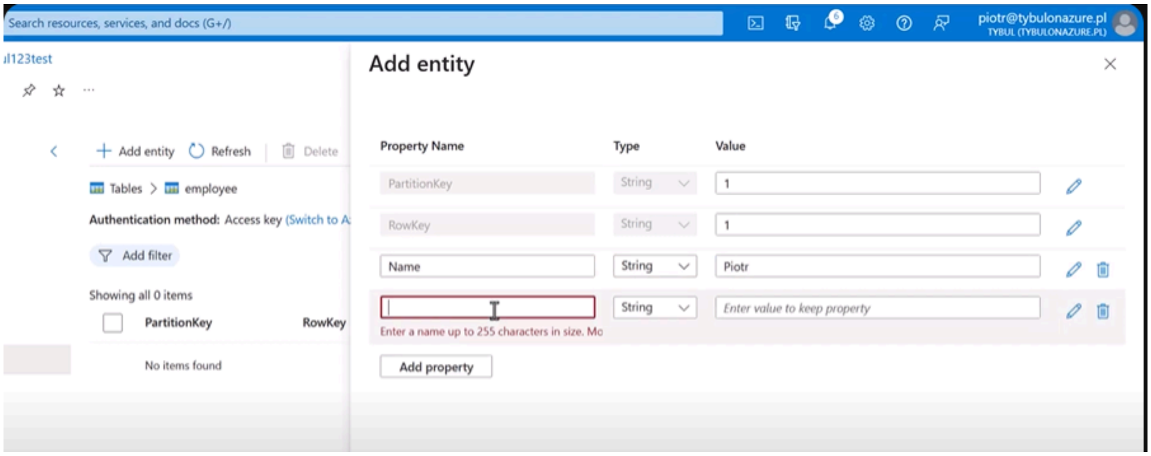
Storage account metrics

The data provided is regularly updated about 2-4 times a day and published hourly. If your account has extremely objects, it may be over a day between updates.

Blob containers	
Number of containers	-
Number of blobs	-
Total data stored	-
File shares	
Number of file shares	-
Number of files	-
Total data stored	-
Tables	
Number of tables	-
Number of entities	-
Total data stored	-
Queues	
Number of queues	-
Number of messages	-

Recently viewed ...

5. Here we create a new table and inside that we can create entities



Add entity

Authentication method: Access key (Switch to A

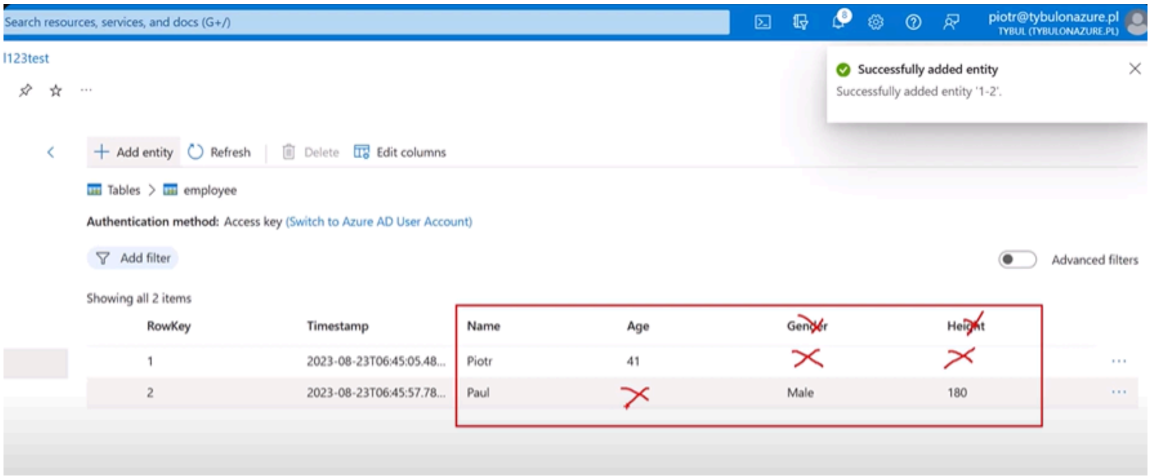
Showing all 0 items

PartitionKey	RowKey
No items found	

Property Name	Type	Value
PartitionKey	String	1
RowKey	String	1
Name	String	Piotr
	String	Enter value to keep property

Enter a name up to 255 characters in size. Mo

6. Here if we can see …we can add whatever values we want



Successfully added entity

Successfully added entity '1-2'.

Authentication method: Access key (Switch to Azure AD User Account)

Showing all 2 items

RowKey	Timestamp	Name	Age	Gender	Height
1	2023-08-23T06:45:05.48...	Piotr	41		
2	2023-08-23T06:45:57.78...	Paul		Male	180

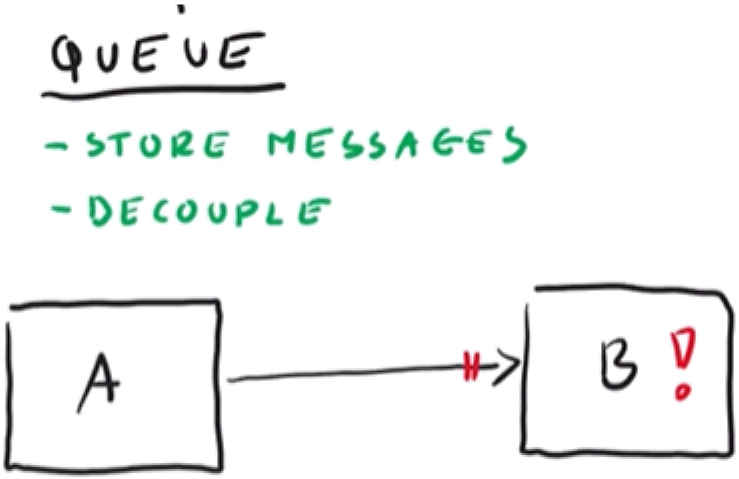
and the table does not have a schema

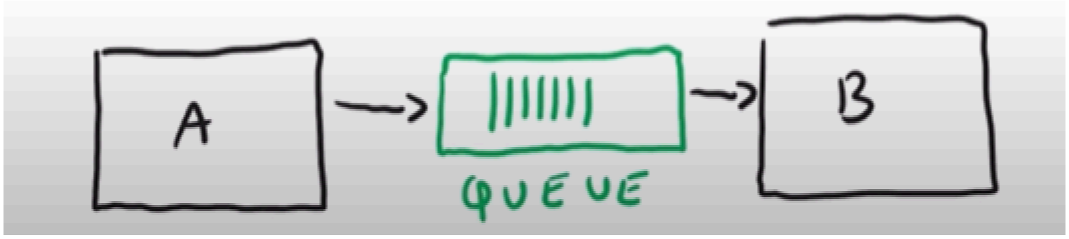
7. SO basically table service is used to store key value pairs in entities

Storage Account - Queue Service

1. They are used to store messages

2. Usecase -



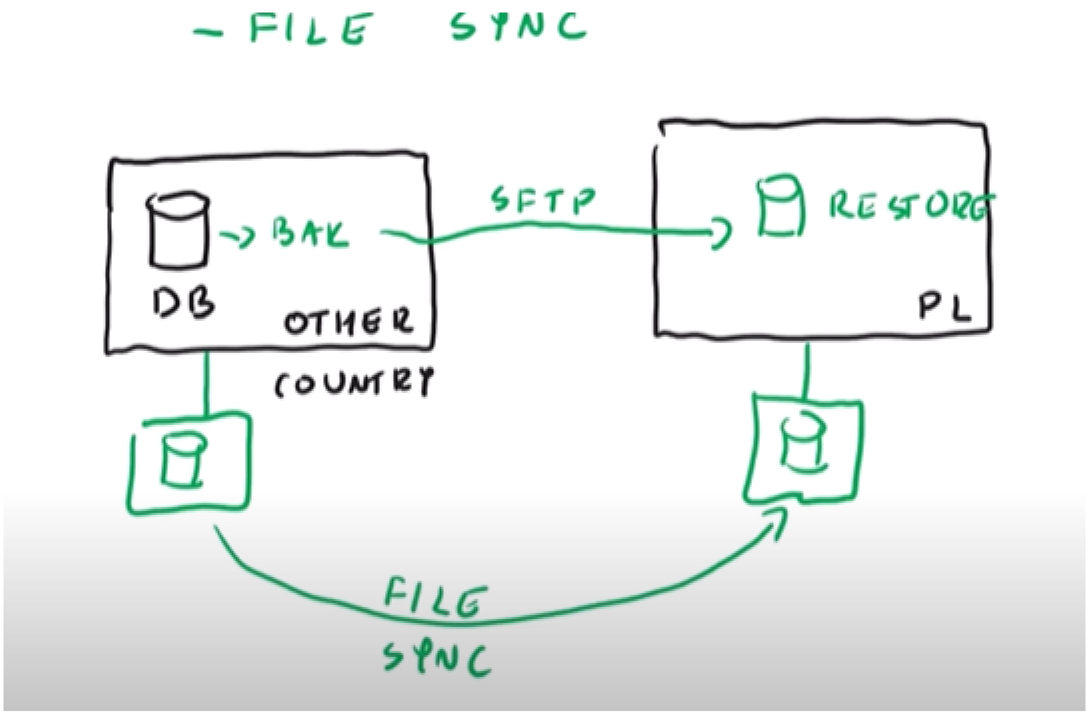


Suppose here we have 2 services…and A is sending msgs to B….what if B's capacity is full or what if there is network issue..then B cannot consume the messages..

3. So to avoid that…we use a Queue service which takes the messages from A ..then B can consume it in own pace
4. But it does not guarantee the FIFO.

Storage account - File Service

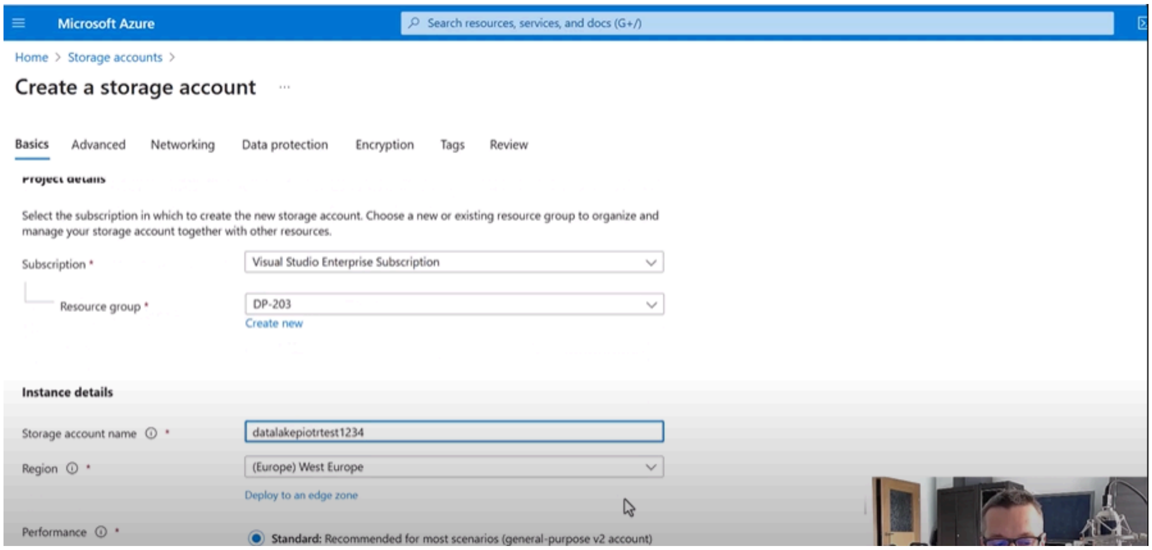
1. It is basically used as fileshare in the cloud
2. Also it helps us in file sync

3. 

4. Here the person wants to restore the data of Db which is in other country…so they have initially used SFTP to send files over the network..but it wasnt the optimal one..due to network issues
5. The workaround was to use File Service…which sync the files as soon as they appear on DB

ADLSg2

1. So here we can configure our blob storage to work as datalake by going into advanced settings and enabling hierarchical Namespace



Microsoft Azure
Search resources, services, and docs (G+/)
Home > Storage accounts >

Create a storage account

Project details

Select the subscription in which to create the new storage account. Choose a new or existing resource group to organize and manage your storage account together with other resources.

Subscription * Visual Studio Enterprise Subscription

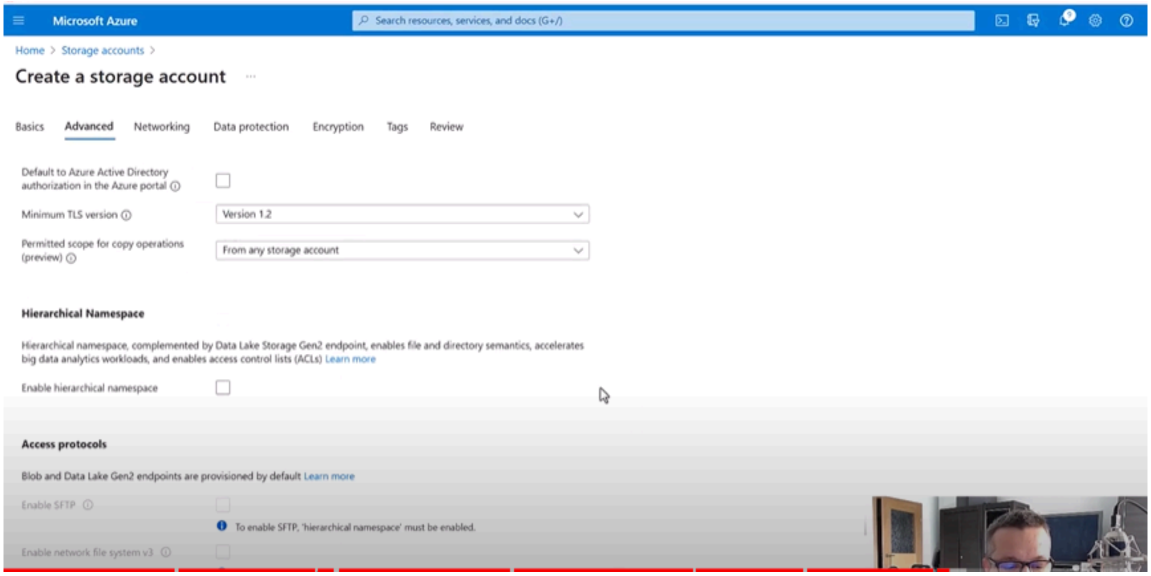
Resource group * DP-203

Instance details

Storage account name * datalakepiotrtest1234

Region * (Europe) West Europe

Performance * ● Standard: Recommended for most scenarios (general-purpose v2 account)



Microsoft Azure
Search resources, services, and docs (G+/)
Home > Storage accounts >

Create a storage account

Default to Azure Active Directory authorization in the Azure portal □

Minimum TLS version Version 1.2

Permitted scope for copy operations (preview) From any storage account

Hierarchical Namespace

Hierarchical namespace, complemented by Data Lake Storage Gen2 endpoint, enables file and directory semantics, accelerates big data analytics workloads, and enables access control lists (ACLs) Learn more

Enable hierarchical namespace □

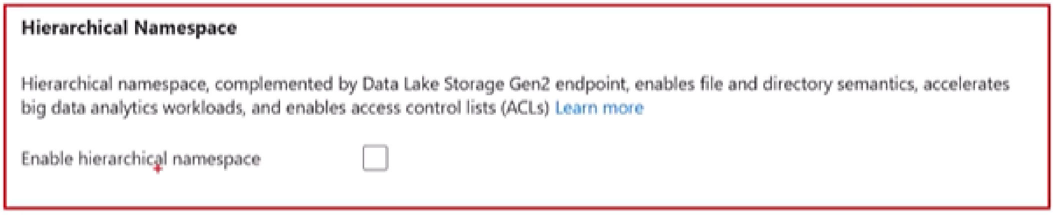
Access protocols

Blob and Data Lake Gen2 endpoints are provisioned by default Learn more

Enable SFTP □

To enable SFTP, 'hierarchical namespace' must be enabled.

Enable network file system v3 □



Hierarchical Namespace

Hierarchical namespace, complemented by Data Lake Storage Gen2 endpoint, enables file and directory semantics, accelerates big data analytics workloads, and enables access control lists (ACLs) Learn more

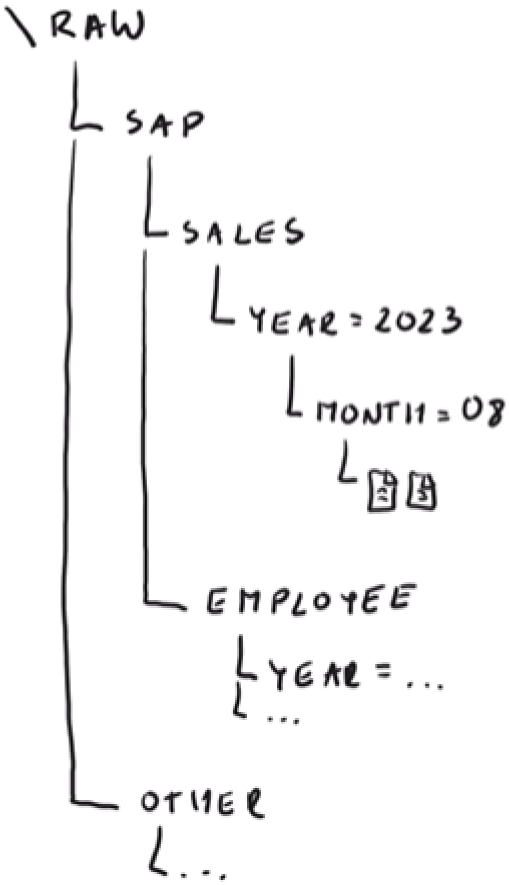
Enable hierarchical namespace □

2. This turns our regular blob storage into datalake
3. Here for analytical purpose we use ADLSg2
4. Blob storage helps us to store images,movies,videos…which we integrate to our website by linking them in blob storage

5. Diff bw ADLSg2 and Blob storage

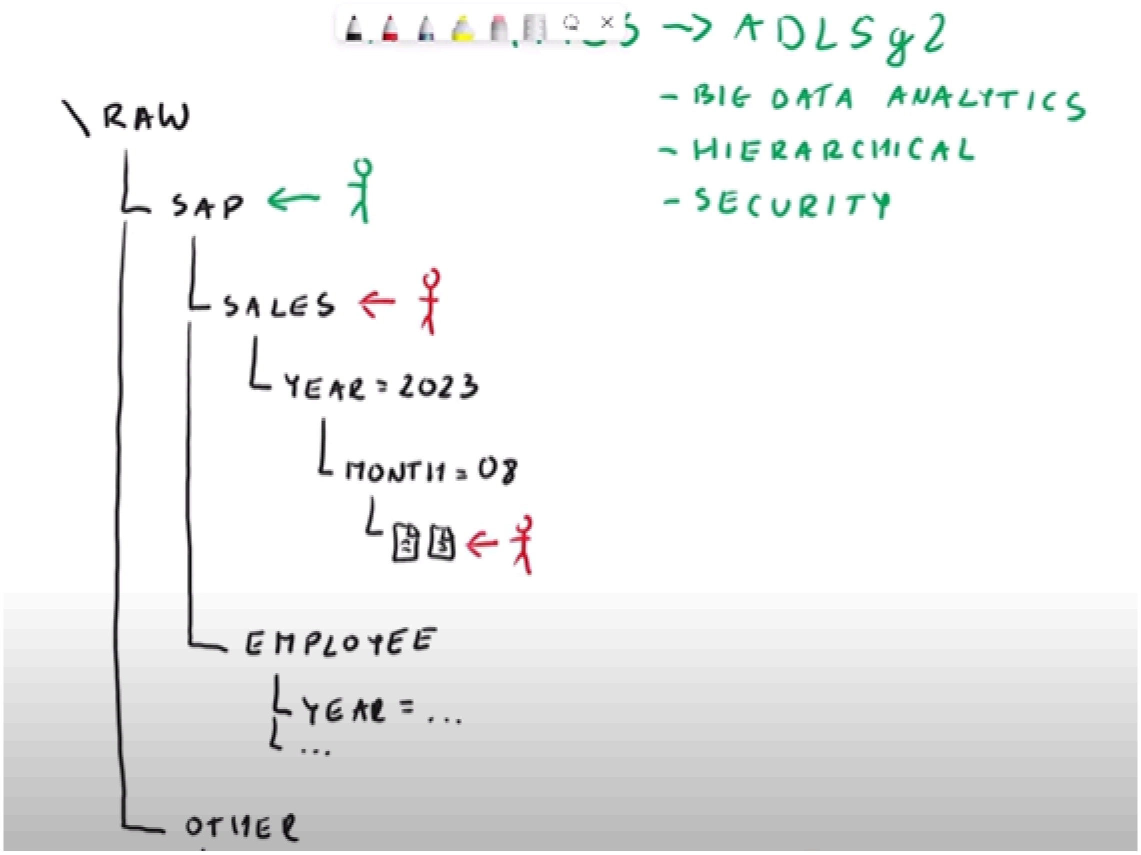
   -ANALYTICS -> ADLSg2

   - BIG DATA ANALYTICS
   - HIERARCHICAL



6. Usecase of hierarchical in real time this improves performance and avoid the mess

7. Here we can also give RBAC to specific users



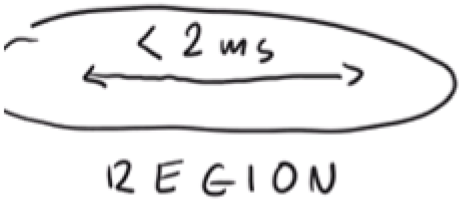
Storage Account Redundancy

Azure regions

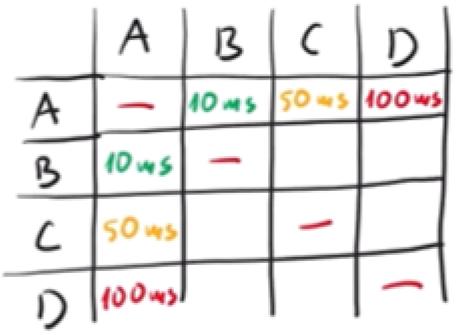
1. Actually Azure services are hosted somewhere physically…but what if they fail? Do we have any options?

2. Here we can see all the azure regions in the world

3. Here region contains one or more data center…and latency within the region is very less



4. We have to choose a region/data center which is closer to our application and which gives us low latency…and if we have many data centers in our region…then we can go ahead with the one that costs less



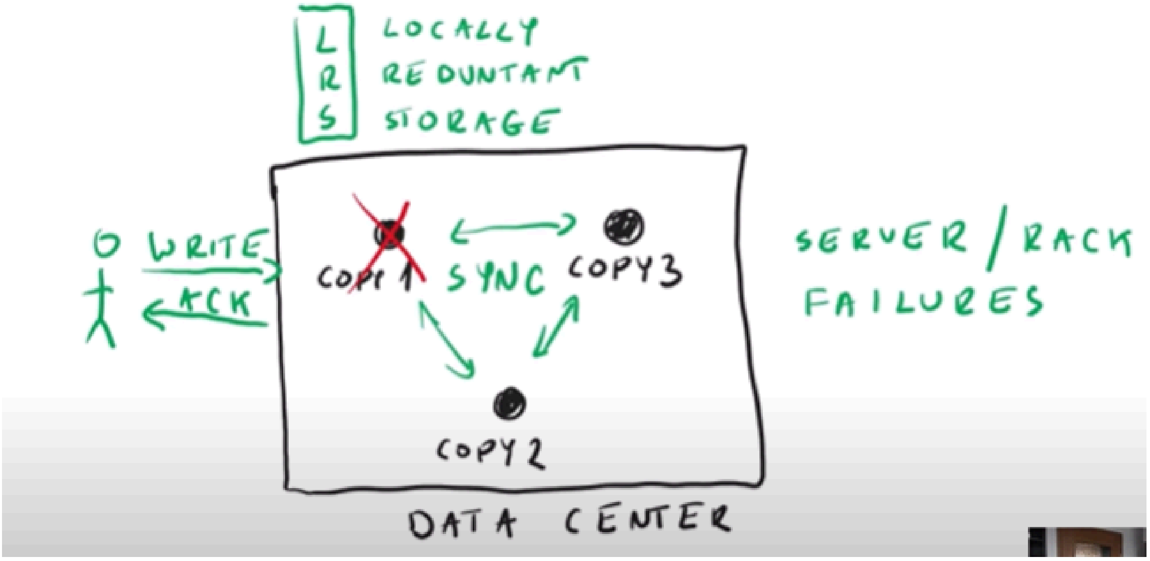
	A	B	C	D
A	—	10 ms	50 ms	100 ms
B	10 ms	—		
C	50 ms		—	
D	100 ms			—

latency bw regions of diff country, diff continent

5. And here we create a datalake in the region which is closer to us…and inside region we may have multiple datacenters…where are datalake gets stored
6. And we have to protect our datalake no matter what

Local Redundancy

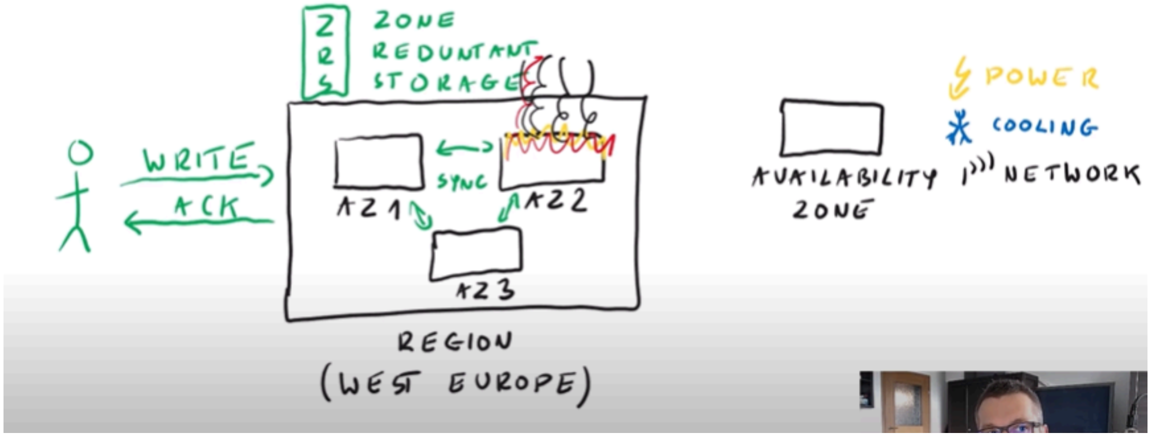
1. Lets assume this as our data center .and when a customer saves(write) his data in a datalake



   The data will be perfectly synced and only after syncing it acknowledges the customer
2. Here what LRS does is it replicates the data in 3 different servers/rack locally in the data center…so if one server fails…we get data from different server/rack
3. It protects us from server failure
4. This is the cheapest redundancy service in storage Account
5. But it is not safest… what if our data center caughts in fire?

Zone Redundancy

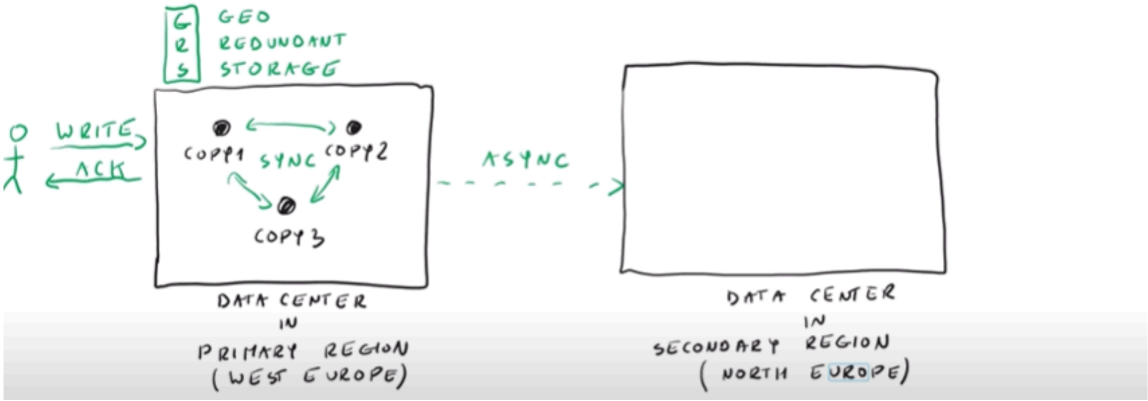
1. Coming to ZRS…here the datalake is replicated in 2 different available zones inside the region



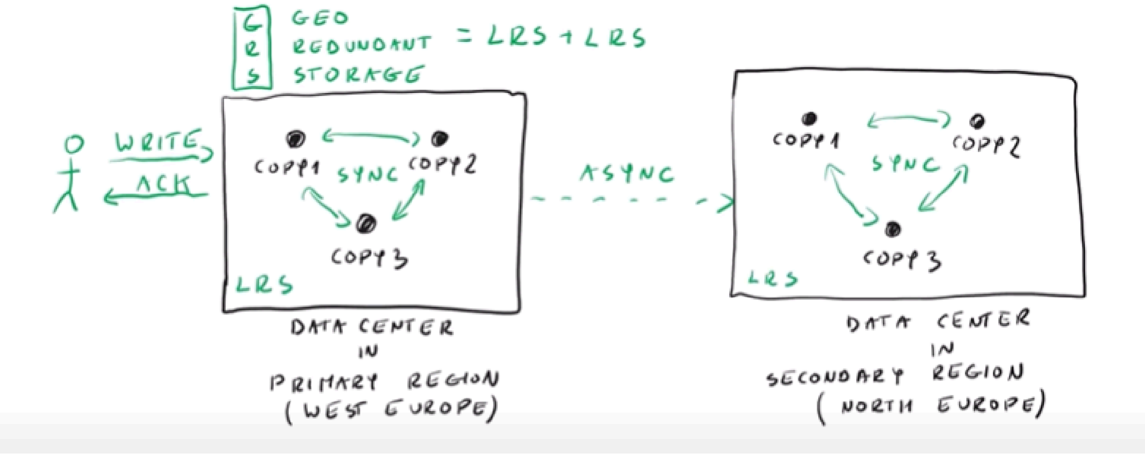
2. And the data will be synced concurrently in all th zones
3. Here each available zone has their own power,cooling system and network
4. So if one AZ gets shutdown …our data will not get lost
5. It Protects us from data centers failures
6. But what if there's a tsunami, earth quack which effects our region?

Geo Redundant Storage

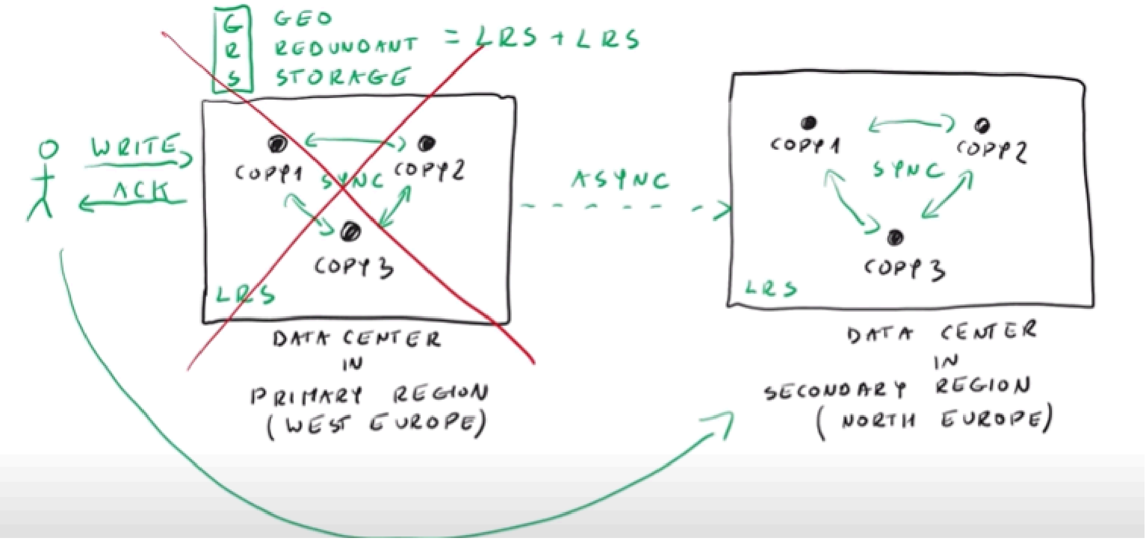
1. In GRS our data is stored in data center which is present in the another region asynchronously



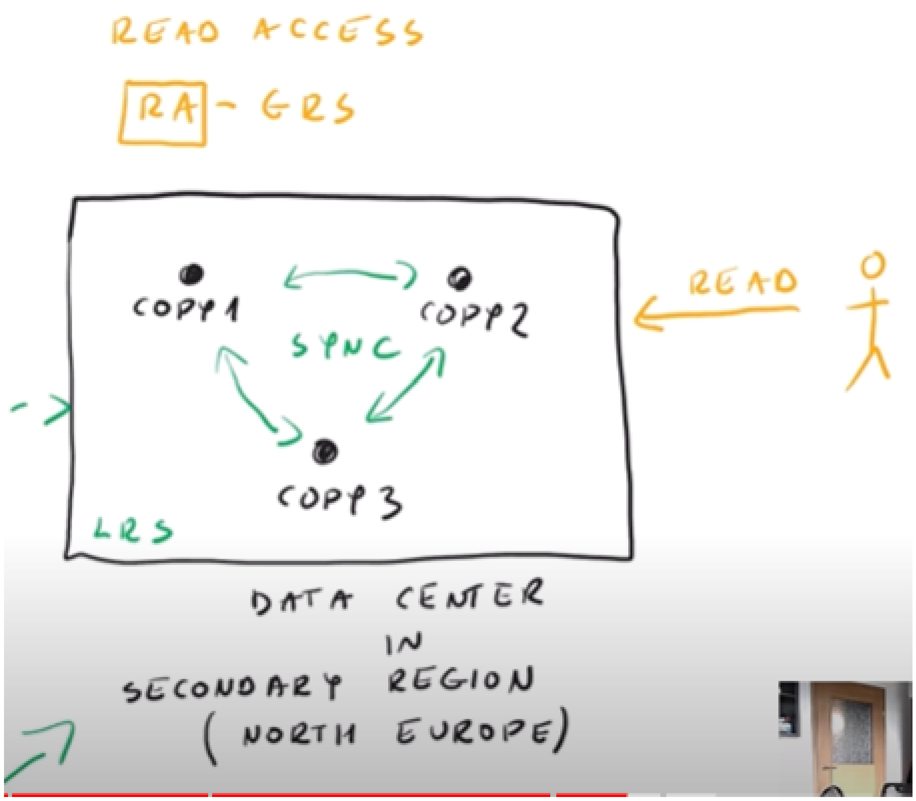
2. Here we also get LRS….



3. Here the one drawback is …suppose a user write some data in a data center at primary region…and what if it immediately shutdown due to unknown reason…then this data might not synced to different region



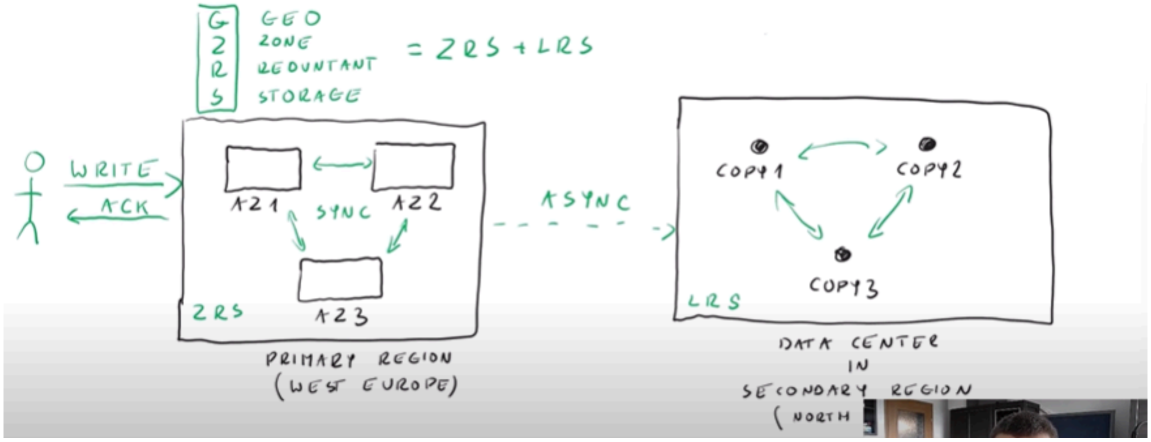
4. Here we can also have read access to the secondary region …for the company’s users who lives in secondary region

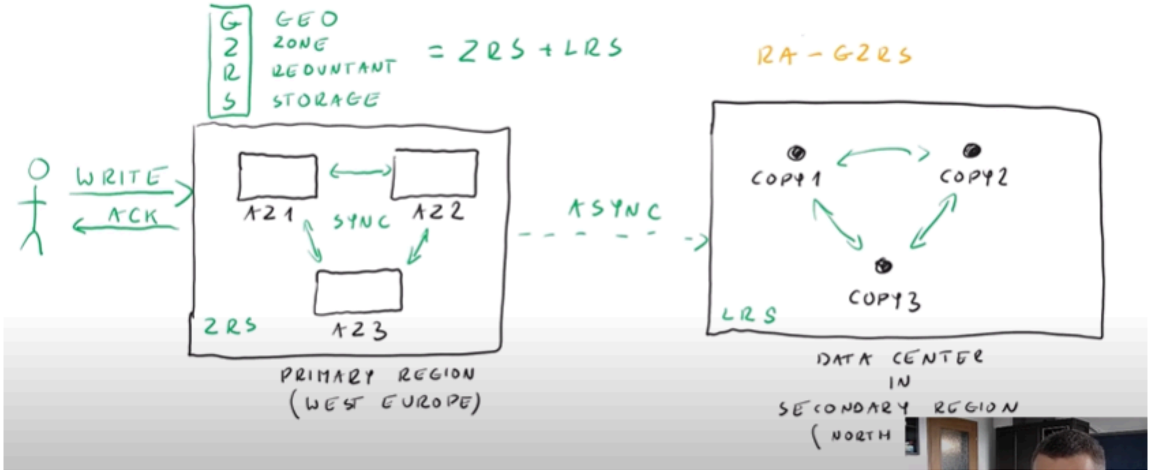


5. But writing happens only in primary region
6. GRS protects us from regional issues

GEO Zone Redundant Storage(the most expensive)

1. SO basically what GZRS does is in Primary region it follows Zone redundancy Storage…and in seocnday region it follows LRS

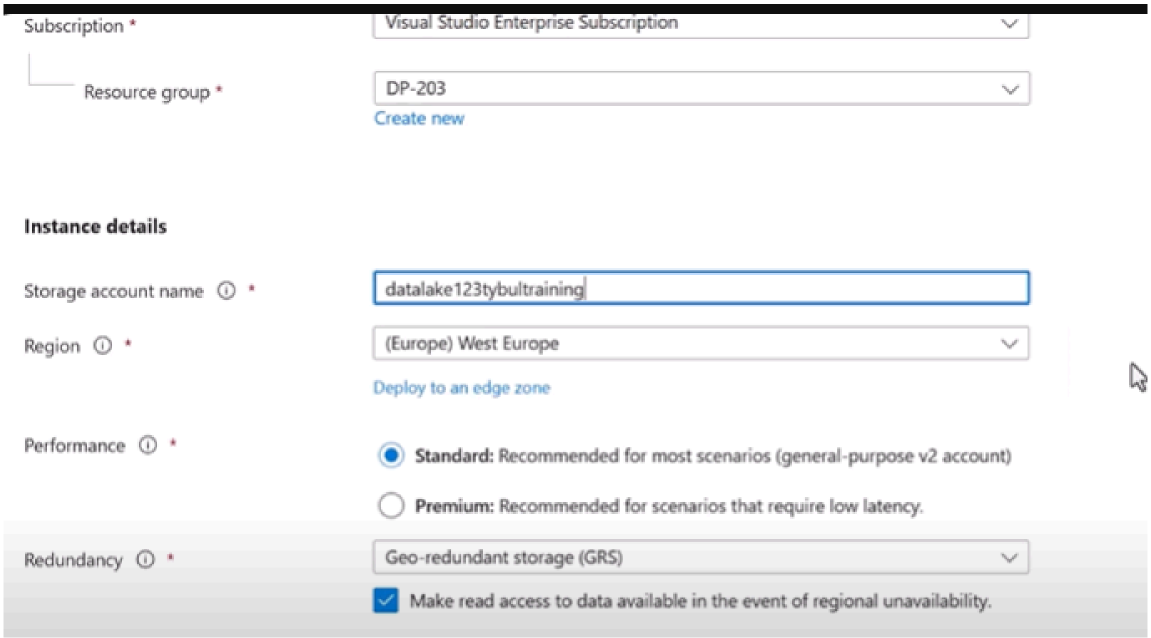


2. 

   we also have RA-GZRS..where a user can get read access from secondary region

Practical

1. Here while creating a storage account we'll be having this redundancy



Subscription * : Visual Studio Enterprise Subscription

Resource group * : DP-203

Create new

Instance details

Storage account name ⓘ * : datalake123tybultraining

Region ⓘ * : (Europe) West Europe

Deploy to an edge zone

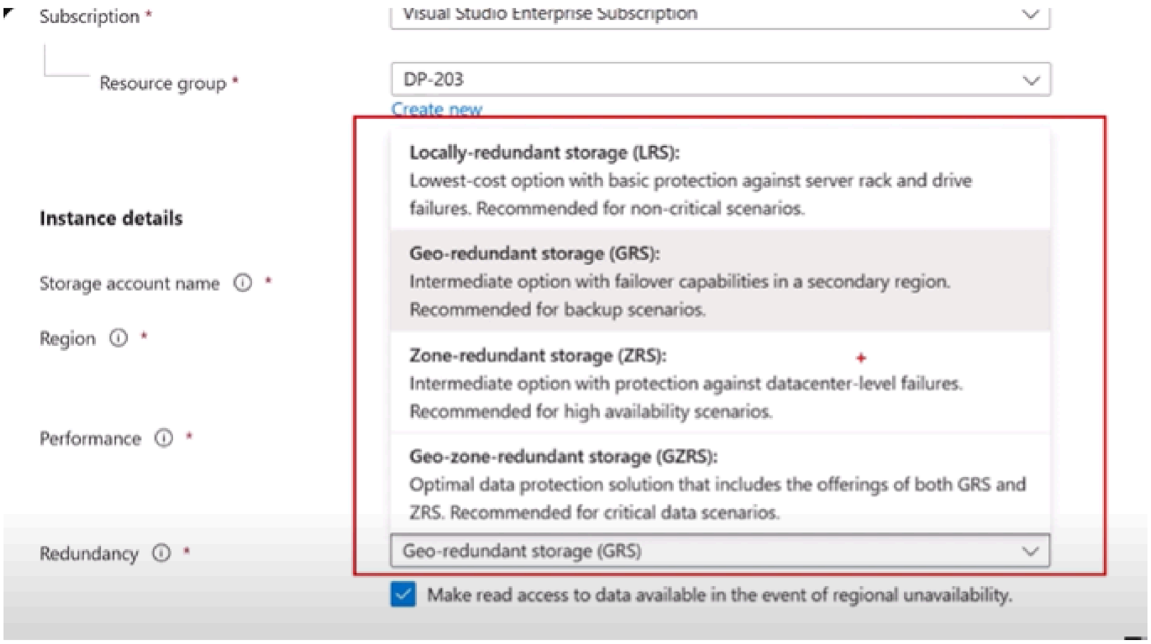
Performance ⓘ * :

● **Standard:** Recommended for most scenarios (general-purpose v2 account)

○ **Premium:** Recommended for scenarios that require low latency.

Redundancy ⓘ * : Geo-redundant storage (GRS)

☑ Make read access to data available in the event of regional unavailability.



Subscription * : Visual Studio Enterprise Subscription

Resource group * : DP-203

Create new

Instance details

Storage account name ⓘ *

Region ⓘ *

Performance ⓘ *

Locally-redundant storage (LRS):
Lowest-cost option with basic protection against server rack and drive failures. Recommended for non-critical scenarios.

Geo-redundant storage (GRS):
Intermediate option with failover capabilities in a secondary region. Recommended for backup scenarios.

Zone-redundant storage (ZRS):
Intermediate option with protection against datacenter-level failures. Recommended for high availability scenarios.

Geo-zone-redundant storage (GZRS):
Optimal data protection solution that includes the offerings of both GRS and ZRS. Recommended for critical data scenarios.

Redundancy ⓘ * : Geo-redundant storage (GRS)

☑ Make read access to data available in the event of regional unavailability.

2.